
Internet Computing Support for Digital Government

Athman Bouguettaya, Abdelmounaam Rezgui, Brahim Medjahed, and Mourad Ouzzani
Virginia Tech



Contents

1 Internet Computing Support for Digital Government	1
<i>Athman Bouguettaya, Abdelmounaam Rezgui, Brahim Medjahed, and Mourad Ouzzani Virginia Tech</i>	
1.1 Introduction	1
1.2 Digital Government Applications: An Overview	4
1.3 Issues in Building E-Government Infrastructures	6
1.4 A Case Study: The WebDG System	8
1.4.1 Ontological Organization of Government Databases	8
1.4.2 Web Services Support For Digital Government	10
1.4.3 Preserving Privacy in WebDG	11
1.4.4 Implementation	12
1.4.5 A Scenario Tour	12
1.5 Conclusion	14



1

Internet Computing Support for Digital Government

Athman Bouguettaya, Abdelmounaam Rezgui, Brahim Medjahed, and Mourad Ouzzani
Virginia Tech

CONTENTS

1.1 Introduction	1
1.2 Digital Government Applications: An Overview	4
1.3 Issues in Building E-Government Infrastructures	5
1.4 A Case Study: The WebDG System	8
1.5 Conclusion	14

The Web has introduced new paradigms in the way data and services are accessed. The recent burst of *Web technologies* has enabled a novel computing paradigm: *Internet computing*. This new computing paradigm has, in turn, enabled a new range of applications built around Web technologies. These Web-enabled applications, or simply, Web applications cover almost every aspect of our everyday life (e.g., e-mail, e-shopping, e-learning). *Digital Government (DG)* is a major class of *Web applications*. This chapter has a twofold objective. It first provides an overview of Digital Government and the key issues and challenges in building DG infrastructures. The second part of the chapter is a description of *WebDG*, an experimental Digital Government infrastructure built around *distributed ontologies* and *Web services*.

1.1 Introduction

The Web has changed many aspects of our everyday life. The *e-revolution* has had an unparalleled impact on how people live, communicate, and interact with businesses and government agencies. As a result, many well-established functions of modern society are being rethought and redeployed. Amongst all of these functions, the *Government* function is one where the *Web impact* is the most tangible.

Governments are the most complex organizations in a society. They provide the legal, political, and economic infrastructure to support the daily needs of citizens and businesses [Bouguettaya et al., 2002]. A government generally consists of large and complex networks of institutions and agencies. The Web is progressively, but radically, changing the traditional mechanisms in which these institutions and agencies operate and interoperate. More importantly, the Web is redefining the government-citizen relationship. Citizens worldwide are increasingly experiencing a new, Web-based paradigm in their relationship with their governments. Traditional, paper and clerk-based functions such as voting, filing of tax returns, or renewing of driver licenses are swiftly being replaced by more efficient Web-based applications. People may value differently this development but they, almost all, appear to be accepting this new, promising form of government called *Digital Government*.

Digital Government (DG) or *E-Government* may be defined as the process of using information

and communication technologies to enable the civil and political conduct of government [Elmagarmid and McIver, 2002]. In a DG environment (Figure 1.1), a complex set of interactions amongst government (local, state, and federal) agencies, businesses, and citizens may take place. These interactions typically involve an extensive transfer of information in the form of electronic documents. The objective of e-government is, in particular, to improve government-citizen interactions through the deployment of an infrastructure built around the “life-experience” of citizens. Digital government is expected to drastically simplify the information flow among different government agencies and with citizens. On-line DG services are expected to result in a significant reduction in the use of paper, mailing and shipping activities, and, consequently, improving the services provided to citizens [Dawes et al., 1999].

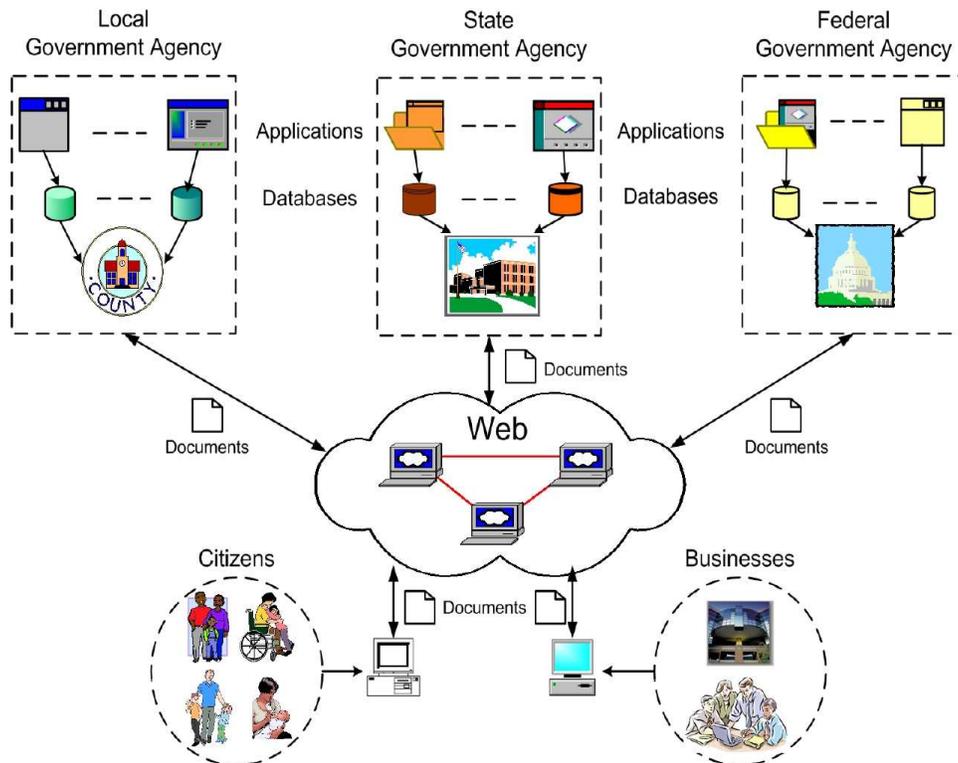


FIGURE 1.1

A Digital Government Environment

From a technical perspective, Digital Government may be viewed as a particular *class* of other classes of Internet-based applications (e.g., e-commerce, e-learning, e-banking). Typically, a DG application is supported by a number of distributed hosts that interoperate to achieve a given government function. The Internet is the medium of choice for the interaction between these hosts. *Internet computing* is therefore the basis for the development of almost all DG applications. Indeed, Internet technologies are at the core of all DG applications. These technologies may be summarized in five major categories: (i) markup languages (e.g., SGML, HTML, XML), (ii) scripting languages (e.g., CGI, ASP, Perl, PHP), (iii) Internet communication protocols (e.g., TCP/IP, HTTP, ATM), (iv) distributed computing technologies (e.g., CORBA, Java RMI, J2EE, EJB), and (v) security proto-

cols (e.g., SSL, S-HTTP, TSL). However, despite the unprecedented technological flurry that the Internet has elicited, a number of DG related challenges still remain to be addressed. These include the interoperability of DG infrastructures, scalability of DG applications, and privacy of the users of these applications. An emerging technology that is particularly promising in developing the next generation of DG applications is *Web services*. A *Web service* is a functionality that can be programmatically accessible via the Web [Tsur et al., 2001]. A fundamental objective of Web services is to enable interoperability amongst different software applications running on a variety of platforms [Medjahed et al., 2003; Vinoski, 2002a; FEA Working Group, 2002]. This development grew against the backdrop of the *Semantic Web*. The Semantic Web is not a “separate Web” but an extension thereof, in which information is given well-defined meaning W3C [2001a]. This would enable machines to “understand” and automatically process the data that they merely display at present.

A Brief History of Digital Government

Governments started using computers to improve the efficiency of their processes as early as the 1950s [Elmagarmid and McIver, 2002]. However, the *real* “history” of Digital Government may be traced back to the second half of the 1960s. During the period from 1965 until the early 1970s, many technologies that would enable the vision of citizen-centered digital applications were developed. A landmark step was the development of packet switching that led to the development of the ARPANET [Elmagarmid and McIver, 2002]. The ARPANET was not initially meant to be used by average citizens. However, being the ancestor of the Internet, its development was undoubtedly a milestone in the history of Digital Government.

Another enabling technology for digital government is EDI (*Electronic Data Interchange*) [Adam et al., 1998]. EDI can be broadly defined as the computer-to-computer exchange of information, from one organization to another. Although EDI mainly focuses on business-to-business applications, it has also been adopted in Digital Government. For example, the U.S. Customs Service initially used EDI in the mid- to late 1970s to process import paperwork more accurately and more quickly.

One of the pioneering efforts that contributed to boosting Digital Government was the 1978 report [Nora and Minc, 1978] presented to then French President *V. G. D’Estaing*. The report aimed at restructuring the society by extensively introducing telecommunication and computing technologies. It triggered the development, in 1979, of the French Télétel/Minitel videotext system. By 1995, Minitel provided over 26000 on-line services, many of which were government services [Kessler, 1995].

The PC revolution in the 1980s coupled with significant advances in networking technologies and dial-up on-line services had the effect of bringing increasing numbers of users to a *computer-based* lifestyle. This period also witnessed the emergence of a number of on-line government services worldwide. One of the earliest of these services was the Cleveland FREENET developed in 1986. The service was initially developed to be a forum for citizens to communicate with public health officials.

The early 1990s have witnessed three other key milestones in the DG saga. These were: (1) the introduction in 1990 of the first commercial dial-up access to the Internet, (2) the release in 1992 of the World Wide Web to the public, and (3) the availability in 1993 of the first general purpose Web browser Mosaic. The early 1990s have also been the years when Digital Government was established as a distinct research area. A new Internet-based form of Digital Government had finally come to life. The deployment of DG systems also started in the 1990s. Many governments worldwide launched large scale DG projects. A project that had a seminal effect was the Amsterdam’s Digital City project. It was first developed in 1994 as a local social information infrastructure. Since then, over 100 digital city projects have started across the world [Elmagarmid and McIver, 2002].

This chapter presents some key concepts behind the development of DG applications. In partic-

ular, we elaborate on the important challenges and research issues. As a case study, we describe our ongoing project named *WebDG* [Bouguettaya et al., 2001b,a; Rezgui et al., 2002]. The *WebDG* system uses *distributed ontologies* to organize and efficiently access government data sources. Government functions are *Web-enabled* by wrapping them with *Web services*.

In Section 2, we describe a number of widely used DG applications. Section 3 discusses some of the most important issues in building DG applications. In Section 4, we describe the major components and features of the *WebDG* system. We provide some concluding remarks in Section 5.

1.2 Digital Government Applications: An Overview

Digital Government spans a large spectrum of applications. In this section, we present a few of these applications and discuss some of the issues inherent to each of them.

Electronic Voting: *E-voting* is a DG application where the impact of technology on the society is one of the most straightforward. The basic idea is simply to enable citizens to express their opinions regarding local or national issues by accessing a government Web-based voting system. Examples of e-voting applications include electronic polls, political votes, and on-line surveys.

E-voting systems are particularly important. For example, a major political race may depend on an e-voting system. The *reliability* of e-voting systems must therefore be carefully considered. Other characteristics of a *good* e-voting system include: (i) Accuracy (a vote cannot be altered after it is cast and only and *all* valid votes are counted), (ii) Democracy (only eligible voters may vote and only once), (iii) Privacy (a ballot cannot be linked to the voter who cast it and a voter cannot prove that s/he voted in a certain way), (iv) Verifiability (it must be verifiable that *all* votes have been correctly counted), (v) Convenience (voters must be able to cast their votes quickly in one session and with minimal special skills), (vi) Flexibility (the system must allow a variety of ballot question formats), and (vii) Mobility (no restrictions must exist on the location from which a voter can cast a vote) [Cranor, 1996].

Many e-voting systems have been developed at a small and medium scale. Examples include the Federal Voting Assistance Program (instituted to allow US citizens who happen to be abroad during an election to cast their votes electronically) [Rubin, 2002] and the *e-petitioner* system used by the Scottish Parliament [Macintosh et al., 2002]. Deploying large scale e-voting systems (e.g., at a country scale), however, is not yet a common practice. It is widely admitted that “the technology does not exist to enable remote electronic voting in public elections” [Rubin, 2002].

Tax Filing: A major challenge being faced by government financial agencies is the improvement of revenue collection and development of infrastructures to better manage fiscal resources. An emerging effort towards dealing with this challenge is the electronic filing (*e-filing*) of tax returns. An increasing number of citizens, businesses, and tax professionals have adopted e-filing as their preferred method of submitting tax returns. According to *Forrester Research*, federal, state, and local US governments will collect 15 % of fees and taxes online by 2006, which corresponds to \$602 billion. The objective set by the U.S. Congress is to have 80% of tax returns filed electronically by 2007 [Golubchik, 2002].

One of the driving forces for the adoption of e-filing is the reduction of costs and time of doing business with the Tax Authority [Baltimore Technologies, 2001]. Each tax return generates several printable pages of data to be manually processed. Efficiency is improved by reducing employees’ costs for manual processing of information and minimizing reliance upon traditional paper-based storage systems. For example, the US Internal Revenue Service saves \$1.20 on each electronic tax return it processes. Another demonstrable benefit of e-filing is the improvement of the quality of

data collected. By reducing manual transactions, e-filing minimizes the potential for error. Estimations indicate that 25% of tax returns filed via traditional paper-based procedures are miscalculated either by the party submitting the return or by the internal revenue auditor [Baltimore Technologies, 2001].

Government Portals: Continuously improving the public service is a critical government mission. Citizens and businesses are requiring on-demand access to basic government information and services in various domains such as finance, healthcare, transportation, telecommunications, and energy. The challenge is to deliver higher-quality services faster, more efficiently, and at lower costs. To face this challenges, many governments are introducing e-government portals. These Web-accessible interfaces aim at providing consolidated views and navigation for different government constituents. They simplify information access through a single sign-on to government applications. They also provide common look-and-feel user interfaces and pre-built templates users can customize. Finally, e-government portals offer *anytime-anywhere* communications by making government services and information instantly available via the Web.

There are generally four types of e-government portals: Government-to-Citizen (G2C), Government-to-Employee (G2E), Government-to-Government (G2G), and Government-to-Business (G2B). G2C portals provide improved citizen services. Such services include transactional systems such as tax payment and vehicle registration. G2E portals streamline internal government processes. They allow the sharing of data and applications within a government agency to support a specific mission. G2G portals share data and transactions with other government organizations to increase operational efficiencies. G2B portals enable interactions with companies to reduce administrative expenses associated with commercial transactions and foster economic development.

Geographic Information Systems (GISs): To conduct many of their civil and military roles, governments need to collect, store, and analyze huge amounts of data represented in graphic formats (e.g., road maps, aerial images of agricultural fields, satellite images of mineral resources). The emergence of Geographic Information Systems (GISs) had a revolutionary impact on how governments conduct activities that require capturing and processing images. A GIS is a computer system for capturing, managing, integrating, manipulating, analyzing and displaying data which is spatially referenced to the Earth [McDonnell and Kemp, 1996].

The use of GISs in public-related activities is not a recent development. For example, the water [Bell, 1993] and electricity [Pecth, 1993] supply industries were using GISs during the early 1990s. With the emergence of Digital Government, GISs have proven to be effective tools in solving many of the problems that governments face in public management. Indeed, many government branches and agencies need powerful GISs to properly conduct their functions. Examples of applications of GISs include: mapmaking, site selection, simulating environmental effects, and designing emergency routes [Geological Survey, 2002].

Social and Welfare Services: One of the traditional roles of government is to provide social services to citizens. Traditionally, citizens obtain social benefits through an excessively effort and time-consuming process. To assist citizens, case officers may have to manually locate and interrogate a myriad of government databases and/or services before the citizen's request can be satisfied.

Research aiming at improving government social and welfare services has shown that two important challenges must be overcome: (i) the distribution of service providers across several, distant locations, and (ii) the heterogeneity of the underlying processes and mechanisms implementing the individual government social services. In [Bouguettaya et al., 2001a], we proposed the *one-stop shop* model as a means to simplify the process of collecting social benefits for needy citizens. This approach was implemented and evaluated in our *WebDG* system described in detail later in this chapter.

1.3 Issues in Building E-Government Infrastructures

Building and deploying an e-government infrastructure entail a number of policy and technical challenges. In this section, we briefly mention some of the major issues that must be addressed for a successful deployment of most DG applications and infrastructures.

Data Integration

Government agencies collect, produce and manage massive amounts of data. This information is typically distributed over a large number of autonomous, heterogeneous, and large databases [Ambite et al., 2002]. Several challenges must be addressed to enable an efficient integrated access to this information. These include: ontological integration, middleware support, and query processing [Bouguettaya et al., 2002].

Scalability

A DG infrastructure must be able to scale to support growing numbers of underlying systems and users. It also must easily accommodate new information systems and support a large spectrum of heterogeneity and high volumes of information [Bouguettaya et al., 2002]. Two important facets of the scalability problem in DG applications must be addressed:

- **Scalability of Information Collection:** Government agencies continuously collect huge amounts of data. A significant challenge is to address the problem of the scalability of data collection, i.e., build DG infrastructures that scale to handle these huge amounts of data and effectively interact with autonomous and heterogeneous data sources [Golubchik, 2002; Wun-nava and Reddy, 2000]. In particular, an important and challenging feature of many DG applications is their intensive use of data uploading. For example, consider a tax filing application through which millions of citizens file (i.e., uploads) their income tax forms. Contrarily to the problem of scalability of data downloading (where a large number of users *download* data from the same server), the problem of scalability of data uploading has not yet found its effective solutions. The *bistros* approach was recently proposed to solve this problem [Golubchik, 2002]. The basic idea is to first route all uploads to a set of intermediary Internet hosts (called *bistros*) and then forward the data from one or more *bistros* to the server.
- **Scalability of Information Processing:** DG applications are typically destined to be used by large numbers of users. More importantly, these users may (or, sometimes, must) *all* use these applications in a short period of time. The most eloquent example for such a situation is certainly that of an e-voting system. On a vote day, an e-voting application must, within a period of only a few hours, process, i.e., collect, validate, and count the votes of tens of millions of voters.

Interoperability of Government Services

In many situations, citizens' needs cannot be fulfilled through one single e-government service. Different services (provided by different agencies) would have to interact with each other to fully service a citizen's request. A simple example is a *Child Support* service that may need to send an inquiry to a *Federal Taxation* service (e.g., IRS) to check revenues of a deadbeat parent. A more complex example is a *government procurement* service that would need to interact with various other e-government and business services.

The recent introduction of *Web services* was a significant advance in addressing the interoperability problem amongst government services. A Web service can easily and seamlessly discover and

use any other Web service irrespective of programming languages, operating systems, etc. Standards to describe, locate, and invoke Web services are at the core of intensive efforts to support interoperability. Although standards like SOAP and WSDL are becoming commonplace, there are still some incoherence in the way that they are implemented by different vendors. For example, SOAP::Lite for Perl and .NET implement SOAP 1.1 differently. In addition, not all aspects of those standards are being adopted [Sabbouh et al., 2001].

A particular and interesting type of interoperability relates to *semantics*. Indeed, semantic mismatches between different Web services are major impediments to achieve full interoperability. In that respect, work in the *semantic Web* is crucial in addressing related issues [Berners-Lee, 2001]. In particular, Web services would need mainly to be linked to ontologies that would make them meaningful [Trastour et al., 2001; Ankolekar et al., 2001]. Description, discovery, and invocation could then be made in a *semantic-aware* way.

Web services *composition* is another issue related to Web services interoperability. Composition creates new value-added services with functionalities outsourced from other Web services [Medjahed et al., 2003]. Thus, composition involves interaction with different Web services. Enabling service composition requires addressing several issues including composition description, service composability, composition plan generation, and service execution.

Security

Digital government applications inherently collect and store huge amounts of sensitive information about citizens. Security is therefore a vital issue in these applications. In fact, countless surveys and polls report that (the lack of) security is the reason that citizens most frequently present for their reluctance in using on-line government services. Applications such as e-voting, tax filing, or social e-services may not be usable if they are not sufficiently secured.

Advances in cryptography and protocols for secure Internet communication (e.g., SSL, S-HTTP) significantly contributed in securing information transfers within DG infrastructures. Securing DG infrastructures, however, involve many other aspects. For example, a service provider (a government agency) must be able to specify *who* may access the service, *how* and *when* accesses are made, as well as any other condition for accessing the service. In other words, *access control* models and architectures must be developed for Web services that support government functions as they must be for any other Web resource.

Part of the security problem in DG applications is also to secure the Web services that are increasingly used in deploying DG services. In particular, the issue of securing the interoperability of Web services is one that has been the focus of many standardization bodies. Many standards for securing Web services have been proposed or are under development. Examples include: WS Security [IBM et al., 2002], XML Encryption [W3C, 2001c], XML Digital Signature [W3C, 2001e], SOAP Digital Signature [W3C, 2001b], XKMS [W3C, 2001d], XACML [OASIS, 2001], and SAML [OASIS, 2002].

Privacy

Government agencies collect, store, process and share information about millions of individuals who have different preferences regarding their privacy. This naturally poses a number of legal issues and technical challenges that must be addressed to control the information flow amongst government databases and between these and third-party entities (e.g., private businesses). The common approach in addressing this issue consists of enforcing privacy by law or by self-regulation and few technology-based solutions have been proposed.

One of the legal efforts addressing the privacy problem was HIPAA, the Health Insurance Portability and Accountability Act passed by the US Congress in 1996. This act essentially includes regulations to reduce the administrative costs of health care. In particular, it requires all health plans that transmit health information in an electronic transaction to use a standard format [Congress,

1996]. HIPAA is expected to play a crucial role in preserving individuals' rights to the privacy of their health information. Also, as it aims at establishing national standards for electronic health care transactions, HIPAA is expected to have major impact on how Web-based health care providers and health insurance companies operate and interoperate.

Technical solutions to the privacy problem in DG have been *ad hoc*. For example, a number of protocols have been developed to preserve the privacy of e-voters using the Internet (e.g., [Ray et al., 2001]) or any arbitrary networks (e.g., [Mu and Varadharajan, 1998]). Another example of *ad hoc* technical solutions is the prototype system developed for the U.S. National Agricultural Statistics Service (NASS) [et al., 2002]. The system disseminates survey data related to on-farm usage of chemicals (fertilizers, fungicides, herbicides and pesticides). It uses *geographical aggregation* as a means to protect the identities of individual farms.

User Interface and User Accessibility

In a report authored by the US *PITAC* (President's Information Technology Advisory Committee) [U.S. PITAC, 2000], the committee enumerates, as one of its main findings, that "major technological barriers prevent citizens from easily accessing government information resources that are vital to their well being". The committee further adds that "government information is often unavailable, inadequate, out of date, and needlessly complicated".

E-government applications are typically built to be used by average citizens with, *a priori*, no special computer skills. Therefore, the user interfaces (UIs) used to access these applications must be easy to use and accessible to citizens with different aptitudes. In particular, some DG applications are destined to specific segments of the society (e.g., citizens at an elderly age or with special mental and/or physical ability). These applications must provide a user interface that suits their respective users' abilities and skills. Recent efforts aim at building "smart UIs" that progressively "get acquainted" to their users' abilities and dynamically adapt to those (typically, decaying) abilities. A recent study [West, 2002] reports that 82% of U.S. government Web sites have some form of disability access, which is up from 11% in 2001.

1.4 A Case Study: The WebDG System

In this section, we describe our research in designing and implementing a comprehensive infrastructure for e-government services called *WebDG* (*Web Digital Government*). *WebDG*'s major objective is to develop techniques to efficiently access government databases and services. Our partner in the *WebDG* project is the *Family and Social Services Administration* (FSSA). The FSSA's mission is to help needy citizens collect social benefits. The FSSA serves families and individuals facing hardships associated with low income, disability, aging, and children at risk for healthy development. To expeditiously respond to citizens' needs, the FSSA must be able to seamlessly integrate geographically distant, heterogeneous, and autonomously run information systems. In addition, FSSA applications and data need to be accessed through one single interface: *the Web*. In such a framework, case officers and citizens would transparently access data and applications as homogeneous resources. This and the next sections discuss *WebDG*'s major concepts and describe the essential components of its architecture.

1.4.1 Ontological Organization of Government Databases

The FSSA is composed of dozens of autonomous departments located in different cities and counties statewide. Each department's information system consists of a myriad of databases. To access

government information, case officers first need to locate the databases of interest. This process is often complex and tedious due to the heterogeneity, distribution, and large number of FSSA databases. To tackle this problem, we segmented FSSA databases into *distributed ontologies*. An *ontology* defines a taxonomy based on the semantic proximity of information interest [Ouzzani et al., 2000]. Each ontology focuses on a single common information type (e.g., disability). It dynamically groups databases into a single collection, generating a conceptual space with a specific content and scope. The use of distributed ontologies elicits the filtering and reduction of the overhead of discovering FSSA databases.

Ontologies describe coherent slices of the information space. Databases that store information about the same topic are grouped together. For example, all databases that may be of interest to disabled people (e.g., *Medicaid* and *Independent Living*) are members of the ontology *Disability* (Figure 1.2). For the purpose of this project, we have identified eight ontologies within FSSA, namely *Family*, *Visually Impaired*, *Disability*, *Low Income*, *At Risk Children*, *Mental Illness and Addiction*, *Health and Human Services*, and *Insurance*. A representative sample of these ontologies is presented in Figure 1.2. In this proposed framework, individual databases join and leave the formed ontologies at their own discretion. An overlap of two ontologies depicts the situation where a database stores information that is of interest to both of them. For example, the *Medicaid* database simultaneously belongs to three ontologies: *Family*, *Visually Impaired*, and *Disability*.

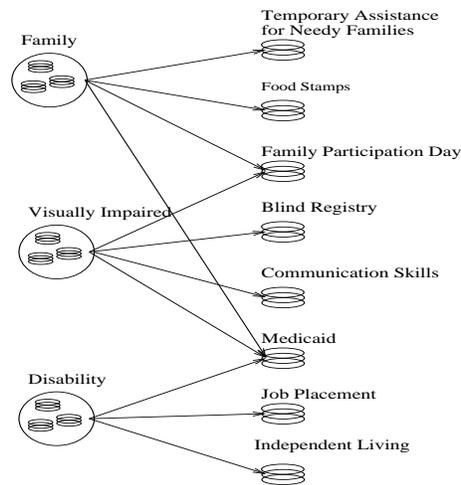


FIGURE 1.2

Sample FSSA Ontologies

The FSSA ontologies are not isolated entities. They are related by *inter-ontology relationships*. These relationships are dynamically established based on users' needs. They allow a query to be resolved by member databases of remote ontologies when it cannot be resolved locally. The inter-ontology relationships are initially determined statically by the ontology administrator. They essentially depict a *functional* relationship that would dynamically change over time.

Locating databases that fit users' queries requires detailed information about the content of each database. For that purpose, we associate with each FSSA database a *co-database* (Figure 1.3). A *co-database* is an object-oriented database that stores information about its associated database, ontologies and inter-ontology relationships. A set of databases exporting a certain type of information (e.g., *disability*) is represented by a class in the co-database schema. This class inherits from

a pre-defined class, *OntologyRoot*, that contains generic attributes. Examples of such attributes include *Information-type* (e.g., “Disability” for all instances of the class *Disability*) and *Synonyms* (e.g., “Handicap” is a synonym of “Disability”). In addition to these attributes, every subclass of the *OntologyRoot* class has some specific attributes that describe the domain model of the underlying databases.

1.4.2 Web Services Support For Digital Government

Several rehabilitation programs are provided within the FSSA to help disadvantaged citizens. Our analysis of the FSSA operational mechanisms revealed that the process of collecting social benefits is excessively time-consuming and frustrating. Currently, FSSA case officers must deal with different situations that depend on the particular needs of each citizen (disability, children health, housing, employment, etc.) For each situation, they must typically delve into a potentially large number of applications and determine those that best meet the citizens’ needs. For each situation, they must manually: (i) determine applications that appropriately satisfy citizens’ needs, (ii) determine how to access each application, and (iii) combine the results returned by those applications.

To facilitate the process of collecting benefits, we wrapped each FSSA application with a *Web service*. Web Services are emerging as a promising middleware to facilitate application-to-application integration on the Web [Vinoski, 2002b]. They are defined as modular applications offering sets of related functions that can be programmatically accessed through the Web. Adopting Web services in e-government enables: (i) *standardized* description, discovery, and invocation of welfare applications, (ii) *composition* of pre-existing services to provide *value added* services, and (iii) *uniform* handling of privacy.

The *providers* of *WebDG* services are bureaus within FSSA (e.g., *Bureau of Family Resources*) or external agencies (e.g., *US Department of Health and Human Services*). They define descriptions of their services (e.g., operations) and publish them in the *registry*. *Consumers* (citizens, case officers, and other e-government services) access the registry to locate services of interest. The registry returns the description of each relevant service. Consumers use this description to “understand” how to use the corresponding Web service.

Composing WebDG Services

The incentive behind composing government e-services is to further simplify the process of searching and accessing these services. We propose a new approach for the *(semi)automatic composition* of Web services. Automatic composition is expected to play a major role in enabling the envisioned *Semantic Web* [Berners-Lee, 2001]. It is particularly suitable for e-government applications. Case officers and citizens need no longer to search for services which might be otherwise a time-consuming process. Additionally, they are not required to be aware of the full technical details of the outsourced services.

WebDG’s approach for service composition includes four phases: *specification*, *matchmaking*, *selection*, and *generation*.

Specification – Users defines high level descriptions of the desired composition via an XML-based language called CSSL (*Composite Service Specification Language*). CSSL uses a subset of WSDL service interface elements and extends it to allow the: (1) description of semantic features of Web services and (2) specification of the control flow between composite services operations. Defining a WSDL-like language has two advantages. First, it makes the definition of composite services as simple as the definition of simple (i.e., non composite) services. Second, it allows the support of recursive composition.

Matchmaking – Based on user’s specification, the matchmaking phase automatically generates *composition plans* that conform to that specification. A *composition plan* refers to the list of out-

sourced services and the way they interact with each other (plugging operations, mapping messages, etc). A major issue addressed by *WebDG*'s matchmaking algorithm is *composability* of the outsourced services [Berners-Lee, 2001]. We propose a set of *rules* to check *composability* of e-government services. These include *operation semantics* composability and *composition soundness*. *Operation semantics* composability compares the categories or domains of interest (e.g., "health-care", "adoption") of each pair of interacting operations. It also compares their types or functionalities (e.g., "eligibility", "counseling"). For that purpose, we define two ontologies *Category* and *Type*. Our assumption is that both ontologies are pre-defined and agreed upon by government social agencies. Each operation includes two elements from the *Category* and *Type* ontologies respectively. *Composition soundness* checks whether combining Web services in a specific way provides an added value. For that purpose, we introduce the notion of *composition template*. A *composition template* is built for each composition plan generated by *WebDG*. It gives the general structure of that plan. We also define a subclass of templates called *stored templates*. These are defined *a priori* by government agencies. Since stored templates inherently provide added values, they are used to test the *soundness* of composition plans.

Selection – At the end of the matchmaking phase, several composition plans may have been generated. To facilitate the *selection* of relevant plans, we propose to define *Quality of Composition (QoC)* parameters. Examples of such parameters include time, cost, and relevance of the plan with respect to the user's specification (based on ranking for example). Composers define (as part of their profiles) thresholds corresponding to *QoC* parameters. Composition plans are returned only if the values of their *QoC* parameters are greater than their respective thresholds.

Generation – This phase aims at *generating* a detailed description of a composite service given a selected plan. This description includes the list of outsourced services, mappings between composite service and component service operations, mappings between messages and parameters, and flow of control and data between component services. Composite services are generated either in WSFL or XLANG [XLANG], two standardization efforts for composing services.

1.4.3 Preserving Privacy in WebDG

Preserving privacy is one of the most challenging tasks in deploying e-government infrastructures. The privacy problem is particularly complex due to the different perceptions that different users of e-government services may have with regard to their privacy. Moreover, a same user may have different privacy preferences associated to different types of information. For example, a user may have tighter privacy requirements regarding medical records than employment history. The user's perception of privacy also depends on the *information receiver*, i.e., who receives the information, and the *information usage*, i.e., the purposes for which the information is used.

To describe our approach to solving the privacy problem, we define three concepts: *privacy profiles*, *privacy credentials*, and *privacy scopes* [Rezgui et al., 2002]. The set of privacy preferences applicable to a user's information is called *privacy profile*. We also define *privacy credentials* that determines the *privacy scope* for the corresponding user. A privacy scope for a given user defines the information that an e-government service can disclose to that user. Before accessing an e-government service, users are granted privacy credentials. When a service receives a request, it first checks that the request has the necessary credentials to access the requested operation according to its privacy policy. If the request can be answered, the service translates it into an equivalent data query that is submitted to the appropriate government DBMS.

When the query is received by the DBMS, it is first processed by a privacy preserving data filter (*DFilter*). The *DFilter* is composed of two modules: the *Credential Checking Module (CCM)* and the *Query Rewriting Module (QRM)*. The *CCM* determines whether the service requester is authorized to access the requested information based on credentials. For example, Medicaid may state that a case officer in a given State may not access information of citizens from another State.

If the credential authorizes access to only part of the requested information, the QRM *redacts* the query (by removing unauthorized attributes) so that *all* the privacy constraints are enforced.

The *Privacy Profile Manager* (PPM) is responsible for enforcing privacy at a finer granularity than the CCM. For example, the local CCM may decide that a given organization can have access to local information regarding a group of citizens' health records. However, a subset of that group of citizens may explicitly request that parts of their records should not be made available to third-party entities. In this case, the local PPM will discard those parts from the generated result. The PPM is a translation of the consent-based privacy model in that it implements the privacy preferences of *individual* citizens. It maintains a repository of privacy profiles that stores individual privacy preferences.

1.4.4 Implementation

The *WebDG* system is implemented across a network of *Solaris* workstations. Citizens and case officers access *WebDG* via a *Graphical User Interface* (GUI) implemented using HTML/Servlet (Figure 1.3). Two types of requests are supported by *WebDG*: querying databases and invoking FSSA applications. All requests are received by the *WebDG manager*. The *Request Handler* is responsible for routing requests to the *Data Locator* (DL) or the *Service Locator* (SL). Queries are forwarded to the *Data Locator*. Its role is to educate users about the information space and locate relevant databases. All information necessary to locate FSSA databases is stored in *co-databases* (*ObjectStore*). The *co-databases* are linked to three different Orbix ORBs (one ORB per ontology). Users can learn about the content of each database by displaying its corresponding documentation in HTML/text, audio, or video formats. Once users have located the database of interest, they can then submit SQL queries. The *Query Processor* handles these queries by accessing the appropriate database via JDBC gateways. Databases are linked to OrbixWeb or VisiBroker ORBs.

WebDG currently includes ten (10) databases and seven (7) FSSA applications implemented in Java (JDK 1.3). These applications are wrapped by WSDL descriptions. We use the *Axis's Java2WSDL* utility in *IBM's Web Services Toolkit* to automatically generate WSDL descriptions from Java class files. WSDL service descriptions are published into UDDI registry. We adopt *Systinet's WASP UDDI Standard 3.1* as our UDDI toolkit. *Cloudscape* (4.0) database is used as a UDDI registry.

WebDG services are deployed using *Apache SOAP* (2.2). *Apache SOAP* provides not only server-side infrastructure for deploying and managing service, but also client-side API for invoking those services. Each service has a *deployment descriptor*. The descriptor includes the unique identifier of the Java class to be invoked, session scope of the class, and operations in the class available for the clients. Each service is deployed using the *service management client* by providing its descriptor and the URL of the *Apache SOAP servlet rpcrouter*.

The *Service Locator* allows the discovery of WSDL descriptions by accessing the UDDI registry. The SL implements *UDDI Inquiry Client* using WASP UDDI API. Once a service is discovered, its operations are invoked through *SOAP Binding Stub* which is implemented using Apache SOAP API. Service operations are executed by accessing FSSA databases (*Oracle 8.0.5* and *Informix 7.0*). For example, TOP database contains sensitive information about foster families (e.g., household income). To preserve privacy of such information, operation invocations are intercepted by a *Privacy Preserving Processor*. The *Privacy Preserving Processor* is based on *privacy credentials*, *privacy profiles*, and *data filters* (Section 1.4.3). Sensitive information is returned *only* to authorized users.

1.4.5 A Scenario Tour

We present a scenario that illustrates the main features of *WebDG*. A demo of *WebDG* is available online at <http://www.nvc.cs.vt.edu/~dgv>. We consider the case of a pregnant teen *Mary* visiting case officer *John* to collect social benefits to which she is entitled. *Mary* would like to apply for a

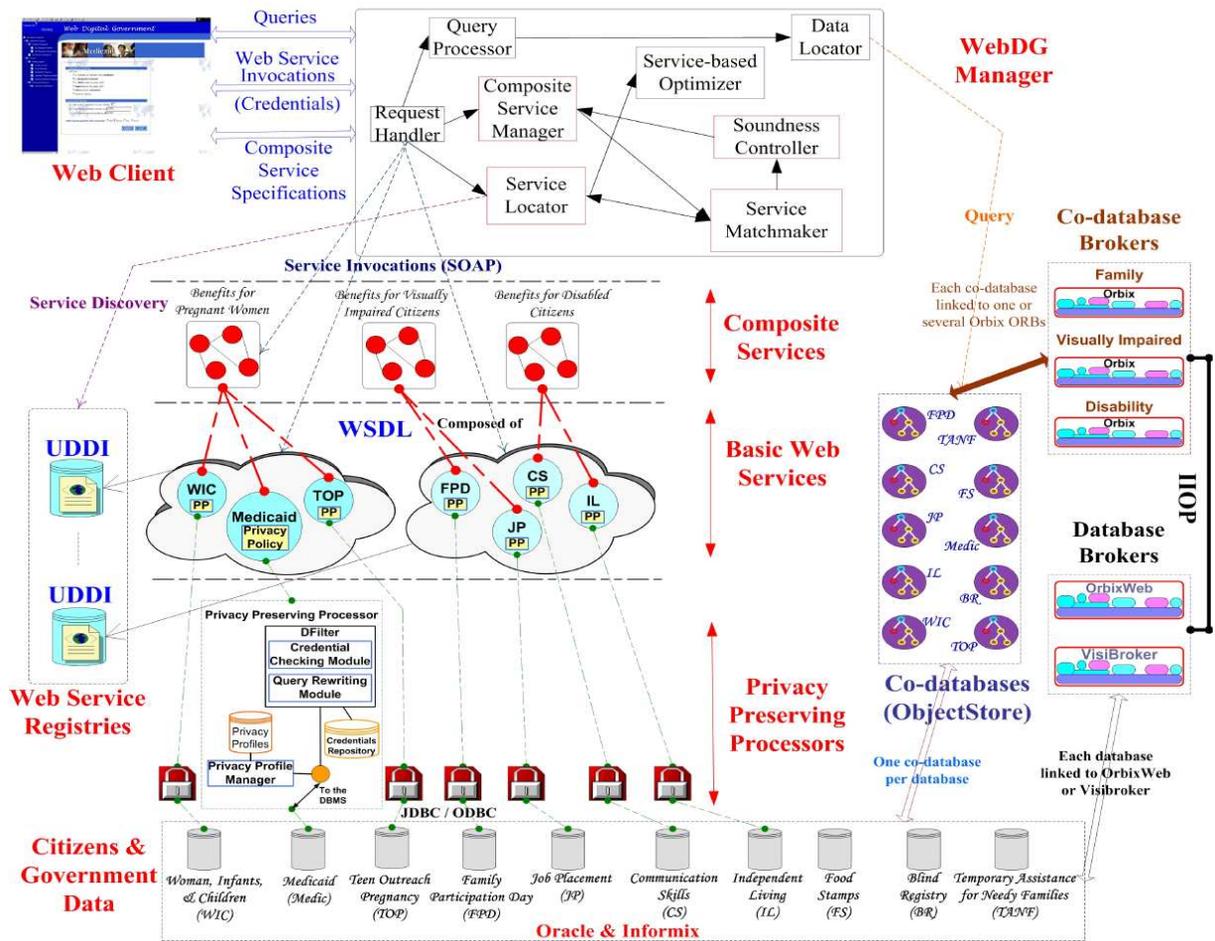


FIGURE 1.3 WebDG Architecture

government funded health insurance program. She also needs to consult a nutritionist to maintain an appropriate diet during her pregnancy. As *Mary* will not be able to take care of the future newborn, she is interested in finding a foster family. The fulfillment of *Mary's* needs requires accessing different services scattered in and outside the local agency. For that purpose, *John* may either look for simple (non composite) Web services that fit specific *Mary's* needs or specify all those needs through one single composite service called Pregnancy Benefits (PB):

- **Step 1: Web Service Discovery** – To locate a specific Web service, *John* could provide either the service name, if known, or properties. This is achieved by selecting the “By Program Name” or “By Program Properties” nodes respectively. *WebDG* currently supports two properties: *Category* and *Agency*. Assume *John* is interested in a service that provides help in finding foster families. He would select the *adoption* and *pregnancy* categories and the *Division of Family and Children* agency. *WebDG* would return the Teen Outreach Pregnancy (TOP) service. TOP offers childbirth and postpartum educational support for pregnant teens.
- **Step 2: Privacy Preserving Invocation** – Assume case worker *John* wants to use TOP service. For that purpose, he clicks on the service name. *WebDG* would return the list of oper-

ations offered by TOP service. As *Mary* is looking for a foster family, *John* would select the `Search Family Adoption` operation. This operation returns information about foster families in a given state (Virginia for example). The value “No right” (for the attribute “Race”) means that *Mary* does not have right to access information about the race of family F1. The value “Not Accessible” (for the attribute “Household Income”) means that family F1 does not want to disclose information about its income.

- **Step 3: Composing Web Services** – *John* would select the “Advanced Programs” node to specify the `Pregnancy Benefits` (PB) composite service. He would give the list of operations to be outsourced by PB without referring to any pre-existing service. Examples of such operations include `Find Available Nutritionist`, `Find PCP Providers` (which looks for primary care providers), and `Find Pregnancy Mentors`. After checking compositability rules, *WebDG* would return *composition plans* that conform to BP specification. Each plan has an ID (number), a graphical description, and a ranking. The ranking gives an approximation about the relevance of the corresponding plan. *John* would click on the plan’s ID to display the list of outsourced services. In our scenario, WIC (a federally funded food program for Women, Infants, and Children), Medicaid (a health-care program for low income citizens and families), and TOP services would be outsourced by PB.

1.5 Conclusion

In this chapter, we presented our experience in developing Digital Government (DG) infrastructures. We first gave a brief history of Digital Government. We then presented some major DG applications. This is followed by a discussion of some key issues and technical challenges in developing DG applications. The second part of the chapter is a description of our experimental DG infrastructure called *WebDG*. During the development of *WebDG*, we implemented and evaluated a number of novel ideas in deploying DG infrastructures. The system is built around two key concepts: *distributed ontologies* and *Web services*. The ontological approach was used to organize government databases. Web services were used as *wrappers* that enable access to and interoperability amongst government services. The system uses emerging standards for the description (WSDL), discovery (UDDI), and invocation (SOAP) of e-government services. The system also provides a mechanism that enforces the privacy of citizens when interacting with DG applications.

Acknowledgment. This research is supported by the National Science Foundation under grant 9983249-EIA and by a grant from the Commonwealth Information Security Center (CISC).

References

- N. Adam, O. Dogramaci, A. Gangopadhyay, and Y. Yesha. *Electronic Commerce: Technical, Business, and Legal Issues*. Prentice Hall, 1998.
- J. L. Ambite, Y. Arens, and W. Bourne et al. Data Integration and Access. In A. K. Elmagarmid and W. J. McIver, editor, *Advances in Digital Government: Technology, Human Factors, and Policy*, pages 85–106. Kluwer Academic Publishers, 2002.
- A. Ankolekar, M. Burstein, J. Hobbs, O. Lassila, D. Martin, S. McIlraith, S. Narayanan, M. Paolucci, T. Payne, K. Sycara, and H. Zeng. DAML-S: Semantic Markup for Web Services. In *Proceedings of the International Semantic Web Working Symposium (SWWS)*, July 30-August 1 2001.
- Baltimore Technologies. Baltimore E-Government Solutions: E-Tax Framework. *White Paper*, <http://www.baltimore.com/government/>, 2001.
- W. T. Bell. Experience of GIS in the Water Industry. In *Proc. of the IEE Colloquium on Experience in the Use of Geographic Information Systems in the Electricity Supply Industry*, May 27 1993.
- T. Berners-Lee. *Services and Semantics: Web Architecture*. <http://www.w3.org/2001/04/30-tbl>, 2001.
- A. Bouguettaya, A. Elmagarmid, B. Medjahed, and M. Ouzzani. Ontology-based Support for Digital Government. In *Proc. of VLDB 2001*, Roma, Italy, September 2001a.
- A. Bouguettaya, M. Ouzzani, B. Medjahed, and J. Cameron. Managing Government Databases. *Computer*, 34(2), February 2001b.
- A. Bouguettaya, M. Ouzzani, B. Medjahed, and A. K. Elmagarmid. Supporting Data and Services Access in Digital Government Environments. In A. K. Elmagarmid and W. J. McIver, editor, *Advances in Digital Government: Technology, Human Factors, and Policy*, pages 37–52. Kluwer Academic Publishers, 2002.
- U.S. Congress. *Health Insurance Portability and Accountability Act*, 1996.
- L. F. Cranor. Electronic Voting. *ACM Crossroads Student Magazine*, January 1996.
- Sharon S. Dawes, Peter A. Bloniarz, Kristine L. Kelly, and Patricia D. Fletcher. Some Assembly Required: Building a Digital Government for the 21st Century. *ACM Crossroads Student Magazine*, March 1999.
- A. K. Elmagarmid and W. J. McIver, editors. *Advances in Digital Government: Technology, Human Factors, and Policy*. Kluwer Academic Publishers, 2002.
- A. F. Karr et al. Web-Based System that Disseminate Information from Databases but Protect Confidentiality. In A. K. Elmagarmid and W. J. McIver, editor, *Advances in Digital Government: Technology, Human Factors, and Policy*, pages 181–196. Kluwer Academic Publishers, 2002.
- FEA Working Group. E-Gov Enterprise Architecture Guidance (Common Reference Model). July 2002.

- U.S. Geological Survey. Geographic Information Systems. 2002.
- L. Golubchik. Scalable Data Applications for Internet-based Digital Government Applications. In A. K. Elmagarmid and W. J. McIver, editor, *Advances in Digital Government: Technology, Human Factors, and Policy*, pages 107–119. Kluwer Academic Publishers, 2002.
- IBM, Microsoft, and Verisign. *Web Services Security (WS-Security)*, <http://www-106.ibm.com/developerworks/webservices/library/ws-secure/>, April 2002.
- J. Kessler. The French Minitel: Is There Digital Life Outside of the "US ASCII" Internet? A Challenge or Convergence? *D-Lib Magazine*, December 1995.
- A. Macintosh, A. Malina, and S. Farrell. Digital Democracy through Electronic Petitioning. In A. K. Elmagarmid and W. J. McIver, editor, *Advances in Digital Government: Technology, Human Factors, and Policy*, pages 137–148. Kluwer Academic Publishers, 2002.
- R. McDonnell and K. Kemp. *International GIS Dictionary*. Wiley Publishers, February 1996. ISBN 0-470-23607-8.
- B. Medjahed, B. Benatallah, A. Bouguettaya, A. H. H. Ngu, and A. Elmagarmid. Business-to-Business Interactions: Issues and Enabling Technologies. *The VLDB Journal (to appear)*, 2003.
- Y. Mu and V. Varadharajan. Anonymous Secure E-Voting over a Network. In *Proc. of the 14th Annual Computer Security Applications Conference*, December 7-11 1998.
- S. Nora and A. Minc. L'informatisation de la Société. *A Report to the President of France*, 1978.
- OASIS. *eXtensible Access Control Markup Language*, <http://www.oasis-open.org/committees/xacml/>, 2001.
- OASIS. *Security Assertion Markup Language*, <http://www.oasis-open.org/committees/security/>, 2002.
- M. Ouzzani, B. Benatallah, and A. Bouguettaya. Ontological Approach for Information Discovery in Internet Databases. *Distributed and Parallel Databases*, 8(3), July 2000.
- J. Pecth. GIS in the Electricity Supply Industry: An Overview. In *Proc. of the IEE Colloquium on Experience in the Use of Geographic Information Systems in the Electricity Supply Industry*, May 27 1993.
- Indrajit Ray, Indrakshi Ray, and Natarajan Narasimhamurthi. An Anonymous Electronic Voting Protocol for Voting Over The Internet. In *Proc. of the 3rd International Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems (WECWIS'01)*, June 21-22 2001.
- A. Rezgoui, M. Ouzzani, A. Bouguettaya, and B. Medjahed. Preserving Privacy in Web Services. In *Proc. of the 4th ACM Workshop on Information and Data Management (WIDM'02)*, pages 56–62, November 2002.
- A. D. Rubin. Security Considerations for Remote Electronic Voting. *Communications of the ACM*, 45(12), December 2002.
- M. Sabbouh, S. Jolly, D. Allen, P. Silvey, and P. Denning. Interoperability. In *W3C Web Services Workshop*, April 11-12 2001.
- David Trastour, Claudio Bartolini, and Javier Gonzalez-Castillo. A Semantic Web Approach to Service Description for Matchmaking of Services. In *Proc. of the International Semantic Web Working Symposium (SWWS)*, July 30 - August 1 2001.
- S. Tsur, S. Abiteboul, R. Agrawal, U. Dayal, J. Klein, and G. Weikum. Are Web Services the Next

- Revolution in e-Commerce? (Panel). In *VLDB Conference*, September 2001.
- U.S. PITAC. Transforming Access to Government Through Information Technology. *Report to the President*, September 2000.
- S. Vinoski. Web Services Interaction Models, Part 1: Current Practice. *IEEE Internet Computing*, 6(3):89–91, February 2002a.
- S. Vinoski. Where is Middleware? *IEEE Internet Computing*, 6(2), March 2002b.
- W3C. *Sematic Web*, <http://www.w3c.org/2001/sw/>, 2001a.
- W3C. *SOAP Security Extensions: Digital Signature*, <http://www.w3.org/TR/SOAP-dsig>, 2001b.
- W3C. *XML Encryption*, <http://www.w3.org/Encryption>, 2001c.
- W3C. *XML Key Management Specification (XKMS)*, <http://www.w3.org/TR/xkms/>, 2001d.
- W3C. *XML Signature*, <http://www.w3.org/Signature/>, 2001e.
- D. M. West. Urban E-Government. *Center for Public Policy, Brown University, USA*, September 2002.
- WSFL. *Web Services Flow Language*, <http://xml.coverpages.org/wsfl.html>.
- S. V Wunnava and M. V. Reddy. Adaptive and Dynamic Service Composition in eFlow. In *Proc. of the IEEE Southeastcon*, pages 205–208, April 2000.
- XLANG. <http://www.coverpages.org/xlang.html>.