JOURNAL OF EUROPEAN ECONOMY
Vol. 2 (№ 3). September 2003
Publication of Ternopil Academy of National Economy

Market of financial and banking services

Alexander ELSAS

E-GOVERNMENT: INTEGRATION WITH WEB SERVICES

Abstract

This paper introduces the concept of Web services as a way to realize integration between distributed applications. These applications can be seen in the context of e-government or e-business, as integration on an economic level has to be supported and reached in the area of business relations and in the area of administrative workflows. Therefore, an introduction into the basic concepts of Web services is given, the main concepts and challenges in the area of e-government are described with the aim to show the integrative aspects and chances of Web services.

Key words:

E-government, integration, web services, SOAP, UDDI, WSDL, XML.

1. Introduction

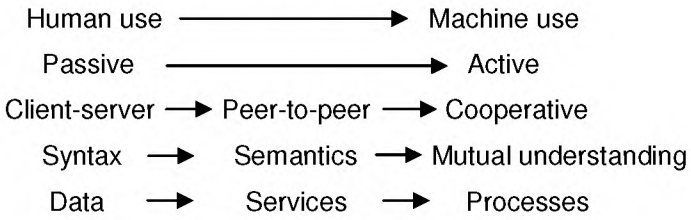
With the popularity of the Internet and especially the World Wide Web (WWW) for realizing electronic business and electronic commerce, there is a growing need of standardization of communication between different applications to allow for interoperability and communication. This need arises from the

Elsas Alexander, Johann Wolfgang Goethe-University, Frankfurt am Main, Germany.

predicted and observed development of the usage of the Internet and the WWW in the direction of automated, cooperative processes, as shown in figure 1[1].

Figure 1.

Trends in the use of the WWW



Source: Huhns / Buell, Trusted Autonomy, 2002, p. 93.

Over the past two years the concept of *Web services* has evolved into a standard way of communication between applications, enabling interaction between previously separated and isolated applications, even the so-called legacy applications. Since the problem of integration is especially relevant in the area of electronic government, this paper proposes the implementation of Web service based e-government applications to allow for easier integration, even in the context of European integration.

Chapter 2 gives a short introduction to e-government, chapter 3 describes the basic concepts of Web services. Chapter 4 shows the possibilities of integrating e-government applications with Web services; the final chapter summarizes the findings.

2. E-Government

Following the adoption of the World Wide Web for commercial applications, the governmental organizations in all European countries are trying to take advantage of this way of interaction and communication. Over the past few years e-government related projects have been in the process of development.

The degree of sophistication of those projects can be categorized into four levels[2]:

1. Static web sites providing information.

[1] See Huhns / Buell, Trusted Autonomy, 2002, p. 92.
[2] See Elsas, E-Government, 2000.

2. Web sites with interaction.
3. Support of internal and external workflows of the administration.
4. Online voting.

Another look at the dimensions of e-government can be achieved by categorizing the involved partners:

- **Government-to-government (G2G) or administration-to-administration (A2A)**: internal processes of the administration.
- **Government-to-business (G2B) or administration-to-business (A2B)**: interactions between governmental organizations and commercial or business organizations, e.g. taxes.
- **Government-to-citizen (G2C) or administration-to-citizen (A2C)**: interactions between governmental organizations and citizens, e. g. marriage.

Most western European countries are engaged in various e-government projects on different levels of administration and sophistication. The reasons for the implementation of e-government applications are based on efficiency and cost developments. Most projects make use of proprietary, often self-developed, standards for communication and interaction[3]. The problem is a lack of coordination and joint efforts between the different projects, which might lead to incompatibilities and integration problems, e. g. with so-called *one-stop-government* portals. This is a single website that acts as a citizen portal, giving access to information originating from different sources or allowing transactions with different local, regional or federal or even foreign authorities[4].

3. Web Services

3.1. Definition

«Over the past few years, businesses have interacted using ad hoc approaches that take advantage of the basic Internet infrastructure. Now, however, Web services are emerging to provide a systematic and extensible framework for application-to-application interaction, built on top of existing Web protocols and based on open XML standards»[5].

[3] See e. g. Elsas, E-Government, 2001.
[4] See BundOnline, One Stop Government, 2002.
[5] Curbera et al, Introduction, 2002, p. 86.

The relationship between Web services and XML is so close (that often the term *XML Web services* is used) to differentiate from other proprietary usages of the term Web services. The main goal of Web services is to achieve application-to-application interaction or communication; therefore, simple retrieval of web pages with a browser can only be seen as a Web service in a very broad sense.

The *World Wide Web Consortium* (W3C)[6] defines a Web service as a software application «whose interfaces and bindings are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via internet-based protocols»[7].

In contrast to traditional IT systems, which can be characterized as tightly coupled systems, Web services implement a loosely coupled approach[8]. A Web services framework consists of three basic services: *communication*, *service description,* and *service discovery*. These basic functionalities have to be implemented by protocols, the three basic standards for these protocols are *SOAP*, *WSDL* and *UDDI*, all based on the common XML meta language[9]. Figure 2 gives a graphical overview over the interaction concept of these protocols, which are briefly described below.

This architecture is often referred to as a *Service Oriented Architecture* (SOA), in the case of Web services, this SOA-approach is implemented with the above mentioned protocols[10]:

- A specific Web service announces its WSDL definition to a UDDI registry (publish).
- A client searches the registry for a service's definition (find).
- The client sends messages or requests directly to the service via SOAP, based on the information of the WSDL definition from the directory.

3.2. XML

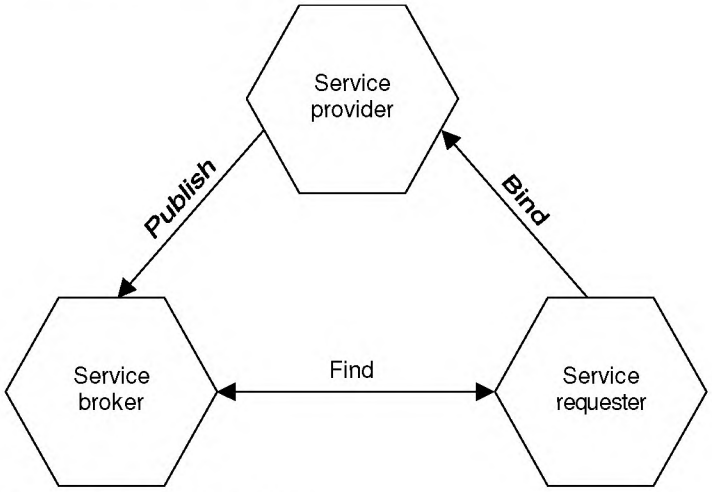
With the success of the *Hypertext Markup Language* (HTML) as basis for the WWW, the so-called *markup languages* became more and more prominent. While HTML describes how a web page should look, it does not give information about the structure and meaning of the data.

---

[6] For a starting point into the various Web services related activities of the W3C see W3C, Web Services, 2002.
[7] Austin et al, Web Services Architecture Requirements, 2002.
[8] See IBM, Overview, 2001.
[9] See Curbera et al, Introduction, 2002, p. 86.
[10] See Vinoski, Interaction Models, 2002, p. 89 – 90.

Figure 2.

Web services concept



Source: IBM, Overview, 2000.

The *Extensible Markup Language* (XML) separates between *content* and *presentation of the content* and allows the definition of markup languages for specific scenarios[11]. So XML is not a language of its own, rather it can be described as a *meta language* or a tool to develop other languages.

Figure 3 shows the difference between HTML and XML: HTML defines the presentation of the words *Alexander* and *Elsas* in a bold typeface, XML defines *Alexander Elsas* as a name.

Figure 3.

HTML vs. XM

HTML:	Alexander Elsas
XML:	<name> Alexander Elsas </name>

[11] For a more detailed introduction into XML and the related standards see (as a starting point) Bos, XML, 1999.

The vocabulary for such a specific markup language can be defined in a *Document Type Definition* (DTD). An alternative to using a DTD is to implement an *XML Schema*, which also describes the structure of an XML document. The XML Schema language is often referred as an *XML Schema Definition* (XSD)[12].

The Web service protocols are defined in XML: they are specific markup languages, defined in XML.

3.3. SOAP

The *Simple Object Access Protocol* (SOAP)[13] is a joint development of Microsoft and IBM and a few other companies, whose further development is in the hands of the W3C[14]. It is a protocol for messaging and remote procedure calls (RPCs)[15] based on XML. SOAP does not define a new transport protocol, it takes advantage of existing Internet transport protocols like HTTP and SMTP[16]. So you can say that SOAP is the link within the Web service concept to the existing Internet infrastructure.

SOAP defines how applications can communicate[17]:

- The communication is message-oriented, meaning that applications communicate with each other by exchanging textual messages, the format for these messages is specified within SOAP.
- SOAP message has a simple structure: an XML element with two child elements, the *header* and the *body*.
- SOAP messages include information for the recipient about how messages should be processed.
- Additional information about who should process a message (the *actor*) is also included in the SOAP message format.
- RPCs, the execution of programs or program fragments (procedures) on remote computers, can be realized with the SOAP protocol.

Figure 4 shows a typical simplified SOAP message defining an electronic ticket for air travel.

---

[12] See W3schools, XML Schema, 2002 for a detailed introduction.
[13] See Mitra, SOAP Primer, 2002.
[14] See Box, SOAP, 2001.
[15] For an introduction into the concepts of RPCs see Webopedia, RPC, 2002.
[16] The Hypertext Transfer Protocol (HTTP) and the Simple Mail Transport Protocol (SMTP) are the foundations of the WWW, respectively the Internet-based email communication. They are part of the so-called TCP/IP protocol suite. For further details see Webopedia, HTTP, 2002 and Webopedia, SMTP, 2002.
[17] See Curbera et al, Introduction, 2002, p. 86–87.

Figure 4.

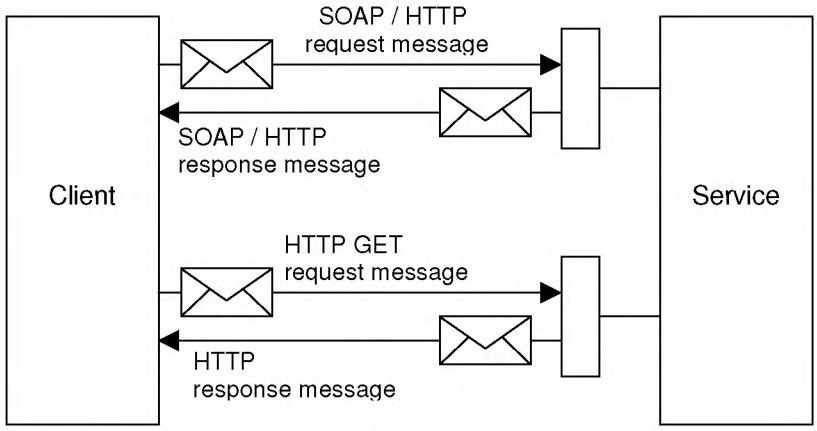
SOAP message for an electronic ticket

```
POST /travelservice
SOAPAction: «http://www.cybertravel.com/checkin»
Content-Type: text/xml: charset=«utf-8»
Content-Length: nnnn

<SOAP: Envelope xmlns:SOAP=«http://schemas.xmlsoap.org/soap/envelope/»>
    <SOAP:Body>
        <et:eTicket xmlns:et=«http://www.cybertravel.com/eticket/schema»>
        <et:passengerName first=«Alexander» last=«Elsas»/>
        <et:flightInfo
                segment=«FRA-SIP»
                airline=«PS»
                class=«L»
                flight=«408»
                departureDate=«2002-09-14»
                departureTime=«1110»
                arrivalDate=«2002-09-14»
                arrivalTime=«1515» />
        </et:eTicket>
    </SOAP:Body>
</SOAP:Envelope>
```

Figure 5.

Invoking Web service



Source: Shohoud, Introduction, 2002.

Figure 5 shows Web service access via SOAP. The client can invoke the Web service in two different ways (SOAP and HTTP), each invocation consists of a request message and a response message[18].

3.4. WSDL

The *Web Services Description Language* (WSDL) describes Web services as a collection of end points for communication that can exchange messages. A complete service description consists of two parts of information[19]:

- The *abstract interface* is a service description on the application-level. This interface has three main components: the *vocabulary*, the *message* and the *interaction*.
  - Vocabulary: Most Web services use XSD as a standard for the data types[20].
  - Messages: They are formed from parts of the vocabulary. Input and output messages together form an *operation*, a collection of operations at an end point form a *portType*.
  - Interactions: They are defined by the operations and portType elements.
- The *protocol-specific details* have to be followed to gain access to services at the end points. These so-called *bindings* describe *what* communication protocol to use, *how* individual service interaction over this protocol is achieved and *where* (network adress) the communication terminates.

Figure 6 adds the introduced WSDL concepts to the example from Figure 4.

3.5. UDDI

The *Universal Description, Discovery, and Integration* (UDDI) can be seen as an online, automated «phone» directory of Web services. The UDDI registry consists of two basic specifications[21]:

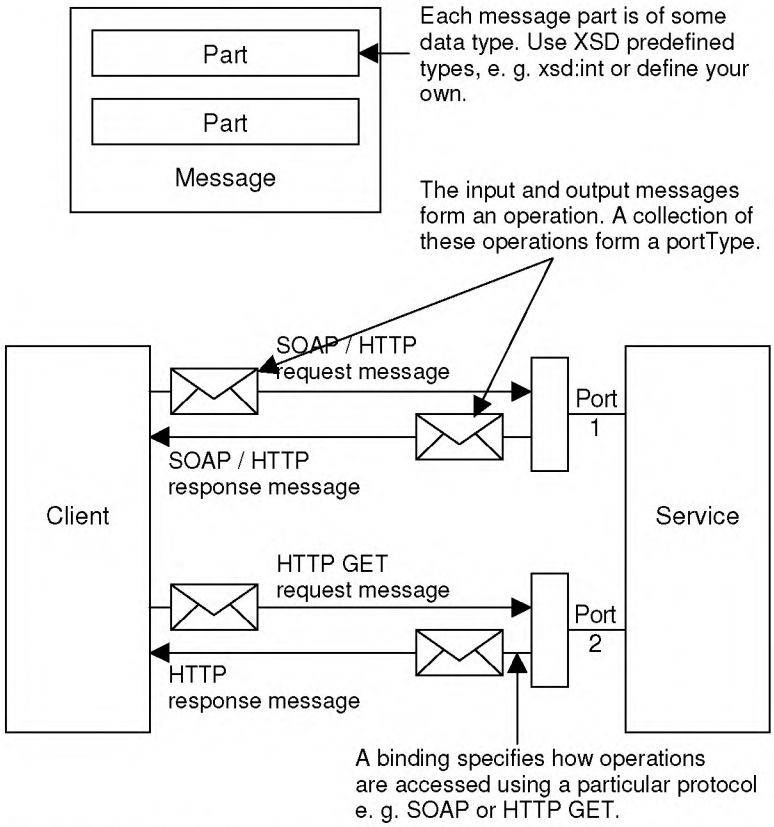
- **Registry structure**: which information has to be provided about a Web service to the registry and how it is encoded.
- **Registry operation**: an application programming interface (API) which allows queries and updates of the registry.

---

[18] See Shohoud, Introduction, 2002.
[19] See Curbera et al, Introduction, 2002, p. 88 -89 and Weerawarana et al, WSDL, 2002.
[20] See Biron / Malhotra, XML Schema, 2001 for a reference to the XSD data types.
[21] See UDDI.org, UDDI, 2000 and UDDI.org, Overview, 2000 and Curbera et al, Introduction, 2002, p. 89 – 91.

Figure 6.

WSDL terminology for Web services



Source: Shohoud, Introduction, 2002.

Three types of information about Web services are encoded in the UDDI registry:

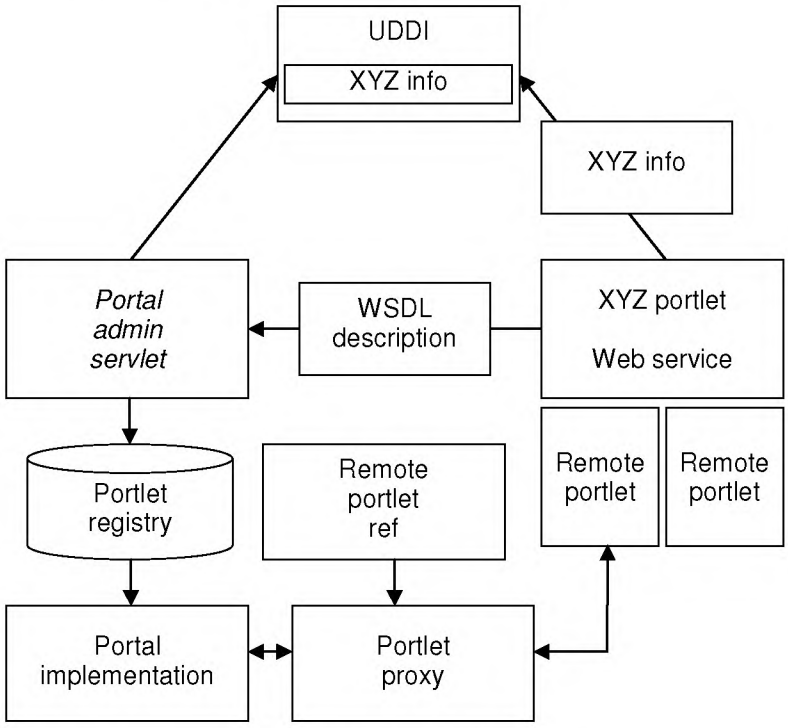
- Names and contact details (*white pages*),
- A categorization of business and service types (*yellow pages*),
- Technical details (*green pages*).

4. E-Government with Web Services

The realization of a citizen portal as a one-stop-government solution is a typical scenario that demonstrates the advantages of using the Web services concept. Web services allow for easy integration of different applications and information sources into one unified portal. A so-called *portlet* is a content container: basically the user's view of the customized content that shall be integrated into the portal[22]. A portal taking advantage of the Web services concept integrates portlets from different content providers into a single portal. Figure 7 gives an example how a portal finds and integrates remote portlets.

Figure 7.

Integration with Web services



Source: Wege, Portal Server Technology, 2002, c. 76.

[22] See Wege, Portal Server Technology, 2002 for an introduction into the concepts.

5. Conclusions

This paper described the basic concepts of Web services and the main related protocols SOAP, WSDL, and UDDI. The implementation of e-government applications which are based on a Web service framework has advantages in interoperability and represents the actual state of the art in the development of distributed systems. While non Web service based (proprietary) e-government applications have been developed over the last few years, most of them are still in an experimental or prototype stage.

New e-government applications should be based on common standards to allow for easier integration with other applications. The issue becomes even more relevant under the aspects of European integration. The number of existing e-government applications is still small, therefore upgrading them to a Web service based framework is not a problematic issue at the moment. The experience gained with the existing applications should go directly into the next generation, which should be based on Web service standards.

This conclusion can also be drawn from recent research activities:

- The German Bertelsmann Foundation has formulated a recommendation for action to establish successful e-government projects which is based on recent best-practice study[23]: An important point within the recommendations is the use of established standards.
- The SAGA project in Germany comes to conclusion that the Web service concept and its related technologies, standards and products are an appropriate basis for integration of different administrative services[24].

The challenge for the forthcoming e-government applications is to learn from the e-commerce experience of the last five years and to avoid the apparent shortcomings. Founding the applications on the Web services framework promises to be a feasible solution. By adopting the Web services technology integration on a European scale can be supported.

[23] See Friedrichs et al, 10-Punkte-Plan, 2002 and Bertelsmann, Balanced E-Government, 2002.
[24] See BundOnline, Web Services, 2002.

Bibliography

1. **Austin et al, Web Services Architecture Requirements, 2002.** Austin, Daniel; Barbir, Abbie; Garg, Sharad (eds.): Web Services Architecture Requirements – W3C Working Draft 29 April 2002; URL: http://www.w3.org/TR/2002/WD-wsa-reqs-20020429 (6.8.2002)
2. **Bertelsmann, Balanced E-Government, 2002.** Bertelsmann Foundation (ed.): Balanced E-Government, 2002; URL: http://www.begix.de/en/ studie.download.html (7.9.2002)
3. **Biron / Malhotra, XML Schema, 2001.** Biron, Paul V.; Malhotra, Ashok (eds.): XML Schema Part: Datatypes – W3C Recommendation 02 May 2001; URL: http://www.w3.org/TR/xmlschema-2
4. **Bos, XMl, 1999.** Bos, Bert: XML in 10 points, 1999; URL:http://www.w3.org/ XML/1999/XML-in-10-points.html.en (6.8.2002)
5. **Box, SOAP, 2001.** Box, Don: A Brief History of SOAP, 2001; URL: http://www.xml.com/pub/a/2001/04/04/soap.html (6.8.2002)
6. **BundOnline, Web Services, 2002.** BundOnline 2005 (ed.): One Stop Governemt; URL: http://www.bund.de/BundOnline-2005/SAGA/Glossar-.6343.htm?suche=W* (7.9.2002)
7. **Curbera et al, Introduction, 2002.** Curbera, Francisco; Duftler, Matthew; Khalaf, Rania; Nagy, William; Mukhi, Nirmal. Weerawarana, Sanjiva: Unraveling the Web Services Web – An Introduction to SOAP, WSDL, and UDDI; in IEEE Internet computing, March / April 2002, c. 86 – 93
8. **Elsas, E-Government, 2002.** Elsas, Alexander: E-Government in Deutschland; in: Schweighofer / Menzel (eds.): E-Commerce und E-Government, Wien 2000; URL: http://elsas.de/papers/e-gov.pdf (6.8.2002)
9. **Elsas, E-Government, 2001.** Elsas, Alexander: E-Government in Deutschland – Version 2.001; Schweighofer / Menzel / Kreuzbauer (eds.): Auf dem Weg zur ePerson – Aktuelle Fragestellungen der Rechtsinformatik 2001, Wien 2001; URL: http://elsas.de/papers/egov2001.pdf (30.8.2002)
10. **Friedrichs et al, 10-Punkte-Plan, 2002.** Friedrichs, Stefan; Hart, Thomas; Welzel, Carolin: 10-Punkte-Plan für gutes EßGovernment – Ein Fahrplan zur Verwaltungsmodernisierung und Stärkung der Bürgergesellschaft, 2002; URL: http://www.begix.de/10_Punkte_Plan.pdf (7.9.2002)

11. **Huhns / Buell, Trusted Autonomy, 2002.** Huhns, Michael N.; Buell, Duncan A.: Trusted Autonomy; in: IEEE Internet Computing, May / June 2002, c. 92 – 95

12. **IBM, Overview, 2002.** IBM (ed.): Web Services Architecture Overview, 2000; URL: http://www-106.ibm.com/developerworks/webservices/library/ws-arc1/ (28.8.2002)

13. **Mitra, SOAP Primer, 2002.** Mitra, Nilo (ed.): SOAP Version 1.2 Part 0: Primer, 2002; URL:

14. http://www.w3.org/TR/soap12-part0/ (26.8.2002)

15. **Shohoud, Introduction, 2002.** Shohoud. Yasser: Introduction to WSDL; URL: http://www.learnxmlws.com/tutors/wsdl.aspx (7.8.2002)

16. **UDDI.org, UDDI, 2000.** UDDI.org (eds.): UDDI Technical White Paper, 2000; URL: http://uuddi.orf/pubs/IRU_UDDI_Technical_White_Paper.pdf (7.8.2002)

17. **Vinoski, Interaction Models. 2002.** Vinoski, Steve: Web Services Interaction Models – Part I: Current Practice; in: IEEE Internet Computing, May / June 2002, c. 89 – 91

18. **W3schools, XML Schema, 2002.** W3schools (eds.): XML Schema Tutorial; URL: http://www.w3schools.com/schema (7.8.2002)

19. **W3C, Web Services, 2002.** W3C (eds.): Web Services Activity; URL: http://www.w3.org/2002/ws (26.8.2002)

20. **Webopedia, HTTP, 2002.** Webopedia (eds.): HTTP; URL: http://www.webopedia.com/TERM/H/HTTP.html (24.8.2002)

21. **Webopedia, RPC, 2002.** Webopedia (eds.): PRC; URL: http://www.webopedia.com/TERM/R/RPC.html (24.8.2002)

22. **Webopedia, SMPT, 2002.** Webopedia (eds.): SMPT; URL: http://www.webopedia.com/TERM/S/SMPT.html (24.8.2002)

23. **Weeravarana et al, WSDL, 2002.** Weeravarana, Sanjiva; Chinnici, Roberto; Gudgin, Martin; Moreau, Jean-Jaques (eds.): Web Services Description Language (WSDL) Version 1.2 – W3C Working Draft 9 July 2002; URL: http://www.w3.org/TR/wsdl12 (6.8.2002)

24. **Wege, Portal Sever Technology, 2002.** Wege, Christian: Portal Sever Technology; in: IEEE INTERNET Comeuting, May / June 2002, c.73 – 77.

Recommended Reading

www.ws-i.org	Web Services Interoperability Organization
www.w3.org	World Wide Web Consortuim
www.xmethds.net	A collection of free Web services
www.webservicesarchtect.com	Web services
www.xml.org	XML
www.webopedia.com	Comprehensive online encyclopedia
www.learmxmlws.com	XML
www.w3schools.com	Tutorials
www.begix.de	Balanced E-Government

The article was received in June 2003