Fabio Casati
Dimitrios Georgakopoulos
Ming-Chien Shan (Eds.)

LNCS 2193

Technologies for E-Services

Second International Workshop, TES 2001
Rome, Italy, September 2001
Proceedings

Springer

Fabio Casati Dimitrios Georgakopoulos
Ming-Chien Shan (Eds.)

Technologies for E-Services

Second International Workshop, TES 2001
Rome, Italy, September 14-15, 2001
Proceedings

Springer

Volume Editors

Fabio Casati
Ming-Chien Shan
Hewlett-Packard
1501 Page Mill Road, 1 U-4, Palo Alto, CA 94304, USA
E-mail: {casati, shan}@hpl.hp.com

Dimitrios Georgakopoulos
Telcordia Technologies
106 E. Sixth Street, Littlefield Building 415, Austin, TX 78701, USA
E-mail: dimitris@research.telcordia.com

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Technologies for E-services : second international workshop ; proceedings / TES 2001, Rome, Italy, September 14 - 15, 2001. Fabio Casati ... (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2193)
ISBN 3-540-42565-9

CR Subject Classification (1998): H.2, H.4, C.2, H.3, J.1, K.4.4, I.2.11

ISSN 0302-9743
ISBN 3-540-42565-9 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by Christian Grosche, Hamburg
Printed on acid-free paper SPIN: 10840622 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2193
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Tokyo

Lecture Notes in Computer Science

For information about Vols. 1–2084
please contact your bookseller or Springer-Verlag

Vol. 1982: S. Näher, D. Wagner (Eds.), Algorithm Engineering. Proceedings, 2000. VIII, 243 pages. 2001.

Vol. 1996: P.L. Lanzi, W. Stolzmann, S.W. Wilson (Eds.), Advances in Learning Classifier Systems. Proceedings, 2000. VIII, 273 pages. 2001. (Subseries LNAI).

Vol. 2085: J. Mira, A. Prieto (Eds.), Bio-Inspired Applications of Connectionism. Proceedings, 2001. Part II. XXVII, 848 pages. 2001.

Vol. 2086: M. Luck, V. Mařík, O. Stěpánková, R. Trappl (Eds.), Multi-Agent Systems and Applications. Proceedings, 2001. X, 437 pages. 2001. (Subseries LNAI).

Vol. 2087: G. Kern-Isberner, Conditionals in Nonmonotonic Reasoning and Belief Revision. X, 190 pages. 2001. (Subseries LNAI).

Vol. 2088: S. Yu, A. Păun (Eds.), Implementation and Application of Automata. Proceedings, 2000. XI, 343 pages. 2001.

Vol. 2089: A. Amir, G.M. Landau (Eds.), Combinatorial Pattern Matching. Proceedings, 2001. VIII, 273 pages. 2001.

Vol. 2090: E. Brinksma, H. Hermanns, J.-P. Katoen (Eds.), Lectures on Formal Methods and Performance Analysis. Proceedings, 2000. VII, 431 pages. 2001.

Vol. 2091: J. Bigun, F. Smeraldi (Eds.), Audio- and Video-Based Biometric Person Authentication. Proceedings, 2001. XIII, 374 pages. 2001.

Vol. 2092: L. Wolf, D. Hutchison, R. Steinmetz (Eds.), Quality of Service – IWQoS 2001. Proceedings, 2001. XII, 435 pages. 2001.

Vol. 2093: P. Lorenz (Ed.), Networking – ICN 2001. Proceedings, 2001. Part I. XXV, 843 pages. 2001.

Vol. 2094: P. Lorenz (Ed.), Networking – ICN 2001. Proceedings, 2001. Part II. XXV, 899 pages. 2001.

Vol. 2095: B. Schiele, G. Sagerer (Eds.), Computer Vision Systems. Proceedings, 2001. X, 313 pages. 2001.

Vol. 2096: J. Kittler, F. Roli (Eds.), Multiple Classifier Systems. Proceedings, 2001. XII, 456 pages. 2001.

Vol. 2097: B. Read (Ed.), Advances in Databases. Proceedings, 2001. X, 219 pages. 2001.

Vol. 2098: J. Akiyama, M. Kano, M. Urabe (Eds.), Discrete and Computational Geometry. Proceedings, 2000. XI, 381 pages. 2001.

Vol. 2099: P. de Groote, G. Morrill, C. Retoré (Eds.), Logical Aspects of Computational Linguistics. Proceedings, 2001. VIII, 311 pages. 2001. (Subseries LNAI).

Vol. 2100: R. Küsters, Non-Standard Inferences in Description Logocs. X, 250 pages. 2001. (Subseries LNAI).

Vol. 2101: S. Quaglini, P. Barahona, S. Andreassen (Eds.), Artificial Intelligence in Medicine. Proceedings, 2001. XIV, 469 pages. 2001. (Subseries LNAI).

Vol. 2102: G. Berry, H. Comon, A. Finkel (Eds.), Computer-Aided Verification. Proceedings, 2001. XIII, 520 pages. 2001.

Vol. 2103: M. Hannebauer, J. Wendler, E. Pagello (Eds.), Balancing Reactivity and Social Deliberation in Multi-Agent Systems. VIII, 237 pages. 2001. (Subseries LNAI).

Vol. 2104: R. Eigenmann, M.J. Voss (Eds.), OpenMP Shared Memory Parallel Programming. Proceedings, 2001. X, 185 pages. 2001.

Vol. 2105: W. Kim, T.-W. Ling, Y-J. Lee, S.-S. Park (Eds.), The Human Society and the Internet. Proceedings, 2001. XVI, 470 pages. 2001.

Vol. 2106: M. Kerckhove (Ed.), Scale-Space and Morphology in Computer Vision. Proceedings, 2001. XI, 435 pages. 2001.

Vol. 2107: F.T. Chong, C. Kozyrakis, M. Oskin (Eds.), Intelligent Memory Systems. Proceedings, 2000. VIII, 193 pages. 2001.

Vol. 2108: J. Wang (Ed.), Computing and Combinatorics. Proceedings, 2001. XIII, 602 pages. 2001.

Vol. 2109: M. Bauer, P.J. Gymtrasiewicz, J. Vassileva (Eds.), User Modelind 2001. Proceedings, 2001. XIII, 318 pages. 2001. (Subseries LNAI).

Vol. 2110: B. Hertzberger, A. Hoekstra, R. Williams (Eds.), High-Performance Computing and Networking. Proceedings, 2001. XVII, 733 pages. 2001.

Vol. 2111: D. Helmbold, B. Williamson (Eds.), Computational Learning Theory. Proceedings, 2001. IX, 631 pages. 2001. (Subseries LNAI).

Vol. 2116: V. Akman, P. Bouquet, R. Thomason, R.A. Young (Eds.), Modeling and Using Context. Proceedings, 2001. XII, 472 pages. 2001. (Subseries LNAI).

Vol. 2117: M. Beynon, C.L. Nehaniv, K. Dautenhahn (Eds.), Cognitive Technology: Instruments of Mind. Proceedings, 2001. XV, 522 pages. 2001. (Subseries LNAI).

Vol. 2118: X.S. Wang, G. Yu, H. Lu (Eds.), Advances in Web-Age Information Management. Proceedings, 2001. XV, 418 pages. 2001.

Vol. 2119: V. Varadharajan, Y. Mu (Eds.), Information Security and Privacy. Proceedings, 2001. XI, 522 pages. 2001.

Vol. 2120: H.S. Delugach, G. Stumme (Eds.), Conceptual Structures: Broadening the Base. Proceedings, 2001. X, 377 pages. 2001. (Subseries LNAI).

Vol. 2121: C.S. Jensen, M. Schneider, B. Seeger, V.J. Tsotras (Eds.), Advances in Spatial and Temporal Databases. Proceedings, 2001. XI, 543 pages. 2001.

Vol. 2123: P. Perner (Ed.), Machine Learning and Data Mining in Pattern Recognition. Proceedings, 2001. XI, 363 pages. 2001. (Subseries LNAI).

Vol. 2124: W. Skarbek (Ed.), Computer Analysis of Images and Patterns. Proceedings, 2001. XV, 743 pages. 2001.

Vol. 2125: F. Dehne, J.-R. Sack, R. Tamassia (Eds.), Algorithms and Data Structures. Proceedings, 2001. XII, 484 pages. 2001.

Vol. 2126: P. Cousot (Ed.), Static Analysis. Proceedings, 2001. XI, 439 pages. 2001.

Vol. 2127: V. Malyshkin (Ed.), Parallel Computing Technologies. Proceedings, 2001. XII, 516 pages. 2001.

Vol. 2129: M. Goemans, K. Jansen, J.D.P. Rolim, L. Trevisan (Eds.), Approximation, Randomization, and Combinatorial Optimization. Proceedings, 2001. IX, 297 pages. 2001.

Vol. 2130: G. Dorffner, H. Bischof, K. Hornik (Eds.), Artificial Neural Networks – ICANN 2001. Proceedings, 2001. XXII, 1259 pages. 2001.

Vol. 2132: S.-T. Yuan, M. Yokoo (Eds.), Intelligent Agents. Specification. Modeling, and Application. Proceedings, 2001. X, 237 pages. 2001. (Subseries LNAI).

Vol. 2136: J. Sgall, A. Pultr, P. Kolman (Eds.), Mathematical Foundations of Computer Science 2001. Proceedings, 2001. XII, 716 pages. 2001.

Vol. 2138: R. Freivalds (Ed.), Fundamentals of Computation Theory. Proceedings, 2001. XIII, 542 pages. 2001.

Vol. 2139: J. Kilian (Ed.), Advances in Cryptology – CRYPTO 2001. Proceedings, 2001. XI, 599 pages. 2001.

Vol. 2141: G.S. Brodal, D. Frigioni, A. Marchetti-Spaccamela (Eds.), Algorithm Engineering. Proceedings, 2001. X, 199 pages. 2001.

Vol. 2142: L. Fribourg (Ed.), Computer Science Logic. Proceedings, 2001. XII, 615 pages. 2001.

Vol. 2143: S. Benferhat, P. Besnard (Eds.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. Proceedings, 2001. XIV, 818 pages. 2001. (Subseries LNAI).

Vol. 2146: J.H. Silverman (Eds.), Cryptography and Lattices. Proceedings, 2001. VII, 219 pages. 2001.

Vol. 2147: G. Brebner, R. Woods (Eds.), Field-Programmable Logic and Applications. Proceedings, 2001. XV, 665 pages. 2001.

Vol. 2149: O. Gascuel, B.M.E. Moret (Eds.), Algorithms in Bioinformatics. Proceedings, 2001. X, 307 pages. 2001.

Vol. 2150: R. Sakellariou, J. Keane, J. Gurd, L. Freeman (Eds.), Euro-Par 2001 Parallel Processing. Proceedings, 2001. XXX, 943 pages. 2001.

Vol. 2151: A. Caplinskas, J. Eder (Eds.), Advances in Databases and Information Systems. Proceedings, 2001. XIII, 381 pages. 2001.

Vol. 2152: R.J. Boulton, P.B. Jackson (Eds.), Theorem Proving in Higher Order Logics. Proceedings, 2001. X, 395 pages. 2001.

Vol. 2153: A.L. Buchsbaum, J. Snoeyink (Eds.), Algorithm Engineering and Experimentation. Proceedings, 2001. VIII, 231 pages. 2001.

Vol. 2154: K.G. Larsen, M. Nielsen (Eds.), CONCUR 2001 – Concurrency Theory. Proceedings, 2001. XI, 583 pages. 2001.

Vol. 2157: C. Rouveirol, M. Sebag (Eds.), Inductive Logic Programming. Proceedings, 2001. X, 261 pages. 2001. (Subseries LNAI).

Vol. 2158: D. Shepherd, J. Finney, L. Mathy, N. Race (Eds.), Interactive Distributed Multimedia Systems. Proceedings, 2001. XIII, 258 pages. 2001.

Vol. 2159: J. Kelemen, P. Sosík (Eds.), Advances in Artificial Life. Proceedings, 2001. XIX, 724 pages. 2001. (Subseries LNAI).

Vol. 2161: F. Meyer auf der Heide (Ed.), Algorithms – ESA 2001. Proceedings, 2001. XII, 538 pages. 2001.

Vol. 2162: Ç. K. Koç, D. Naccache, C. Paar (Eds.), Cryptographic Hardware and Embedded Systems – CHES 2001. Proceedings, 2001. XIV, 411 pages. 2001.

Vol. 2164: S. Pierre, R. Glitho (Eds.), Mobile Agents for Telecommunication Applications. Proceedings, 2001. XI, 292 pages. 2001.

Vol. 2165: L. de Alfaro, S. Gilmore (Eds.), Process Algebra and Probabilistic Methods. Proceedings, 2001. XII, 217 pages. 2001.

Vol. 2166: V. Matoušek, P. Mautner, R. Mouček, K. Taušer (Eds.), Text, Speech and Dialogue. Proceedings, 2001. XIII, 452 pages. 2001. (Subseries LNAI).

Vol. 2170: S. Palazzo (Ed.), Evolutionary Trends of the Internet. Proceedings, 2001. XIII, 722 pages. 2001.

Vol. 2172: C. Batini, F. Giunchiglia, P. Giorgini, M. Mecella (Eds.), Cooperative Information Systems. Proceedings, 2001. XI, 450 pages. 2001.

Vol. 2176: K.-D. Althoff, R.L. Feldmann, W. Müller (Eds.), Advances in Learning Software Organizations. Proceedings, 2001. XI, 241 pages. 2001.

Vol. 2177: G. Butler, S. Jarzabek (Eds.), Generative and Component-Based Software Engineering. Proceedings, 2001. X, 203 pages. 2001.

Vol. 2181: C. Y. Westort (Eds.), Digital Earth Moving. Proceedings, 2001. XII, 117 pages. 2001.

Vol. 2184: M. Tucci (Ed.), Multimedia Databases and Image Communication. Proceedings, 2001. X, 225 pages. 2001.

Vol. 2186: J. Bosch (Ed.), Generative and Component-Based Software Engineering. Proceedings, 2001. VIII, 177 pages. 2001.

Vol. 2188: F. Bomarius, S. Komi-Sirviö (Eds.), Product Focused Software Process Improvement. Proceedings, 2001. XI, 382 pages. 2001.

Vol. 2189: F. Hoffmann, D.J. Hand, N. Adams, D. Fisher, G. Guimaraes (Eds.), Advances in Intelligent Data Analysis. Proceedings, 2001. XII, 384 pages. 2001.

Vol. 2190: A. de Antonio, R. Aylett, D. Ballin (Eds.), Intelligent Virtual Agents. Proceedings, 2001. VIII, 245 pages. 2001. (Subseries LNAI).

Vol. 2191: B. Radig, S. Florczyk (Eds.), Pattern Recognition. Proceedings, 2001. XVI, 452 pages. 2001.

Vol. 2193: F. Casati, D. Georgakopoulos, M.-C. Shan (Eds.), Technologies for E-Services. Proceedings, 2001. X, 213 pages. 2001.

Preface

The workshop on Technologies for E-Services (TES) is a forum for the exchange of ideas, results, and experiences in the area of e-services. The first workshop (TES 2000) was organized in Cairo, Egypt in conjunction with the VLDB conference. Following the success of the first workshop and pushed by many requests, we organized a second one, also in conjunction with VLDB.

TES 2001 took place in Rome, Italy, in the Pontificia Università Urbaniana, a university belonging to the Catholic Church, located in the center of Rome, and overlooking St. Peter's Cathedral. The TES workshop began right after VLDB, and lasted two days. It featured the presentation of 15 papers focused on several aspects of e-services, including e-service description, e-service composition, peer-to-peer execution of e-services, transactional issues in e-services, e-services infrastructures, and e-services for mobile users. In addition, the workshop included overview and discussion papers that presented and summarized the issues that still need to be addressed to realize the e-services vision.

We take this opportunity to thank the many people that contributed to the organization of the workshop. The TES organization was coordinated by Mariagrazia Fugini (TES organization chair), in conjunction with the wonderful support provided by the local VLDB organization. VLDB 2001 officers, and in particular Paolo Atzeni (general chair) and Stefano Ceri (program chair and workshop coordinator), also provided invaluable help and support. We are very grateful to all of them.

We also thank the members of the program committee and the additional reviewers for their thorough work, which greatly contributed to the quality of the final program. Special thanks goes to Stefanie Chatelain, who coordinated many aspects of the paper review process. The quality of all the accepted papers was very high, and we are grateful to the authors for submitting their interesting results and innovative contributions to the TES workshop.

We hope that the participants found the workshop interesting and stimulating, and we thank them for attending the workshop and for contributing to the discussions, despite the many temptations provided by the beautiful city of Rome. We are also confident that the readers of this book will find that the time and money have been well invested.

July 2001

Fabio Casati
Dimitrios Georgakopoulos
Ming-Chien Shan

TES 2001 Organization

Workshop Officers

General Chair
Ming-Chien Shan
Hewlett-Packard
Palo Alto, California, USA

Program Chair
Dimitrios Georgakopoulos
Telcordia Technologies
Austin, Texas, USA

Organization Chair
Mariagrazia Fugini
Politecnico di Milano
Milan, Italy

Program Committee

Gustavo Alonso	ETH Zurich, Switzerland
Alejandro Buchmann	Darmstadt University of Technology, Germany
Christoph Bussler	Oracle corp., USA
Fabio Casati	Hewlett-Packard, USA
Andrzej Cichocki	Telcordia, USA
Panos Chrysanthis	University of Pittsburgh, USA
Anindya Datta	Georgia Inst. of Technology, USA
Umesh Dayal	Hewlett-Packard, USA
Claude Godart	LORIA, France
Paul Grefen	University of Twente, The Netherlands
Rick Hull	Lucent Labs, USA
Alfons Kemper	Passau University, Germany
Hong Jun Lu	Hong Kong Univ. of Science and Technology, China
Heiko Ludwig	IBM TJ Watson Research Center, USA
John Mylopoulos	University of Toronto, Canada
Maria Orlowska	University of Queensland, Australia
Tamer Ozsu	University of Waterloo, Canada
Mike Papazoglou	Tilburg University, The Netherlands
Louiqa Raschid	University of Maryland, USA
Hans-Jörg Schek	ETH Zurich, Switzerland
Cyrus Shahabi	USC, USA

Afroditi Tsalgatidou	University of Athens, Greece
Gerhard Weikum	University of the Saarland, Germany
Lizhu Zhou	Tsinghua University, China

Additional Referees

Mariagrazia Fugini	Heiko Schuldt
Ralf Muhlberger	Wijnand Derks

Table of Contents

E-Services: Current Technology and Open Issues

Thomi Pilioura* and Aphrodite Tsalgatidou†

University of Athens
Department of Informatics & Telecommunications
TYPA Buildings, Panepistimiopolis, Ilisia
GR-157 84, Athens, Greece
{thomi,afrodite}@di.uoa.gr

Abstract. The Internet changes the way business is conducted. It provides an affordable and easy way to link companies with their incorporating trading and distribution partners as well as customers. However, the Internet's potential is jeopardized by the rising digital anarchy: closed markets that cannot use each other's services; incompatible applications and frameworks that cannot interoperate or build upon each other; difficulties in exchanging business data; lack of highly available servers and secure communication. One solution to these problems is a new paradigm for e-business in which a rich array of modular electronic services (called e-services) is accessible by virtually anyone and any device. This new paradigm is currently the focus of the efforts of many researchers and software vendors. This paper presents the e-services architecture, its advantages as opposed to today's applications and gives an overview of evolving standards. It then presents the related technical challenges, the way some of them are addressed by existing technology and the remaining open issues.

1 Introduction

Companies face a number of challenges in choosing and implementing the right software and technology solutions in order to better serve their needs and support their business endeavors. This has become particularly problematic in recent years as companies attempt to leverage existing practices, systems and resources across the Web. Critical to success in this environment is to find an integrated, robust e-business solution that allows a company to leverage existing applications, rapidly adapt to the unique needs of the business, and continually evolve as business requirements change over time.

Nowadays, the current trend in the application space is moving away from tightly coupled systems (e.g. DCOM based business solutions [1]) and towards systems of loosely coupled, dynamically bound components (e.g. Jini [2] or Enterprise Java

* Ms. Thomi Pilioura is also with the National Bank of Greece, email: tpilioura@nbg.gr.
† Corresponding Author.

F. Casati, D. Georgakopoulos, M.-C. Shan (Eds.): TES 2001, LNCS 2193, pp. 1-15, 2001.

Beans [3]). Systems built with these principles are more likely to dominate the next generation of e-business systems, with flexibility being the overriding characteristic of their success.

A new paradigm for e-business [4, 5], called e-services, seems to be able to help in this direction. E-services are self-contained, modular applications that can be described, published, located and invoked over a network. The e-services framework enables an application developer who has a specific need to cover it by using an appropriate e-service published on the Web, rather than developing the related code from scratch.

The e-service architecture is the logical evolution from object-oriented systems to systems of services. As in object-oriented systems, some of the fundamental concepts in e-services are encapsulation, message passing and dynamic binding. The e-service approach can be considered as a component-based approach where components are large and loosely coupled. Furthermore, the e-service approach advances the component-based paradigm a step beyond signatures, since information related to the quality of service and to what it does is also published in the service interface.

The goal of this paper is to give a technology overview of the challenging area of e-services and to draw some conclusions regarding the status, the applicability and the future of e-service technology. It is therefore organized as follows. Section 2 presents the e-services architecture and its anticipated benefits as compared to today's applications. Section 3 gives an overview of evolving standards. Section 4 presents the technical challenges deriving from the e-services architecture, the way some of them are addressed by current technology and the remaining open issues. Finally, we present our concluding remarks.

2 The E-Services Concept

E-services constitute a new model for using the Web. It allows the publishing of business functions to the Web and enables universal access to these functions. The architecture that enables it, is presented in the following paragraphs along with the benefits that this architecture could bring to e-business.

2.1 The E-Services Architecture

Several basic activities need to be supported by any service-oriented environment:

1. An e-service needs to be created and described.
2. An e-service needs to be published to one or more Intranet or Internet repositories for potential users to locate.
3. An e-service needs to be located by potential users.
4. An e-service must be invoked to be of any benefit.
5. An e-service may need to be unpublished when it is no longer available or needed, or may need to be updated to satisfy new requirements.

In addition to these basic activities there are some other activities that need to take place in order to take full advantage of the e-services architecture. Such activities include e-services integration with existing infrastructure, e-services management and

maintenance. However, we consider that the e-services architecture requires at least the following basic operations: describe, publish, unpublish, update, discover and invoke, and contains 3 roles: service provider, service requester and service broker [6]. Fig. 1illustrates the e-services architecture.

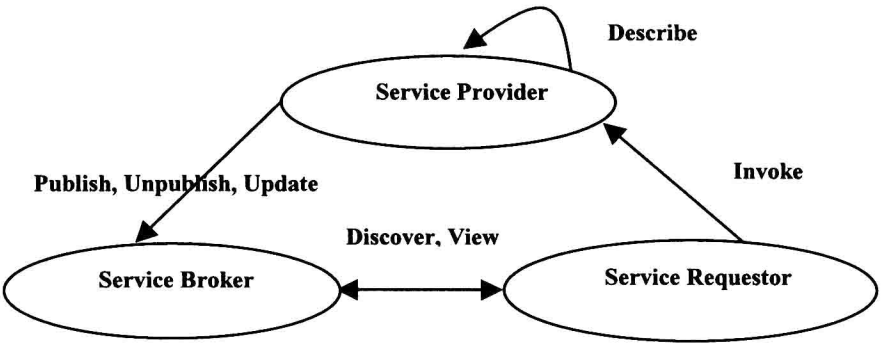

Fig. 1. E-Services Architecture.

Service Provider: A service provider is the party that provides software applications for specific needs as services. Service providers publish, unpublish and update their services so that they are available on the Internet. From a business perspective, this is the owner of the service. From an architectural perspective this is the platform that holds the implementation of the service.

Service Requestor: A requestor is the party that has a need that can be fulfilled by a service available on the Internet. From a business perspective, this is the business that requires certain function to be fulfilled. From an architectural perspective, this is the application that is looking for and invoking a service. A requestor could be a human user accessing the service through a desktop or a wireless browser; it could be an application program; or it could be another e-service. The requestor finds the required services via the Service Broker and binds to services via the Service Provider.

Service Broker: This party provides a searchable repository of service descriptions where service providers publish their services and service requesters find services and obtain binding information for services. It is like telephone yellow pages. Such examples of service brokers are the e-speak E-services Village [7], UDDI (Universal Description, Discovery, Integration) [8], [9], [10] and XMethods [11].

2.2 Anticipated Advantages of E-Services

The following paragraphs focus on the anticipated benefits of e-service approach as compared to today's applications.

Interoperability: Any e-service can interact with any other e-service. This is achieved through an XML-based interface definition language and a protocol of

communication. By limiting what is absolutely required for interoperability, interacting e-services can be truly platform and language independent. This means that developers should not be expected to change their development environments in order to produce or consume e-services. Furthermore by allowing legacy applications to be exposed as services, the e-services architecture easily enables interoperability between legacy applications or between e-services and legacy applications.

Just-in-Time Integration: Traditional system architectures incorporate relatively brittle coupling between various components in the system. These systems are sensitive to change. A change in the output of one of the subsystems or a new implementation of a subsystem will often cause old, statically bound collaborations to break down. E-services systems promote significant decoupling and just-in-time integration of new applications and services, as they are based on the notion of building applications by discovering and orchestrating network-available services. This in turn yields systems that are self-configuring, adaptive and robust with fewer single points of failure.

Easy and Fast Deployment: Enterprises using the e-service model are expected to provide new services and products without the investment and delays a traditional enterprise requires. They may utilize the best-in-their-class component services without having to develop them (outsourcing).

Efficient Application Development: Application development is also more efficient because existing e-services can be reused and composed to create new e-services. An e-service can aggregate other e-services to provide a higher-level set of features.

Strong Encapsulation: All components are services. What is important is the type of behavior a service provides, not how it is implemented. This reduces system complexity, as application designers do not have to worry about implementation details of the services they are invoking.

3 Evolving Standards for E-Services

Today there is a lot of activity in the e-services area. We are currently witnessing the rapid development and maturation of a stack of interrelated standards that are defining the e-services infrastructure. In this section we will outline the key standards of WSDL (Web Services Description Language) [12], SOAP (Simple Object Access Protocol) [13] and UDDI (Universal Description, Discovery, Integration) [14, 15]. Each one of these standards supports some of the basic operations of a service-oriented environment, namely describe, publish, unpublish, update, discover and invoke.

3.1 Web Services Description Language (WSDL)

For an application to use an e-service, the programmatic interface of the e-service must be precisely described. In this sense, WSDL plays a role analogous to Interface Definition Language (IDL) used in distributed programming. It is an XML grammar for specifying properties of an e-service such as *what* it does, *where* it is located and *how* to invoke it.

A WSDL document defines *services* as collections of network endpoints, or *ports*. In WSDL, the abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. This allows the reuse of abstract definitions: *messages*, which are abstract descriptions of the data being exchanged, and *port types* that are abstract collections of *operations*. The concrete protocol and data format specifications for a particular port type constitute a reusable *binding*. A port is defined by associating a network address with a reusable binding, and a collection of ports defines a service. Hence, a WSDL document uses the following seven elements in the definition of network services:

- *Types:* A container for data type definitions using some type system (such as XSD).
- *Message:* An abstract, typed definition of the data being communicated.
- *Operation:* An abstract description of an action supported by the service.
- *Port Type:* An abstract set of operations supported by one or more endpoints.
- *Binding:* A concrete protocol and data format specification for a particular port type.
- *Port:* A single endpoint defined as a combination of a binding and a network address.
- *Service:* A collection of related endpoints.

3.2 Simple Object Access Protocol (SOAP)

SOAP [13] is a standard for sending and receiving messages over the Internet. It is independent of the programming language, object model, operating system and platform. It uses HTTP as the transport protocol and XML for data encoding. However, other transport protocols may also be used. For example, Simple Mail Transport Protocol (SMTP) can be used to send SOAP messages to e-mail servers.

SOAP defines two types of messages, Request and Response, to allow service requestors to request a remote procedure and to allow service providers to respond to such requests. A SOAP message consists of two parts, a header and the XML payload. The header differs between transport layers, but the XML payload remains the same. The XML part of the SOAP request consists of three main portions:

- The *Envelope* defines the various namespaces that are used by the rest of the SOAP message.
- The *Header* is an optional element for carrying auxiliary information for authentication, transactions, and payments. Any element in a SOAP processing chain can add or delete items from the Header; elements can also choose to ignore items if they are unknown. If a Header is present, it must be the first child of the Envelope.