Frank Eliassen
Alberto Montresor (Eds.)

LNCS 4025

Distributed Applications and Interoperable Systems

6th IFIP WG 6.1 International Conference, DAIS 2006
Bologna, Italy, June 2006
Proceedings

ifip

Springer

Frank Eliassen Alberto Montresor (Eds.)

Distributed Applications and Interoperable Systems

6th IFIP WG 6.1 International Conference, DAIS 2006
Bologna, Italy, June 14-16, 2006
Proceedings

Springer

Volume Editors

Frank Eliassen
University of Oslo
Department of Informatics
P.O. Box, 1080 Blindern, Oslo, Norway
E-mail: frank@ifi.uio.no

Alberto Montresor
University of Trento
Department of Information and Communication Technology
via Sommarive 14, 38050 Povo (TN), Italy
E-mail: alberto.montresor@dit.unitn.it

Library of Congress Control Number: 2006926954

CR Subject Classification (1998): D.2, C.2.4, I.2.11, D.4, H.4

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN 0302-9743
ISBN-10 3-540-35126-4 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-35126-9 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11773887 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 4025

Preface

This volume contains the proceedings of the IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems VI held in Bologna, Italy, on June 14-16, 2006.

The conference program presents the state of the art in research on distributed and interoperable systems. In recent years, distributed applications have indeed gained a practical and widely-known footing in everyday computing. Use of new communication technologies have brought up divergent application areas, including mobile computing, inter-enterprise collaborations, and ubiquitous services, just to name a few. New challenges include the need for service-oriented architectures, autonomous and self-managing systems, peer-to-peer systems, grid computing, sensor networks, semantic enhancements, and adaptivity and dynamism of distribution constellations.

Following the evolution of the field, DAIS 2006 focuses on architectures, models, technologies and platforms for interoperable, scalable and adaptable systems that are related to the latest trends towards service orientation and self-* properties. The papers presented at DAIS 2006 cover methodological aspects, tools and language of building adaptable distributed and interoperable services, fault tolerance and dependability, peer-to-peer systems, mobility issues, web services applications and performance issues and composition, semantic web and semantic integration, and context- and location-aware applications. Also included in these proceedings is an invited paper by Jan Bosch and colleagues (Nokia Research Center, Finland) addressing the apparent conflict between usability and the architectural drivers that drive success or failure of mobile services.

This year, the technical program of DAIS drew from 99 submitted papers, among which 10 were explicitly submitted as work-in-progress papers. From these 21 regular and 5 work-in-progress papers were selected for inclusion in the proceedings. As a rule, each paper was reviewed by three reviewers. The DAIS 2006 conference was sponsored by IFIP (International Federation for Information Processing), and it was the sixth conference in the DAIS series of events organized by the IFIP Working Group 6.1. The previous conferences in this series took place in Cottbus, Germany (1997), Helsinki, Finland (1999), Krakow, Poland (2001), Paris, France (2003), and Athens, Greece (2005). Since Paris, DAIS has been organized in conjunction with the FMOODS conference (Formal Methods and Open Object-Based Distributed Systems). This time the COORDINATION conference (Conference on Coordination Models and Languages) joined the federated event of DAIS and FMOODS.

Finally, we would like to take this opportunity to thank the numerous people whose work made this conference possible. We wish to express our deepest gratitude to the authors of submitted papers, to all program committee members for their active participation in the paper review process, to all external reviewers

for their help in evaluating submissions, to the University of Bologna for hosting the event, and to Gianluigi Zavattaro for acting as a general chair of the joint event, who also provided the Conference Management System and support. Ketil Lund took care of the publicity for the event. The Steering Committee with Lea Kutvonen, Hartmut König, Kurt Geihs, and Elie Najm extended their helping hand for making DAIS 2006 a successful conference.

June 2006 Frank Eliassen and Alberto Montresor

Conference Committees and Organization

Chairs

Steering Committee:	Lea Kutvonen, University of Helsinki, Finland
	Elie Njm, ENST, Paris, France
	Hartmut König, BTU Cottbus, Germany
	Kurt Geihs, University of Kassel, Germany
General Chair	Gianluigi Zavattaro, University of Bologna, Italy
Program Co-chairs	Frank Eliassen, University of Oslo, Norway
	Alberto Montresor, University of Trento, Italy
Publicity Chair	Ketil Lund, Simula Research Laboratory, Norway

Sponsoring Institutions

University of Bologna, Italy
IFIP WG 6.1

Program Committee

N. Alonistioti	University of Athens, Greece
D. Bakken	Washington State University, USA
A. Bartoli	University of Trieste, Italy
Y. Berbers	Katholieke Universiteit Leuven, Belgium
A. Beugnard	ENST-Bretagne, France
G. Blair	Lancaster University, UK
A. Corsaro	Alenia Marconi System, Italy
I. Demeure	ENST, France
P. Felber	Université de Neuchâtel, Switzerland
K. Geihs	University of Kassel, Germany
K.M. Goschka	Technical University of Vienna, Austria
S. Graupner	HP Labs, USA
R. Grønmo	SINTEF ICT, Norway
D. Hagimont	INP Toulouse, France
S. Hallsteinsen	SINTEF ICT, Norway
J. Indulska	University of Queensland, Australia
A. Keller	IBM Thomas J. Watson Research Center, USA
H. König	BTU Cottbus, Germany
R. Kröger	University of Applied Sciences Wiesbaden, Germany
H. Krumm	University of Dortmund, Germany
L. Kutvonen	University of Helsinki, Finland

W. Lamersdorf	University of Hamburg, Germany
C. Linnhof-Popien	University of Munich, Germany
K. Lund	Simula Research Laboratory, Norway
R. Meier	Trinity College Dublin, Ireland
E. Najm	ENST, France
R. Oliveira	Universidade do Minho, Portugal
K. Raymond	University of Queensland, Australia
R. Schantz	BBN Technologies, USA
A. Romanovsky	University of Newcastle upon Tyne, UK
W. Schreiner	Johannes Kepler University Linz, Austria
T. Senivongse	Chulalongkorn University, Thailand
K. Sere	Abo Akademi University, Finland
J.B. Stefani	INRIA, France
N. Wang	Tech-X Corporation, USA

Lecture Notes in Computer Science

For information about Vols. 1–3925

please contact your bookseller or Springer

Vol. 4044: P. Abrahamsson, M. Marchesi, G. Succi (Eds.), Extreme Programming and Agile Processes in Software Engineering. XII, 230 pages. 2006.

Vol. 4041: S.-W. Cheng, C.K. Poon (Eds.), Algorithmic Applications in Management. XI, 395 pages. 2006.

Vol. 4039: M. Morisio (Ed.), Reuse of Off-the-Shelf Components. XIII, 444 pages. 2006.

Vol. 4038: P. Ciancarini, H. Wiklicky (Eds.), Coordination Models and Languages. VIII, 299 pages. 2006.

Vol. 4037: R. Gorrieri, H. Wehrheim (Eds.), Formal Methods for Open Object-Based Distributed Systems. XVII, 474 pages. 2006.

Vol. 4034: J. Münch, M. Vierimaa (Eds.), Product-Focused Software Process Improvement. XVII, 474 pages. 2006.

Vol. 4027: H.L. Larsen, G. Pasi, D. Ortiz-Arroyo, T. Andreasen, H. Christiansen (Eds.), Flexible Query Answering Systems. XVIII, 714 pages. 2006. (Sublibrary LNAI).

Vol. 4025: F. Eliassen, A. Montresor (Eds.), Distributed Applications and Interoperable Systems. XI, 355 pages. 2006.

Vol. 4024: S. Donatelli, P. S. Thiagarajan (Eds.), Petri Nets and Other Models of Concurrency - ICATPN 2006. XI, 441 pages. 2006.

Vol. 4021: E. André, L. Dybkjær, W. Minker, H. Neumann, M. Weber (Eds.), Perception and Interactive Technologies. XI, 217 pages. 2006. (Sublibrary LNAI).

Vol. 4011: Y. Sure, J. Domingue (Eds.), The Semantic Web: Research and Applications. XIX, 726 pages. 2006.

Vol. 4010: S. Dunne, B. Stoddart (Eds.), Unifying Theories of Programming. VIII, 257 pages. 2006.

Vol. 4007: C. Àlvarez, M. Serna (Eds.), Experimental Algorithms. XI, 329 pages. 2006.

Vol. 4006: L.M. Pinho, M. González Harbour (Eds.), Reliable Software Technologies – Ada-Europe 2006. XII, 241 pages. 2006.

Vol. 4004: S. Vaudenay (Ed.), Advances in Cryptology - EUROCRYPT 2006. XIV, 613 pages. 2006.

Vol. 4003: Y. Koucheryavy, J. Harju, V.B. Iversen (Eds.), Next Generation Teletraffic and Wired/Wireless Advanced Networking. XVI, 582 pages. 2006.

Vol. 4001: E. Dubois, K. Pohl (Eds.), Advanced Information Systems Engineering. XVI, 560 pages. 2006.

Vol. 3999: C. Kop, G. Fliedl, H.C. Mayr, E. Métais (Eds.), Natural Language Processing and Information Systems. XIII, 227 pages. 2006.

Vol. 3998: T. Calamoneri, I. Finocchi, G.F. Italiano (Eds.), Algorithms and Complexity. XII, 394 pages. 2006.

Vol. 3997: W. Grieskamp, C. Weise (Eds.), Formal Approaches to Software Testing. XII, 219 pages. 2006.

Vol. 3996: A. Keller, J.-P. Martin-Flatin (Eds.), Self-Managed Networks, Systems, and Services. X, 185 pages. 2006.

Vol. 3995: G. Müller (Ed.), Emerging Trends in Information and Communication Security. XX, 524 pages. 2006.

Vol. 3994: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part IV. XXXV, 1096 pages. 2006.

Vol. 3993: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part III. XXXVI, 1136 pages. 2006.

Vol. 3992: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part II. XXXV, 1122 pages. 2006.

Vol. 3991: V.N. Alexandrov, G.D. van Albada, P.M.A. Sloot, J. Dongarra (Eds.), Computational Science – ICCS 2006, Part I. LXXXI, 1096 pages. 2006.

Vol. 3990: J. C. Beck, B.M. Smith (Eds.), Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. X, 301 pages. 2006.

Vol. 3989: J. Zhou, M. Yung, F. Bao, Applied Cryptography and Network Security. XIV, 488 pages. 2006.

Vol. 3987: M. Hazas, J. Krumm, T. Strang (Eds.), Location- and Context-Awareness. X, 289 pages. 2006.

Vol. 3986: K. Stølen, W.H. Winsborough, F. Martinelli, F. Massacci (Eds.), Trust Management. XIV, 474 pages. 2006.

Vol. 3984: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part V. XXV, 1045 pages. 2006.

Vol. 3983: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part IV. XXVI, 1191 pages. 2006.

Vol. 3982: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part III. XXV, 1243 pages. 2006.

Vol. 3981: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part II. XXVI, 1255 pages. 2006.

Vol. 3980: M. Gavrilova, O. Gervasi, V. Kumar, C.J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo (Eds.), Computational Science and Its Applications - ICCSA 2006, Part I. LXXV, 1199 pages. 2006.

Vol. 3979: T.S. Huang, N. Sebe, M.S. Lew, V. Pavlović, M. Kölsch, A. Galata, B. Kisačanin (Eds.), Computer Vision in Human-Computer Interaction. XII, 121 pages. 2006.

Vol. 3978: B. Hnich, M. Carlsson, F. Fages, F. Rossi (Eds.), Recent Advances in Constraints. VIII, 179 pages. 2006. (Sublibrary LNAI).

Vol. 3976: F. Boavida, T. Plagemann, B. Stiller, C. Westphal, E. Monteiro (Eds.), Networking 2006. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems. XXVI, 1276 pages. 2006.

Vol. 3975: S. Mehrotra, D.D. Zeng, H. Chen, B.M. Thuraisingham, F.-Y. Wang (Eds.), Intelligence and Security Informatics. XXII, 772 pages. 2006.

Vol. 3973: J. Wang, Z. Yi, J.M. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks - ISNN 2006, Part III. XXIX, 1402 pages. 2006.

Vol. 3972: J. Wang, Z. Yi, J.M. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks - ISNN 2006, Part II. XXVII, 1444 pages. 2006.

Vol. 3971: J. Wang, Z. Yi, J.M. Zurada, B.-L. Lu, H. Yin (Eds.), Advances in Neural Networks - ISNN 2006, Part I. LXVII, 1442 pages. 2006.

Vol. 3970: T. Braun, G. Carle, S. Fahmy, Y. Koucheryavy (Eds.), Wired/Wireless Internet Communications. XIV, 350 pages. 2006.

Vol. 3968: K.P. Fishkin, B. Schiele, P. Nixon, A. Quigley (Eds.), Pervasive Computing. XV, 402 pages. 2006.

Vol. 3967: D. Grigoriev, J. Harrison, E.A. Hirsch (Eds.), Computer Science – Theory and Applications. XVI, 684 pages. 2006.

Vol. 3966: Q. Wang, D. Pfahl, D.M. Raffo, P. Wernick (Eds.), Software Process Change. XIV, 356 pages. 2006.

Vol. 3965: M. Bernardo, A. Cimatti (Eds.), Formal Methods for Hardware Verification. VII, 243 pages. 2006.

Vol. 3964: M. Ü. Uyar, A.Y. Duale, M.A. Fecko (Eds.), Testing of Communicating Systems. XI, 373 pages. 2006.

Vol. 3963: O. Dikenelli, M.-P. Gleizes, A. Ricci (Eds.), Engineering Societies in the Agents World VI. X, 303 pages. 2006. (Sublibrary LNAI).

Vol. 3962: W. IJsselsteijn, Y. de Kort, C. Midden, B. Eggen, E. van den Hoven (Eds.), Persuasive Technology. XII, 216 pages. 2006.

Vol. 3960: R. Vieira, P. Quaresma, M.d.G.V. Nunes, N.J. Mamede, C. Oliveira, M.C. Dias (Eds.), Computational Processing of the Portuguese Language. XII, 274 pages. 2006. (Sublibrary LNAI).

Vol. 3959: J.-Y. Cai, S. B. Cooper, A. Li (Eds.), Theory and Applications of Models of Computation. XV, 794 pages. 2006.

Vol. 3958: M. Yung, Y. Dodis, A. Kiayias, T. Malkin (Eds.), Public Key Cryptography - PKC 2006. XIV, 543 pages. 2006.

Vol. 3956: G. Barthe, B. Grégoire, M. Huisman, J.-L. Lanet (Eds.), Construction and Analysis of Safe, Secure, and Interoperable Smart Devices. IX, 175 pages. 2006.

Vol. 3955: G. Antoniou, G. Potamias, C. Spyropoulos, D. Plexousakis (Eds.), Advances in Artificial Intelligence. XVII, 611 pages. 2006. (Sublibrary LNAI).

Vol. 3954: A. Leonardis, H. Bischof, A. Pinz (Eds.), Computer Vision – ECCV 2006, Part IV. XVII, 613 pages. 2006.

Vol. 3953: A. Leonardis, H. Bischof, A. Pinz (Eds.), Computer Vision – ECCV 2006, Part III. XVII, 649 pages. 2006.

Vol. 3952: A. Leonardis, H. Bischof, A. Pinz (Eds.), Computer Vision – ECCV 2006, Part II. XVII, 661 pages. 2006.

Vol. 3951: A. Leonardis, H. Bischof, A. Pinz (Eds.), Computer Vision – ECCV 2006, Part I. XXXV, 639 pages. 2006.

Vol. 3950: J.P. Müller, F. Zambonelli (Eds.), Agent-Oriented Software Engineering VI. XVI, 249 pages. 2006.

Vol. 3948: H.I Christensen, H.-H. Nagel (Eds.), Cognitive Vision Systems. VIII, 367 pages. 2006.

Vol. 3947: Y.-C. Chung, J.E. Moreira (Eds.), Advances in Grid and Pervasive Computing. XXI, 667 pages. 2006.

Vol. 3946: T.R. Roth-Berghofer, S. Schulz, D.B. Leake (Eds.), Modeling and Retrieval of Context. XI, 149 pages. 2006. (Sublibrary LNAI).

Vol. 3945: M. Hagiya, P. Wadler (Eds.), Functional and Logic Programming. X, 295 pages. 2006.

Vol. 3944: J. Quiñonero-Candela, I. Dagan, B. Magnini, F. d'Alché-Buc (Eds.), Machine Learning Challenges. XIII, 462 pages. 2006. (Sublibrary LNAI).

Vol. 3943: N. Guelfi, A. Savidis (Eds.), Rapid Integration of Software Engineering Techniques. X, 289 pages. 2006.

Vol. 3942: Z. Pan, R. Aylett, H. Diener, X. Jin, S. Göbel, L. Li (Eds.), Technologies for E-Learning and Digital Entertainment. XXV, 1396 pages. 2006.

Vol. 3941: S.W. Gilroy, M.D. Harrison (Eds.), Interactive Systems. XI, 267 pages. 2006.

Vol. 3940: C. Saunders, M. Grobelnik, S. Gunn, J. Shawe-Taylor (Eds.), Subspace, Latent Structure and Feature Selection. X, 209 pages. 2006.

Vol. 3939: C. Priami, L. Cardelli, S. Emmott (Eds.), Transactions on Computational Systems Biology IV. VII, 141 pages. 2006. (Sublibrary LNBI).

Vol. 3936: M. Lalmas, A. MacFarlane, S. Rüger, A. Tombros, T. Tsikrika, A. Yavlinsky (Eds.), Advances in Information Retrieval. XIX, 584 pages. 2006.

Vol. 3935: D. Won, S. Kim (Eds.), Information Security and Cryptology - ICISC 2005. XIV, 458 pages. 2006.

Vol. 3934: J.A. Clark, R.F. Paige, F.A. C. Polack, P.J. Brooke (Eds.), Security in Pervasive Computing. X, 243 pages. 2006.

Vol. 3933: F. Bonchi, J.-F. Boulicaut (Eds.), Knowledge Discovery in Inductive Databases. VIII, 251 pages. 2006.

Vol. 3931: B. Apolloni, M. Marinaro, G. Nicosia, R. Tagliaferri (Eds.), Neural Nets. XIII, 370 pages. 2006.

Vol. 3930: D.S. Yeung, Z.-Q. Liu, X.-Z. Wang, H. Yan (Eds.), Advances in Machine Learning and Cybernetics. XXI, 1110 pages. 2006. (Sublibrary LNAI).

Vol. 3929: W. MacCaull, M. Winter, I. Düntsch (Eds.), Relational Methods in Computer Science. VIII, 263 pages. 2006.

Vol. 3928: J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.), Smart Card Research and Advanced Applications. XI, 359 pages. 2006.

Vol. 3927: J. Hespanha, A. Tiwari (Eds.), Hybrid Systems: Computation and Control. XII, 584 pages. 2006.

Table of Contents

Mobile Service Oriented Architectures (MOSOA)

Jilles van Gurp, Anssi Karhinen, and Jan Bosch

Software and Application Technologies Laboratory
Nokia Research Center
P.O. Box 407, FI-00045 NOKIA GROUP, Finland
{jilles.vangurp, anssi.karhinen, jan.bosch}@nokia.com

Abstract. Mobile services hold a promise of utilizing the phone also for other purposes than purely communication. However, repeated attempts at realizing mobile services in the market place have been met with limited success. This article (1) defines the architectural drivers that drive success or failure of mobile services, (2) analyzes three different architectural styles of realizing such a mobile service using the example of a movie ticket selling service and (3) presents the results of this analysis. The main result of the analysis is that a serious conflict exists between usability and essentially all the other architectural drivers included in our analysis, i.e. portability, deployability and scalability. This is due to the fact that, because of the restricted state of the art technology, only native client applications offer satisfactory usability, but these do not satisfy the other drivers.

1 Introduction

Mobile services hold a promise of utilizing the phone also for other purposes than voice and SMS communication. Turning the promise into reality has proven to be more complex than what was anticipated in the dawn of digital mobile communication. Offering highly usable, value adding services to consumers and enterprise users has challenged the technology developers, business developers and the mobile service concept developers. The mobile services field became divided into content services and added value functional services early on. The main function of content services is to deliver media content such as ringing tones, greeting cards and background pictures into the mobile phone. Such services were originally built on top of mobile messaging technology such as SMS and have achieved continuously growing success in consumer segment. A more challenging area has turned out to be the services that aim to provide end users with added value functionality such as, for example, route planning, e-commerce and mobile payment. Some of these services were originally built on top of mobile messaging technology and then later on a browser based approach using WAP technology. However, consumer acceptance and business model continued to be a challenge.

An example of technology that aims to deliver ready-to use services for consumers out-of-the-box is SIM-ATK (Application ToolKit). SIM-ATK works in GSM networks. It allows the operators to include pre-installed service menus in the SIM

F. Eliassen and A. Montresor (Eds.): DAIS 2006, LNCS 4025, pp. 1 – 15, 2006.

card delivered with the subscription. The service menus are automatically integrated with the native menus of the phone for seamless access by the end-user. These services use the short message protocol (SMS) to implement access to server functions. This sets some limitations for the implementation of the user experience as the interaction between the user and the server is not synchronous. Success of mobile service technology seems to have depended much on the level of integration into the native phone user interface and the ease with which end users can access the additional functionality offered by the services. An important landmark has been the iMode technology originally launched by NTT-DoCoMo in Japan around the turn of the millennium. iMode offers end-users seamless integration of basic services such as email, weather forecasts, sports results, various content services, online banking, stock trading, online gaming and ticketing services. An important factor for the success of iMode has been the "always on" aspect of the services, which is based on mobile packet data technology.

As the processing capability of the terminal devices has kept growing and faster data protocols have been deployed in the mobile networks it has become feasible to closely emulate the user experience of a PC with fast Internet access. Some of the recent devices include HTML browsers that are capable of displaying dynamic HTML pages with interactive scripts thus bringing the idea of "Internet in a pocket" closer to reality.

Browser based services are evolving rapidly as the capability of the terminals is growing. Traditional problems that plagued the early WAP/WML based solutions where the appearance of a service had to be tightly optimized for mobile device through custom design are slowly vanishing. To publish an existing WWW/HTML service for WAP access typically requires either a separate implementation of the user interface of the service or heavy automatic adaptation by a gateway which in many cases leads results in usability problems for mobile users of the service.

Another approach to offer highly usable mobile services and applications is to open the phone software platform for user installable native applications. This approach is available for example on phones using Symbian platform. A user installable application has access to all phone resources and can integrate with the native phone UI with no restrictions. This enables the creation of mobile services through a client-server pattern that are similar to the native functions available on a phone. A challenge in this approach is to ensure the compatibility of the phone software platform and installed applications. Additionally, the application installation process itself can be more complex than in a pre-installed services or browser based paradigms described earlier. The contribution of this paper is that it provides a comprehensive overview of the architectural alternatives that are available for the implementation of mobile services and analyses the different trade-offs of them.

The remainder of the paper is organized as follows. The next section presents the problem statement, followed by a section outlining the design drivers. Subsequently, three mobile service-oriented architectural styles are presented and analyzed. Related work and a summary section conclude the paper.

2 Problem Statement

Different brands, run-time platforms, screen sizes, networks and many other factors all contribute to the fact that there is a large variety of mobile devices on the market. To provide the best user experience, software developers of mobile applications must specialize their software for specific devices in order to take advantage of device specific features or device specific implementations of common features. A number of problems with device specific software exist:

- The number of devices that can be supported by specialized software is much smaller than the total number of devices. Specializing software for device specific features in any way causes the potential target market to shrink.
- Devices are sold on the market only for brief periods of time and are generally replaced by new devices months or at most a year after their introduction on the market.
- Developing software for a group of similar devices with a common set of specific features requires testing the software on all those devices.
- Device specific software may be hard to port to other devices that do not support the device specifics.

The goal of mobile services is to allow users to access these services through a mobile device. That means that a mobile service consists of a *client-side* component and a *server-side* component. The server-side component offers the feature, the client-side component makes it available to the user.

Necessarily, features, and usability of those features, in the client-side component depend strongly on the capabilities of the client device. Unfortunately, as outlined above making the client component usable by introducing device specific feature dependencies limits the potential market for the server.

As discussed in the introduction, the conflict between usability on one hand and market share on the other hand underlies the limited success in the market of mobile services so far. Solutions in the market so far have either suffered from poor usability (e.g. WAP [16]) or had the problem that the potential group of users able to adopt the solution was only a (small) subset of the entire group of mobile device owners.

Mobile service oriented architectures need to address the following goals:

- **Number of devices.** The service must be provided on a wide variety of mobile devices. The more devices it supports, the larger the market is.
- **Native features.** The service must make full use of native features. Native features add value to the phone and therefore to the services provided on that phone. Native features include both software (e.g. text input methods) and hardware features (softkeys, camera's, display resolution). Native features relevant to the service should preferably be used.
- **Time to market.** The service must have a quick time to market. It is important to reach the market before the competition.
- **Window of opportunity.** The service must not miss its window of opportunity. It is important to be able to target the devices the service is developed for as soon as these devices become available on the market. The devices are on the market for a relatively short period of time and the potential revenue of a service is constrained by this period of time.

- **Forward compatibility.** The service must be forward compatible with the successors of the devices it is targeting. Users of the service will want to continue using the service when they purchase a new phone (assuming the service is useful to them).

Any successful architecture solution for mobile services will need to address these goals explicitly to the extent that is technically feasible. We have the following reasons to believe that it is now possible to define such an architecture:

- **Device performance.** Moore's law [13] has gradually improved device speed, bandwidth and capabilities. These abilities may be exploited to provide an acceptable end user experience while meeting most of the goals outlined above.
- **Consolidation.** There is a growing set of common features supported across a wide variety of devices that may be of use when implementing mobile services. This common feature set is good enough for a wide range of applications so an increasing number of mobile services can be implemented for a (relatively) broad set of devices.
- **Market size.** Adoption of mobile devices in the market has grown exponentially. A large part of the world population now has access to mobile technology.

The problem in mobile service oriented architectures so far has been that existing architectures fail to reach all aforementioned goals due to the fact that they are conflicting given the current state of the art. In this article we evaluate three architectural alternatives against the goals outlined in this section. In the next section we translate these goals into architectural drivers.

3 Architectural Drivers

In this section we look at the architectural drivers that influence the design, development, deployment and ultimately the success of mobile services. In the next section we will compare three architectures and analyse how they are affected by the architectural drivers.

3.1 Usability

In order to be adopted by users, mobile services need to be usable. By usable we mean that:

- It must be easy for the user to find and access the service.
- It must be easy for the user to make use of the service (learnability, ease of use).
- It must be convenient for the user to make use of the service (performance, usefulness).

In practical terms this means that the service needs to be integrated with the mobile user interface because this provides users with the fastest route to the service and the best performance possible on the device. Access points to the service may be embedded in menus, associated with soft keys, etc.

Experience with mobile user interfaces has shown that it is important to minimize the number of navigation steps to particular features [12]. For example, access to the