Consuming Web Services on Mobile Platforms

Alin COBÂRZAN
Faculty of Economic Sciences,
Babeş-Bolyai University, Cluj-Napoca, Romania
alin.cobarzan@econ.ubbcluj.ro

Web services are an emerging technology that provides interoperability between applications running in different platforms. The Web services technology provide the best approach to Service Oriented Architecture envision of component collaboration for better business requirements fulfilment in large enterprise systems. The challenges in implementing Web services consuming clients for low-resources mobile devices connected through unreliable wireless connections are delimited. The paper also presents a communication architecture that moves the heavy load of XML-based messaging system from the mobile clients to an external middleware component. The middleware component will act like a gateway that lightly communicates with the device in a client-server manner over a fast binary protocol and at the same time takes the responsibility of solving the request to the Web service.
***Keywords**: Web Services, SOA, Mobile Computing, Mobile Devices*

1 Introduction

The current trend in the application space is moving towards *Service-Oriented Architecture* (SOA) paradigm [1] which describes a flexible set of design principles for building interdependent and interoperable application components that act like services by implementing a specific functionality and publishing a communication interface to access it. The paradigm envisions component collaboration for better fulfilment of business requirements in large enterprises systems. Systems built with these principles will be more flexible to changes in business requirements as opposed to traditional monolithic systems that are extremely sensitive to any change in one of its subsystems [2].

The *Web services* technology [3] evolved from the most logical approach of applying the SOA vision to the Web space. This technology provides interoperability over the Web between applications running on different platforms, ensuring a standardized protocol for business partners to exchange information. Being built over the *eXtensible Mark-up Language* (XML) and *Hypertext Transfer Protocol* (HTTP) standard protocols, Web services can be easily deployed and accessed from anywhere over the Web.

Despite the obvious benefits that Web services technology brings to enterprise business initiatives by ensuring easy integration of highly interoperable and deployable service-like components, it all comes with a payoff. Web services consumers may suffer from poor communication performance due to the verbosity of XML-based messaging system. This challenge alone needs to be seriously addressed as mobile application development trends are towards taking the benefits of consuming the plethora of available Web services [4]. While mobile devices hardware capabilities have greatly increased in the last years of the ongoing mobile revolution, the capability of easily consuming Web services when in motion is far from a standard feature on most of the mobile platforms due to existing performance issues and lack of native development support on most of the important mobile platforms.

The paper starts with a short description of the Web services technology, the next standard way of providing interoperability between applications running on different platforms. The major limitation of the SOAP-based Web services is also explored from the point of view of consuming them on mobile handheld devices. The overhead of the XML processing that the SOAP Web services rely on does not match with the limited hardware capabilities that mobile devices present nowadays. Network bandwidth limitation and the unreliable wireless communication on mobile devices are also decreasing the overall support for Web services consuming on handheld devices. An overview of the native support on most of mobile platforms reveals another possible issue towards fast mobile Web service consuming. Related work in this area of research [16] [17] are shortly summarized mainly to enforce the success of solutions that introduce in the communication scheme a middleware component that takes the heavy load of the communication with the Web service. Those middleware components will act like *gateways* that communicate lightly with the device while ensuring the responsibility of retrieving the response from the Web service. The proposed communication architecture will emphasize introducing a gateway between the mobile client and the Web service that will take all the burden of the heavy load XML-based communication with the service. The mobile client will instead have to sustain a lightweight and simple client-server communication over a fast binary protocol. The paper details the component of the communication architecture and emphasizes both on the problems solved and the issues introduced by this architecture.

2 Description of Web services

Web services are software components build upon Web-based technologies including HTTP and XML and allow standard means of interoperability [2] over the Internet or intranets between application running on a large variety of hardware and platforms. Web services act like self-contained components that are published, located and invoked over the Web. The key concept behind Web services is to provide a standard platform and operating system independent mechanism for application communication over the Web.

The *Web Service Architecture* (WSA) [3] proposed by W3 organization relies on a number of Web standards like XML, *Simple Object Access Protocol* (SOAP), *Web Service Description Language* (WSDL) and *Universal Description Discovery and Integration* (UDDI) that allow services to be described, searched and integrated by any application. Main Web services implementations fall into one of the two categories: SOAP Web services and *REpresentational State Transfer* (REST) Web services – introduced for the first time by [5]. SOAP-based Web services are the preferred way to implement the SOA initiative in today's complex and heterogeneous computing environment. SOAP-based Web services present greater flexibility at lower integration costs over RESTful Web services that instead offer great performance through lighter messaging system [6].

There are two major limitations that SOAP-based Web services have: performance issues due to XML processing overhead and lack of support for transaction in communication. Communication with a Web service may suffer poor performance over busy or unstable networks in comparison with other traditional approaches to distributed computing like *CORBA* or *DCOM*, due to the verbose nature of the XML-based messaging system that was simply not intended for efficiency. Moreover, extra overhead when encoding/decoding XML requests/responses definitely count on the overall Web application performance. Lack of transaction support in the communication with the Web service makes this kind of data exchange protocols rather *stateless* as the Web service provider and Web service consumer don't have knowledge of each other's state.

As wireless network access becomes a standard feature in the nowadays mobile devices, mobile Web services consuming challenges are worth investigating from two points of view: from the point of *mobile hardware capabilities* (both processing power and network capabilities) and the point of *native support on the current mobile platforms*. Also, adapting the existing standards to the limitations of resource-constrained mobile devices should be considered.

3 Limitations of mobile devices

About fifteen years ago the mobile phones had huge sizes and carrying them was an issue. They were about phone calls only. Today the mobile revolution brings mobile devices that can easily fit in your back pocket and feature incredible entertainment and multimedia capabilities. The capabilities of mobile devices are constantly increasing to satisfy an even more increasingly need for flexibility in terms of mobility and connectivity. Cellular networks are expected to sustain a 40 time increase of data traffic in the next 5 years [7]. Data traffic will come from Web browsing in general and video watching in particular.

Sites like [8] give an up to date hardware features comparison of existing smartphones – the most appealing mobile devices on the market in terms of mobility. Besides the obvious advances in computational capabilities (e.g. faster processor, greater amount of ROM and RAM), the trend is towards providing wireless network communication capabilities and high connectivity features. The main hardware limitations of mobile devices that need to be addressed in the near future are huge battery consumption - that translates into poor battery life, low wireless network bandwidth and wireless connection instability. Decent levels of mobility and connectivity can be achieved only with a high battery autonomy while being constantly connected to a wireless access point and consuming a service with little or no delays.

4 Existing support on mobile platforms

This section is intended to compare the development effort required by Web services client implementation on most important mobile platforms and discuss the various levels of support that they provide.

Sun moved recently towards enabling Java technology-capable mobile devices with a standardized model to access the existing Web Services by extending J2ME platform with JSR (Java Specification Request) 172 [9]. The request also named *J2ME Web Services Specification* (WSS) leverages the J2ME for the standard Java Web services platform thus allowing developers to easily create clients for Java-enabled mobile devices also. The WSS specification is part of the *Mobile Service Architecture* platform designed to meet the evolution of market towards incorporating new technologies and services in mobile devices [10] by providing a standardized application environment for all Java technology-enabled mobile devices.

Apple has made little effort in supporting Web services consuming on its mobile iPhone platform. The SDK does not present native Objective-C libraries for simple creation of Web services clients and moreover, the existing NSXML library of Cocoa framework is rarely used when implementing the needed XML-based messaging mechanism behind. Libraries like *libxml2* or *KissXML* are definitely preferred when it comes for parsing the XML bloat. Workarounds on iPhone platform include the usage of existing utility applications like *wsdl2objc* [11] that generate stubs proxy classes to access the service from the WSDL specification. The generated classes contain all the methods that the Web service exposes. Nevertheless noticeable efforts have been conducted by open source community to create easy to use frameworks for Web services access [12].

Google has also shown no interest in making SOAP Web services consuming an easy task for the Android platform developer community. The SDK is not even bundled with tools to generate stubs for the Web service's interface. Workarounds are more like do-it-yourself solutions based on *kSOAP 2* [13] more flexible and complete XML parsing library as Android is indeed a Java-based platform but not fully a J2ME one. The overall feeling in the development community is that the Android platform best fits as a deliverer of the Google's services in the mobile space [14].

Consuming Web services is an easy job on Windows Mobile as .NET Compact Framework is well suited for both synchronous and asynchronous access [15]. By simply referencing a Web service, proxy classes that expose the service's interface are generated. This is also the case for the Windows Phone 7 platform.

Nokia provides the Serene framework (from Symbian Series 60 platform) for easy creation of application clients that consume Web services. The framework relies on the J2ME JSR 172 specification to provide full support in creating relevant and rich web applications.

Except for the Windows and Nokia platforms that provide a level of support that allows the developer to concentrate on design and creativity rather than finding dirty workarounds, developing SOAP Web services clients requires a lot of extra effort mostly due to the lack of native support. Moreover, specific workarounds are required for specific platforms making very resource consuming the task of implementing a Web application on all mobile platforms.

5 Related work

This section is intended to present two types of solutions that research community has proposed as simple adaptations to the limitations of direct SOAP-based Web services consuming from most of the mobile platforms.

5.1 Migration to RESTful architectures

The Web Service architecture for mobile clients presented in [16] deals with the problem of adapting an existing SOAP Web Services implementation to the low resource nature of the mobile devices by replacing two resource consuming issues of underlying SOAP messaging system: communication payload over HTTP due to thick SOAP request enveloping mechanism and SOAP response parsing on mobile device side. The adopted solution relies on the RESTful architecture and raises a series of problems due to the transport neutral nature of the SOAP protocol. The existing SOAP interface can be even optimized to a RESTful architecture by clearly allocating a distinct URL for each type of interaction. Performance tests have been conducted with both synchronous and asynchronous RESTful implementations and the results are promising in terms of HTTP payload reduction: up to 96% payload reduction for synchronous invocation and up to 75% in case of asynchronous invocation.

5.2 Using mobile Web service agents

Another category of solutions introduces a mobile Web service agent in the network that acts like a gateway between the mobile device and the Web service [17]. The agent receives the input parameters from the mobile device, invokes the required service and returns the result back to the mobile device. Results are encouraging as improved performance is obtained by eliminating the XML processing from the mobile device. The gained performance is measured both in little response time and lower data load.

The reason for presenting two of the existing proposed solutions in the research area is to enforce the belief that any feasible communication architecture for mobile devices must include a middleware component that exists outside the device and takes the responsibility and heavy load of XML processing needed in communicating with the Web service. Those middleware components will act like gateway servers that communicate lightly with the device (thus ensuring a small bandwidth usage – for limited GPRS data communication - and little chance for failure in unreliable wireless networks) and take the load of retrieving the response from the Web service. This type of architecture can bring many more opportunities towards ensuring a more reliable communication with the Web service. As underlined before, the wireless communication is prone to fluctuant behaviour on mobile devices thus making wireless network access prone to failure. The gateway will most likely run on dedicated hardware that will justify the search of solutions for ensuring some kind of *state* of the communication with a Web service. Retry mechanisms can be explored in case of connection failures.

6 Proposed solution

The proposed communication architecture introduces a middleware component that acts like a gateway between the SOAP service and a thin client that is stripped of the heavy load of XML processing. The gateway will in fact be a server for the mobile client and will take the burden of request resolution from the mobile client. The mobile client will have to sustain instead a light client-server communication with the gateway based on binary protocols like [18] (see Fig. 1). Of course that any other lighter protocol than SOAP can be used (like REST for instance) but binary protocols are known to roughly deliver the same functionality as the SOAP protocol. Binary protocols are simply not as "standard" as XML or SOAP but they can provide the same level of extensibility and security at a lower communication footprint. Moreover, from the point of view of development support, successful protocols like [18] have out-of-the-box clients ported for all important programming languages and platforms. Thus, the underlying data communication processing remains as transparent as it is for the generated proxy client classes from a WSDL for SOAP clients (see tools like [11]).

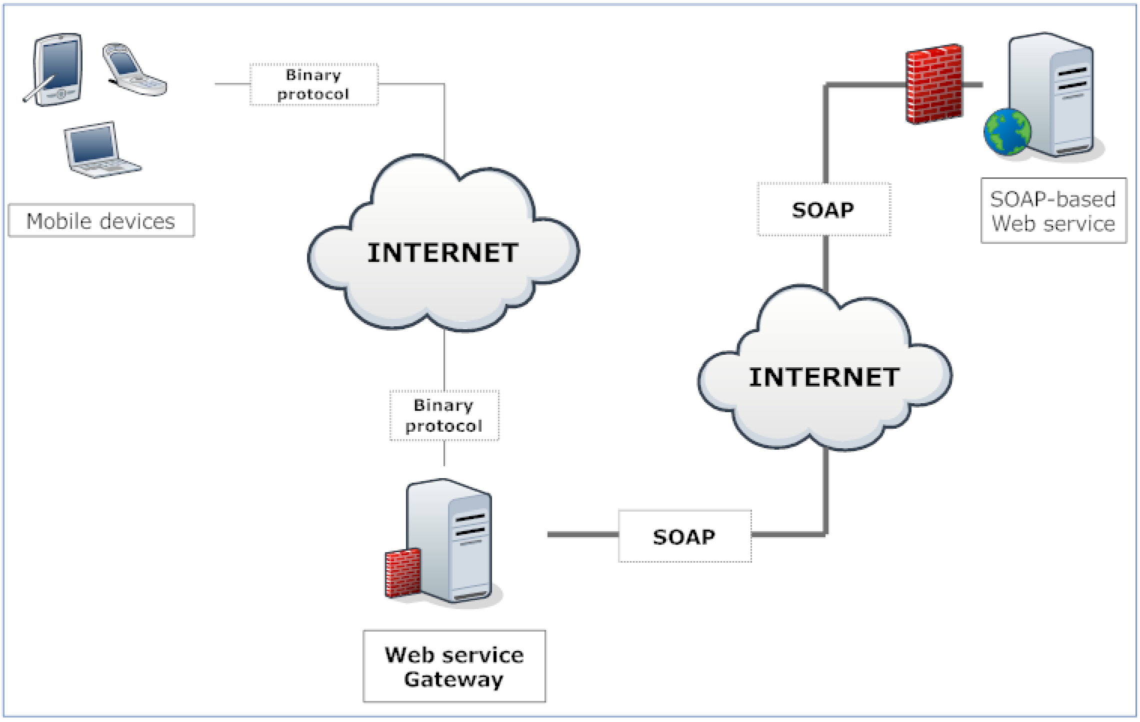


Fig. 1. General overview of the architecture

Introducing a middleware obviously increases the overall duration of a request sent to the Web service. This is one of the many of issues introduced by this solution. The performance of the system can be increased by deploying the gateway in the same network with the actual Web service. The heavy HTTP communication from gateway to Web service will bypass the firewall of the network thus adding an extra boost in the communication (see Figure 2).

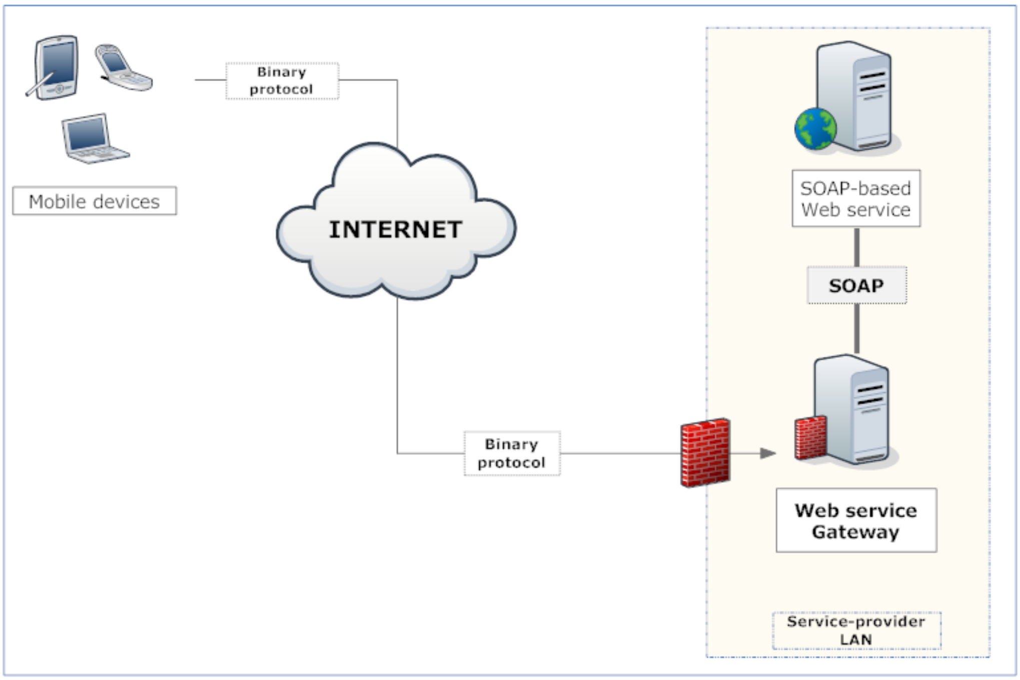


Fig. 2. Improved communication system

Figure 3 further details the components in the system and the way they communicate. Mobile devices will communicate in a client-server fashion with the gateway based on a fast binary protocol and the gateway will rely on automatically generated proxy classes from the WSDL of the service to relay the requests from the mobile client.

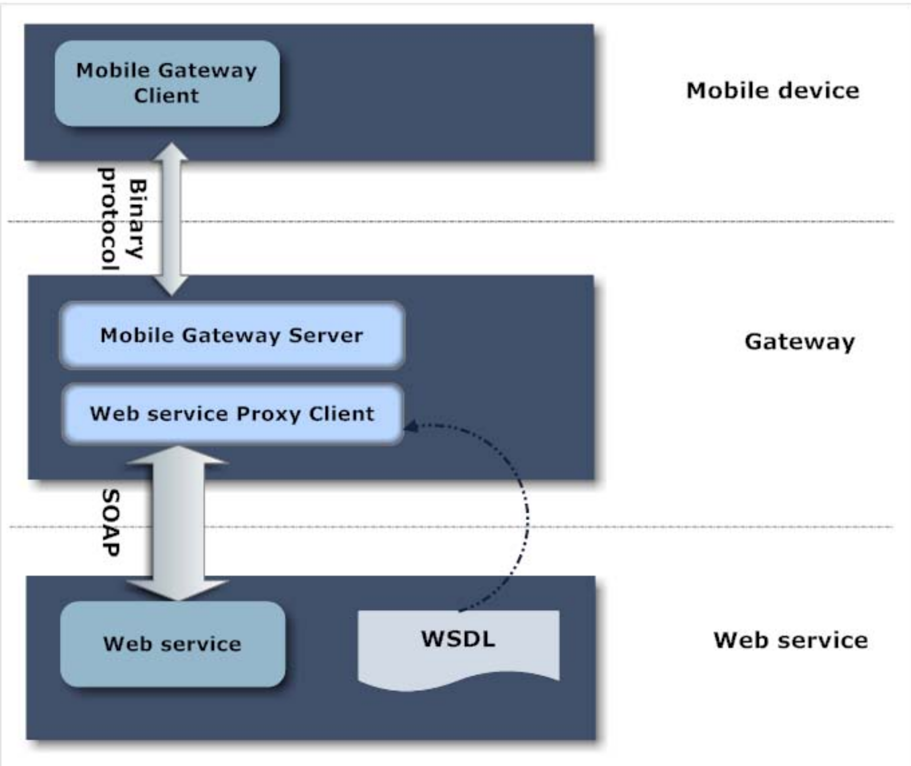


Fig. 3. Proposed architecture for mobile access to Web services

The advantages of this architecture are:

- lighter communication that the mobile client has to sustain: small bandwidth footprint and lesser processing power,
- advanced security features can be explored as the gateway middleware will act on behalf of the mobile client,
- some sort of state can be ensured for the communication, transparent to the mobile client: in case any of the communications fails (mobile device to gateway or gateway to Web service), the gateway can retain the state of the overall communication and retry to continue when all parties come back online,
- from the point of view of implementing the architecture, tools can be designed to easily and automatically create virtually the entire gateway and mobile client components starting from the WSDL document - like [11] does for instance; the backbone of the architecture can be easily generated and tested prior to any real scenario implementing and using this kind of system.

The only major concern for this kind of architecture is a possible increase in the overall duration of a request resolution. The system might earn some time if the XML processing bloat is performed on the gateway but loose because two communications lines have to be established and maintained so prior to any measurement of an actual implementation there is no conclusion on this direction.

7 Conclusions and future work

Moving towards implementing Web applications that consume a large variety of Web services is the next hype in application space and the mobile application market is searching for solutions to empower mobile devices with Web services integration while minimizing the existing performance issues. Researches in the area of Web services [17] show that there are no real magic workarounds that will improve the performance of the mobile clients that try to consume a Web service by means of faster XML parsers, overall protocol optimization or data compression. Researches like [16] and [17] validate the effort of designing communication architectures that include a middleware component that exists outside the device and takes the responsibility and heavy load of XML processing needed in the communication with the Web service.

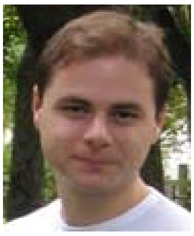
Future work will include creating a more detailed picture of the existing support for Web application development on major mobile platforms. The main effort however will be invested in implementing the proposed architecture for the upcoming Windows Phone platform. This platform ensures easy creation of clients for Web services consuming and therefore comparisons with the implementation of the gateway-based architecture performance will be very relevant. Tests will mainly be conducted to measure bandwidth footprint on the Windows Phone client in the gateway-based architecture implementation. Data will be compared against measures obtained from the easy to create SOAP Web service clients with .NET Framework. Overall system performance and reliability of the two solutions will be monitored and compared.

Acknowledgements

Investing in people! PhD scholarship, Project co-financed by the
SECTORAL OPERATIONAL PROGRAMME HUMAN RESOURCES DEVELOPMENT 2007 - 2013
Priority Axis 1 "Education and training in support for growth and development of a knowledge based society"
Key area of intervention 1.5: Doctoral and post-doctoral programmes in support of research.
Contract **POSDRU 6/1.5/S/3 – „**DOCTORAL STUDIES: THROUGH SCIENCE TOWARDS SOCIETY**"**
Babeş-Bolyai University, Cluj-Napoca, Romania
The authors wish to thank for the financial support provided from programs co-financed by The SECTORAL OPERATIONAL PROGRAMME HUMAN RESOURCES DEVELOPMENT, Contract **POSDRU 6/1.5/S/3 – „**Doctoral studies: through science towards society**".**

References

[1] “Reference Architecture Foundation for Service Oriented Architecture Version 1.0”, 2009. Available: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-cd-02.pdf
[2] IBM, Software Group, “Web Services architecture overview”, 2000. Available: http://www.ibm.com/developerworks/library/w-ovr/
[3] W3C, “Web Services Architecture”, 2004. Available at: http://www.w3.org/TR/ws-arch/
[4] E. Al-Masri and Q. Mahmoud, “Investigating Web Services on the World Wide Web”, 2008. Available at: http://www2008.org/papers/pdf/p795-almasriA.pdf
[5] R. Fielding and R. Taylor, "Principled Design of the Modern Web Architecture", *ACM Transactions on Internet Technology (TOIT) (New York: Association for Computing Machinery)* 2 (2), pp. 115–150, 2002. Available: http://www.ics.uci.edu/~taylor/documents/2002-REST-TOIT.pdf
[6] C. Pautasso, O. Zimmermann and F. Leymann, “RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision”, *17th International World Wide Web Conference (WWW2008),* 2008
[7] E. Schonfeld, “Mobile Data Traffic Expected To Rise 40-Fold Over Next Five Years”. Available at: http://techcrunch.com/2010/03/30/mobile-data-traffic-rise-40-fold/
[8] Wikipedia, “Comparison of smartphones", 2010. Available at: http://en.wikipedia.org/wiki/Comparison_of_smartphones
[9] “JSR 172: J2ME™ Web Services Specification”. Available at: http://jcp.org/en/jsr/detail?id=172
[10] “JSR 248: Mobile Service Architecture”. Available at: http://jcp.org/en/jsr/detail?id=248
[11] wsdl2objc library homepage. Available at: http://code.google.com/p/wsdl2objc/
[12] iPhoneWebServicesClient library homepage. Available at: http://github.com/akosma/ iPhoneWebServicesClient
[13] kSOAP library homepage. Available at: http://code.google.com/p/ksoap2-android/
[14] S. Herren, “Android SOA (Service Oriented Architecture)”, 2009. Available at: http://www.androidguys.com/2009/09/24/android-soa-service-oriented-architecture/
[15] N. Cowburn, “Consuming Web Services with the Microsoft .NET Compact Framework”, 2003. Available at: http://msdn.microsoft.com/en-us/library/aa446547.aspx
[16] F. Aijaz, S. Zahid Ali, M. Chaudhary and B. Walke, “Enabling High Performance Mobile Web Services Provisioning”, *Vehicular Technology Conference Fall (VTC 2009-Fall),* IEEE, pp. 1-6, 2009
[17] M. Adaçal and A. B. Bener, “Mobile Web Services: A New Agent-Based Framework”, *Internet Computing,* IEEE 10(3), pp. 58-65, 2006
[18] Hessian Library Homepage. Available at: http://hessian.caucho.com/



Alin COBÂRZAN has graduated the Faculty of Mathematics and Computer Science from the Babeş-Bolyai University in 2003. Since 2008 he holds a Master’s Degree from the Faculty of Economic Sciences in the study of Applied Computer Sciences in Economics. Since 2008 he attends a PhD scholarship sustained by the co-financed project “SECTORAL OPERATIONAL PROGRAMME HUMAN RESOURCES DEVELOPMENT”. His research focuses on the analysis of architectures for decision support systems that use mobile technologies.