Hybrid Solution for Integrated Trading

Vlad DIACONIȚA
Academy of Economic Studies, Bucharest, Romania
diaconita.vlad@csie.ase.ro

Integrated applications are complex solutions, whose complexity are determined by the economic processes they implement, the amount of data employed (millions of records grouped in hundreds of tables, databases, hundreds of GB) and the number of users. Service oriented architecture (SOA), is now the most talked-about integration solution in mainstream journals, addressing both simple applications, for a department but also at enterprise level. SOA can refer to software architecture or to a way of standardizing the technical architecture of an enterprise and it shows its value when operating in several distinct and heterogeneous environments.

Keywords: *System Integration, Data Integration, Web Services, Java, XML, Stock Market*

1 About integrated systems

We can say that integration means building systems out of systems. As discussed in [11], the beginning in IT integration can be considered the emergence of the integrated circuit in 1959, which brought together other discoveries such as transistors, resistors and capacitors on a single silicon chip. In a 1965 work Gordon Moore, a founder of Intel predicted that the numbers of transistors on a microchip will double every 18 months [2]. This law is still surprisingly real now, more than 40 years after its formulation. This may be why we need to integrate: to manage in terms of the increased complexity. In this context we recall the principles of complexity management which are: decomposition into parts smaller and more easily manipulated (*divide et impera*), building a standard interface for these parts to communicate and then develop a hierarchical structure where information is increasingly abstracted once we ascend in the hierarchy.

By the 90s, the concept of integration related more to the interlinking problems of hardware and computer components. Gradually, the concept of integration has expanded covering equipment, software, data, communications and technology.

IT development and global economic development, specific to early XXI century have accentuated the trend of organizing information systems in models of increasingly complexity. Integration leads to increased complexity, but also to increased quality for the systems involved because bringing system together implies adding evolving and emerging components.

The information system of an organization is a constantly increasing entity. Yet there is a risk that the new requirements, the new business requirements determine the adoption of new specialized systems with the immediate consequence of generating a redundancy in business logic, data and responsibilities [9].

As shown in [6], collaboration between organizations, based on communications between systems and their applications has been identified as a key factor for success in an evolving international environment, enabling companies to develop partnerships and strengthen their market position. While each organization wants to self-develop, experience has shown that this is not possible without the development of relations and business processes, easing the exchange of information and documents with partners [1]. Even if many of these business processes are created for internal use within the company, their evolution leads to sharing logic between the companies and the outside partners. Using services exposure infrastructure, not only integrates applications with the aim of sharing information, but also provides the infrastructure for reusing the business logic in inter-company processes (by using a

business process oriented approach) [12]. Problems are generally due to incompatibility in the representation of information and in the methods adopted [7]. The causes of these problems are often rooted in past investments in hardware and software. Usually, these purchases were aimed at solving particular problems, but often led to fragmentation of information and difficulties in integrating due to incompatibilities or differences in format [8].

Use of effective standards for data representation, knowledge and services are crucial for achieving interoperability between systems.

The consequent need of a middleware that integrates dynamic services running on distributed, heterogeneous platforms led to different but in some parts interleaving solutions. The problem, in its most general significance, has been addressed trough the concept of Service Oriented Architecture (SOA) [14].

The service-oriented architecture offers mechanisms of flexibility and interoperability that allow different technologies to be dynamically integrated, independently of the system's platform in use. This architecture promotes reusability, and it has reduced the time to put available and get access to new system's functionalities, allowing enterprises to dynamically publish, discover and aggregate a range of Web services through the Internet [13].

The possibility of getting access to the core services of an application - and as a result, the further integration of these applications can be an important benefit. However, this benefit is followed by a real risk because of high implementation costs of service based integration, as these costs could, in some cases, exceed the value created by the integration.

Integration helps in the development of business intelligence solutions, increasing the systems capability of providing representative information to the high-level management in support of strategic activities such as goal setting, planning and forecasting [10].

Before using a type of integration, one should well understand the aspects of the opportunities they offer and the risks involved. Only then, an objective evaluation can be made.

Currently, complexity continues to increase, for hardware, software and information systems. The trend is to use services orientation in as many areas of automation.

2 Brief analyze of business requests

To plan a general, integrated architecture for the stock exchange broker firm, we begin by briefly analyzing its departments and applications to be integrated:

- Front-office, functions: management of the company's potential customers, bringing new ones;
- Back-office, functions: open customers account in the market, brokers management, reporting to the NSC (National Securities Commission), supervision the consolidation at the end of each day, generating lists of fees, printing cards and money orders for customers with unique sales, archiving;
- Accounting information system, functions: all activities related to the company's book keeping;
- Trading, functions: the actual trading done on markets, both for customers who are not using the web platform and the firm (*House* account), investment management, may take, in some cases functions from the back-office and front-office;
- Web trading platform: the tool by which some of the company's customers interact with the stock exchanges through the vision provided by the company;
- In Romania, trading can be done on two markets: one that can deal spot stocks, bonds and government securities, derivatives (BSE) and the second one specialized on derivatives (SCE Sibex).

Together with the broker firm, they were established facilities that the system must provide:

- User access to the platform must be made using a web browser;
- Possibility of trading, depending on the options each customer, on more financial markets;
- Real-time financial markets message-processing;
- Building on then broker's system a reliable image of the integrated financial markets;
- Real-time acquisition and transmission of orders from the customers to the financial markets;
- Flexible queries as: historical quotes, transaction history, technical analysis etc.;
- The reporting required by law, for the brokerage firm and for the user: portfolio, transactions, portfolio sheet etc.

The success of the implementation is assessed primarily by the increase of the volume traded by the intermediary, and thus the increase commission charged.

To integrate the systems I will use a mixed approach, the proposed architecture is shown in figure 1.

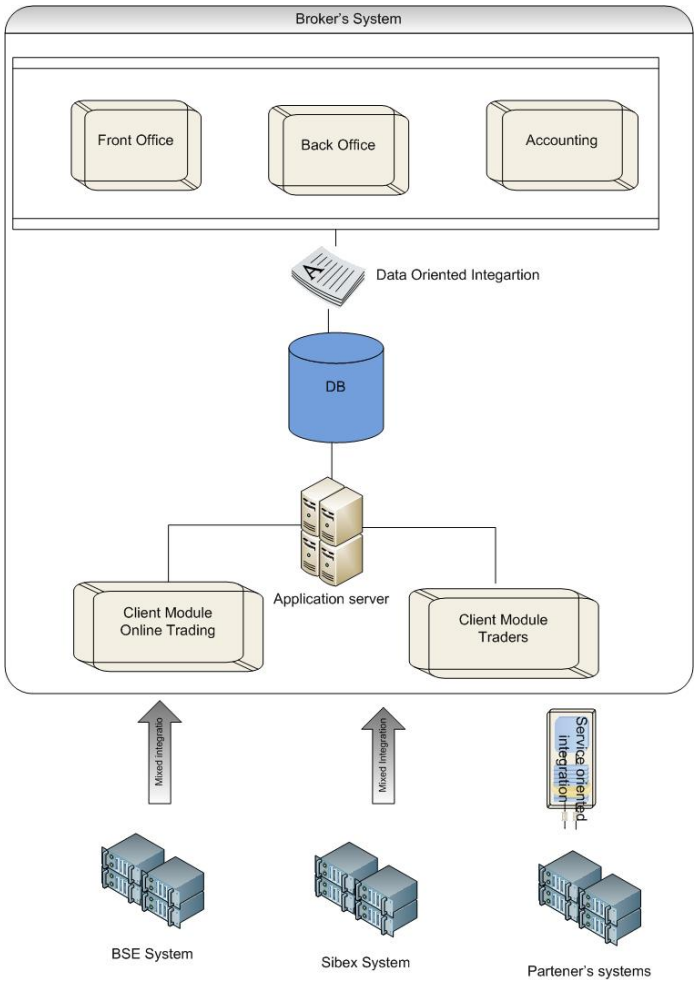

Fig. 1. The general architecture

This architecture takes into account the present situation but also the plans for short and medium term development of the brokerage firm.

For integrated the Back Office, Front Office and Accounting I opt for a data-oriented integration with the database as a central point.

To achieve integration between different modules I use stored functions and procedures with simple or compound parameters, organized in packets by their common function. If it is desired, the

migration to a service oriented solution these functions can be the basis on which to build the web services.

The proposed solution for the client module (online trading + trading and control) consists of an application built using a three tier architecture: application server, database and client. This solution will contain a set of web services that will be used in building integration solutions to the financial markets.

The application server is built using *Oracle Application Server 10g*. This is a suite of programs and tools used for deploying and integrating Internet applications. While maintaining the convenience of earlier versions, *Oracle Application Server* 10g enables its users to more easily respond to the ever-changing requirements of business processes, coming up with new features to increase the computing resources and the integration of applications and Web services. Enterprise grid computing architecture can dramatically reduce the time, energy work and IT administration costs by integrating standard servers, storage devices and existing software. This platform supports the latest Web development technologies such as *Java 2 Platform Enterprise Edition* (*J2EE*), Web services, XML, or standard Grid. For this solution, from this suite, we use Apache web server with *Forms and Reports Services* to run web applications built using *Oracle Developer Suite*. *Forms and Reports Services* generates Java applets that run on the user's browser. Any computer user who has a browser with Java can run this application from any network: Internet, intranet or extranet. He receives messages from the server about what to display and in turn, sends information about the user's actions to be processed by the server. The client is a generic applet that can run any form, regardless of its size, and must be downloaded once. Little algorithm is run on the client's computer, it usually displays the information served by the application server.

For the database tier I have chosen an Oracle *Database Enterprise Edition* (EE) solution which is the most complex version, with facilities that helps in managing large volumes of data in transactional environments and critical applications, optimizing data mining of data warehouses, improving administration and security. In addition to data, on the database level, we store part of the algorithm in the form of stored PL/SQL. Also, the database and serves as the integration process and generate XML messages. For applications that use XML to be integrated, they have to outsource information as XML, which can be achieved by the combination Oracle Database and Oracle JDeveloper [3].

The security system used by the application is implemented by group-wide rights to view, edit and delete using roles. User authentication will be done using two passwords, one for access to the platform and the other for confirmation on the order of operations. The encryption of the permanent communication channel is done using the DES standard.

For the client tier, the user can use a web browser that supports Java to access the trading platform. Using the interface, the client can inform himself about the situation of his portfolio but can also view data regarding the market, like: stock levels 1 and 2, historical quotes, technical analysis etc. Acquisition of new orders or updates is easy and free of technical limitations. As an alternative to using a web browser, because the OAS Forms and Reports Services generates Java applets, the client will be able to use the JNLP technology (Java Network Launch Protocol). This is an XML based protocol that can be used to run applications from a local cache directly using the JAVA machine.

3 Integration with the stock exchange systems (BSE and Sibex)

In this section I will refer to integration with the BSE and Sibex systems. The proposed solution has some elements of SOA, such as using XML documents, the WSDL language, use of adapters at the application level or use of granular methods for various operations (placing orders, change orders, acquiring market level one and level two prices etc.)

but lacks a key element of this architecture that is easy coupling. In this case we speak of a quasi-permanent coupling between the two servers who know their prior identification data (IP address). This coupling is mainly done for two reasons. First reason, authentication is done using beside, the username and the password, the IP address. This prevents the connection from another to the gateway even if the username and the password were exposed. The second reason is the speed, when the market is opened there is a great volume of information that is being exchanged. The solution is shown in figure 2.

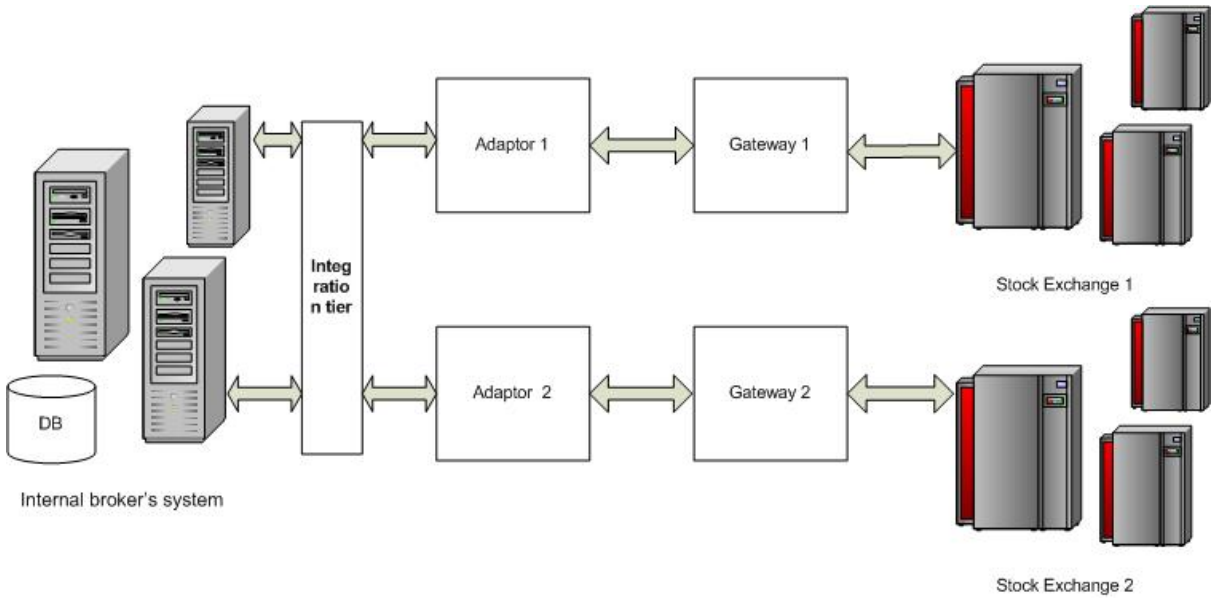

Fig. 2. Adaptors and gateways

Both adapters are built using the Java language. The adapters are built so they will use multiple threads. Upon receiving a message a new thread of processing will be opened enabling the adapter to process multiple messages at a time, given a multi-core/multi-CPU server hardware architecture. If we choose not to work with threads, the times when traffic is heavy, the adapter will be blocked by the messages that gather on a single thread, and the application would not respond to control messages (PING) and will not transmit, for a period up to several seconds, instructions from the users, which is considered unacceptable in such situations. Also due to the specific type of connection I chose to use a timer for taking instructions from the broker's system.

I chose processing XML documents at the database level. The processing is also possible in the adapter but where we have an efficient database that can process these files natively, I believe that it is desirable because it ensures greater independence regarding implementing the interface. The adapters and the database level XML processing were discussed in depth in previous works: [16] and [17].

At the database level there are retrieved and stored customer's orders and those who fulfill the conditions of validation are transmitted to the level of integration that forwards them to the market. Received by the Java built adapters and sent to the database, the responses from the market to the outgoing messages are processed. Also, they are processed other messages necessary for the customers to trade (for example, the market picture). The messages are treated according to their type. Every call to this procedure processes a single XML message, which is called for each message received from grant (fig. 3), making the above talked multi-thread architecture necessary.

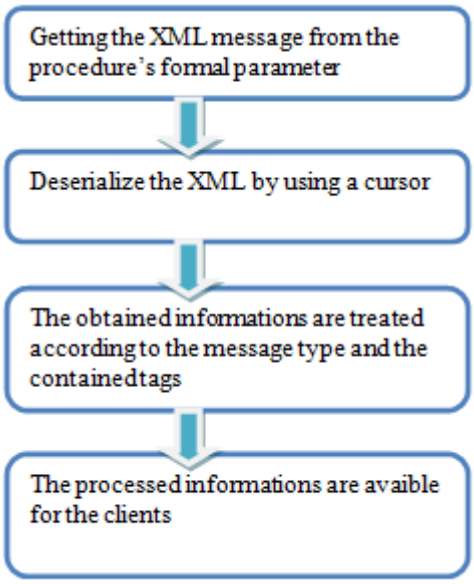

Fig. 3. Database tier processing

4 Integration with the partners' systems using web services

Like shown in [18], web services are self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces. They allow flexible and dynamic software integration that is often referred to as the *Find-Bind-Execute* paradigm. Using standard Internet technology, Web services facilitate cross-organizational transactions and thus outsourcing of software functionality to external service providers. Thus, service-oriented computing requires an infrastructure that provides a mechanism for coordination between service requesters and providers.

Three main technologies are currently used to implement Web services: SOAP , WSDL and UDDI. SOAP is an XML-based, stateless, one-way message exchange protocol for interacting with Web services over HTTP. WSDL is an XML-based format for describing Web services as collections of network endpoints or ports. UDDI is a standard for defining a registry, which allows clients to find Web services through descriptions of business entities, business services or via pre-defined business categories [19].

Web-services are the main ingredient when building complex or basic SOA solutions like the one presented in figure 4.

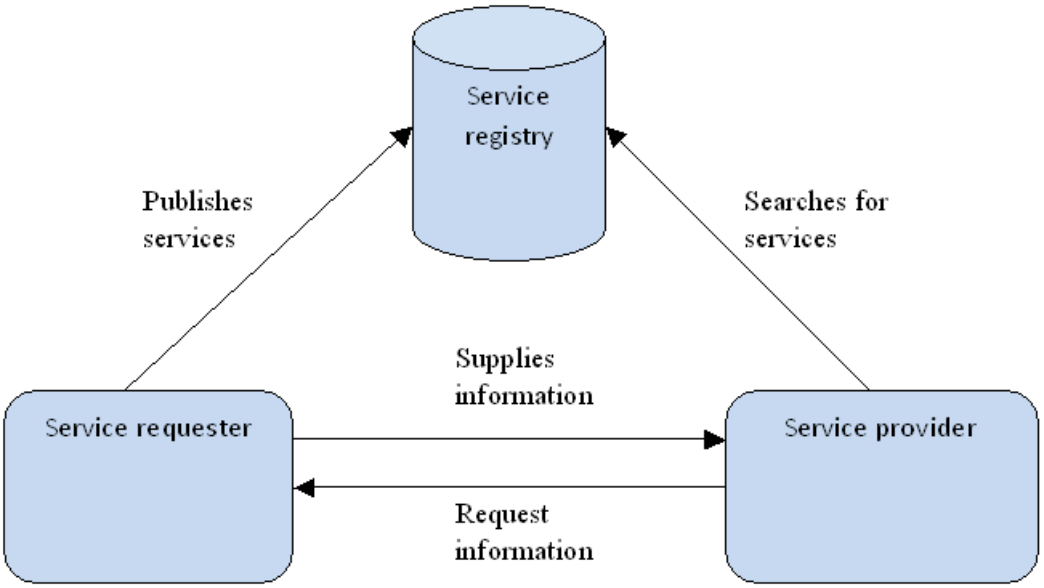

Fig. 4. Generic SOA

In this basic architecture, WSDL is used for service publishing and searching, SOAP for exchanging information between the provider and the requester and UDDI provides the service registry format.

I present below a web service, to be included in the service registry, of the many designed

to interface with the systems of the brokerage firm's partners. These web services are part of the integration tier and are to be used by other systems or applications that wish to communicate with the trading platform. This web services are built using Oracle JDeveloper from PL / SQL database stored packages.

The flow in a web service that can be used to query quotes on a symbol is presented in figure 5:

Fig. 5. The flow inside a web service

Security for client-server or three-tier architecture doesn't raise many problems, users, user groups and roles can be used, at database tier level. Access to various parts of the application can also be controlled at the interface with the customer and can incorporate technologies like single sign-on. When using web-services, as part of service oriented architecture, things are not so simple because there isn't a single point of authentication. Security can be achieved by several technologies, many of these being found in WS-Security standard. WS-Security describes enhancements to SOAP messaging to provide quality of protection through message integrity, message confidentiality, and single message authentication. These mechanisms can be used to accommodate a wide variety of security models and encryption technologies.

Like shown in [15], WS-Security also provides a general-purpose mechanism for associating security tokens with messages. No specific type of security token is required by WS-Security. It is designed to be extensible (e.g. support multiple security token formats). For example, a client might provide proof of identity and proof that they have a particular business certification.

In the example described by above in figure 5, we have a web service based on a PL/SQL procedure stored in the database. We can attach security information in the header by using a user and a password. For the services that we want to be public, we should skip this part. The service requires a stock's symbol to be passed. If the date is not included in the message, the service will return the price of the last transaction on the respective market. If we include a date in the message, the service will return the reference price at that time (figure 6).

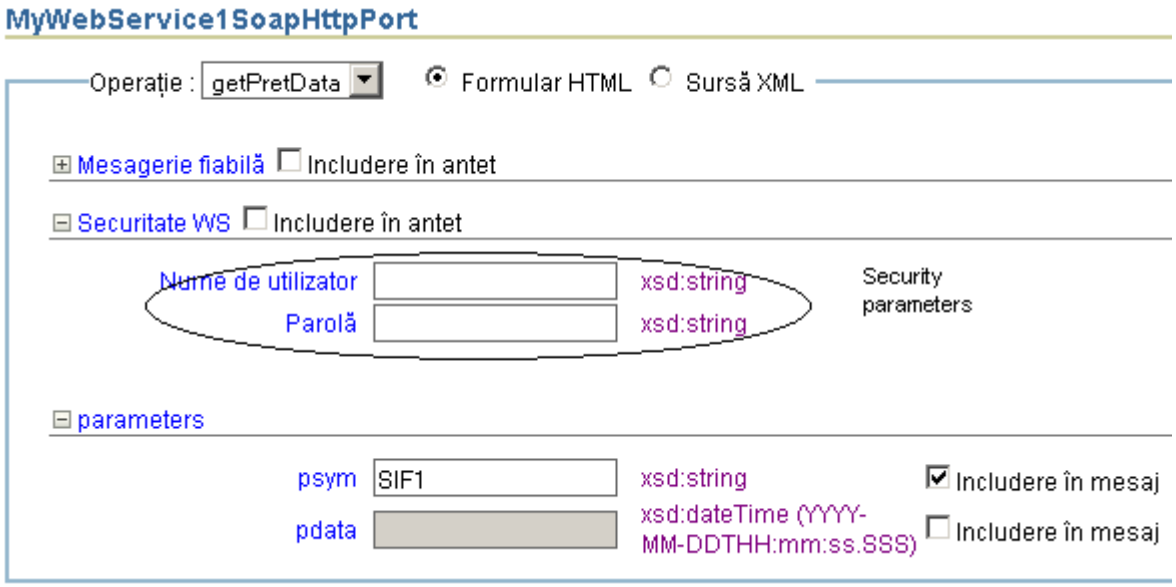

Fig. 6. A web service with security parameters

For example, the result of calling this web service to query for SIF1 quote is the one presented in figure 7.

Fig. 7. Web service response

5 Conclusions

I consider the hybrid solution optimal in this case. The link between trading platform and the markets are made using the adapters built using advanced technical solutions (Oracle Database, Java platform, multiple threads, etc.), communications is accomplished using XML messages, and the system as a whole is capable of processing a great volume of data in a short time. Meanwhile, Web services are available as part of the integration tier, which can be used in an ESB, which allows accessing various features of the brokerage firm's services by its partners.

References

[1] E. Moore and E. Gordon, "Cramming more components onto integrated circuits," *Electronics Magazine*, pp. 4, 1965.

[2] I. Lungu, A. R. Bologa, V. Diaconita, A. Bara and I. Botha, *Integrarea sistemelor informatice*, ASE Publishing House, Bucharest, 2007.

[3] I. Lungu, M. Velicanu, A. Bara, V. Diaconita and I. Botha, "Raport de cercetare: Testarea unor produse software (in special SGBD-uri) care permit integrarea datelor. Testarea unor produse software care permit integrarea aplicatiilor," *Tehnologii informatice pentru integrarea sistemelor institutiilor publice*, GRANT CNCSIS (tip A) între Academia de Studii Economice şi CNCSIS, Faza 2, 2006.

[4] V. Diaconita and I. Lungu, "TCP/IP XML Integration," *The Proceedings of the 1st International Conference - Economics and Information Technology - e society Knowledge and Innovation*, pg 51-58, 26-27 Sep. 2008.

[5] *Arena Gateway 1.1.7 Documentation*, August 2009.

[6] J. Greenfield and K. Short, "Software factories assembling applications with patterns, models, frameworks and tools," *Proceedings of the 18th Annual ACM SIGPLAN Conference OOPSLA2003*, USA, 2003.

[7] Z. J. Brown, "Extended and virtual enterprises—similarities and differences," *International Journal of Agile Manufacturing Systems*, Vol. 1, No. 1, 1999.

[8] K. Kosanke, J. G. Nell, F. Vernadat and M. Zelm, "Enterprise integration—international consensus (EI-IC), în changing the ways we work. Shaping the ICT-solutions for the next century," *Proceedings of the Conference on Integration în Manufacturing*, IOS Press, Sweden, Amsterdam, 1998.

[9] I. Lungu, D. Popescu and A. Velicanu, "Multi Channel Architecture Model Based on Service Oriented Integration," *Informatica Economică Journal*, Vol. 12, No. 3, 2008, pp. 82-86.

[10] A. Bâra, I. Botha, V. Diaconița, I. Lungu, A. Velicanu and M. Velicanu, "A model for Business Intelligence Systems' Development," *Informatica Economică Journal*, Vol. 13, No. 4, 2009, pp. 99-108.

[11] V. Diaconița and I. Botha, "The Level of Information Systems Integration in Romania," *Informatica Economica Journal*, Vol. XI, No. 3 (43)/2007.

[12] *Arena Gateway 1.1.7 Documentation*, August 2009

[13] R. Jardim-Goncalves, A. Grilo, A. Steiger-Garcaox, "Challenging the interoperability between computers în industry with MDA and SOA," *Computers in Industry*, Vol. 57, pp. 679–689, 2006.

[14] L. Bocchi and P. Ciancarini, "On the Impact of Formal Methods in the SOA," *Electronic Notes in Theoretical Computer Science*, Vol. 160, pp. 113–126, 2006.

[15] Web Services Security, "SOAP Message Security 1.1," *WS-Security*, 2004.

[16] V. Diaconita, I. Lungu and A. Bara, "Technical solutions for integrated trading on spot, futures and bonds stock markets (extended version)," *WSEAS Transactions on Information Science and Applications*, Vol. 6, No. 5, 2009, pp. 798-808.

[17] V. Diaconita, I. Lungu and A. Bara, "A practical solution for stock market integration," *Proceedings of the 10th WSEAS international conference on Mathemetics and Computers in Business and Economy (MCBE'09 Prague)*, ISBN 978-960-474-063-5, ISSN 1790-5109, Prague, 2009.

[18] S. Agarwal, S. Lamparter and R. Studer, "Making Web services tradable: A policy-based approach for specifying preferences on Web service properties," *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 7, No. 1, January 2009, pp. 11-20.

[19] J. Domingue, L. Cabral, S. Galizia, V. Tanasescu, A. Gugliotta, B. Norton and C. Pedrinaci, *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 6, No. 2, April 2008, pp. 109-132.

Vlad DIACONIȚA is an Assistant Lecturer at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Bucharest. He has graduated the faculty at which he is now teaching in 2005 and studies for his PhD in the field of Cybernetics and Statistics. He is the co-author of 2 books in the domain of economic informatics, 2 articles in ISI journals, 4 articles in Scopus journals, 4 articles in ISI proceedings, 5 papers in B+ journals and 6 papers in the proceedings of international conferences. He participated as team member in 3 research projects that have been financed from national research programs. He is a member of the IEEE Computer Society and the INFOREC professional association. Domains of competence: Database systems, Data warehouses, OLAP and Business Intelligence, Integrated Systems, SOA.