Integration of Information Technologies in Enterprise Application Development

Iulia SURUGIU
TotalSoft, Bucharest, Romania
isurugiu@totalsoft.ro, iulia.surugiu@yahoo.com

Healthcare enterprises are disconnected. In the era of integrated information systems and Internet explosion, the necessity of information systems integration reside from business process evolution, on the one hand, and from information technology tendencies, on the other hand. In order to become more efficient and adaptive to change, healthcare organizations are tremendously preoccupied of business process automation, flexibility and complexity. The need of information systems integration arise from these goals, explaining, at the same time, the special interest in EAI. Extensible software integration architectures and business orientation of process modeling and information systems functionalities, the same as open-connectivity, accessibility and virtualization lead to most suitable integration solutions: SOA and BPM architectural styles in a cloud computing environment.
Keywords: *EAI (Enterprise Application Integration), BPM (Business Process Management), SOA (Service Oriented Architecture), ESB (Enterprise Service Bus), WOA (Web Oriented Architecture)*

1 Introduction

The importance and the actual state of knowledge regarding the integration of informatic technologies in enterprise application development, rise from the current research on information processes, concepts and technological approaches. The essential mechanisms regarding enterprise software platforms and integration requirements of heterogeneous platforms at different levels (data, functionalities, services, processes), depend on business specific and existing software products. Integration processes should be oriented on capturing the development stages and aspects of enterprise applications on the one hand, and information technology integration processes, on the other.
Basic EAI principles and concepts should be highlighting issues in defining complete and coherent solutions for enterprise application integration.

2. Analysis of enterprise software platforms in relation with business requirements

EAI technologies are emerging, not yet reached a consensus on the ideal approach that businesses should adopt the correct formula and complete integration. Moreover, the processes of definition of original and revolutionary EAI solutions are based on the existing integration solutions, that are ussually optimized and extended in order to meet new requirements and approaches in the field of integration technologies.
The best known commercial EAI solutions are: IBM WebSphere Message Broker, Jonah Technologies, Microsoft BizTalk Server, Oracle BPEL Process Manager, Pervasive Software, SAP Exchange Infrastructure, Sterling Commerce, TIBCO Software, webMethods, Sun Microsystems. EAI open source solutions, often preferred by software integrators, are: OpenESB, Virtuoso Universal Server, Jitterbit Integration Server, JBossESB, Mule.
Table below presents a comparison between different EAI platforms, given some basic criteria: price, implementation, software solution maturity and support.

EAI solution/Criteria	Mule (MuleSource)	BizTalk (Microsoft)	WebSphere (IBM)	OpenESB (SUN)
Price	-	44,228 $ EE[1] 10,138 $ SE[2]	85,000 $ EE 25,000 $ SE	-
Implementation	ESB Messaging[3]	ESB	Messaging Middleware	ESB Messaging
Releases	often	rare	rare	rarely
Training & Support	yes	yes	yes	LogiCoy

Fig. 1. EAI solutions comparison

Concluding on the table, open source EAI solutions provide the same functionality as commercial solutions, also offered support developers is supported by communities or ad-hoc solutions for open source to commercial, which is purchased at cost. In terms of planning versions, open-source solutions tend to be updated more often. My choise in term of integration platform for this research is Mule, an open source solution that offers all the benefits of a mature development integration platform and a proper work environment.

EAI implementations and modularisation of software applications, regardless of business processes addressed, are based on applying some architectural principles in SOA approaches: applications are transformed in independent service bus, ESB architectures are used for application integration, XML is used for data sharing, operational architecture and logical database structure are redesigned, according modularization rules, services can be consumed independently, which allows more flexibility in deployment and software configuration capabilities, compared to solutions for stand-alone applications. Different versions of the integrated system are simplified by EAI modularization schemas, as an effect to a weak binary coupling, while versioning data and interfaces will be compliant to the overrall integrated solution. I will briefly identify some general characteristics of services and processes integration, focusing on the advantages and disadvantages of integrating information technologies in enterprise application development.

Advantages:

- natural and proper software solutions;
- development projects must follow the standards, procedures and working methodologies at all levels;
- independence between services and processes, and also business functionalities;
- project development and production rollout are efficient and easy to handle;
- systems are scalable and availability is maximum, restricted to user types, business processes and business reasons;
- project development life cycle is much reduced, while adapting to new requirements of business process is faster and easier.

Disadvantages:

- hard work on business analysis and technical analysis in the development of projects;
- consistent initial efforts to modify and adapt existing solutions and to implement the new system;
- additional and detailed management activities;
- not all software configuration options are possible for rollouts;

- some business processes that require interoperation may induce a response latency of communication between integrated subsystems.

3. Study of compatibility between different software platforms for enterprise application development

The latest technology approaches in enterprise application integration are SOA, WOA and cloud computing architectures.

Two current concepts and frequently used as an approach to integrate computer technology, cloud computing and technologies based on XML Web Services. Cloud architectures dominate distributed computing solutions that require access to hardware, data and functionalities. Architectures based on XML Web Services technology is the main trend in heterogeneous information integration technologies, the integration of functional capabilities and operational business process flows, the definition of complex models and consistent level of service integration and process. XML Web Services architecture is a starting point for integration of information technology-based processes, as programmable atomic entities underlying the business logic for definition of workflows and business processes.

Established enterprise application development platforms, Java and. NET allow the natural operation of SOA architecture and Web services consumption by implementing specific invokers. Both in Java and .NET platform, XML Web services use, in the context of client applications, is done through remote procedure calls, implemented in Java by invoking RPC on EJBs servers or Servlets and in .NET through listener implementation of type SOAP listener or .NET Remoting. Regardless of the technological approach and operating system being heterogenious, service-oriented architectures offer support for interoperability and integration, encouraging the development of ESB components in various programming languages designed for bussiness application connectors.

Integration architecture based on web services has led to aggregation of strong integrated solutions across business processes and enable design of complex models of business processes, the starting point for new technologies such as grid computing or WOA. In the context of orientation towards web technologies, the topic of cloud computing is the most powerfull approach in software integration, presenting series of capabilities and facilities of such software and hardware architecture models. I will define a comparative study between SOA and WOA architectures, highlighting specific factors of interest.

Table 1. Comparison between SOA and WOA

Criteria/ Architecture	**SOA**	**WOA**
Approach	Focus on simple functionality, but properly implemented.	Focus on simple and natural information technology and architectural concepts.
Reusability	Encourages use of Web services to standardize interoperability.	Opt for simple web technologies at the expense of complex protocols.
Software architecture	SOA services functionality aligns the business processes by mapping and modeling operations specific business logic.	The emphasis is on gradual implementation of functionality for business process modeling as their need arises.
Information technologies	Leverages declarative interfaces and insists on standardization of implementation tools	It promotes the implementation to use standard development tools, but encourages the use of information technology as the best suited typology

		applications tools

The main conclusion is that integration technologies such SOA is one of the most natural EAI solution to implement, while providing the possibility of defining reliable, stable, decoupled, complete architectures and, not least coherent and interoperable, beyond any limitations related software platforms or complex data structures. Exposing SOA integration solution through web, in a cloud computing environment, for instance, lead to extreme accesibility and virtualization power of business processes modeling business functionalities.

4. Development of a methodology for design, implementation and integration of information systems based on heterogeneous software platforms

Following standard EAI methodologies analysis and undertaken research, with the starting point of such process oriented methodology Waterfall TBI, I decided to use an iterative, process based methodology for developing the solution for applications integration. Each stage of software solution development will be functional validated and quality and experimental tested so that the work for the new phase to have the previous results as a starting point. By the very nature of the research project and the subject matter, iterative approach has proven to be an appropriate choice.

I consider that the most appropriate way to develop an integration project would be based on an Agile type EAI methodology, enhanced with RUP (Rational Unified Process) influences, MSF (Microsoft Solution Framework) and XP (eXtreme Programming) techniques. As iterative and incremental process model with object-oriented architecture, I propose RUP influences as a starting point, being suitable to use for a product with a high complexity and dynamism, with complex activities, continuously changing and developing.

Generally, choosing a methodology for developing a software system will be made according to each project specificity and to the nature of the business model. Methodologies and procedures, working rules will be not simply applied step by step, but they will intervene during each phase as required by the current project, as guidelines. Standard methodologies provide only starting points and directions to a complete and perfect methodological approach.

As in this case I combined three software methodologies and I used established templates to define work phases that will optimize the process of software integration solution, in a similar way, in any project, according to a whole work context and also according to factors outlined above, we will choose to use our own and specific methodology, which will be the most appropriate option to meet requirements and project necessities.

In terms of practical usage for application integration solution, I have used the proposed development methodology for an experimental prototype, as following: analysis and design phase has been extended in time so that the bussiness requirements on one hand, and objectives of the integrated system - on the other hand, is completely defined. After defining the technical solution and starting the development and implementation phases, I repeated, in iterative cycles, functional validation and acceptance testing of the modules, so that after completion of one iteration to get to know the real state of the project and to re-estimate total work effort, as well as refactoring / adapting the solution according to the feedback and validation of intermediate versions at the end of iteration. This incrementally work model contributed to greater predictability in project completion and to a smaller refactoring effort as compared to the requirement

changes that would have occurred at the end of implementation, if there were no partial validation, along the way, for atomic features.

Proposals to improve standard software integration methodologies:

- increased understanding of the business area to identify key business processes and the least important processes;
- give greater importance to the implementation of logical flows, described in terms of functional, operational and integrated business flows;
- understanding of data structures - many EAI projects are integrated only at the data level; it is very important to correctly identify data structures, descriptive data schemas and to accordingly apply business principles to determine which data streams coexist in an integrated system, where and why to use these data in order to make data mappings and transformation;
- understanding of the processes is essential to obtain a correct and functional model of the integrated system; project definition is dependent on how business processes are perceived, but also on the accuracy of the data model definition;
- increased attention to developing methods and technologies, rather than work processes;
- identification of user interfaces has great importance for understanding the integration points between disparate applications;
- understanding of events is a key factor for the functional model accuracy of the integrated system; events can be synchronous and, moreover, asynchronous calls in the automated EAI solutions;
- business process modeling using traditional techniques such as UML will refine them as to achieve the process, according to the specific integration project;
- „continous integration" process, especially automated testing in order to validate system quality;
- appropriate use of information technologies in the integration solutions development; it is a basic need to select the best technologies in order to develop integrated solutions, otherwise any mistake is costly in time – an appropriate option would be to build a pilot project;
- integrated system performance and scalability is a major concern with integration projects;
- importance of the prototype installation of intermediate versions of the integrated system, before running the final installation and launching it into production environments;
- maintenance effort and support are considerable in software integration solutions and should be treated separately.

I also made a research on the risks faced by software integration projects, by highlighting some key issues and implementing a risk analysis example for e-Health prototype. I highlighted the major risk factors and proposed a risk management plan, as a measure to reduce the negative impact on implementation processes of an integrated system.

Risk	Probability (1–low, 5-high)	Impact (1–low,5-high)	Exposure degree	Action
Data updating gap	1	3	3	Acceptance
Misuse of the system	2	4	8	Avoidance
Data non-compliance	1	2	2	Prevention
Incomplete business requirements	3	3	9	Contingency plan
System unavailability	1	4	4	Transfer

Fig. 2. Risk assesment

For each of these risks I have probably, given the appropriate points of occurrence and expected impact, causing a degree of exposure of project risks. Prioritizing, then, risks after exposure, we see that the most dangerous risk for the integration project is the lack and incompletion of requirements analysis. For each risk noted in the figure above, I have shortly proposed actions to take in order to minimize negative impact and effects, actions that should be the most appropriate strategy to consider. Extending risk analysis, we can discuss here also about positive risks, events that can bring benefits and new capabilities to project success.

In other words, the proposed methodology was applied naturally in the e-Health prototype, proved to be both a way of control and predictability, offering in the same time operational support for carrying out a complex integration project.

5. Implementation of a cross-platform software system

The proposed solution for enterprise applications integration is located at the middleware level and aims to integrate functionalities and heterogeneous data sources, by interoperation of heterogeneous information technology, data sources of different origins and different business functionalities. Also, in the practical experiment I will emphasize the use of uniform policies for communication between applications, by implementing software development over established protocols and standards. Case study performed and presented in this paper starts from the goal of systems integration process, the complexity of its component phases, meaning the creation of an integrated information system, having at least the following attributes: comprehensive, coherent, independent of geographical, social, national or business processes logic limitations.

I will describe in this section a platform for software systems integration, developed in Mule, which aims to highlight aspects of the application integration, such as: integration at data sources level, service based integration, business processes integration, through implementation of components focused on the integration of data, services and processes. I choose Mule platform for software prototype development because it is an efficient, complex and frequently used development environment for integration of information technologies. To illustrate the theoretical and practical concepts presented in previous sections, according to the implementation which is further described in detail, I present the prototype of an integrated system for e-Health. From functional point of view, the experiment represents an interoperability model between a medical system and financial system. In terms of working context, the integration prototype combines development technologies / platforms like Java, .NET, Microsoft SQL Server, MySQL, Mule, Apache, web-service based architecture (SOA, WOA), service and process based,

ODRA, ORM. The project aims to achieve an integrated sistem and consistency at the level of data, services and processes, by combining and integrating heterogeneous information technologies.

Besides service level integration prototype and specific calls that orchestrate the communication between the two heterogeneous systems, the next level of complexity is demonstrating a non-trivial mechanism of communication through messages. I will present architectural and functional model underlying messaging communication between software systems located on both sides of the enterprise web service bus.

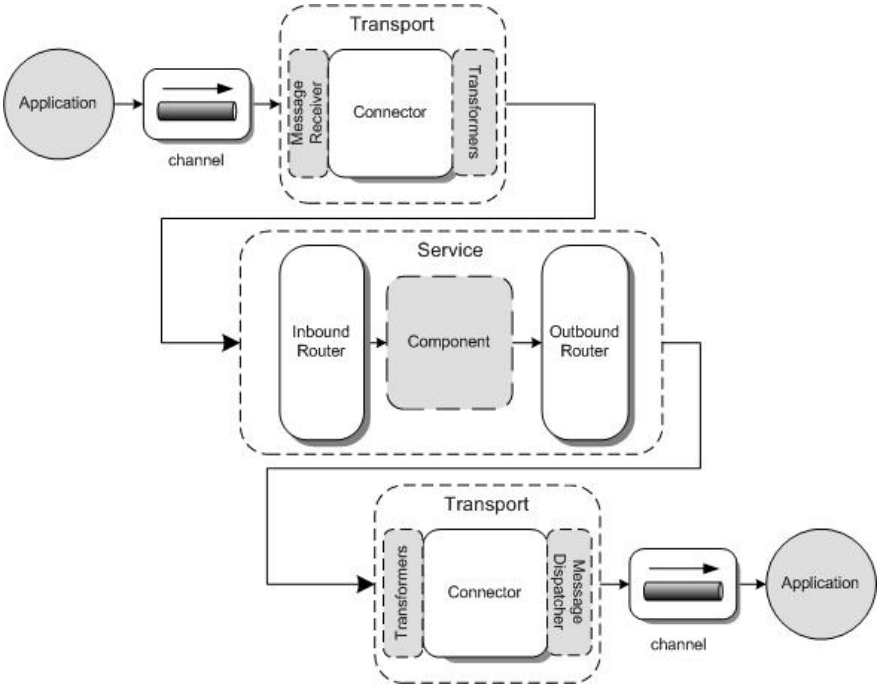

Fig. 3. Mule architecture

Mule supports transport protocols used in distributed programming, WS-* standards, JMS, File, SMTP. Client applications publish client requests on pre-defined communication channels. Requests are processed by the integration server application through services. Client applications access the services of ESB through secure communication channels and the flow transfer is done by specific invocations of connectors which implements the connection between client and server calls integration. Processing activities that the integration server performs in order to operate on incoming messages, for communication reasons with other client applications, are a result of transformation calls that specialized components perform inside ESB. ESB messages arriving through protocols / mechanisms of transport services are configured in Mule and processed on the application server side. A service component includes two processing end-points: an entry point and one exit point. The entry point is an endpoint that specifies the communication channel can be used, what messages can be transmitted and how to transfer such calls to Web services through a .NET proxy or a through a JMS message queue.

As a case study based on theoretical elements on integration of information technology and XML web services already mentioned, I propose a prototype implementation of an integrated system through web services technologies and Web 2.0 standards. This practical experiment

highlights ways to integrate enterprise applications based on integration of information technologies, in addition to a complex architecture based on an integration server. In this respect, I followed techniques and integration between Java and .NET platforms at the level of database, functionality and user interface applications. Working context is given by an architecture based on XML Web services implemented using Java platform. Data sources are heterogeneous, relational DBMS, MySQL and Microsoft SQL Server and will be integrated into a global catalog on Microsoft SQL Server. The purpose of the prototype is to demonstrate how we can integrate up to the level of services, applications built on heterogeneous data sources.

Coming back to the e-Health integration prototype, the business flows architecture can be described as following: in the medical system we have dictionaries of patients, delivered medical services under financial articles mapping and information on financial status of the services performed by physicians. On the other hand, the financial system have invoiceable items – corresponding to medical services from healthcare application – along with prices and financial attributes, customers - corresponding to patients from healthcare application, financial transactions - the equivalent of medical consultations, financial form and generated financial documents - invoices and details.

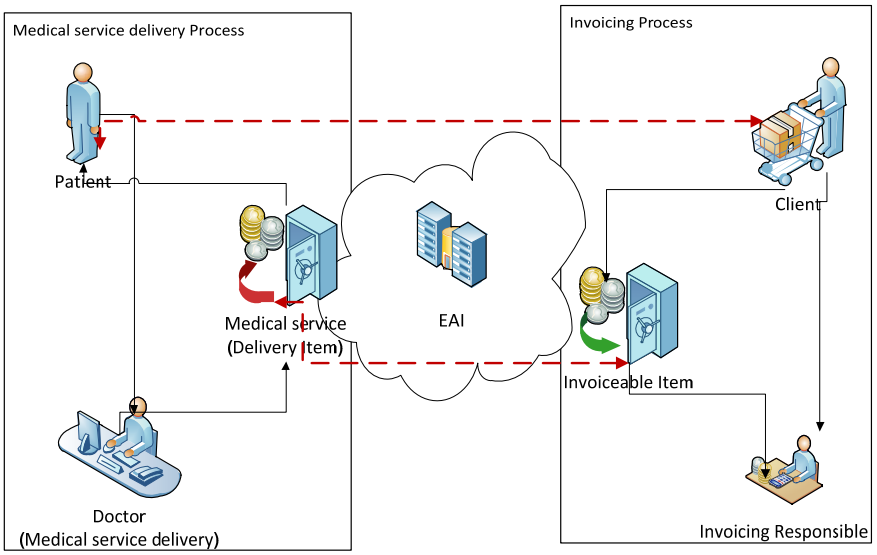

Fig. 4. Business flow design

Medical system manages entities like patients and consultations and receives from the financial system billable items, deliverable items/services, notification about patient billing / payment. At the other endpoint of the enterprise service bus there are integration services and business processes of the financial system, which manages items and bills and communicate both with the medical system for receiving patients as the equivalent of customers and for intercepting transaction execution as the deliverance of medical services in the consultations.

Using this functional model, I have designed an integrated system implemented in Mule, using some design principles to work in the following integration schema: declarative addressing and XML configuration for service invoker, SOAP formatting messages sent inside the service bus and web proxy service calls to end-point. I used the .NET environment developer tools and Visual Studio 2010 to generate references to proxy for Web Services.

After generating web service brokers, client applications generate in the form of .NET proxy, on the one hand, and within ESB, Apache proxy, on the other hand, the distributed communication architecture is the following: a client application, i.e. the medical system, calls a local method to process entity "delivered service into consultation"; local method calls a web service exposed by Mule integration server, it makes medical entity to be passed internally within the service bus, which will be given to a web service component type to the other endpoint of the integration server. This service will call an internal method of the other client application, i.e. the financial system, that will process information from another business perspective: invoicing patient medical services. Between the exchanges of messages in the web service bus, working entities are subject to transformations and serialized/de-serialized. Generic objects are serialized in SOAP messages, then transformed into XML data, SOAP and retransmitted as serializable objects to client applications, to be deserialized for internal usage purposes.

Software architecture used for exposing functionalities of business processes as web services is based on a SOA virtualization engine, which allows the definition of services that can be accessed via FTP, XFB, EJB, JMS, SOAP / HTTP. The access to web services is done through UDDI or graphical interfaces provided by the development environment tools, the main purpose of SOA being service re-usability. UDDI registry allow the attachment of metadata objects that virtualize access to services, such as status or version identifiers for each called service instance.

The conclusion of the project of integrating information technologies Java and .NET using the Mule platform, is that I have proposed an efficient and interesting way of building a reliable, scalable and extensible prototype for enterprise application integration. The project can be easily extrapolated and adapted to complex business processes, in order to achieve a fully integrated IT solution. Developing this prototype I reached to demonstrate an original formula for a solution of integrating information technologies for enterprise application development.

6. Business process modelling and optimization based on integrated software systems and workflow analysis

The need for enterprise application integration result from the need for networking and communication between different types of applications. Relative to disconnected systems in terms of costs versus benefits, there are two solutions to meet the need to integrate existing software products: rewriting them for intercommunication or software integration for their existing IT solutions - that means integration through coupling / decoupling mechanisms governed by EAI principles.

Various aspects of business processes modelling and optimization in terms of information systems, start from the use of existing information systems, restructured/refactored systems or integrated systems.
Based on the comparative approach in terms of cost and efficiency, depending on circumstances and context of business (designed business area, dynamism of the business domain, scale and complexity of software product, architectural design of the underlying software systems, the degree of use and efficiency) will be chosen one of these options: to rewrite existing software products, according to new requirements, or to define a solution for enterprise application integration, starting fro legacy applications.
Whatever the choice made, system development related activities require additional resources and maintenance efforts. As the level of abstraction of a system is higher, the more simple the process of reengineering becomes and there will be available more information about program and data structure, models of business entities, design models for data and control flow, UML class diagrams, states, actions, workflows.
The basic conclusion is that systems require additional efforts and extensible, automated architectures, in order to maintain and simplify the structure and to deal with the dynamism of business environment. Any chosen solution must be optimal no matter the considerents taken into account. In particular, integrated software solutions are subject to software reengineering, in order to optimize business processes and to better exploit the new system.
I will define, based on personal experience and information systems efficiency theories, an own assessment of the effectiveness system for an enterprise application integration, in the medical field. The main advantages of integrated information system usage are:

- removes difficulties and low speed in the operation of data, restricting users to be proficient;
- offers real time data acces and operation;
- removes multiple data operations and the need of greater allocation in order to check data consistency;
- eliminates gaps in information system, information redundancy and long time development of various internal processes (billing, contracting, commissioning, monthly reporting);
- reduces cumbersome tracking business costs distributed on cost center and budget lines, allocated to business size.

Conclusions

Research done in integrating enterprise applications, as described in this paper, captures more fundamental aspects, theoretical and practical, existing and new concepts of EAI. Based on existing theories, but also practical and methodological aspects of the existing implementations of EAI, I defined a solution for enterprise application integration on multiple descriptive levels: software product development methodology, technical implementation and functional analysis. I also stressed the context of these approaches, methods of optimization of business processes through an integrated approach, options for achieving a fully integrated information system respecting the principle of maximizing efficiency. Based on personal experience, but also on standard theories in software integration area, I outlined the advantages and disadvantages of various solutions and I have highlighted generally accepted principles in case of dynamic software systems over a certain level of complexity. Software prototype for enterprise applications integration for e-Health was designed and developed with the latest information technologies and it is a functional system that demonstrates the practical and experimental utility, of an integrated information system, built with a personal vision.

References

[1] Iulia Surugiu, *Integration of Information Technologies in Enterprise Application Development*, PhD. Thesis, 2012

[2] Faisal Aburub, Mohammed Odeh and Ian Beeson, *Modelling non-functional requirements of business processes*, Information and Software Technology, Issue 49, 2007, pp. 1162–1171, ISSN: 0950-5849

[3] Joseph Barjis, *The importance of business process modeling in software systems design*, Science of Computer Programming, Issue 71, 2008, pp. 73–87, ISSN: 0167-6423

[4] Bih J., *Service Oriented Architecture (SOA) - A New Paradigm to Implement Dynamic E-business Solutions*, ACM Ubiquity, Vol. 7, Issue 30, 2006, ISSN: 1530-2180

[5] S. Chandra, J. de Vries, J. Field, H. Hess, M. Kalidasan, K. V. Raghavan, F. Nieuwerth, G. Ramalingam, J. Xue, *Using logical data models for understanding and transforming legacy business applications*, IBM Systems Journal, Vol 45, Issue 3, pp. 647-655, 2006, ISSN 0018-8670

[6] David Dossot, John D'Emic, *Mule in action,* Manning Publications, Greenwich, 2009, ISBN 978-1-933988-96-2

[7] Kevin B. Hendricks, Vinod R. Singhal, Jeff K. Stratman, *The impact of enterprise systems on corporate performance: A study of ERP, SCM, and CRM system implementations*, Journal of Operations Management, Issue 25, 2007, pp. 65–82, ISSN: 0272-6963

[8] Xin Jin, *Research on the Model of Enterprise Application Integration with Web Services*, Proceedings of the 3rd WSEAS International Conference on COMPUTER ENGINEERING and APPLICATIONS (CEA 2009), ISSN: 1790-5117, ISBN: 978-960-474-41-3

[9] Shahed Latif, *Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance (Theory in Practice)*, O'Reilly Media, 2009, ISBN:0596802765.

Iulia SURUGIU has graduated The Bucharest Academy of Economic Studies (Romania), Faculty of Cybernetics, Statistics and Economic Informatics in 2007. She is a Ph.D. Candidate in Economic Informatics with the Doctor's Degree Thesis: "Integration of Information Technologies in Enterprise Applications Development". Her research activity can be observed in scientific papers and proceedings achievements, among which:

- "Information Technology Standards – A Viable Solution to Reach the Performance", The 12th WSEAS International Conference on Automation & Information (ICAI '11), April 2011, Brasov, Romania;
- "Information Systems Integration, A New Trend In Business", The 10th WSEAS International Conference on Applications of Computer Engineering (ACE'11), March 2011, Spain;
- "Ways to Increase the Efficiency of Information Systems", The 10th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Databases (AIKED'11), February 2011, Cambridge, UK;
- "Advanced e-Government Information Service Bus (eGov-Bus)", The 9th International Conference on Informatics in Economy, May 2009, Bucharest, Romania;

- "Academic Research Information System", Journal: JAQM (Journal Of Applied Quantitative Methods), issue 1, vol. 3, 2008;
- "Medical Virtual Public Services", Journal: Informatics Economics, issue 1(45), 2008.