ACADEMIC RESEARCH INTEGRATION SYSTEM

Iulia SURUGIU[1]
PhD Candidate, University of Economics, Bucharest, Romania

E-mail: : iulia_surugiu2003@yahoo.com

Manole VELICANU
PhD, University Professor, Department of Computer Science
University of Economics, Bucharest, Romania

E-mail: mvelicanu@ase.ro

Abstract: *This paper comprises results concluding the research activity done so far regarding enhanced web services and system integration. The objective of the paper is to define the software architecture for a coherent framework and methodology for enhancing existing web services into an integrated system. This document presents the research work that has been done so far in this aspect by applying the proposed architecture for system integration in the academic field. The basics concepts used are Enterprise Application Integration (EAI) and Service Oriented Architecture (SOA), as the most commonly used approaches to information systems integration lately.*
SOA is an Information Technology (IT) architectural style that supports the transformation of businesses into a set of linked services or repeatable business tasks that can be accessed when needed over a network. This may be a local network, it may be the Internet, or it may be geographically and technologically diverse, combining services in New York, Paris, and Beijing as though they were all installed on a local desktop. These services can coalesce to accomplish a specific business task, enabling businesses to adapt quickly to changing conditions and requirements.

Key words: *Service Oriented Architecture; Enterprise Application Integration; Business Process Management*

1. Introduction

One of the most important activities inside a university is scientific research. Academy accreditation is one of the main goals for university, as it attests the quality and performances of the academic activities and its position through similar educational institutions. I consider tremendously important to mention here a phrase we read once in Lewis Carroll's book, *Alice in Wonderland*, which we found interesting to apply in academic field: "*If you want to stay where you are, you must keep moving*". In my opinion, the management of information is the only key factor capable to assure success continuality and

real performances. Education and research are fields where science facts grow with astonishing dynamicity and flexibility and adaptive characters become vital for surviving.

Research activities grave the name and fame of the university on educational international context through its member submitted, accepted and pleaded papers, approved and appreciated research projects, the quality and properness of scientific research results, the viability of proposed solutions.

The first step in organizing research is being informed about conferences, symposiums, workshops, scientific research magazines and ISI quoted publications that allow an efficient dissemination of the results of research and attract investors and research financing, another important piece in the activity of research.

Given the previously mentioned context, my proposal is to create an integrated system that allows educational and scientific research services recovery. The target user group of the system is academic environment and research institutes that need quick and organized information on scientific research projects and related research topics.

Wouldn't be useful for a university to have access to a system that, given the present situation of the academy, to come up with a complete and coherent program about all the activities that need to be done in order to obtain an accreditation? All that the user of the system must do is to authenticate with the system, specify the university traits and its educational background features, maybe other distinctions that the academy already has gained, then choose a program that the academy wants to attend, eventually a commission for evaluation or an accreditation institute. The result will be a list of steps to make in order to achieve the goal, which could be: leadership of European research projects, involvements in educational projects financed by governments, usage of required standardization politics, papers and documents given to evidence to certain international educational and/or research organisms.

All these tasks and activities will be identified using Web Services, the existing ones or some others wrapped onto existing applications that have the permission to integrate with our system. Service recovery and integration will be achieved by creating a web services repository that will serve client requests by wrapped web responses.

2. System Architecture

We will present further a detailed architecture of a Web Services Bus, proposed as a whole in another personal paper called "*Health Information Service Bus*", being in course of publishing to "*Informatica Economica*" journal**.**

In general, a UDDI registry is quite focused on Web Services. Users or applications describe Web Services inside the registry using the UDDI interface (based on Web Services). The description of a Web Service inside a UDDI registry covers different aspects such as where is located its WSDL and/or its access point, or what metadata is attached to a service or to its access points.

The Virtualization Engine extends the concept of services to other middleware than just Web Services. It is possible to define services that can be accessed through FTP, XFB, EJB, JMS, SOAP/HTTP … The ways to access it are still the UDDI Web Services and the GUI. The major goal of SOA is service reusability. A described service can be reused if people know about it and if they can find it inside the registry. UDDI allows attaching metadata to the objects managed by the registry in order to ease the search operations. The Virtualization

Engine comes with a pre-defined set of categorization identifiers. There identifiers can be used to describe in a standardized way the metadata that can be attached to a service (for example its status or the version of one of its instances). They can allow the users to tell if a particular service is in production or not, what is its version number, what kind of middleware it uses and many other information.

As described inside the UDDI standard, the categorization data can be added to any level of the UDDI structure. The Virtualization Engine will use specific metadata that will be inserted at the Service level and at the Instance (access point) level.

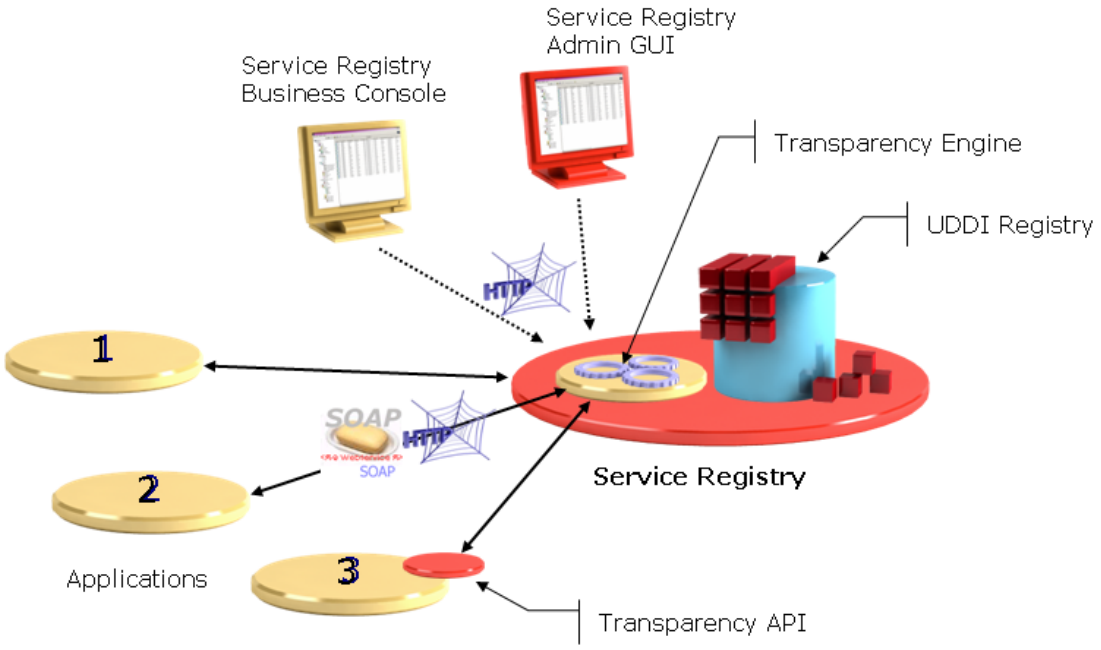

Figure 1. Virtualization Engine

2.1. Architectural overview

The proposed solution is the most obvious for the situations where there are already implemented web services for education and research topics, because it will require minimal but strict modifications on the existing web services. Still, some Web Services (WS) standards might not be directly supported and other problems with standardization could appear and must be solved by modifying existing service structure.

The main disadvantage of the integration solution would be that:

- Not all services fulfill framework requirements.
- Requires implementation effort to cope with unsupported standards.

An architectural overview of the solution proposed would be the following:

- The framework integrates e-Research and e-Learning services and applications through the Enhanced Web Service Framework
- Each e-Research service is published into the UDDI Registry which is running on the integration server
- The UDDI Registry stores metadata into the UDDI Repository along with state
- When a request is made, the public portal calls the front-end Web Services Engine using its service client mode

- The Web Services Engine calls the Virtualization Engine to get the best instance for the requested service
- This returns an endpoint using registry metadata or raises an exception if none was found
- The front-end server forwards the initial call to the given back-end instance which has another Web Services Engine working as a service provider
- If the service is fully compliant, then it will be invoked directly
- If it is partially compliant, then it will be invoked using the Relay Proxy to insure compliancy
- If it is a legacy application, first it will be published as a Web service using the Service Integration Layer and then invoked through the Relay Proxy
- Each Web Services Engine will log its service activity using the Auditing and Monitoring Engine, which updates the global UDDI Repository and its Registry
- Both the front-end and the back-end servers are controlled by the Management Engine, which is used also to manage their Web Services Engines and Auditing and Monitoring Engine

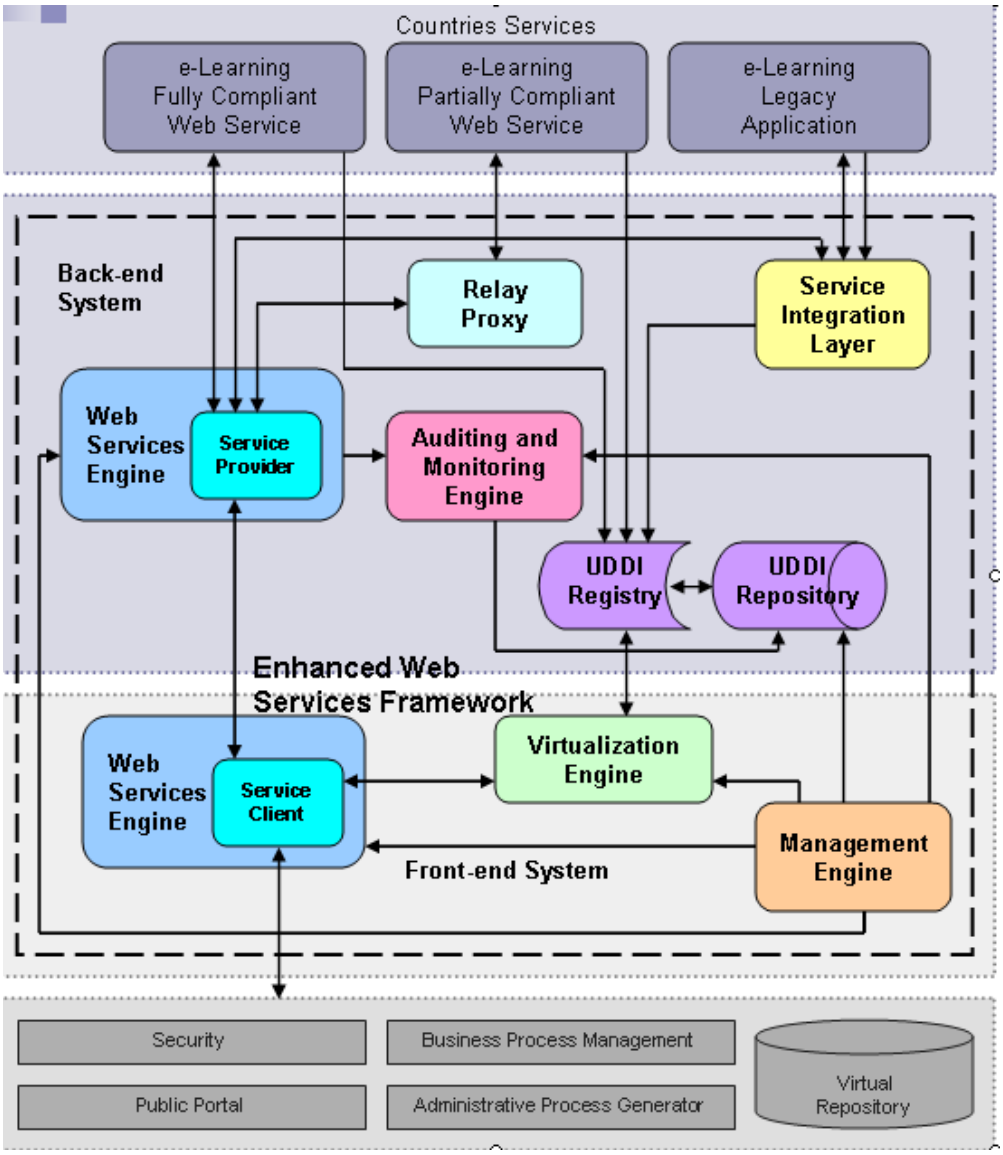

Figure 2. Architectural overview

2.2. Securing Communication with Server Relay Mode

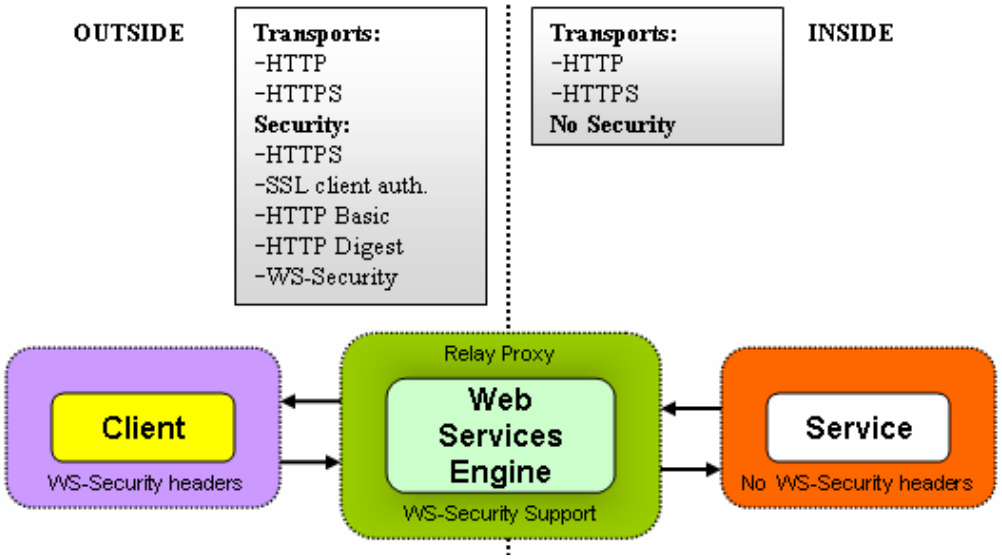

Figure 3. Security with Server Relay Mode

- The client produces the SOAP message with security headers and sends it to the Service Security Proxy.
- The Service Security Proxy strips out security headers from the message and forwards pure SOAP message to the destination Service, injecting security tokens into custom headers.
- When the response arrives, the server extracts custom headers and applies security policy.
- Then, the server delivers the response message to the client.

2.3. Securing Communication with Client Relay Mode

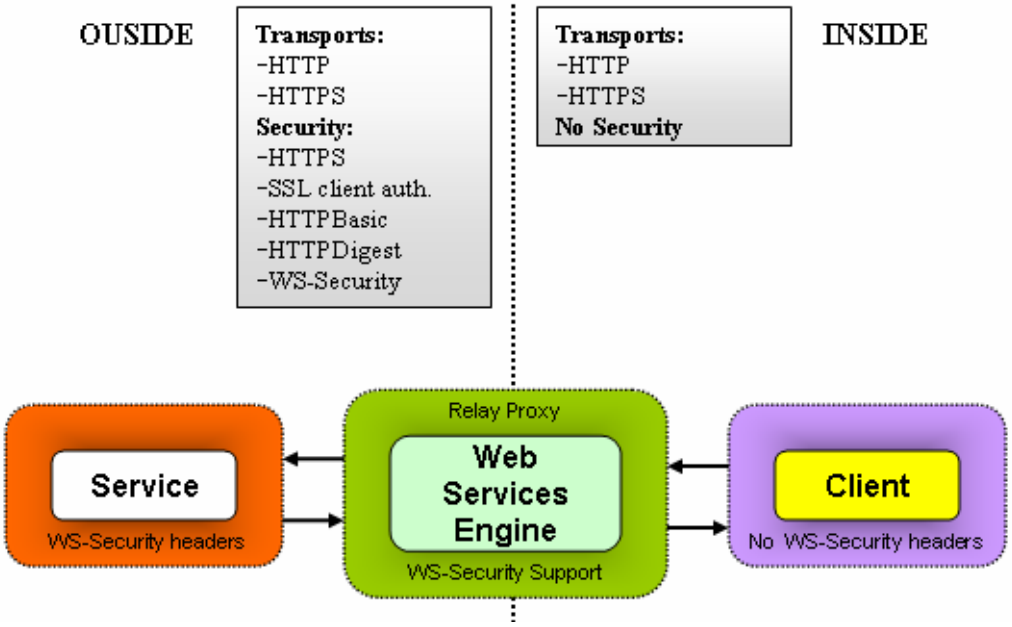

Figure 4. Security with Client Relay Mode

- The client produces the SOAP message with custom headers containing security tokens and sends it to the Service Security Proxy.
- The Service Security Proxy extracts custom headers from the message, applies security policy and forwards SOAP message to the destination service
- When the response arrives, the server strips out security headers and checks security policy
- Then, the server delivers the response to the client

2.4. Managing Communication with Server Relay Mode

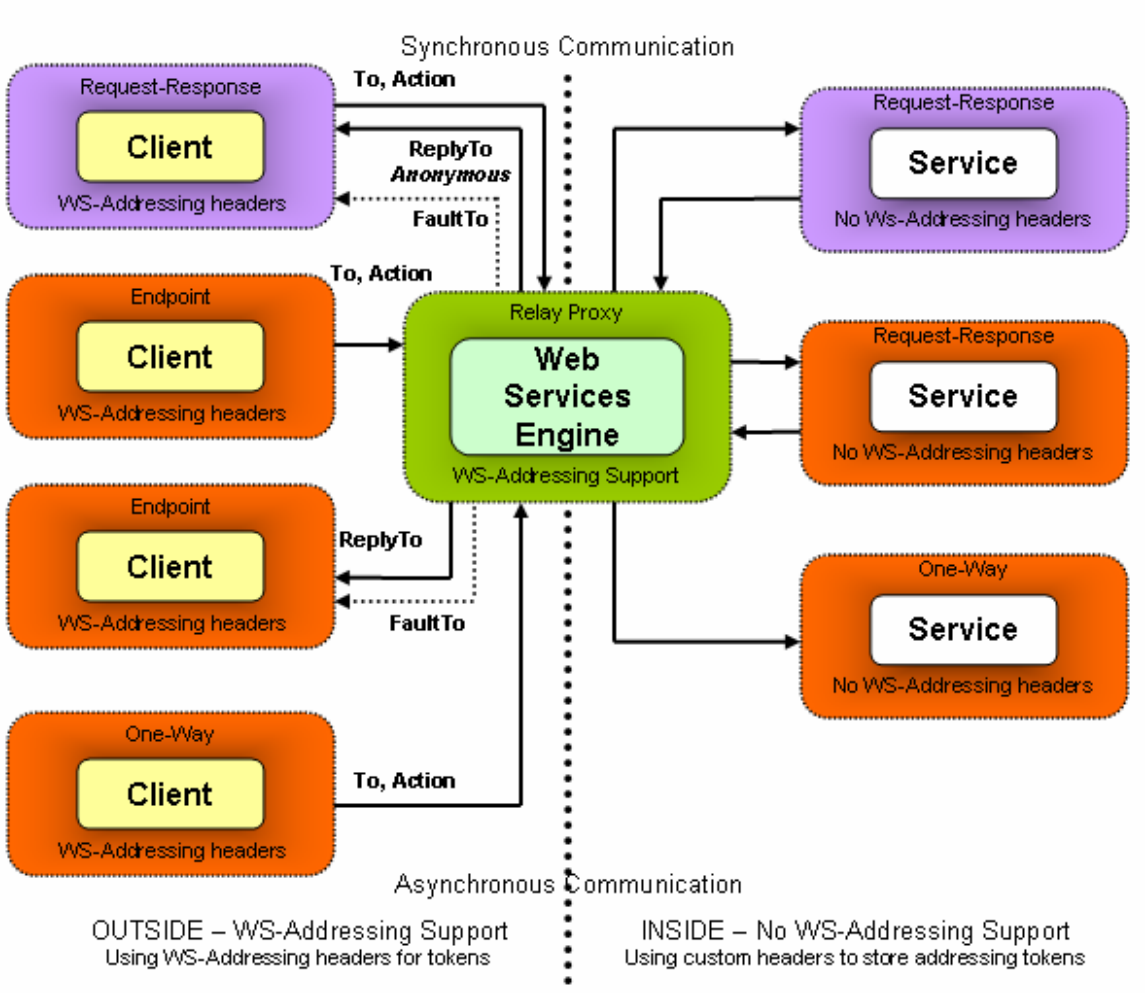

Figure 5. Communication with Server Relay Mode

Asynchronous Communication:

- Client sends a SOAP message to the server containing addressing headers, having a *ReplyTo* other than the anonymous URI
- Server translates addressing headers into custom headers, and then invokes the service
- Service sends back a response message to the server forwarding custom headers with addressing tokens
- Server extracts addressing tokens and uses them to delivery the SOAP message to ReplyTo endpoint
- If the response was a SOAP fault, then the server delivers a it to the *FaultTo* endpoint
- Client can send a One-Way message containing addressing headers

- Server strips out the addressing headers and forwards the One-Way call to the service

Synchronous Communication:

- Client sends a SOAP message to the server containing addressing headers, including an anonymous URI for ReplyTo
- Server translates addressing headers into custom headers, and then invokes the service
- Service sends back a response message to the server forwarding custom headers with addressing tokens
- Server extracts addressing tokens and uses them to deliver the SOAP message to ReplyTo, the same with the initial client endpoint
- If the response was a SOAP fault, then the server delivers it to the FaultTo endpoint

2.5. Managing Communication with Client Relay Mode

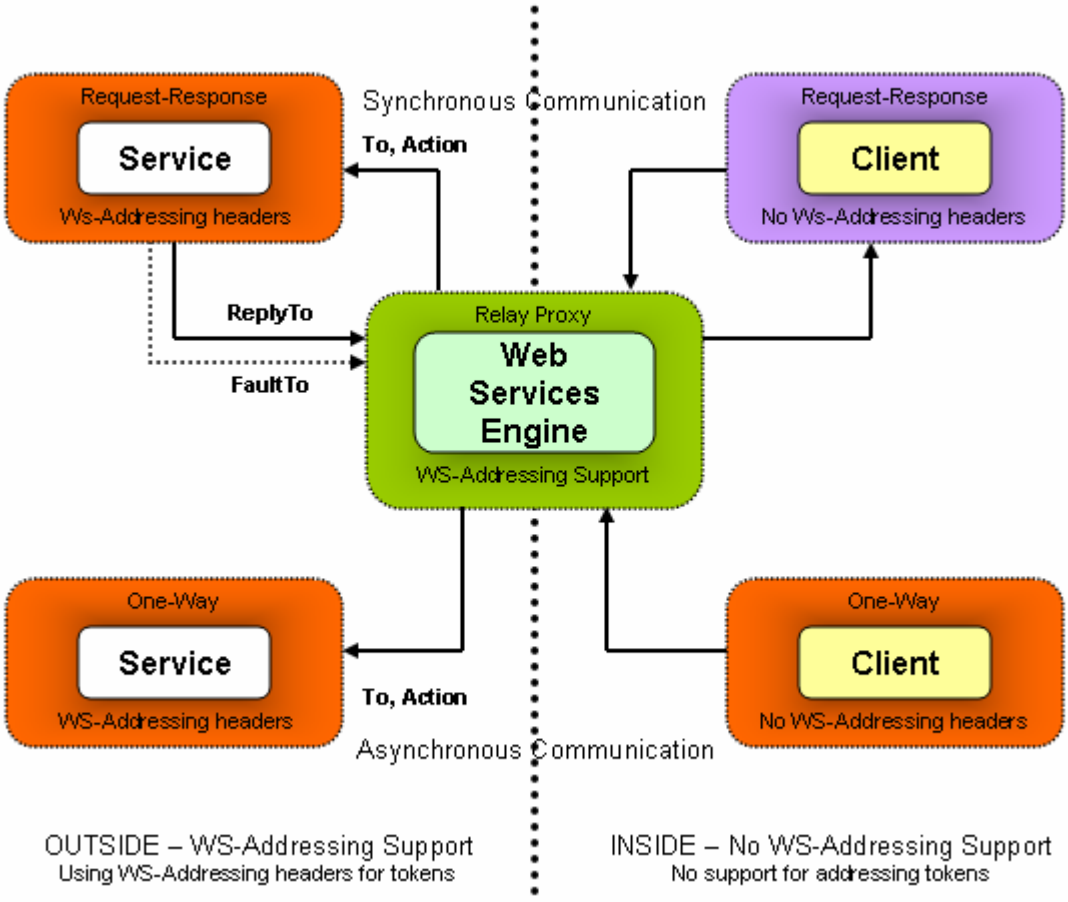

Figure 6. Communication with Client Relay Mode

Asynchronous Communication:

- Client sends a One-Way message to the server without addressing headers
- Server injects addressing headers and forwards the One-Way message to the service

Synchronous Communication:

- Client sends a Request-Response message to the server without addressing headers
- Server injects addressing headers, setting its endpoint as the target of the response and invokes the service

- Service sends back the response message to the server, being the reply endpoint
- Server delivers the response to the client endpoint
- If the response was a SOAP fault, then the server delivers it to the client endpoint

3. Conclusions

As different academic research activities are spread all over the Internet through Web sites, Web Services and portals, we found a way to bring all this information together inside an integrated system in order to be used by universities when involving into scientific research activities. In order to achieve that, it is needed help and support from the research service providers, that must allow access to their services and cooperation in case already existing systems need to be modified for being integrated.

The main objectives set for the proposed WS integration architecture are the following:

- Combining several Web Services additional standards and ad hoc data structures to ensure management of contextual information in order to provide Business Process Management (BPM) support, audit trail and error management.
- Asynchronous behaviour of the Web Services
- Defining a balance between Web Services security capabilities and the underlying security architecture
- Providing location transparency of Web Services in order to provide redundancy and scalability
- Proposing new ways to present Web Services inside a service registry in order to make them easier to understand to business analysts
- Checking with selected technology providers that their implementation of the Web Services standards (basic & additional) allow perfect interoperability within the functional perimeter defined for Web Services usage inside the proposed architecture.

As research and educational services are very heterogeneous and universities disconnected, the architecture proposed by this paper comes as a lien between universities and research institutions and commissions for educational evaluation. The combination of information technologies and concepts such as Web Services, Service Oriented Architecture, Business Process Management, in the way presented, leads to the integration of all those services in order to provide academies with easier and efficient access to a unified educational and research system, no matter the position from which they use system's services.

4. References

1. Sadter, C, Huber, P. and Sangmin, Y. **Enabling SOA Using WebSphere Messaging**, IBM, March 2006
2. Stamos, A. and Stender, S. **Web Services Security – Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0**, Microsoft, November 2005
3. Surugiu, I. **Health Information Service Bus**, sent to publication to "Informatica Economica" Journal, University of Economics, Bucharest, Romania

4. * * * **Business Process Management,** 4th International Conference, BPM 2006, Vienna, Austria
5. * * * **Patterns: SOA Foundation Service Creation Scenario,** IBM publication
6. * * * **Web Service Atomic Transaction Protocol (WS-AtomicTransaction)**, http://www-128.ibm.com/developerworks/library/specification/ws-tx/#atom
7. * * * **Web Services Addressing Protocol (WS-Addresing)**, http://www.w3.org/Submission/ws-addressing
8. * * * **Web Services Secure Conversation Protocol (WS-SecureConversation),** http://specs.xmlsoap.org/ws/2005/02/sc/WS-SecureConversation.pdf
9. * * * **Web Services Security Protocol (WS-Security)**, http://www.verisign.com/wss/wss.pdf
10. * * * **Web Services Trust Protocol (WS-Trust)**, http://www.verisign.com/wss/WS-Trust.pdf

[1] Iulia Surugiu, Software Engineer, TotalSoft S.A., Bucharest, Romania