WEB-SERVICES FOR EXCHANGE OF DATA ON COOPERATION AND MOBILITY BETWEEN HIGHER EDUCATION INSTITUTIONS

Fabio Arcella[2], Janina Mincer-Daszkiewicz[1], Simone Ravaioli[2]

[1]University of Warsaw, Banacha 2, Warszawa, 00-927 Poland

[2]KION, Via Cristoni 84, Casalecchio di Reno, RA 40033 Italy.

Keywords
Student and staff mobility, web-services, data exchange, student management information systems

1. EXECUTIVE SUMMARY

Student and staff mobility is one of the most important priorities of the European Higher Education Area promoted by the Bologna process. According to the Mobility Barometer the most recent posted statistics (for academic years 2005 and 2006) for Italy and Poland are quite substantial: the number of students incoming to Poland in 2005 was 11 365, to Italy in 2006 was 48 771, the number of students outgoing from Poland in 2006 was 31 132, from Italy in 2005 was 33 855.

Even these old numbers give some idea on the huge amount of letters, emails, faxes which had to be sent to make it happen, not mentioning the time and cost involved. In the age of the Internet, web services and digital signatures, European universities keep data concerning students and study programs in an electronic form, yet exchange it on paper.

What is needed to change the way in which information is handled? Applications can transfer data electronically through web-services architecture; however, they need common formats for data concerning cooperation and mobility. There are on-going initiatives aiming to set European standards for student digital data portability (see e.g. joint works by CEN/WSLT and RS3G) and although an official European Norm has not been released so far, the specifications already produced might serve as a starting point.

Authors of this paper come from two countries and represent two European Higher Education consortia – MUCI in Poland and CINECA in Italy – which develop software for their domestic higher education institutions. After meeting at Eunis 2008 MUCI and CINECA agreed to engage in a project to develop a prototype infrastructure to send data about student mobility between the two systems.

The project started with setting format for data like *HEI information record*, *Bilateral agreement between two HEIs*, *Students nominated for mobility*, *Learning agreement and Transcript of records of a student*. Then WSDL file for web services was defined. The format was tested by preparation of XML files with sample data from the University of Warsaw in Poland and Parma University in Italy. WSDL was used to generate web-server and web-client for testing data transfer. Each part has to implement middleware between the web-server and the local database. Various solutions are possible, also in respect of how the exchanged data is being handled inside the local system. UDDI registry was set up for keeping binding information.

The experiment is in a very early stage, but results look promising. Prototype implementation already enables to send data between the two systems and to test transfer for any set of data stored on the test web-server and web-client. Open questions have been recognized and possible solutions are discussed. There is an issue of scalability and security if we think about data exchange between all European HEIs. The problem of standard for data format should be solved by organizations like CEN and RS3G but the RAD-like (or as some say *quick and dirty*) approach of this project may help in setting some practical perspective. The idea is to call out for participation to other consortia and/or student systems implementers to join this project, thereby achieving critical economies of scale, which can result in a substantial impact toward the creation of the European Higher Education Area.

2. INTRODUCTION

Student and staff mobility is one of the most important priorities of the **European Higher Education Area** promoted by the Bologna process. Let us have a look at some numbers. According to the Mobility Barometer available on the web (http://www.letsgocampaign.net/) the statistics for Italy and Poland, home countries of the authors of this paper, are as shown in the following table:

	Incoming students		**Outgoing students**	
	Year	Number	Year	Number
Poland	2005	11 365	2006	31 132
Italy	2006	48 771	2005	33 855

Newer data are not yet officially available but may be expected to be much higher.

Data found in the information portal of the European Commission in Education & Training Section (http://ec.europa.eu/education/programmes/llp/erasmus/stat_en.html) present long-term trend for Erasmus programme which is the most important European project promoting mobility. The curve showing student mobility during the period 1987/88 – 2006/07 grows almost linearly from 3 244 to 159 324.

Even this three-year old sample data gives us some idea on effort, time and cost involved in management of mobility between higher education institutions in Europe. Unfortunately this effort is perceived by traveling students and academic teachers as bureaucratic, and the huge amount of letters, emails, faxes to be delivered and sent between involved parties is regarded as an obstacle which makes good ideas cumbersome and tedious in implementation. No wonder that recommendations for national and academic-level agencies, formulated in a summary of **2008 Study on the Impact of Erasmus on European Higher Education: Quality, Openness and International-isation** (http://ec.europa.eu/education/erasmus/doc/publ/impact08.pdf) stress importance of reducing administrative requirements and bureaucracy associated with the participation in international programmes and in particular around mobility, supporting simpler, more efficient and uniform procedures.

In our opinion this literally means less paperwork and more automatic handling of data exchange. Proposals of bilateral agreements circulate on paper between partner institutions to be finally signed and entered independently into both electronic systems. Students wanting to perform part of studies abroad are asked to deliver – again on paper – their history of academic achievements, courses passed, grades. Their personal data are obtained from the local student management system, to be printed and sent by fax to the receiving institution, where they are retyped and entered to the other database. Paper lists of nominated students circulate between partner institutions before being finally approved. Learning agreements are prepared by students on paper, approved and signed by coordinators from both involved institutions. Transcript of records (in fact the same learning agreements but with grades attached) are printed, signed and sent back to the home university of the student to be retyped there again to become a part of student's history and finally be printed on a diploma supplement.

In the age of the Internet, web services and digital signatures, European universities keep data concerning students and study programs in an electronic form, yet exchange it on paper.

What is needed to change the way in which information is handled? Applications can transfer data electronically through web-services architecture; however, they need common formats for data concerning cooperation and mobility. There are on-going initiatives aiming to set European standards for student digital data portability (see e.g. joint works by CEN/WSLT and RS3G) and although an official European Norm has not been released so far, the specifications already produced might serve as a starting point.

Authors of this paper come from two countries and represent two consortia – MUCI in Poland and CINECA in Italy – which develop software for their domestic higher education institutions –

University Study-Oriented System (USOS) in Poland [USOS] and ESSE3 in Italy. After meeting at Eunis 2008 where they participated in sessions on *Bologna, standards and integration*, MUCI and CINECA agreed to engage in a project to develop a prototype infrastructure for sending data about student mobility between the two systems.

In the chapters to follow we describe the most important steps and issues involved in building a prototype distributed web-based system for electronic exchange of data on cooperation and mobility between any higher education institutions.

3. PROCESSES INVOLVED IN INTERNATIONAL COOPERATION AND MOBILITY

The project started with recognizing sets of data being exchanged between cooperating partner institutions. They imply directly from the main business areas of International Relations Offices, which are agenda inside universities responsible for international programmes: cooperation and agreement management (governmental, bilateral and multilateral agreements, educational programmes), and mobility management (short-term arrivals and departures of students, academic teachers, and other staff, for studies, internships, teaching, research) [JMD2008]. The mentioned processes involve the following sets of data:

1. International cooperation.

   Cooperation conditions of the agreements between partner institutions concerning short-term studies are a source of important data which settle context for all mobility activities. Sets of data comprise:

   a. HEI description like names of both institutions, personal data of coordinators (administrative and institutional), addresses, Erasmus codes etc.

   b. Agreement data – description of both partner institutions and data on cooperation conditions, like Socrates codes of disciplines, unit coordinators, number of places and months for each study program level (bachelor, master, doctoral).

2. Mobility (incoming and outgoing).

   a. List of students nominated for mobility. It comprises student personal data, period of stay, exact arrival and departure dates.

   b. Course catalog – details of courses are a basis for *Learning Agreements* and *Transcripts of Record*s. It comprises information like course code, course name, course description, edition, total number of hours, lecturer, ECTS points.

   c. *Learning Agreement* – the proposed programme of study for an outgoing student, approved by coordinators from sending and receiving institutions. It is a list of courses from the course catalog of the receiving institution a student intends to take during his/her *Erasmus* study period.

   d. *Transcript of Records* – it lists all the courses mentioned in the *Learning Agreement* with details about obtained grades (usually given in a local scale and ECTS scale). Based on the *Transcript of Records* the home university recognizes the results achieved by the student at the receiving university.

These data are gathered in local databases of both institutions and handled electronically. Students incoming for short-term studies should be treated like local students: get electronic student identity card, register for courses, obtain grades which have to be registered in the system, get printed version of a transcript of records. If the data are sent from one place to another on paper they have to be manually entered into the system, running the risk of typing errors, which are common when texts in foreign languages are handled.

Our purpose is to build tools for electronic transfer of data processed along mobility lifecycle to avoid data reentry. The beneficiaries of such tools would be those institutions which have a local student management information system, are active partners in international mobility programmes, and want to automate the process of data exchange.

What is needed is a standard format for data like *HEI information record*, *Bilateral agreement between two HEI*s, *Students nominated for mobility*, *Learning agreement and Transcript of records of a student*.

4. ESTABLISHING FORMAT FOR DATA EXCHANGE

4.1. Standardisation committees and on-going projects

Standardization in data exchange is going to play a relevant role in this project, particularly in terms of its potential of becoming an early-adopter of the work being carried out around the establishment of a common format to describe data being exchanged between the various European Higher Education stakeholders. In recent years, CEN (European Commission for Standardization) has been putting more and more attention to supporting the creation of a European Higher Education Area, through the activities of two main projects: Metadata for Learning Opportunities (MLO) and European Learner Mobility (ELM).

MLO [MLO] is a standard addressing metadata sufficient for advertising a learning opportunity, a programme of study for example. The goal of MLO is to provide information about a learning opportunity, to enable the learner to make a decision if there is a need for more information about the learning opportunity, and where to find that information. MLO is a lightweight standard that fits well with existing business processes and technologies, and is designed in full compliance with ECTS requirements. The MLO standard facilitates semantic technologies and web architectures to support several mechanisms for exchange of information and aggregation of information by third party service suppliers, while being easy to implement to ensure a rapid uptake by the European countries. The MLO work stemmed from the need to harmonize different regimes around Europe which describe and exchange information about courses, e-learning offerings, and learning opportunities. MLO concept mapping is an effort to map how the existing specifications for metadata about courses relate to the MLO model – if and how.

ELM [ELM] is a standard data model for the expression and exchange of European Learner Mobility information, as defined by the European transparency instruments, starting with the Europass portfolio (Diploma Supplement, Certificate Supplement, Curriculum Vitae, Mobility, Language Passport) [EUROPASS]. This standard will support the interoperability of European-wide IT systems that manage and exchange Europass related information. The standard will build on existing learning technology specifications, such as MLO, and take into account related national application profiles. As a multi-part standard project, the initial focus of ELM will be to create the specification to describe the Diploma Supplement document – starting with an abstract model and all the way to providing XML schemas, application profiles, and implementation guidelines.

The ongoing formal standardization efforts are based on two fundamentals principles: they build upon previously developed specifications and leverage the experience of actual implementers and vendors of student systems.

For example, the same team who worked on the XCRI (eXchange of Course-Related Information) standard for Course Description (used as a base for MLO), has been working on defining the structure for 'achievement records documents' (i.e. Diploma Supplement, a project called HEAR – Higher Education Achievement Report [HEAR]), which is heavily referenced in the ELM project as well.

As mentioned, real implementers have also been involved in the development of the standard; this is particularly critical in overcoming the so-called 'adoption-dilemma' whereby the official standardization authorities end with producing specifications which do not reach the adoption phase. In this respect, the working groups developing MLO and ELM do include representatives from a specific implementation community, R3SG (Rome Student System and Standard Group [RS3G]) to capture the real market perspective and obtain eventual buy-in from them. RS3G is a self-established group of software implementers and stakeholders in the European Higher Education domain which is focused on contributing to the definition and adoption of standards and procedures for the exchange of data to facilitate student mobility and lifelong learning. KION is a founding member of RS3G and MUCI has been always informed and indirectly involved in its activities – this link between CEN, RS3G and the KION-MUCI project creates a very fertile ground for an early adoption opportunity of new standard specifications for the exchange of student and course data; this is a distinctive and promising feature of the presented endeavor.

4.2. WSDL and web-services for the MUCI-KION project

Web Service Definition Language (WSDL) is an XML format for describing network services as collections of communication endpoints capable of exchanging messages. In our case these messages transfer data on cooperation and mobility, as described in chapter 3.

A very important aspect of data exchange is a choice of unique identifiers for objects handled inside the system. For some objects the choice is easy since they have natural unique identifiers. For others some common agreement is needed since every institution probably has its own set of values for identifiers. The most important objects to be identified are:

a. **Higher Education Institutions** – Erasmus code (like **PL WARSZAW01** or **I PARMA01**) seems to be a good choice, however it is limited to the institutions which have Erasmus University Charter awarded be the European Commission. Institutions may want to use the same electronic tools to support mobility with partners which do not belong to this group;

b. **Countries** – we have chosen **ISO 3166-1-alpha-2**, adopted also for Europass (see http://europass.cedefop.europa.eu/xml/EUROPASS_ISOCountries_V1.1.xsd);

c. **Languages** – similarly the choice is **ISO 639-1:2002**, adopted also for Europass (see http://europass.cedefop.europa.eu/xml/EUROPASS_ISOLanguages_V1.1.xsd);

d. **Persons** – we suggest to use person identifiers being in force in a country of citizenship. Each country defines such type of unique code for its citizens, a pair (country code, person id) would be a global unique identifier of a person in a system. Such identifiers could be verified using a validation procedure delivered by each country involved in the project;

e. **Courses** – course code is a string which uniquely identifies a course inside the university course catalog, a pair (HEI Erasmus code, course code) would identify a course globally in the system;

f. **Disciplines of study** – Socrates code is a commonly accepted identifier for a discipline of study, e.g. 11.3 corresponds to Computer Science, 15.1 to Journalism.

There are also other sets of data for which some commonly accepted dictionaries could be used, like types of organization units inside HEI (faculties, departments, institutes etc.), types of study activities for a course (lectures, laboratories, seminars etc.), grades (ECTS scale is easy to define, but local scales may vary substantially and may be difficult to translate automatically to another local scale or to ECTS scale).

The proposed set of web-services for the project comprises the following methods (arguments are omitted, some involved data types are also described):

a. **sendHeiData()**, **getHeiData()** – official information on HEIs which sign the cooperation agreement is a basis for further electronic handling of involved documentation and activities. An agreement signed once is very often extended for next years. Keeping track of partner university authorities and contact persons helps in daily communication (e.g. in USOS an email may be send straight from the system if an email address is available in the database);

b. **sendAgreementData()**, **getAgreementData()** – this information is vital for all further mobility handling and tracking since it defines the most basic details of cooperation: disciplines, number of places and months for each study program level (bachelor, master, doctoral). These details are negotiated between partners and then settle a range for all practical activities. In USOSweb, which is a web part of USOS for students and academic staff, during recruitment for outgoing mobility students can browse agreements while choosing a place to go for short-term studies. Erasmus coordinators can assign places to students only up to the defined limits. The discipline code of the agreement should be consistent which the discipline of study of the applying student;

c. **sendNominatedStudents()**, **getNominatedStudents()** – each institution should send to the partner a list of students nominated for the mobility. Personal data of these students should be entered to the other database since after arrival these students will be treated like local ones. The current practice is that either data are retyped on the basis of a paper documentation or students are asked to enter the data themselves. Both ways are often sources for many typing errors. In the University of Warsaw incoming students enter their personal data into a special web form, from which data are transferred to the central database. A system tries to uniquely identify a foreign student by first name, last name, passport number (or any other document id entered through the form). However these are textual data, any misspelling or difference in a way string is typed is the reason the matching fails. As a result the person may get duplicated in the system since quite often a student coming for a summer course later on returns as an Erasmus student or even – after some time – as a lecturer. Such data duplication can cause many problems. A record of personal data consists of the following fields: **familyName, firstName, secondName, birthDate, birthCity, gender, email, citizenNumber, countryCode, permanentAddress, correspondenceAddress, stationaryTelephone, mobileTelephone,**

d. **sendArrivalDate()**, **getArrivalDate()**, **sendDepartureDate()**, **getDeparture-Date()** – these dates are important since stipends paid to traveling students depend on the exact dates of arrival and departure to/from a receiving institution;

e. **getCourseData()** – a course catalog is needed for two purposes. First, when delivered by a partner and posted locally it may help students and Erasmus coordinators during negotiations on Learning Agreements. Second, when Learning Agreement or Transcript of Records is defined in the local system for each outgoing student, the course catalog may be used to validate some data like course code, course name, lecturer, and ECTS points. Again this may help to avoid possible typing errors. The main fields of the course record are the following: **courseCode**, **courseDescription** (possibly in English, local language, course language), **name**, **language** (in which the course is taught), **URL**, **ectsPoints**, **totalHours**, **period**, **lecturer**;

f. **sendLA()**, **getLA()** – Learning Agreement is an official document required by the Erasmus Agency. Having an approved LA is a necessary condition for any transfer of stipend money to a student account. LA should definitely be kept in a local system for further reference;

g. **sendToR()**, **getToR()** – Transcript of Record is the most important document for the outgoing student. It is issued by the receiving institution and has to be delivered to home institution where it will be entered to the local system to become a part of history of student achievements. Eventually it will be printed on the diploma supplement. It lists all the courses mentioned in the *Learning Agreement* with details about obtained grades (usually given in a local scale and ECTS scale). In USOS a student after coming home from the partial study abroad is obliged to enter information from the Transcript of Records to the system through USOSweb and simultaneously deliver a paper version to the students office. The officer from the students office has to validate the paper version against electronic and finally approve the data in the system. This procedure would be much simpler for both parties if data would be delivered electronically from the institution which issued this document;

h. **validateUID()** – this is a helper function for validating a person country identifier to ensure correctness of delivered data. Every electronic system needs a unique identifier of a person to make sure that personal record is not duplicated.

In most cases two symmetric functions are available: **send()** for sending data to the partner and **get()** for getting data from the partner. Depending on scenarios of cooperation either one or the other may be more suitable in particular cases.

Coding is set to UTF-8. Current version of WSDL is available on the net at URL:

http://usosphp.mimuw.edu.pl:8080/ErasmusService/ErasmusService?wsdl.

We are aware that eventually WSDL definition will have to be updated after European standards concerning formats of description of courses, personal data, and university data are officially approved.

In the meantime cooperating partners may test how relevant is the suggested WSDL to their needs by defining by hand sets of sample data to be transferred by the delivered tool. During the course of the project we prepared two such sets of sample data – for the University of Warsaw and for Parma University – and used them for testing before the new tool was integrated with the university student management information system.

Depending on the final version of WSDL the problem of data mapping from common ‘mobility format’ to/from local format already in use may be less or more difficult to handle. Tests are needed before any final conclusions can be drawn.

5. ARCHITECTURE

The general overview of the prototype architecture is shown on Fig. 1. The overall system is a multi-node peer-to-peer network. The main elements are:

- UDDI (see chapter 5.1) – only one UDDI is needed in the system;
- Web server embedded in the application server (see chapter 5.2) – each institution participating in the project should have its own web-server and WSDL, not all web-services need to be implemented in each node;
- Web client embedded in the same application server (see chapters 5.3 and 5.4) – web-client is needed generally for testing purposes and during initialization phase, it helped us to test connections between system elements and data formats at various stages of the project;
- Local student management information system with a module for International Relations Office – in chapter 5.5 we describe a solution built for USOS (based upon an Oracle database).

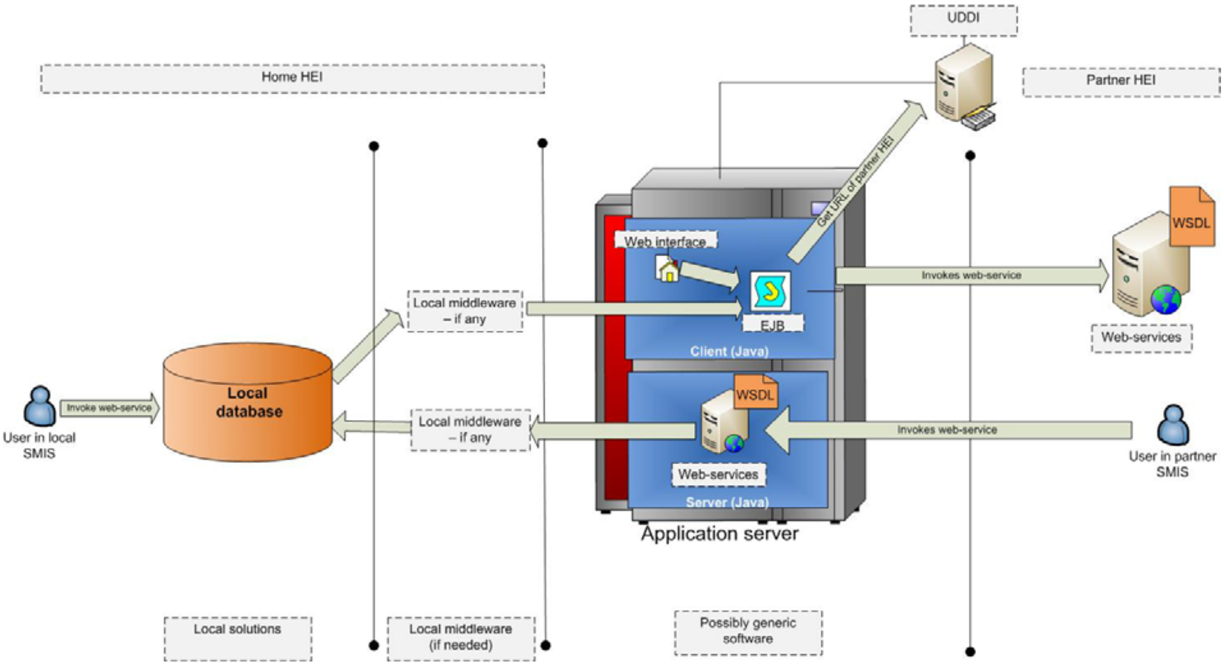


Figure 1 Architecture of the prototype

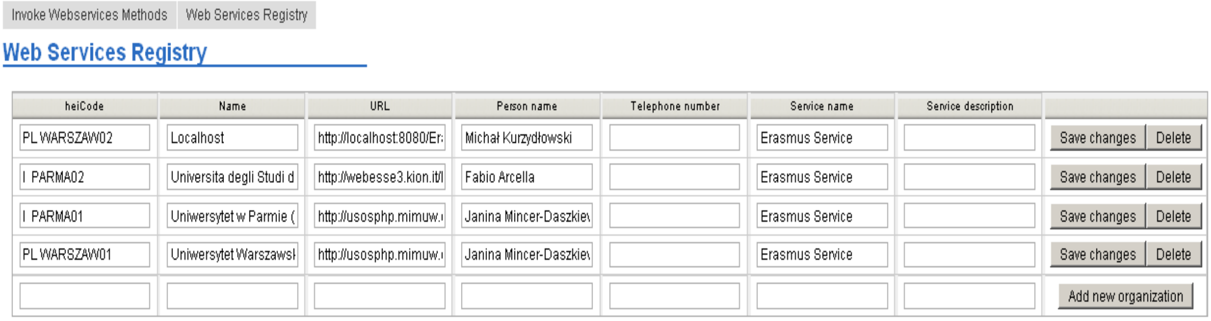
WSDL was used to generate raw versions of a web server and a web client. Their code does not depend on local solutions and may be shared between universities. Each partner has to integrate handling of web services into the interface of local SMIS and possibly also implement middleware between the web server and the local database. Various solutions are possible at this part of the system.

5.1. UDDI

Universal Description, Discovery and Integration (UDDI) is an XML-based service registry of universities participating in the project. It is designed to be interrogated by *Simple Object Access Protocol* (SOAP) messages and to provide access to WSDL documents describing the protocol bindings and message formats required to interact with the web services listed in its directory. HEI Erasmus code is an identifier which should be used to locate partner university on the net. The answer for the inquiry contains an URL of the WSDL document describing the services implemented by the partner (which might be a subset of the full version).

The UDDI server for the project was build using JUDDI-web-2.0rc5, which is an open source Java implementation of the UDDI specification. We deployed it on the GlassFish application server for the Java EE platform (GlassFish has been developed as an open source project led by Sun Microsystems). The registry data is stored in the MySQL database. Communication is carried out via SOAP protocol, in transport layer via HTTP protocol, by POST method (thus cannot be called directly from a web browser).

UDDI registry can be inquired and updated using web-client described in chapter 5.3. The web browser interface to UDDI is demonstrated in fig. 2.



heiCode	Name	URL	Person name	Telephone number	Service name	Service description	
PL WARSZAW02	Localhost	http://localhost:8080/Er	Michał Kurzydłowski		Erasmus Service		Save changes Delete
I PARMA02	Universita degli Studi d	http://webesse3.kion.it/l	Fabio Arcella		Erasmus Service		Save changes Delete
I PARMA01	Uniwersytet w Parmie (	http://usosphp.mimuw.	Janina Mincer-Daszkiev		Erasmus Service		Save changes Delete
PL WARSZAW01	Uniwersytet Warszawsk	http://usosphp.mimuw.	Janina Mincer-Daszkiev		Erasmus Service		Save changes Delete
							Add new organization

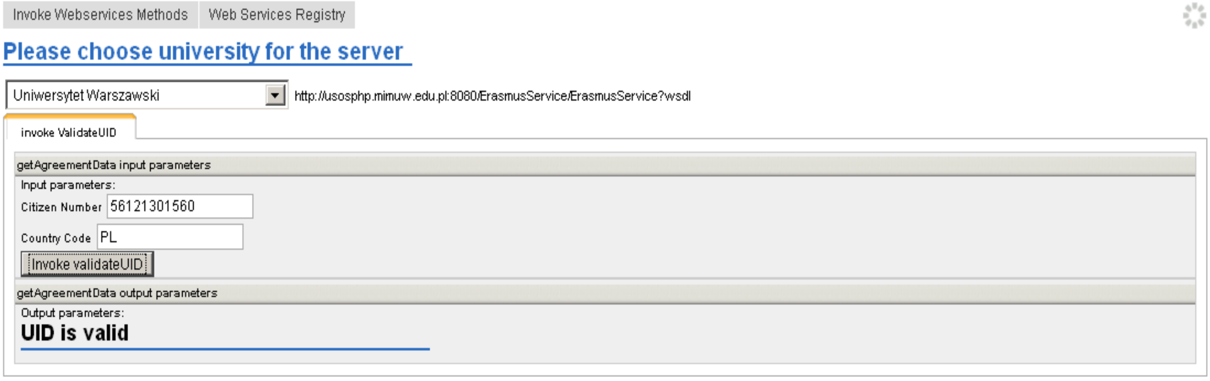
Figure 2 Web interface to UDDI

5.2. Web server

Web server accepts web service calls from the net and forwards them to the local system (possibly through the middleware layer). In our prototype implementation we used Java platform for embedding web server – it is the same GlassFish application server which has been used for the UDDI. However each partner institution may use it own solution (e.g. KION installed Apache with Tomcat and Axix). The only obligation is to implement a subset of web services described by the commonly agreed WSDL document.

5.3. Web client

Web client was built to support the project during the testing phase. It allows for calling web services from a web browser. The main parameters are URL of a web server and URL of a web client, e.g. it may be used to simulate a call made by the University of Parma of a web service delivered by the University of Warsaw web server. The interface is demonstrated on fig. 3 and 4. More details about the web client are given in chapter 5.4.



Please choose university for the server

Uniwersytet Warszawski http://usosphp.mimuw.edu.pl:8080/ErasmusService/ErasmusService?wsdl

getAgreementData input parameters

Input parameters:

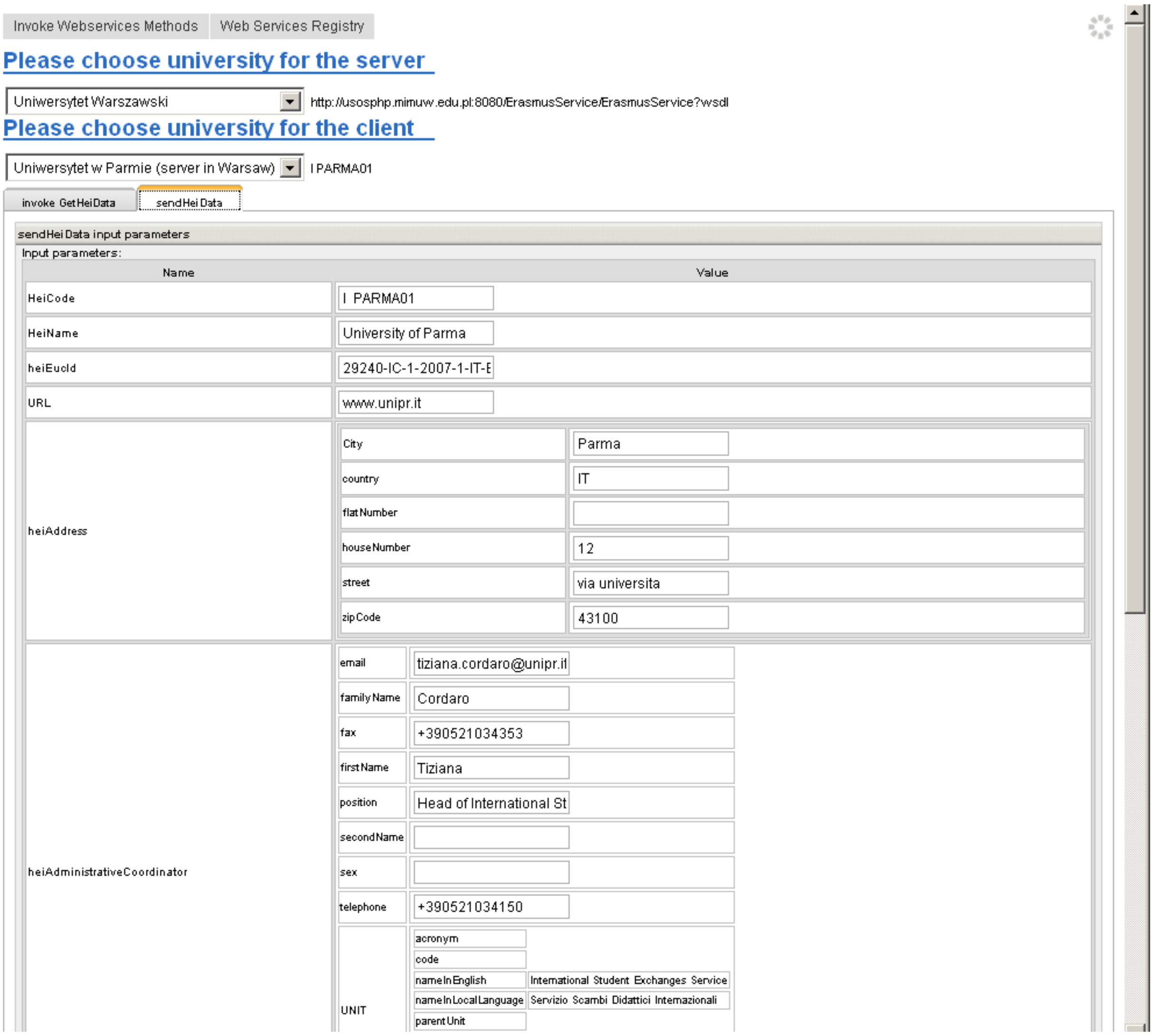
Citizen Number	56121301560
Country Code	PL

getAgreementData output parameters

Output parameters:

UID is valid

Figure 3 Calling validate() for Polish citizen id from the web client interface



Please choose university for the server

Uniwersytet Warszawski http://usosphp.mimuw.edu.pl:8080/ErasmusService/ErasmusService?wsdl

Please choose university for the client

Uniwersytet w Parmie (server in Warsaw) I PARMA01

sendHei Data input parameters

Input parameters:

Name	Value		
HeiCode	I PARMA01		
HeiName	University of Parma		
heiEucId	29240-IC-1-2007-1-IT-E		
URL	www.unipr.it		
heiAddress	City	Parma	
	country	IT	
	flatNumber		
	houseNumber	12	
	street	via universita	
	zipCode	43100	
heiAdministrativeCoordinator	email	tiziana.cordaro@unipr.it	
	familyName	Cordaro	
	fax	+390521034353	
	firstName	Tiziana	
	position	Head of International St	
	secondName		
	sex		
	telephone	+390521034150	
	UNIT	acronym	
		code	
		nameInEnglish	International Student Exchanges Service
		nameInLocalLanguage	Servizio Scambi Didattici Internazionali
		parentUnit	

Figure 4 Calling sendHeiData() from the web client interface

5.4. TESTBED

We built a testbed to help us during the initial stages of the project. Its architecture is shown in fig. 5. It comprises UDDI, web client, one web server at port 8080 (corresponding to PL WARSAW01) embedded in one GlassFish application server, and another web server at port 9090 (corresponding to I PARMA01) embedded in the second GlassFish application server. All this is installed in Warsaw. The third web server was available in Bologna (hosted by KION) playing the role of Parma as I PARMA02. Important elements of the testbed environment are also:

- Sample data stored in a file system which simulate data obtained from a database.

  The sample set of files in XML format with data from both universities involved in the project was stored on the client and on each tested server.

- Logs at the web servers.

  Data being sent to the web server are always logged in a file system and may be viewed using a web browser.

A web method invocation can be made either from the web client or (eventually) from a local SMIS. When a **send()** method is called from the web client interface (see fig. 4), data to be sent are read from a relevant set of files, can be updated through the web page and then are send to the server. A **get()** method called from the web client interface displays the obtained data on a web page.

A called web service after reaching an appropriate web server can either be handled on the server or (eventually) delegated to a local SMIS. When a **send()** invocation is handled by a server, the obtained data is simply logged in the file system. When a **get()** invocation is handled, data are read from a relevant set of files and send to the caller.

Eventually the local SMIS should suport method invocations made by local users and method invocations made by remote users from partner institutions.

New institutions joining the project might in a similar way start with preparing XML files with sample data, implement methods one by one and replace a test invocation by the invocation made from/delegated to the local SMIS. Technically this is obtained by replacing one DAO implementation (accessing the file system) with another (accessing the database in the local SMIS).

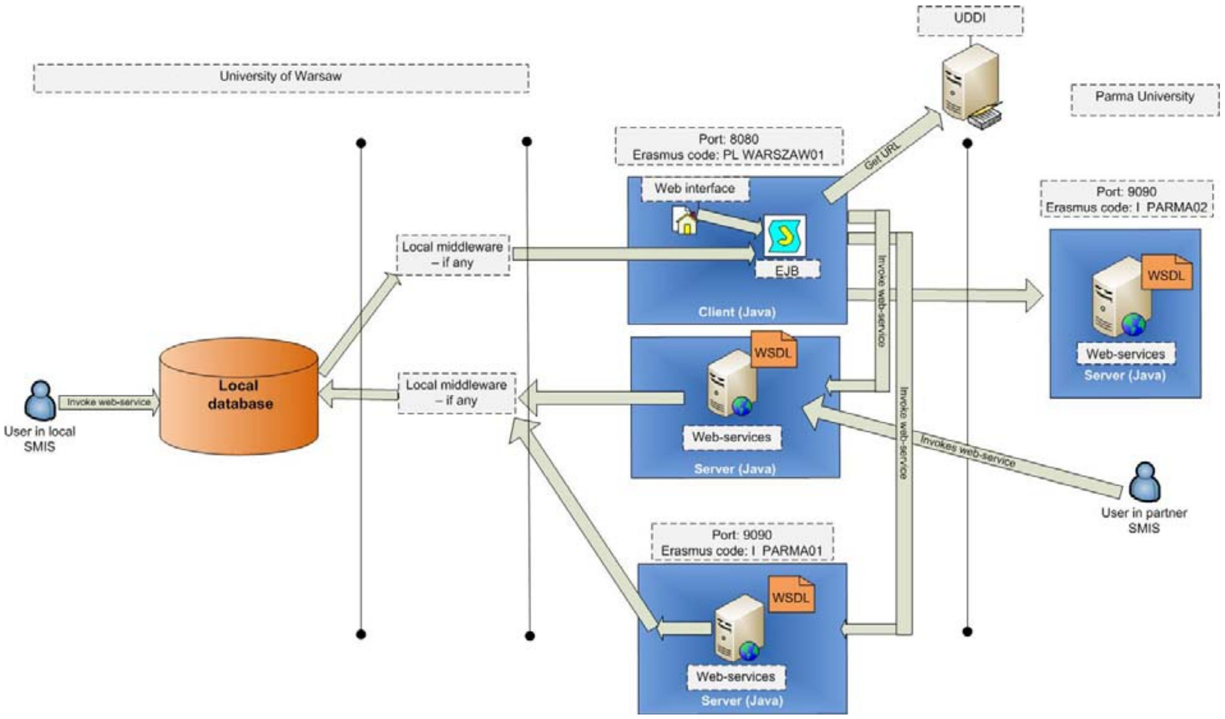
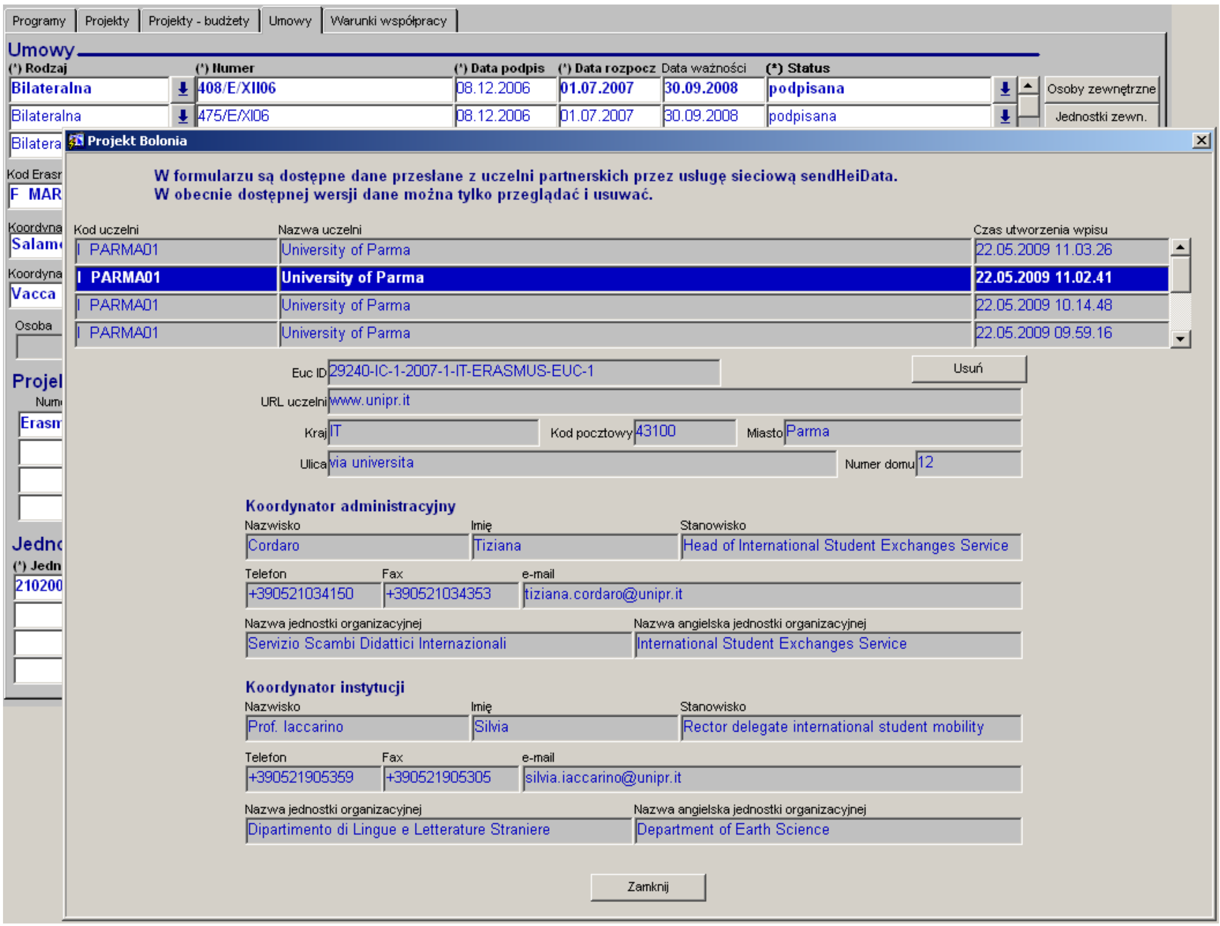


Figure 5 Testbed for the project

5.5. Integration into local SMIS

Interface for calling web services should be integrated into the local student management information system. It depends on local solutions so the effect of scale may be achieved if national consortia built software for their products. In case of USOS it was added to the module for International Relations Office (described in [JMD2008]). USOS is built around an Oracle database. From the database perspective two technologies were used for implementing data exchange depending on the site which initiates the communication:

1. Web service called by the local user
   - For **send()** – based on the context, data is selected from the database, an XML message is formulated and enqued into *Advanced Message Queue* for outgoing messages, the answer is read from *Advanced Message Queue* for incoming messages.
   - For **get()** – an XML message is constructed and enqued into the *Advanced Message Queue* for outgoing messages, the answer is read from another *Advanced Message Queue* for incoming messages and displayed in Oracle form, where it can be viewed and deleted (other actions, like transferring data from the queue to the local table, will be implemented later).
2. Web service called by the remote user
   - For **send()** – records which have been sent are stored in the XML buffer inside Oracle database, special views are implemented for selecting messages corresponding to specific web methods, they are displayed in Oracle forms, can be browsed, read, and deleted (see fig. 6).
   - For **get()** – request for data initiates a call of a procedure from an Oracle package, which selects data from the database and sends to the caller.



Umowy

(*) Rodzaj	(*) Numer	(*) Data podpis	(*) Data rozpocz	Data ważności	(*) Status
Bilateralna	408/E/XII06	08.12.2006	01.07.2007	30.09.2008	podpisana
Bilateralna	475/E/XI06	08.12.2006	01.07.2007	30.09.2008	podpisana
Bilatera					

W formularzu są dostępne dane przesłane z uczelni partnerskich przez usługę sieciową sendHeiData.
W obecnie dostępnej wersji dane można tylko przeglądać i usuwać.

Kod uczelni	Nazwa uczelni	Czas utworzenia wpisu
I PARMA01	University of Parma	22.05.2009 11.03.26
I PARMA01	University of Parma	22.05.2009 11.02.41
I PARMA01	University of Parma	22.05.2009 10.14.48
I PARMA01	University of Parma	22.05.2009 09.59.16

Euc ID 29240-IC-1-2007-1-IT-ERASMUS-EUC-1

URL uczelni www.unipr.it

Kraj IT Kod pocztowy 43100 Miasto Parma

Ulica via universita Numer domu 12

Koordynator administracyjny

Nazwisko Cordaro Imię Tiziana Stanowisko Head of International Student Exchanges Service

Telefon +390521034150 Fax +390521034353 e-mail tiziana.cordaro@unipr.it

Nazwa jednostki organizacyjnej Servizio Scambi Didattici Internazionali Nazwa angielska jednostki organizacyjnej International Student Exchanges Service

Koordynator instytucji

Nazwisko Prof. Iaccarino Imię Silvia Stanowisko Rector delegate international student mobility

Telefon +390521905359 Fax +390521905305 e-mail silvia.iaccarino@unipr.it

Nazwa jednostki organizacyjnej Dipartimento di Lingue e Letterature Straniere Nazwa angielska jednostki organizacyjnej Department of Earth Science

Figure 6 Oracle form with data obtained from partner universities calling sendHeiData()

There are some open questions concerning the way the new functionality should be implemented. Let us consider some real life scenarios. When a list of students nominated for a short-term studies at one of the partner institutions is ready, an administrative officer from the IRO pushes the button and sends it to the other site. After a day or two some corrections may have to be made. Should the whole list be sent again or just an upgrade? Definitely a timestamp should be attached to each sent record of data, so that the receiver can sort it chronologically. If the same list is needed from the partner, should we rather wait for his **send()** to come or call **get()** ourselves? How to implement such **get()**? A list of students nominated by the partner may not yet be ready when our **get()** arrives. Web services are symmetric in a sense that the same set of functions can be called by either of the partners but it seems that sends are more in line with real life data exchange.

Other problems arise when the data is obtained from the partner. It waits for our approval, can be read from the buffer and viewed. But then we may want to store it in the local database. Some data may depend on some other e.g. cooperation conditions may involve a particular unit from the partner university, like Faculty of Chemistry or Institute of Social Sciences. This unit may already be present in a local table or it may have to be inserted first, before other, dependent data can be added. Can we uniquely recognize such unit in the database? How to avoid data duplication? If we keep relations between tables (and most probably we do) the obtained record of data may have to be divided into components and handle separately (e.g. it may happen that Faculty of Chemistry is already available in the local table, however Institute of Social Sciences may have to be inserted).

Real life scenarios should be observed to find answers to such questions. Transfer of data to the local database have not been implemented in the prototype system. More work is needed before final solutions are delivered.

5.6. Deployment

There are some logistic problems concerning the deployment of the proposed solution on a large scale, more challenging than pure technical issues. The following problems should be addressed:

1. Joining the project

   International cooperation between two higher education institutions is always preceded by signing the official agreement. This is also a good moment to discuss and settle conditions for electronic data exchange. The designed distributed architecture operates on a peer-to-peer basis so bilateral agreements between partners should be signed.

2. Authorization and authentication

   Authorization and authentication might be supported by some Federated Identity Management infrastructure similar to Eduroam. PKI can be regarded if higher level of security becomes necessary, but in our opinion this is not an obligatory requirement.

3. Security

   Secure encrypted transfer protocol should be used since in many countries personal data are protected by law and their privacy can not be jeopardized.

6. CONCLUSIONS

There have been some earlier approaches concerning electronic exchange of data on student academic achievements. In [STKA2007] authors describe a system for sending Transcripts of Records which is based on e-government standards being in force in Germany. In [SITI2006] a distributed system is described for sending Europass documents – it is going to be deployed with the support of Cedefop. An attempt to integrate two software systems for data exchange on international cooperation is also described in [VeRa2007]. Our approach differs in some respect since the main goal is the support for processes running at IROs of cooperating higher education institutions, strictly connected with the student and staff mobility. The solution is needed now and here, and – what is very important – may be deployed now and here. Two partners who gather data in their local systems and are willing to cooperate may start exchanging data electronically almost immediately, others may join when they are ready. An effect of scale can be obtained relatively easily, if only national consortia developing student management systems join the project and integrate needed interface into their products. MUCI develops USOS which is deployed in 25 higher education institutions in Poland which group about 30% of students in Poland (40% of state owned HEIs). USOS contains a module for IRO with full support for electronic handling of processes of student and staff mobility ([JMD2008]). KION owned by CINECA, which is the leading consortium of Italian Universities, develops ESSE3 which is a student management system in place in over 75% of all Italian Higher Education Institutions. There are other consortia in European countries which cover high percentage of domestic universities like Ladok in Sweden (almost 100% of the market), HIS in Germany (230 universities), OCU in Spain, and also companies like unisolution which deliver software for many institutions all over Europe (moveon is the standard software used by 260 institutions in 12 European countries to manage their international activities). Let us imagine that these companies add support for data exchange into their systems. Not so much implementation work, tremendous effect of scale – it would be interesting to count how many students might travel across Europe not accompanied by a flow of papers.

This will not happen without common data format, adopted and agreed on by all partners, compliant with the official standards. This may be the most difficult step to make. The suggested approach is not to wait passively for the final approval of the official standards (European Norm) but to actively start with some preparatory steps and work on the practical implementation. Some parts like UDDI or web servers are already available and may be obtained as an open-source. Others, like local arrangements, have to be done by software vendors but – as it was said – the effect of scale may be obtained rather quickly. RAD-like approach of our project may help in setting some practical perspective. The idea is to call out for participation to other consortia and/or student systems implementers to join this project, extending it and making it sounder with their requirements, thereby achieving critical economies of scale which can result in a substantial impact toward the creation of a European Higher Education Area.

Acknowledgements

Many thanks to Michał Kurzydłowski, Piotr Pietrzak, Artur Popławski, and Janek Rudziński who implemented the Warsaw part of the prototype system; and Valentino Sarli on KION's side.

7. REFERENCES

[ELM] European Learner Mobility,
http://wiki.teria.no/confluence/display/EuropeanLearnerMobility/European+Learner+Mobility

[EUROPASS] Europass, http://europass.cedefop.europa.eu

[HEAR] Information regarding HEAR (Higher Education Achievement Report),
http://wiki.cetis.ac.uk/Achievement_Information_Working_Group

[JMD2008] Mincer-Daszkiewicz, J. (2008). International Cooperation and Mobility in Higher Education from IT Perspective, EUNIS'2007. Retrieved May 1, 2009, from: http://eunis.dk/papers/p21.pdf

[MLO] Metadata for Learning Opportunity, http://wiki.teria.no/confluence/display/CIF/Home

[RS3G] Home of Rome Student Systems and Standards Group,

http://wiki.teria.no/confluence/display/RS3G/Home+of+RS3G

[SITI2006] Siaperas A., Tissot P. (2006). A Distributed System for Issuing Europass Mobility Documents. Retrieved May 1, 2009, from:
http://wiki.teria.no/confluence/download/attachments/13697206/Distributed+System+for+EUROPASS.doc?version=1

[STKA2007] Strack H., Karich Ch. (2007). A Distributed Architecture for the Management of Data Flows within the Bologna Process Using Egovernment Standards. Retrieved May 1, 2009, from:
http://www.eunis.org/events/congresses/eunis2007/CD/pdf/papers/p205.pdf

[USOS] Home page of the University Study-Oriented System, http://usos.edu.pl

[VeRa2007] Velay S., Ravaioli S. (2007). Powering International Mobility – A Software Integration Experience Across Europe, EUNIS'2007. Retrieved May 1, 2009, from:
http://www.eunis.org/events/congresses/eunis2007/CD/pdf/abstracts/p141.pdf