

The Mobility Project - Building network of web-servers for exchange of data on student mobility

Janina Mincer-Daszkiewicz¹

¹Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Banacha 2, 00-927 Warszawa, Poland, jmd@mimuw.edu.pl

Keywords

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

1. ABSTRACT

The Mobility Project started in 2008 as a joint initiative of two national consortia, developers and suppliers of student management systems for their home countries, MUCI from Poland and Cineca from Italy. The initial goals of the project were as follows:

1. Recognizing requirements concerning exchange of data on international cooperation between higher education institutions and in particular on student and staff mobility, describing business processes of such data exchange, consulting them with employees of university International Relations Offices;
2. Defining format of the data to be exchanged in some formal language, like WSDL, and preparing sample data for universities from two involved countries;
3. Building a prototype network of web-servers and testing data exchange.

The results of this preliminary phase were presented at [AMR2009]. The parties involved decided to broaden the scope of the project and organized the workshop [AMR2009b]. The leading force was RS3G (Rome Student Systems and Standard Group) and among participants were national consortia and suppliers of student management systems in Europe. The participants acknowledged the importance of the project goals and suggested that RS3G should take a lead of the project.

After EUNIS'2009 the development team from the University of Warsaw started working on a new version of WSDL for web-services and software for web-servers, taking into account the experience gained during the preliminary phase and also comments and suggestions gathered during the EUNIS presentations and discussions. This new version was ready in November 2009 and is fully documented at <http://usos.edu.pl/Mobility>.

The next step was the workshop in Uppsala in November 2009 and another one in Bologna in March 2010. They brought together consortia and companies willing to take an active part in the project. Among these were some national consortia (OODI - Finland, SIGMA - Spain, CINECA - Italy, Ladok - Sweden, FS - Norway, MUCI - Poland, HIS - Germany, SURF - Netherlands, VHS - Sweden), companies (Digitary - Ireland, AcademyOne - USA, KION - Italy, QS unisolution - Germany) and individual universities. The most important outcome of the Uppsala workshop was the creation of Work Teams responsible for further investigation of various aspects of the Mobility Project. In Bologna the Mobility Group met with TF-EMC2 (*TERENA Task Force on European Middleware Coordination and Collaboration*) to discuss the aspects of security and identity management for the Mobility Project. Possible solutions were recognized and further actions planned. It was also decided that the new version of the software would be deployed in a couple of nodes to make more tests with sending the sample data between countries.

The initiative of the Mobility Group seems to us an important and practical step towards the creation of the European Higher Education Area. We would like to present the project to the wider audience gathered at EUNIS, to report on the undertaken activities, to describe the ideas behind the new WSDL and software, and to invite cooperation of other system vendors. The final success of the project depends on its scope which is proportional to the number of student management systems able to exchange mobility data by web-services.

2. INTRODUCTION

Student and staff mobility is one of the most important priorities of the European Higher Education Area promoted by the Bologna process. Long-term trend for Erasmus programme (which is the most important European project promoting mobility) shows almost linear growth of the number of students travelling to other universities for short-term studies. In Europe as a whole, this programme involves over 4000 higher education institutions (HEI) while the number of participating students approaches two million. The European Commission introduced the new Lifelong Learning Programme which is aimed to further stimulate mobility and achieve a number of 3 million Erasmus students by 2012. Therefore it becomes an important part of HEIs' activity. Important also in a sense of the amount of administration work needed.

Management of mobility consumes a lot of effort, time and cost. 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 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.

Currently, the vast majority of higher educational institutions are equipped with IT infrastructure which enables them to keep all the necessary data needed to run the studies, i.e. personal data, learning achievements data, accounting, etc. However, the process of exchanging the necessary data in order to actually perform the student mobility is still carried out in a manual way.

In [AMR2009] we suggested how to change the way in which information is handled. We designed a common format for the mobility data and implemented the software for sending it by web-services delivered by web-servers installed at universities engaged in international cooperation. We deployed a prototype systems in Warsaw and Bologna to make some test data transfers.

The ideas presented at EUNIS'2009 were positively accepted by many national consortia, companies and individual universities, developers and suppliers of student management systems. Some of them decided to join the project. The workshop for the prospective partners was scheduled for November. In the meantime the Warsaw development team started working on a new version of WSDL for web-services and software for web-servers, taking into account the experience gained during the preliminary phase and also comments and suggestions gathered during EUNIS presentations and discussions. The new version is described in section 4.

The workshop for consortia and companies willing to take active part in the Mobility Project took place in Uppsala in the middle of November. Another workshop followed, in Bologna in March 2010, bringing together the Mobility Group and TF-EMC2. Results of both meetings are described in section 5. Final section contains conclusions.

3. REAL LIFE SCENARIOS

Before we describe the second version of the data format and software, let's recall the goal we want to achieve. Let us follow the real-life scenario. In Warsaw (and in 30 other higher education institutions in Poland) we use student management information system USOS ([USOS]) which - in particular - contains a module for handling international cooperation and mobility. Recruitment for outgoing mobility is done at faculties through USOSweb, the web part of USOS, and at the end the system contains the list of nominated students with all details, like personal data, program of study, destination, period of stay (eventually also Learning Agreements and stipends), all available in the electronic form ready for being send to the partner universities - see Figure 1.

What happens next? Nominations are either typed manually into a partner's system (in practice many different systems), or scanned, faxed, delivered by phone calls - usually many times since updates are common. What might happen instead? We proposed in [AMR2009]: the operator might click the button (like the one inside the red rectangle in Figure 2) and send the list to the partner - the figure shows the result of the remote invocation of the web-service **sendNominated-**

Students(). Data obtained from the home institution would be delivered straight to the database of the partner institution and be immediately ready for further processing.

Erasmus outgoing student mobility coordination
at the faculty/unit: Faculty of Mathematics, Informatics, and Mechanics [choose another]

Assigned outgoing student mobilities

→ print list
→ export list to file
→ Send email to all 24 persons from the table below.
→ send list to coordinators

FILTER OPTIONS

Status: (all)
Academic year: 2009
Institution name:
Erasmus Code:
Country:
Assigned not earlier than: YYYY_MM_DD
Assigned not later than: YYYY_MM_DD

OR

SELECTED STUDENT
Student's name: APPLY

Items 1..24 of 24

No.	Surname	Name	Country	Erasmus Code	Institution name	Date of assignment	Academic year	Status	Options
1	Konrad		Hiszpania	E BARCEL02	Universitat Autònoma de Barcelona		2009	Finished	view learning agreement
2	Przemysław		Francja	F PALAIS01	Ecole Polytechnique		2009	Finished	view learning agreement
3	Aleksander		Francja	F PARIS013	Université Paris Nord (Paris XIII)		2009	Finished	view learning agreement
4	Kamil		Szwecja	S UPPSALA01	Uppsala Universitet		2009	Finished	view learning agreement
5	Kamil		Holandia	NL AMSTERD02	Vrije Universiteit Amsterdam		2009	Finished	view learning agreement
6	Dominika		Austria	A WIEN01	Universität Wien		2009	Finished	view learning agreement
7	Jan		Holandia	NL AMSTERD02	Vrije Universiteit Amsterdam		2009	Finished	view learning agreement
8	Weronika		Dania	DK KOBENH01	Københavns Universitet		2009	Finished	view learning agreement
9	Anna		Dania	DK ARHUS01	Aarhus Universitet		2009	Finished	view learning agreement
10	Hanna		Francja	F PARIS013	Université Paris Nord		2009	Finished	view learning agreement

Figure 1. USOSweb - A list of students nominated for outgoing mobility

USOS - [Przyjazdy]

Akcja Edycja Zapytanie Blok Rekord Pole Okno Pomoc

Program <wszystkie> Etap <wszystkie> Cykl dydaktyczny <wszystkie> Miejsce <wszystkie> Filtruj po etapach

Osoby

Projekt MOBILITY

Kod dyscypliny	Poziom studiów	Rok	akad przyjazdu	Okres przyjazdu	Jednostka przyjmująca	Czas utworzenia wpisu
08.4	2	2008	S1		Facolta di Lingue e Letterature straniere	19.06.2009 12.38.33
10.0	4	2008	S2		Facolta di Giurisprudenza, Istituto di Diritto e	19.06.2009 12.38.33
10.0	2	2008	S1		Facolta di Lingue e Letterature straniere	20.06.2009 17.42.52
10.0	4	2008	S2		Facolta di Giurisprudenza, Istituto di Diritto e	20.06.2009 17.42.52

Jednostka wysyłająca
Instytut Archeologii

Nazwisko i imię koordynatora jednostki przyjmującej
Prof. Modrzewska-Pianetti Iwona

Nazwisko	Imię	Data urodzenia	Kraj urodzenia	Płeć	Czas utworzenia wpisu
Kurzydowski	Michał	1985-10-26	PL	M	19.06.2009 12.38.33
Kowalski	Fabio	1985-01-14	PL	M	19.06.2009 12.38.33

Miasto urodzenia: Warszawa Email: michal.kurzydowski@students.mimuw.edu.pl

Zamknij

MOBILITY w. 1
MOBILITY w. 2

Figure 2. USOS - A list of incoming students obtained from the partner university

Other possible scenarios are: signing bilateral agreement (sending organization's information record and cooperation conditions of the agreement); sending Learning Agreements, getting Transcripts of Records (grades), sending departure/arrival dates of mobile students (which are the basis for calculation of scholarships), sending course catalog.

During the first phase of the project we identified business processes and recognized sets of data being exchanged between cooperating partner institutions. We defined WSDL for data like *HEI information record*, *Bilateral agreement between two HEIs*, *Students nominated for mobility*, *Learning agreement and Transcript of records of a student*. The proposed set of web-services for the project comprised the methods like `sendOrganizationData()`, `sendAgreementData()`, `sendNominatedStudents()`, `sendArrivalDate()`, `sendDepartureDate()`, `sendLearningAgreement()`, `sendTranscriptOfRecords()`, `validateNationalPersonalId()`. In most cases two symmetric functions were delivered: `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.

4. SECOND PHASE OF THE PROJECT (JUNE 2009 - APRIL 2010)

During the first phase of the project we gathered experience concerning structure and scope of data, software design, technology used, system architecture. During the presentation at EUNIS'2009 and the workshop of RS3G we got valuable feedback from the audience. As concerning the data format we realized that there are much more various types of "standards" being used by countries (Finland, Norway, Germany), companies (QS unisolution) or associations (TERENA). We decided to have a closer look at these standards and incorporate them into the Mobility Project. We also recognized some specific requirements of the project: since we are going to send data from one database to another, all exchanged objects should be uniquely identified in both systems. There is also an important issue of internationalization - many languages should be handled simultaneously. Some aspects of the new WSDL are described in section 4.1.

As concerning software we realized that - due to the foreseen changes in the format of the mobility data - we should design it, in particular transport middleware, to be highly vulnerable to such changes, i.e. modification of the data format should have possibly minimal impact on the source code. Software should also be relatively easy to install, especially in diverse environments of higher education institutions, and to maintain. The solution should be scalable. Tools and technologies used to implement the software should be:

- freely available; preferably licensed in a way which allows redistribution - ideally the licenses should be open source compatible,
- maintained and widely used, with strong, active community,
- independent of any specific operating system or hardware architecture.

There was also an important non-functional requirement specific to the University of Warsaw concerning interoperability with Oracle Database (PL/SQL and Oracle AQ).

The new software is described in section 4.2. This section is based on [Nag2009]. Details, in particular current version of WSDL, can be found in [Mobility].

4.1. WSDL version 2.0

There are some key aspects to consider when proposing a new standard. First, there is a need to provide a vocabulary of well-defined terms which model the problem domain well. Second, there is a need to avoid unnecessary complexity and keep things possibly simple. Third, a good standard leaves a way to extend it easily. And the last but not least, employment of existing standards and practices to the maximum extent would be highly desired, so the effort of converting the data already formatted in compliance to these standards is minimized.

There are on-going initiatives aiming to develop standards covering various aspects of student mobility. There are some projects having a European scope, like *Metadata for Learning Opportunities* (MLO) which aims to describe programmes of study and course catalogs, or *European Learner Mobility* (ELM) for processing Europass portfolio (Diploma Supplement, Certificate Supplement, Curriculum Vitae, Mobility, Language Passport). Both projects are run by CEN (*European*

Commission for Standardization). At EUNIS'2009 we also learned about numerous national initiatives like that of OODI Finland, FS Norway, or German universities. Last but not least there are vendor "standards", for example the one used by QS unisolution to implement web services for users of the moveonnet. There is also a standard called SCHAC (*SCHema for ACademia*), developed by TERENA, which defines a set of attributes to describe individuals in the academic and research institutions and contains an appropriate LDAP profile. Yet there is no official or unofficial standard regarding electronic data in the context of mobility scenarios.

The Mobility Project has its specific objectives and needs:

- data should be well structured;
- pieces of data should be uniquely identifiable (on a database level);
- each party may want to use own identifiers and own language;
- data format should be generic, i.e. facilitate the data exchange not only in terms of Erasmus programme but mobility programmes in general.

We decided to reuse some ideas of SCHAC because its purpose is the closest to the problem to be solved. Besides, SCHAC leverages ISO and RFC norms and provides precise definitions. We have also made a few definitions connected with domain classification, grade, ECTS credits to some extent similar to those found in Europass.

For types of information which already have an ISO standard (like country code, language code, gender) we use ISO. We introduced a helper type `internationalizedStringT` to handle a list of strings in different languages.

To uniquely identify HEI we follow SCHAC ideas and adopt *domain names* (for example for the University of Warsaw that is `uw.edu.pl`, for the University of Parma - `unipr.it`, for the Autonomous University of Barcelona - `uab.cat`, for the University of Oslo - `uio.no`). This identifier is more generic than for example Erasmus code (like `PL WARSAW01`) since it is available for all HEI's, not only participants of the Erasmus Programme, and also for non-educational institutions, like the ones offering internships to students. We also reuse SCHAC's `schacHomeOrganization` type.

Identifiers of students and employees are coded as `organizationalPersonIdT` type which consists of an `organizationIdT`-like prefix, a colon and an id given by the organization, e.g. `uw.edu.pl:60225`. Type `organizationalPersonIdT` would have a more elegant definition, if XML Schema was able to use values of complex types with key definitions (`xsd:key`).

In case a national id of a person is needed, there is the type `nationalPersonIdT` which is a string based on SCHAC's attribute `schacPersonalUniqueID` but does not use a prefix `urn:mace:terena.org:schac:personalUniqueID` and id type information. Value consists of a country prefix (exactly as in `countryCodeT`), a colon and an identification number, e.g. for Polish citizen that might be `PL:85102702439`. The national identifier is the most convenient kind of personal identifier possible but due to the legal issues connected with personal data protection regulations in some countries it may not be possible to share such data. Therefore its use is optional.

Courses are identified by codes unique within an organization but use of pairs [`organizationId`, `courseCode`] is not a necessity due to the fact that course codes are always placed within a context of an organization.

In order to identify a bilateral agreement between organizations, type `agreementIdT` is provided. It consists of two sub-ids for both organizations. These sub-ids comprise of an `organizationIdT` value like `uw.edu.pl` along with an internal agreement identifier - `localAgreementIdT`, like `1207/E/2009`.

For describing a person there is a type `personalCharacteristicsT` which contains a sequence of personal data. A type `employeePersonalCharacteristicsT` extends it with data specific for an employee, and `studentPersonalCharacteristicsT` with data specific for a student (see Figure 3).

There are also organization-related types (`organizationDataT`), course-related types (`studyCreditsT`, `contactHoursT`, `courseDataT`, `gradeT`, `academicYearT`), and also agreement-related types (`cooperationConditionsT`). Last but not least there are web-service methods (listed in section 3). For example a method `sendNominatedStudents()` sends to a partner organization a list of students from a home organization, using `nominations` element (see Figure 4 and Figure 5).

We used free version of *Liquid XML Studio* to generate a HTML documentation for the WSDL document enriched with graphical representation of its parts (see Figure 3 and Figure 4).



Figure 3 Datatype for describing a student

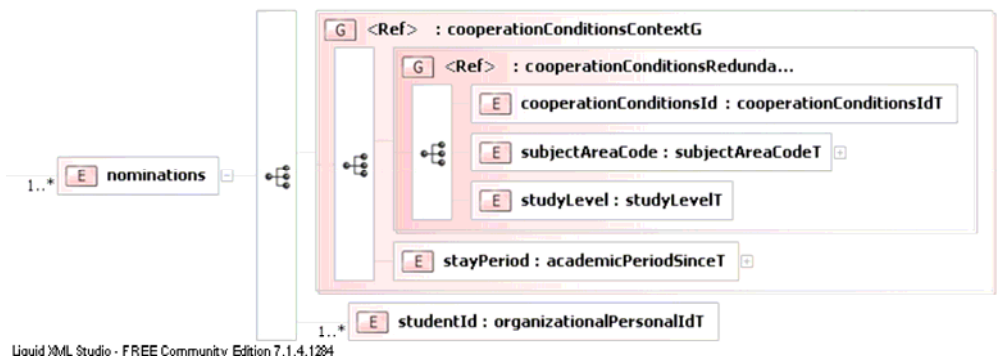


Figure 4 Graphical representation of a nomination element

```

<tns:nominations>
  <tns:cooperationConditionsId>1</tns:cooperationConditionsId>
    <tns:subjectAreaCode>
      <tns:value classification="eu">11.3</tns:value>
    </tns:subjectAreaCode>
  <tns:studyLevel>1</tns:studyLevel>
  <tns:stayPeriod>
    <tns:academicYear>2008</tns:academicYear>
    <tns:academicPeriod>S1</tns:academicPeriod>
    <tns:duration>1</tns:duration>
  </tns:stayPeriod>
  <tns:studentId>unipr.it:9001</tns:studentId>
  <tns:studentId>unipr.it:7042</tns:studentId>
  <tns:studentId>unipr.it:4603</tns:studentId>

```

Figure 5 Sample list of nominations in WSDL

4.2. Software version 2.0

We have chosen P2P-like architecture, where every node acts as a server and client at the same time. UDDI (*Universal Description Discovery and Integration*) registry stores addresses of endpoints. The protocol used to perform data exchange is SOAP defined in terms of a WSDL document.

The software is divided into 4 independent modules (distributed as war archives): 2 transport modules, one for a client and one for a server side of communication, and 2 web interfaces for each transport module. An overview is presented in Figure 6. The modules are all located in the middle part of the diagram, which is labelled “Transport middleware”. The top box represents client transport and web modules while the one at the bottom - server modules respectively.

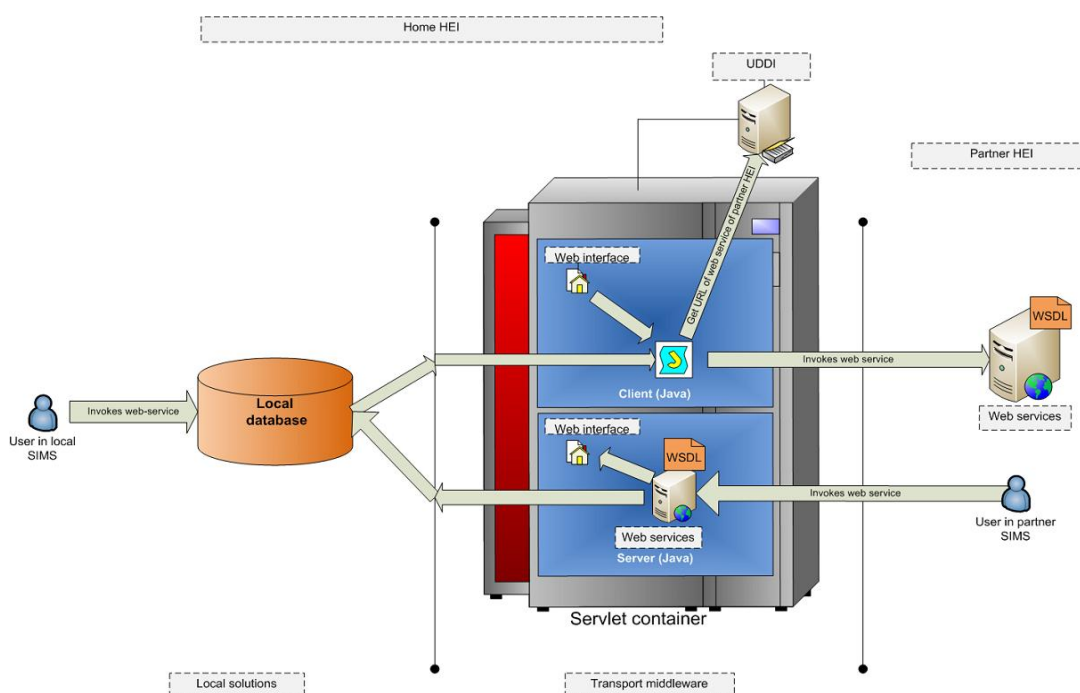


Figure 6 Architecture of the prototype version 2.0

4.2.1. Technologies used in the project

The software platform chosen for the project is Java with Spring framework to manage and configure application objects.

Apache acts as HTTP server and Apache Tomcat as a servlet container capable of running Java applications (the first prototype used Glassfish but we had problems to make it interoperate with Oracle using only open-source solutions).

Apache Camel is a routing and mediation engine which provides an Enterprise Integration Pattern implementation. It can work directly with numerous types of transport or messaging models, including but not limited to HTTP, JMS, AMQP, CXF - the project actually utilizes Camel's CXF support for web services, Oracle *Advanced Queueing* access through JMS (*Java Message Service*) API and AMQP (*Advanced Message Queuing Protocol*) support. Interaction is done through a uniform API for all types of transport, yet it does not prevent access to specific characteristics of the underlying transport layer.

Apache CXF is a web services framework allowing to build SOAP and WS-* standards-based services through JAX-WS API (*Java API for XML Web Services*). It also integrates with Spring framework and is easily embeddable in custom solutions.

Apache jUDDI is a UDDI Registry implementation. MySQL acts as a DBMS backend for jUDDI.

Apache Scout is an implementation of the JAXR API which allows to interoperate with a UDDI version 2 compliant Registry instance.

Apache Qpid is an AMQP implementation.

ICEfaces is a server-based RIA (*Rich Internet Application*) framework. It is an extension to the standard JSF specification. It supports AJAX and thus it can handle rendering phase a bit differently. Other reasons to be in favour of ICEfaces are attractive default look and a rich component library.

Mobility-client-transport and mobility-server-transport applications act as a transport layer between HEI's RDBMS and the web service. Both of them use Camel's Java DSL to route the messages from database and vice versa. The transport layer interfaces to the SIMS with XML payload of the web service messages.

Because of the requirement of interoperability with USOS which has its logic coded in Oracle procedures, the software communicates with it through PL/SQL procedures and Oracle *Advanced Queuing* messaging system abstracted with JMS interface. Other solutions may expose a different interface but there is a great chance that it would be relatively easy to integrate with Camel due to its extensive communication technologies support.

4.2.2. Web client and server

Mobility-client-web and mobility-server-web applications are mainly testing tools. Mobility-client-web enables a user to invoke operations of the web service manually and to edit UDDI Registry. 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 Figure 7.

The screenshot shows the 'Mobility Client' web application interface. On the left is a navigation menu with categories like 'Information', 'Web Service Methods', 'UDDI Registry', and 'Help'. The main area is divided into 'Client' and 'Server' configuration sections. The 'Client' section has fields for Id (uw.edu.pl), Name (pl:Uniwersytet Warszawski), and WSDL URL. The 'Server' section has fields for Id (unipr.it), Name (it:Universita degli Studi di Parma), and WSDL URL. Below these is the 'Request' section, which includes a 'Resource Path' and a 'Sample Data File' (sendNominatedStudentsRequest.xml). A text area displays the XML request payload, which includes details for a student named Michal Harek. At the bottom, a 'Response' section shows the XML response from the server, indicating a successful call to sendArrivalDateResponse.

Client
Id: uw.edu.pl
Name: pl:Uniwersytet Warszawski
WSDL URL: http://usos.php.7879/mobility-server-transport/webservices/mobility-service?wsdl

Server
Id: unipr.it
Name: it:Universita degli Studi di Parma
WSDL URL: http://usos.php.7878/mobility-server-transport/webservices/mobility-service?wsdl

Request
Resource Path: xml/client=uw.edu.pl /server=unipr.it
Sample Data File: sendNominatedStudentsRequest.xml

```
<?xml version="1.0" encoding="UTF-8"?><tns:sendNominatedStudentsRequest
xmlns:tns="http://mobility.usos.edu.pl" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://mobility.usos.edu.pl MobilitySchema.xsd">
  <tns:agreementId>
    <tns:homeId>
      <tns:organizationId>uw.edu.pl</tns:organizationId>
      <tns:value>982/E/II08</tns:value>
    </tns:homeId>
    <tns:partnerId>
      <tns:organizationId>unipr.it</tns:organizationId>
      <tns:value>1207/E/XI08</tns:value>
    </tns:partnerId>
    </tns:agreementId>
    <tns:academicYear>2008</tns:academicYear>
    <tns:nominatedStudentsWithPersonalData>
      <tns:uniquePersonalData>
        <tns:student>
          <tns:nationalPersonalId>PL:85
          </tns:nationalPersonalId>
          <tns:givenName1>Michal</tns:givenName1>
          <tns:givenName2>Harek</tns:givenName2>
          <tns:surname1>Kurzydowski</tns:surname1>
          <tns:gender>1</tns:gender>
        </tns:student>
      </tns:uniquePersonalData>
    </tns:nominatedStudentsWithPersonalData>
  </tns:agreementId>
  <tns:studentId>uw.edu.pl:8001</tns:studentId>
</tns:sendNominatedStudentsRequest>
```

Data successfully sent.

Type	Line	Column	Report
Response	1		
	2		
	3		
	4		
	5		
	6		
	7		

```
<?xml version="1.0" encoding="UTF-8"?><tns:sendArrivalDateResponse xmlns:tns="http://mobility.usos.edu.pl"
xmlns:xsi="http://mobility.usos.edu.pl MobilitySchema.xsd" xmlns:soap="http://schemas.xmlsoap.org
/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
```

Figure 7 Calling sendNominatedStudents() using the web client interface

Mobility-server-web enables a user to preview requests coming to a server. It accepts web service calls from the net and forwards them to the local system, storing them also in the local file system. The interface is demonstrated on Figure 8.



Figure 8 Browsing results of the web-service call using the web client interface

4.2.3. Test deployments of the software

To test the new version of the software we deployed it first in the University of Warsaw, then at three other sites, KION in Bologna (hosting a server for the University of Parma), SIGMA in Barcelona (hosting a server for the Autonomous University of Barcelona) and in the University of Oslo (as for the end of April, 2010). The test nodes of the Mobility network are shown on the Google map in Figure 9.

We found it relatively easy to adapt the software to new versions of WSDL (of course reinstallation is needed each time WSDL is being changed). We encountered some problems with deploying the software in system environments of KION and SIGMA. First, it turned out that although SIGMA uses Oracle as the database engine for the local student management information system, they prefer Glassfish over Tomcat since they use it for other purposes. It thus means that we should have various options possible, i.e. to distribute software which can be used with any servlet containers which might be chosen by deployment teams. In KION it turned out that there are problems when trying to install jUDDI on Oracle. We found out that versions of jUDDI for various databases are not compatible. This needs further investigation - we plan testing the newest 3.0 version of jUDDI.

New installations are on the way which will give us more experience with deployment in various system environments and new requirements concerning the chosen technologies.

To ease the installation of the software we prepared a preconfigured installation package, with the web client and the web server settled inside Tomcat and a small database for UDDI entries. Installation and deployment of the software from the new package takes no more than 5 minutes. We also plan to prepare a package with a virtual platform, like VMware (free version).

For testing purposes we have registered mobility web services at the public, centralised registry hosted by EdUnify at <https://demo.edunify.pesc.org/> (they are available at <https://demo.edunify.pesc.org/services/11>). The EdUnify Registry allows to easily register, discover, annotate, monitor and use web services. Service providers, expert curators and the wider community are invited to enter metadata which is indexed and then used to support searching. It is thus easy to find the registered web service based on its type, category, location, descriptions, tags, input type etc.

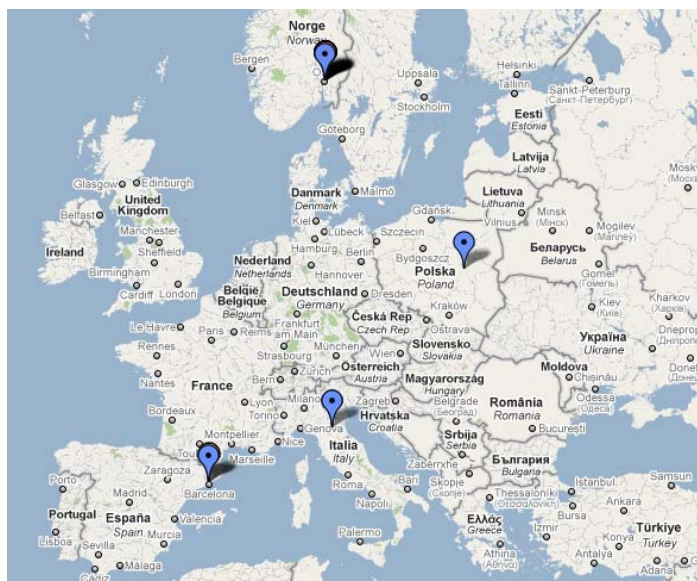


Figure 9 Google map showing the nodes of the mobility network

We are aware that the real challenge is the integration of the mobility web services with local student management information systems, which are built in variety of technologies, each of them needing specific solution. In [AMR2009] we described a solution for USOS, which is used in Polish universities. USOS is built around Oracle database so we utilize Oracle based technologies: *Oracle Advance Queueing* to implement web services invoked by local users and XML buffers and stored procedures to handle web services invoked by remote users. The module of USOS for International Relations Office is equipped with extra buttons and windows for invoking web services and browsing the received data.

The other foreseen challenge is how to ensure, in the production environment, compatibility of nodes running various versions of WSDL (which is unavoidable).

5. WORKSHOPS IN UPPSALA AND IN BOLOGNA

The next step was the workshop in Uppsala in November 2009 which brought together consortia and companies willing to take an active part in the project. Among these were some national consortia (OODI - Finland, SIGMA - Spain, CINECA - Italy, Ladok - Sweden, FS - Norway, MUCI - Poland, HIS - Germany, SURF - Netherlands, VHS - Sweden), companies (Digitary - Ireland, AcademyOne - USA, KION - Italy, QS unisolution - Germany) and individual universities. Presentations were devoted to various aspects of mobility, standards for electronic data exchange, relevant software systems:

1. Business processes in the area of international cooperation and mobility at the University of Thessaloniki.
2. Europass Mobility System produced by CEDEFOP (*The European Centre for the Development of Vocational Training*) to support handling of Europass documents in a distributed environment.
3. QS unisolution model for exchange of the mobility data by users of *move-on*.
4. Update on the MLO/ELM standardization processes.
5. Update on European higher education Identity and Access Management.
6. Update on the development of WSDL and software for the Mobility Project.

The most important outcome of the Uppsala workshop was formation of the Work Teams responsible for further investigation of various aspects of the Mobility Project: Business Processes, WSDL format, Identity and Access Management, Security and LifeCycle Management, System Architecture. Team

leaders were appointed to animate the work. Some participants volunteered to take part in practical activities, deploy software and test suitability of WSDL by preparing files with sample data.

The workshop in Bologna took place in March 2010. Its main purpose was to bring together the Mobility Group and TF-EMC2 to discuss the aspects of security and of identity and access management of the Mobility Project. There was a presentation on the Security and Document Lifecycle given by the leader of the Security Team. Secure encrypted transfer protocol should be used for sending data since in many countries personal data are protected by law and their privacy cannot be jeopardized. Digital signatures and PKI can be considered if higher level of security becomes necessary. TF-EMC2 shared with the participants the results of the recent Andalusian experience in FIAM implementation (*Federated Identity Access Management*). Vivid discussion followed, the debaters tried to assess how IAM could be brought into the Mobility Project. Possible solutions were recognized and further actions planned.

6. CONCLUSIONS

The initiative of the Mobility Group seems to us an important and practical step towards the creation of the European Higher Education Area. At EUNIS'2008 MUCI and CINECA met for the first time and agreed to start a project on electronic exchange of data on international cooperation and mobility. At EUNIS'2009 the running software system was presented, capable of transferring data between the two countries. We are ahead of EUNIS'2010, having the new more mature version of the software, four nodes in four countries (Poland, Italy, Spain and Norway) ready to transfer data, and a large group of national consortia, software vendors and universities willing to cooperate. The final success of the project depends on its scope which is proportional to the number of student management systems able to exchange mobility data by web-services. There is still a lot to be done but we believe that by such small steps we will eventually make it possible and some day the button inside the red rectangle in Figure 2 will become ready to send data to any European Higher Education Institution.

Acknowledgements

Many thanks to Rafał Nagrodzki, Michał Kurzydłowski, Artur Poptawski, and Janek Rudziński who implemented the second version of the prototype system.

7. REFERENCES

- [AMR2009] Arcella, F., Mincer-Daszkiewicz, J., Ravaioli, S. (2009). Webservices for Exchange of Data on Cooperation and Mobility between Higher Education Institutions, EUNIS'2009. Retrieved January 25, 2009, from: <http://www.eunis.es/programme/>
- [AMR2009b] Arcella, F., Mincer-Daszkiewicz, J., Ravaioli, S. (2009). Harmonizing Higher Education: "Standard" perspectives in support of the Bologna process toward the creation of a European Higher Education Area. Workshop at EUNIS'2009 organized by RS3G
- [Mobility] Home page of the Mobility Project, <http://usos.edu.pl/Mobility>
- [Nag2009] Nagrodzki, R. (2009). The Mobility Project, Master Thesis, University of Warsaw
- [RS3G] Home of Rome Student Systems and Standards Group (RS3G), <http://wiki.teria.no/confluence/display/RS3G/Home+of+RS3G>
- [Uppsala 2009] From Idea to Action, Workshop of the Mobility, organized by RS3G, Uppsala, November 16-17, <http://wiki.teria.no/confluence/display/RS3G/UPPSALA>
- [USOS] Home page of the University Study-Oriented System, <http://usos.edu.pl>