**Title: Educational Certificates Data Exchange**

**Abstract:**

The student data is currently exchanged manually between educational institutes in Pakistan. The thesis is an effort in suggesting a prototype infrastructure that will enable these institutes to inter-exchange student degree record digitally. Infrastructure represents data format and architecture.

Keywords: data exchange format, bologna process

**Abbreviations and Acronyms**

|  |  |
| --- | --- |
| SCHAC | Schema for Academia |
| LDAP | Lightweight Directory Access Protocol |
| FVUSPEC | Finnish Virtual University Specifications |
| MLO | Meta-data for Learning Opportunities |
| WSDL | Web Service Description Language |
| Requester |  |
| Provider | University or Board or any other Degree Issuing Authority which serves the requests |

**Introduction**

**Literature Review:**

**Bologna Process**

It aims to develop european educational framework of standards that will enable different countries to compare and make compatible their educational systems.

<http://www.ehea.info/Uploads/Irina/Bologna%20beyond%202010.pdf>

There exists many related data exchange standards and related systems;

**Data Exchange Standards**

These are;

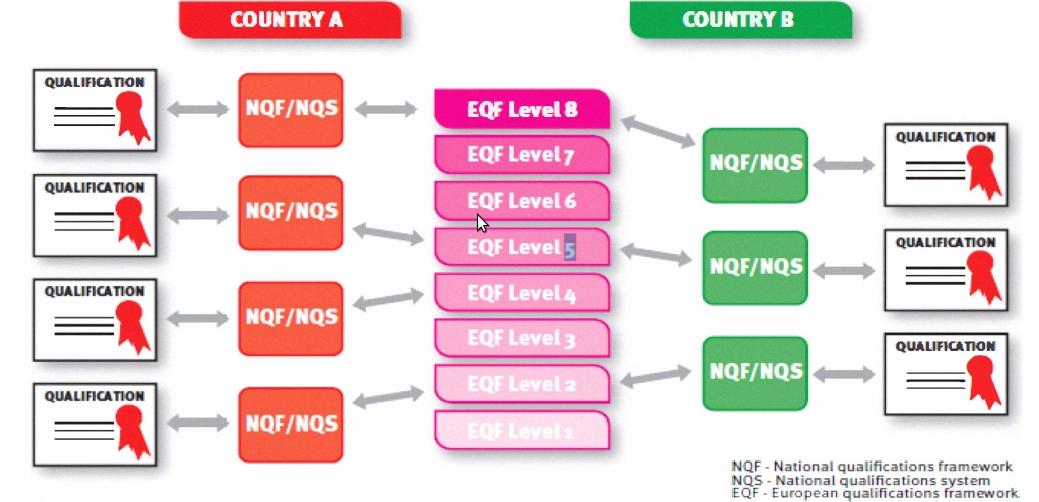
**European Initiatives**

European Qualifications Framework (EQF), Europass, and European Learner Mobility.

**European Qualifications Framework (EQF)**

EQF is a common reference framework that will help participating countries to compare national qualifications and make them more readable across europe. The aim is to promote mobility of workers and learners. This was agreed upon by european universities in 2008 to relate their national qualifications to EQF. The new qualifications from 2012 carry a reference to suitable EQF level.

EQF consists of eight reference levels representing what a learner knows and is able to understand. National qualifications of the participating countries relate to these eight levels ranging from basic level (level 1) to advanced level (level 2). This enables comparison of qualifications in different countries easy, thus providing support for mobility for learners and enabling them to not repeat what they have already learned. The figure below is from [3].



EQF focuses on learning outcomes instead of focusing on learning inputs. It covers all types of education including vocational, professional, school education and qualifications. It tries to validate formal as well as informal education.

EQF SPREAD PROJECT

**Europass**

It is collection of five documents which intend to ease mobility in European job market. These include the Curriculum Vitae, the Language Passport, the Mobility, the Diploma Supplement, and the Certificate Supplement. One can fill himself the Curriculum Vitae, and the Language Passport but the rest of the documents are issued by the corresponding authorities. It follows a template format system. All documents having same format help to achieve neutrality and transparency while presenting one’s skills.

The motto as mentioned on the europass website’s homepage is as follows;

“Five documents to make your skills and qualifications clearly and easily understood in Europe”

Europass has defined XML schemas for CV and Language Passport. The documents can be exported in XML format when created on Europass. These exported XML documents can be imported to Europass and converted to HTML, PDF, Microsoft Word or ODT templates. The Europass XML version 3 is available at [this link](http://europass.cedefop.europa.eu/TechnicalResources/XML/xsd/Europass%20XML%20Schema%20V3.0-Documentation-And-Changelog.pdf). The CV vocabulary sample XML, I generated on March 03, 2013 is [here](https://docs.google.com/file/d/0BxY0BVINEp7NMl9SOWljMnpUMGc/edit?usp=sharing).

Europass specifies JSON schema according to Internet Engineering Task Force’s JSON specifications [draft](http://tools.ietf.org/html/draft-zyp-json-schema-02). The europass JSON vocabulary is close and similar to europass XML schema. The JSON objects for europass documents (CV and Language Passport) can be validated using [europass JSON validator](http://europass.cedefop.europa.eu/en/resources/for-developers/json/validation).

All these documents have some common XML schema attributes which describe document type, printed preferences.

<DocumentInfo>

<DocumentType>ECV</DocumentType>

<CreationDate>2012-03-01T00:00:00.000+03:00</CreationDate>

<LastUpdateDate>2012-03-01T00:00:00.000Z</LastUpdateDate>

<XSDVersion>V3.0</XSDVersion>

<Generator>XML Editor</Generator>

<Comment>Example CV XML according to XSD v3.0</Comment>

</DocumentInfo>

Europass does not explain details related to degrees or educational certificates in XML certificate.

**ECV**

Europass Curriculum Vitae is a template which one can create online and it can be exported in xml format. The ECV XML schema contains vocabularies related to document type, printing preferences, personal details, contact details, skills, and educational degrees and institutes. The XML vocabulary related to degree details is very little only to cover the scope of a CV.

**ELP**

Europass Language Passport is a template. One can create it online and export it in europass xml format. It contains XML vocabulary related to language skills and the scale of six values to score proficiency.

**SCHAC**

Schema for Academia describes vocabulary related degrees and courses. The schema is written for LDAP (Lightweight Directory Access Protocol). It aims at promoting a common framework to inter-exchange data between educational institutes. It defines attributes that describe individuals and their LDAP profile.[4] For individual attribute specification, use [this link](http://www.google.com/url?q=http%3A%2F%2Fwww.terena.org%2Factivities%2Ftf-emc2%2Fschacreleases.html&sa=D&sntz=1&usg=AFQjCNE2FTKeZ9yozJMc3E4r7EGzUVa5Mw).

(FVUSPEC, MLO)

**Dublin Core**

The Dublin core is a simple meta-data standard consisting of set of elements to describe information resources on the network. There are two type of elements; simple and qualifiers. It has 15 simple elements and qualifiers which have additional three elements namely Audience, Provenance and RightsHolder. Qualifiers help in resource discovery.

**European Learner Mobility**

Some related work has been done recently and systems have been proposed based on the above mentioned standards. These are “The Mobility Project” and “The REST Mobility” projects.

**The Mobility Project [1] by Rafal Nagrodzki**

It aimed to provide a platform and infrastructure for exchange of electronic data exchange between educational institutes. Infrastructure includes data format, architecture and the prototype software. The system will called The Mobility later in this thesis.

The Mobility is peer to peer like architecture. Nodes exchange data using SOAP base web service. Other web services like XML-RPC and REST were not used due to their limitations. XML-RPC not have developer defined data-types and character set. REST does not imposes a standard specification, instead it follows set of rules and is used for speedy development of web service interface.

The nodes represented the universities, and their number tend to change. So there was a need for system to maintain this record and UDDI was used. He does not recommended the central or delegated private registry instead gave advantages and disadvantages of both. Central single registry have all information at one place but also it a single point of failure.

The software has two transport modules and each have web interface.

Nagrozki proposed a new standard, defined its vocabulary re-using ideas taken from SCHAC to leverage ISO and RFC rules. Some like grade, credits were taken in inspiration from Eropass Mobility.

Although The Mobility project was started by MUCI and CINECA, two European Higher Education Consortia. Many universities consortia, individual universities and companies joined in later on.

**The REST Mobility by Karol Kanski [2]**

This is alternative implementation of Nagrozki’s mobility system. Nagrozki’s system used SOAP web service for data exchange. Karol created a RESTful implementation of the Mobility. The Mobility lacked data model. In The REST Mobility a data model is proposed since REST is resourceful. The model proposed not represents or intends to be a standard.

**Information Manyfold [Helvey’s Paper]**

Providing a uniform interface for querying data from many sources is the aim of Information Manyfold. It enables a simple user to not worry about locating sources and manually combining results. This leads to concept of Deep Web. Data integration systems give users a common global schema called mediated schema for posting queries. To answer these queries semantic relationships called mappings are needed between mediated schema and the sources schema.

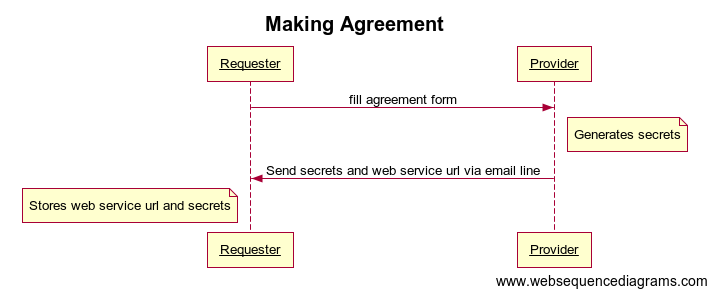
Local as View (LAV) and Global as View (GAV).

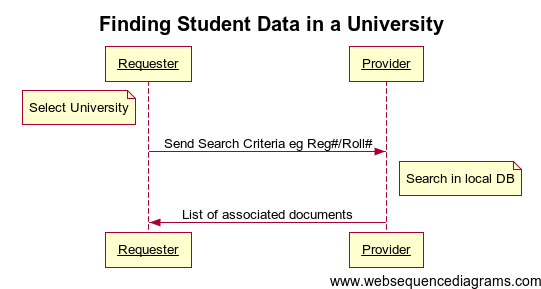
**Requirement Analysis**

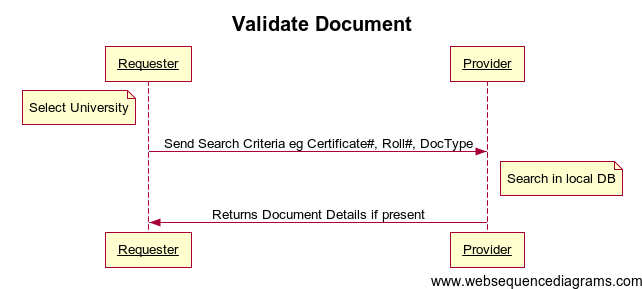
**Business Process**

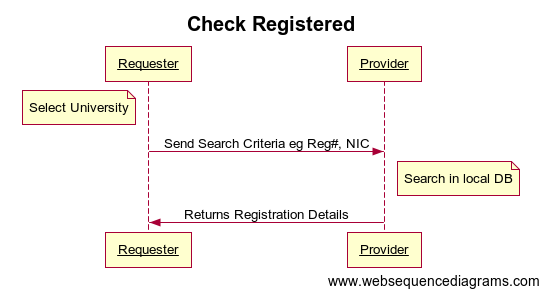
There are many universities. For two universities to exchange data, they have to create an agreement first. The agreement will have the list of documents for which data exchange is available and exchange secrets will be generated.

**Sequence Diagrams**



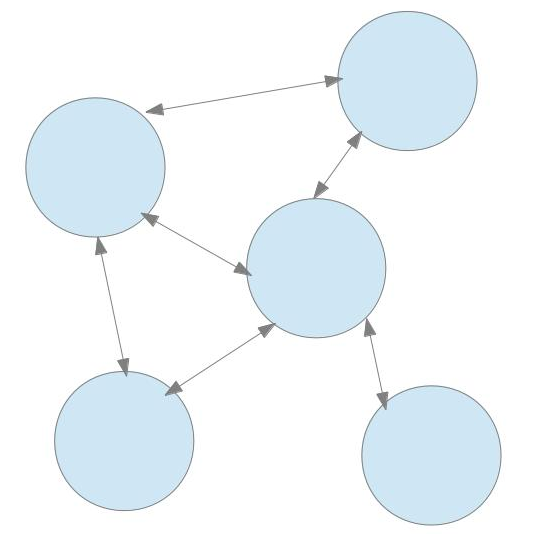






**Architecture and Design**

From use cases and the business requirements, we suggest student exchange system will have distributed architecture. Each university has its own data and signs agreements independently for exchanging data with other universities. Each can be a requester plus a provider of data. The circles/nodes in the figure below represents universities. The arrows represent exchange of data.



This peer to peer like distribute architecture has benefits over adding a middle agent or central server in the system.

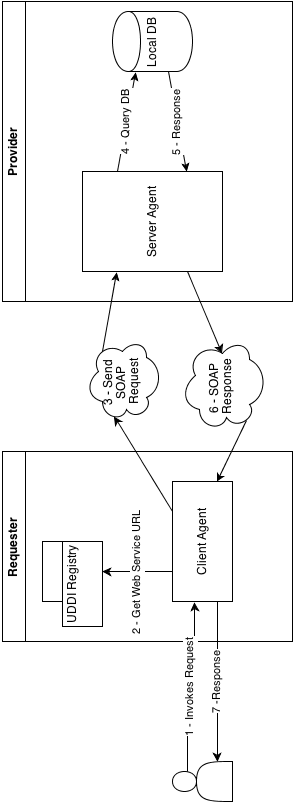
1. Avoidance from single point of failure.
2. Less load.
3. Each university having control over its own data and thus building trust in the system.

Now we need to decide how these nodes will exchange data. Which protocol will be used for data exchange. There are some choices to be made at this point. We will be using web services for exchanging data as they provide a high abstraction from network issues and use well known standards like XML over HTTP. There are some XML based data exchange protocols on web. These are XML-RPC, SOAP, and REST.

The nodes will exchange data using SOAP based web service in our system. We chose SOAP as it forces to follow a formal standard and supports developer defined data types.

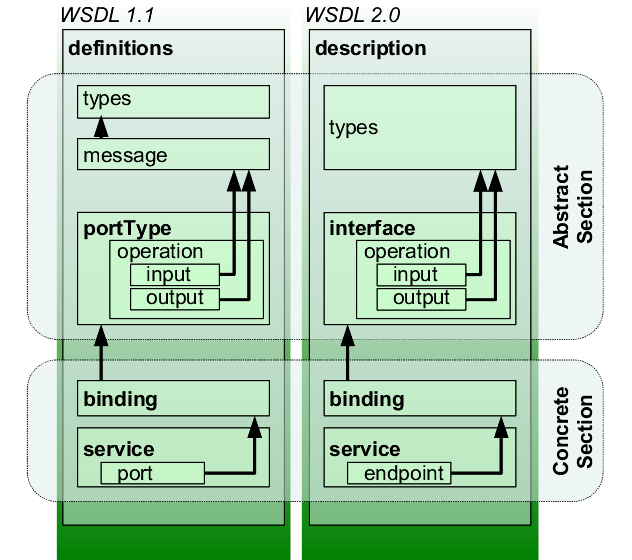
The number of universities can increase when agreements are signed with new universities for exchange data. The web service URLs need to be saved so that requester can retrieve this URL and request that university. This can be achieved by developing a custom system or using UDDI. UDDI is a standard used to discover and save web services URLs. Now we have to make a choice. UDDI can be global or each requesting node can have its own private UDDI registry. We will use private registry to avoid single point of failure and to minimize load.

**System Architecture**



**WSDL Document**

vocabulary



**Implementation**

First we setup a UDDI registry server to store the URLs of the web-services provided by different universities. This is done by making an agreement with each university. We will be using PHP UDDI Registry for this purpose.

Then take a sample university database having degree record. For sample database we will be using MySQL DBMS. Apply operations on the data-set and return corresponding response. This exchange of information is done in XML SOAP format as it forces to follow a specification. The SOAP server will be implemented in PHP.

Below are the list of operations available;

1. Make agreement: Storing web-service URL in UDDI registry.
2. Find student data in university using SOAP service.
3. Validate a degree document.
4. Check registration of a student in a university.

When using SOAP format, it is optional to use WSDL. It is a web service description language document. We will generate this document using php2wsdl tool. This is an open-source php library used for generating WSDL document.

Setting Up PHP UDDI Registry Server

Data Provider Module / Library

class DataProvider

{

public function findStudent ($param, $criteria) {

$dwarves = array("Bashful", "Doc", "Dopey", "Grumpy", "Happy",

"Sneezy", "Sleepy");

return $dwarves;

}

public function validateDocument ($param, $criteria) {

$dwarves = array("Bashful", "Doc", "Dopey", "Grumpy", "Happy",

"Sneezy", "Sleepy");

return $dwarves;

}

public function checkRegistered ($name) {

return array("message" => "Hello, " . $name);

}

public function greetUser ($name) {

return array("message" => "Hello, " . $name);

}

}

SOAP Server

require('dataprovider.php');

$options = array("uri" => "http://localhost");

$server = new SoapServer(null, $options);

$server->setClass('Library');

$server->handle();

USING PHP2WSDL : Generating WSDL Document

SOAP Client

$client = new SoapClient('http://api.radioreference.com/soap2/?wsdl&v=latest');

$countries = $client->getCountryList();

/\*$options = array("location" => "http://localhost/sdrdes/soap-server.php",

"uri" => "http://localhost");

try {

$client = new SoapClient(null, $options);

$dwarves = $client->getDwarves();

var\_dump($dwarves);

} catch (SoapFault $e) {

var\_dump($e);

}\*/

**References**

1. The Mobility Project
2. The REST Mobility
3. MAPQFTOOL
4. The Mobility Project - Building network of web-servers for exchange of data on student mobility