



UPC - FIB
Universitat Polytecnica de Catalunya - Facultat de Informatica de Barcelona

Final projet assignement report

Performed at BULL SAS

Implementation of a digital workspace

Aurore Weber

Master in Information Technology (MTI) Software Engineering and Information Systems

February 2009, the 1st – July 2009, the 31th

BULL SAS

Espace Bellegrade
Bat A 985,Chemin du Mas de Rochet
34174 Cedex - Castelnau le lez

Company tutor

Jerome Ricard

School tutor

Xavier Franch

Contents

Abstract	4
Acknowledgments	4
1 Context of the internship	5
1.1 Overview of the company	5
1.2 Detail of the project	9
2 Digital workspace : definitions and implementations	11
2.1 Portal	12
2.1.1 Web portal	12
2.1.2 Portlet	14
2.1.3 Portlet specifications	16
2.2 Requirements	19
2.2.1 Specification	19
2.2.2 Available modules	21
2.2.3 Appearance of the portal	26
2.3 Technical points	27
2.3.1 Portal comparison	27
2.3.2 Architecture and organization of the portal	39
3 Data integration	48
3.1 Amount of data	48
3.2 Source of the data	49
3.3 Data transfer to the LDAP directory	51
3.3.1 Functionalities	51
3.3.2 The problematics and analysis	52
3.3.3 The Sarapis project	55
3.3.4 Sarapis Analysis	60
3.3.5 Modification to the Sarapis project	61
3.3.6 Test case	62
3.4 Management of external datas : Initialization	64
3.4.1 Functionalities	64
3.4.2 Architecture	65
3.4.3 Problems and solution	66
3.4.4 Test cases	67
3.5 Management of external datas : Propagation	68
3.5.1 Functionalities	68
3.5.2 Analysis	68
3.5.3 Implementation	70
3.5.4 Test case	71

<i>CONTENTS</i>	3
3.6 Integration of pedagogical resource and Gibii	72
3.6.1 CAS-ifying and integrate	72
3.6.2 Data transfer	73
4 Planning and budget	75
5 Conclusion	84
A Glossary	86
B Accessible modules	87
C Databases schema	89
D CAS life-cycle	92
E XML extract	95
E.1 DTD associated to the XML	95
E.2 XML School example	95
E.3 XML Student example	96
E.4 XML Teacher example	97

Abstract

Implementation of a digital workspace for secondary school pupils of the region Languedoc-Roussillon in the scope of a final project assignment.

In this report, I introduce the context of the internship (company, client, subject,..), then present what are digital workspaces, and I finally consider the problems related to data integration.

Keywords: Portal, uPortal, ESUP, JBOSS, CAS, Portlet, LDAP, Hibernate, Springs

Acknowledgments

I would like to offer my most sincere thanks

- to the ENSIMAG and the FIB for their education
- to the Bull company and more specially Mr Ricard as he hire me as a work experience student
- to my two tutors, Mr Nicollin and Mr Franch for tutoring and supporting me
- to all of my co-workers, Mr Breyton, Mr Carriere, Mrs Pasquier and Mr Vibert for the friendly atmosphere they create and their availability.

1 Context of the internship

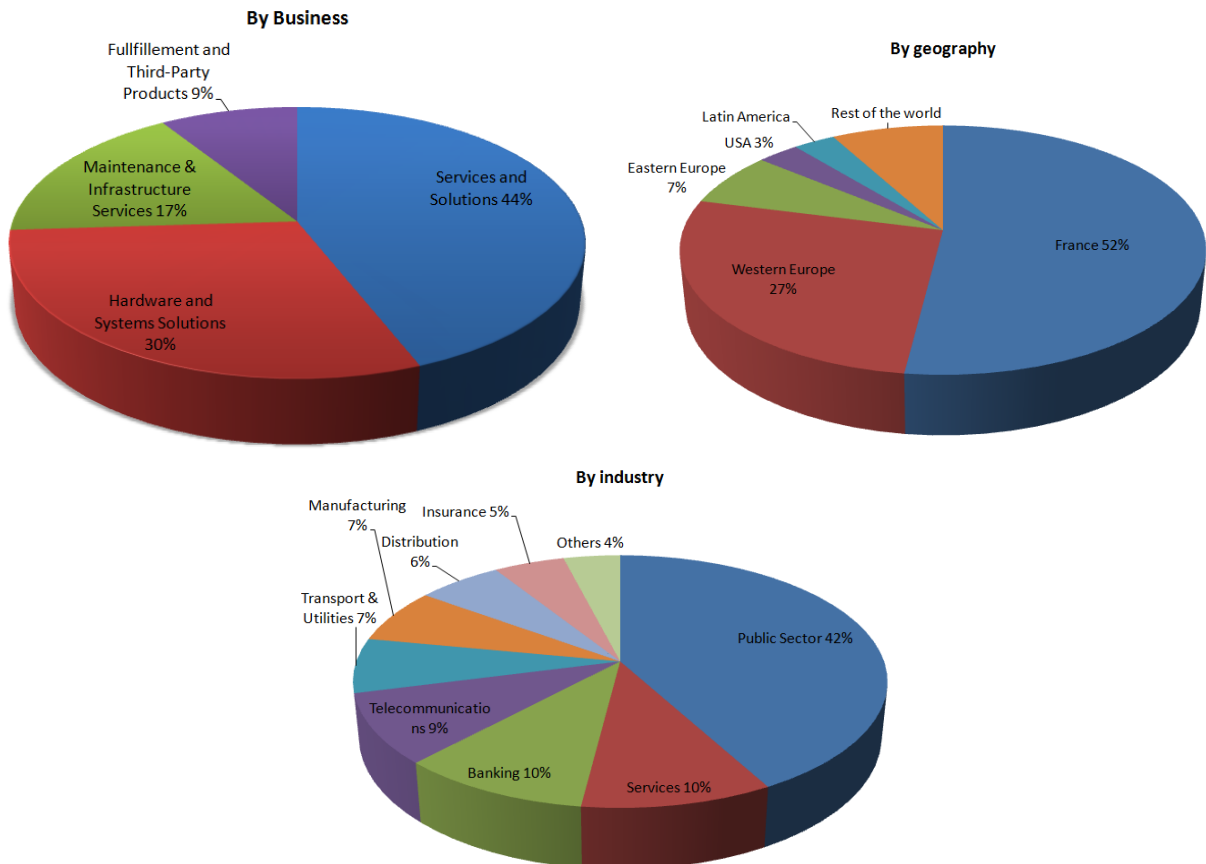
I performed my final project study in Montpellier's division of the Bull Company. I was asked to integrate a team in order to implement a digital workspace for the region Languedoc-Roussillon. In this chapter, I propose an in depth analysis of the company and the subject.

1.1 Overview of the company

Groupe Bull (also known as Bull Information Systems or simply Bull) is a French owned computer company headquartered in Les Clayes-sous-Bois, outside Paris.

Bull - the only European-based company that masters all the key elements of the IT value chain - is a specialist of open Information Systems (IS). It helps enterprises and public sector bodies to build, optimize, run and ensure return on investment in their business-critical IS, so they can operate in total security. They are used to modernize IS by applying open and secure solutions.

Revenue breakdown



Architect of an open World

Bull puts its skills and know-how at the service of its customers to help them develop and fine-tune their IS. Thanks to its versatile offering of open solutions and technologies, Bull helps them set up IS that slot smoothly into their business strategies, optimize costs and build trust.

Building open servers and secure storage solutions for new-generation data centers

Bull's NovaScale open and scalable server family is designed around standard components for data centers and high-performance computing (HPC). As a result, Bull has equipped some of the most prominent R&D centers across the globe, with some of the world's most powerful supercomputers. Bull also offers its powerful Escala range of AIX servers, co-developed with IBM, which is particularly relevant for large IT infrastructure consolidation and virtualization projects.

And with its StoreWay offering - a broad palette of end-to-end storage solutions - Bull meets the challenges of data protection and the exponential growth in data volumes. In partnership with industry leaders, StoreWay combines Bull's innovative technologies with its expertise in IT security and major projects.

Developing middleware for applications and service-oriented architectures (SOA)

As well as developing applications for specific industry sectors, Bull is a leading supplier of open infrastructure solutions. Bull is a founder of OW2 - the world's leading consortium dedicated to open middleware - and a key partner of JBoss/Red Hat in Europe. Bull is also one of the founding members of QualiPSo, the first Open Source quality software platform.

NovaForge, Bull's pioneering software development factory, brings together the best software development resources in a shared, structured environment. By professionalizing and automating these development processes, NovaForge enables projects to be delivered on time, on budget, and with the right levels of quality. Thanks to these emerging practices, Bull helps organizations enter the Virtual Shore era, and optimize the governance of their projects at a time when speed and flexibility are essential.

Providing the security and Open World needs

As European leader in IT security, Bull provides a comprehensive range of solutions: identity and access management (through its subsidiary Evidian), encryption, e-signature, payment systems, secured mobile computing, and Web services security. With globull, Bull has invented the world's most secure mobile computing platform. A large number of European organizations with particularly demanding security requirements - governments, manufacturers and companies in the finance sector, defense, healthcare... - rely on Bull for their IT security solutions.

A solutions-oriented IT services provider consulting

Consulting

With services ranging from IT architecture and system planning all the way through to project management, Bull helps organizations to design or re-engineer their information systems to support their business strategies.

Integration

These services can involve complete information systems or specific business applications (ERP, business intelligence, CRM and HR), computer simulation or industry-specific applications. With NovaForge and its Virtual Shore services, Bull helps its customers to accelerate the development of their applications and industrialize their information systems for optimum availability and flexibility. In the telecoms sector, where Bull's revenues have doubled in just three years, Bull provides a unique mix of skills combining speed, innovation and robustness.

Operation and outsourcing

With its support, operation and 24x7 outsourcing services, Bull helps its customers hand over all or part of their IT operations so they can focus on their core business. Numerous organizations in the public sector, healthcare and telecoms outsource their emergency centers or their entire information system to Bull and its subsidiary Agarik, which specializes in hosting critical Web infrastructures and on-line services.

A unique range of skills for an Open World

Bull, the IT innovator, is:

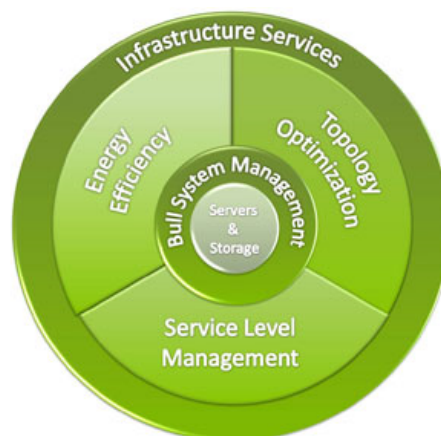
- European leader of enterprise systems and HPC solutions: Its open and innovative technologies have been selected by major research centers and industries across the world.
- Europe's leading integrator of Open Source software: Bull is not only a major provider of consulting, integration and facilities management services, but also one of European leading providers of Open-Source-based integration services.
- Europe's leading supplier of integrated security solutions: Bull is the top European developer of identity and access management solutions. We also provide cutting-edge integration services for end-to-end security solutions.

A supplier of choice for demanding organizations

Every day, in over 50 countries worldwide, hundreds of organizations depend on Bull's products, solutions and services to support their business activities and critical processes, especially in:

- Public sector, to simplify public services and stimulate local development. Bull also delivers specific solutions for Customs and tax authorities to modernize declaration and collection systems, for local and regional authorities as well as social and healthcare services to help them personalize and optimize the service they deliver to beneficiaries.
- Telecommunications and media, to streamline roll-out of innovative services and assure their continuity and robustness.
- Banking and insurance, to improve customer loyalty and guarantee total security of transactions.
- Transport and utilities, to align information systems to new priorities due to deregulation.
- Manufacturing, to accelerate the development of innovative and energy efficient products using computer simulation, optimize the supply chain, and reduce time and costs using ERPs.
- Research, to support innovation with cost-effective HPC solutions.
- Postal services, to automate mail processing and raise reliability.

Bull, as "Architect of an Open World", is committed to helping the most demanding of public and private organizations to grow in an open world with absolute confidence, while respecting the environment.



1.2 Detail of the project

The 'Conseil Regional Languedoc Roussillon' (Languedoc Roussillon' County Council), in partnership with Montpellier' education office, initiated a project in order to provide to every single secondary school pupils of the region a digital extension to their own academic institution's life.

All academic institutions are being equipped with computers and connected to the Internet. Many Technologies of Information and Communication for Education (TICE) have been developed in the teaching area and in the scholar life. School is currently modernizing both formation and education.

Now, the evolution speeds up and new possibilities show up. It's time to give a new impulsion to the TICE. In the meantime, new offers have appeared in terms of education: scholar help, preparation to exams, pedagogical on line resources or even orientation information.

This particular project aimed at producing a digital workspace which will provide for all pedagogical and management needs in the educational context.

It is more precisely in line with the '4-5-6' plan which is aiming at deploying the solution in high school (4 years), secondary school (5 years) and primary school (6 years) of the region. The '4-5-6' plan is an attempt to successfully integrate those new usages and services by taking the best of them. All students, must eventually get an access and participate to the e-learning.

This digital workspace is designed to propose a unique and secure gateway to all on line services offered to the users. It will respect the single authentication principle and each application will manage by itself its own access right to its services.

The project will be materialized by a website available 24/7 on the Internet for all the stakeholders of the educational world (Students, Teachers, Student family, Educational staff, Educational Inspector, Companies involved in education, ...). Thanks to this portal, users are offered many different services including:

- Web mail
- Address book
- Management of scholar life (timetables, marks, homework notebook, ...)
- ...

Within the team of developers, I was in charge of the data integration. It included the introduction of new data in the main database through an automated process, the management of data modification and the data forwarding to other application. This

central process allows all the application to be synchronized at any moment with the data provided by the government.

2 Digital workspace : definitions and implementations

Our digital workspace is designed as an communication platform. It aims to gather every single stakeholder and every piece of on line application.

A portal perfectly fit theses necessity. It will provide:

- single-sign-on (SSO) : A session/user authentication process that lets a user enter one name and password once and for all in order to access multiple applications.
- multi-institution : The portal must be made available for each academic institution of the region AND allow customization by institutions. According to the institution of the concerned user, the portal must provide a different appearance.
- managing profile : The portal must interact differently according to the profile of the user. The actions allowed to a teacher are not the same than for a student.

I will first introduce portal and key functions of them, with a brief comparison of four of them. Then I will detailed our portal requirement and finally describe and comment a few technical points.

2.1 Portal

2.1.1 Web portal

A Web portal is a **unique gateway** to an aggregation of **personalized information and-or resources** for a **specific domain or a given community**.

The three main points of a portal:

- unique gateway : It allows to connect to multiple application through a unique identification service. As a result, the user only get one login and one password for all available resources.
- personalized information and/or resources : There can be as many resources and information as needed, whose access is restricted to people having interest in them.
- a specific domain or a given community : A portal is designed for a specific purpose and focused for a set of person.

Here is a list of the main capabilities expected from any portal solution:

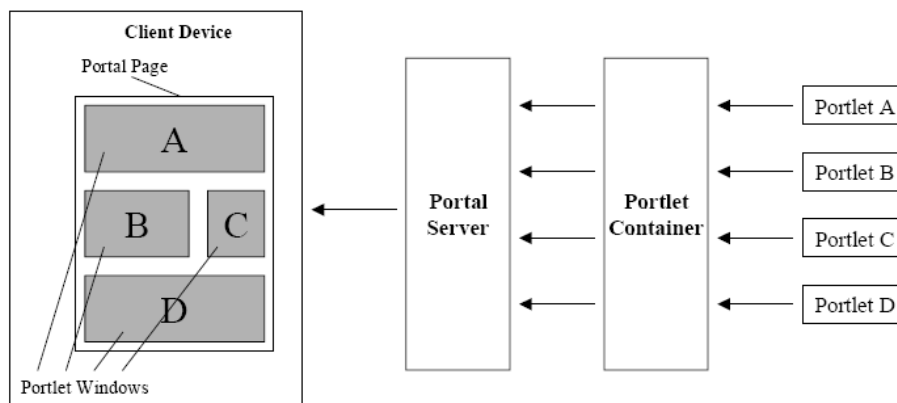
- Aggregation of application and content into a single view
- Integration of data from multiple sources
- Security to restrict access only to authorized users
- Collaboration tools such as shared resources, instant messaging, and chat tools
- Personalization to organize their own on line workspace
- Search technologies to retrieve information
- User-friendly interface
- Categorization of content (taxonomy)
- Content management & aggregation
- Robust application integration
- Development tools
- Single Sign-On

The applications integrated into a portal are referred as 'module'. A module can be integrated by several ways. The best method is the portlet which is outlined in the following paragraph. Other possibilities include:

- Servlet : A Java Application which helps to create dynamically data inside an HTTP server. Most of the time, they are HTML data, although they can be any kind of format accepted by browser. The servlet use the API 'Java Servlet' and runs into a container: the portal.
- iFrame (Inline Frame): A method to insert into a web page (the portal) another HTML page. In an iFrame, we can have information from another server than the portal.
- Web Proxy : Allows incorporation of web-based services regardless of the nature of the technology used to implement them. It provides mechanisms for connecting to and rendering HTML and XML services: When a portal incorporates web applications or web services, it must intercept the communication to tailor it for the portal environment. This is done by rewriting the application's output appropriately. In particular, rewriting any URLs so that they will go through the portal rather than directly to the back-end application or elsewhere.

2.1.2 Portlet

Portlet is a unitary component that is pluggable into user interface of a Web portal. A portlet is displayed into a portlet window which is displayed by a portal page. Each portlet is independent of the others like a web application is.

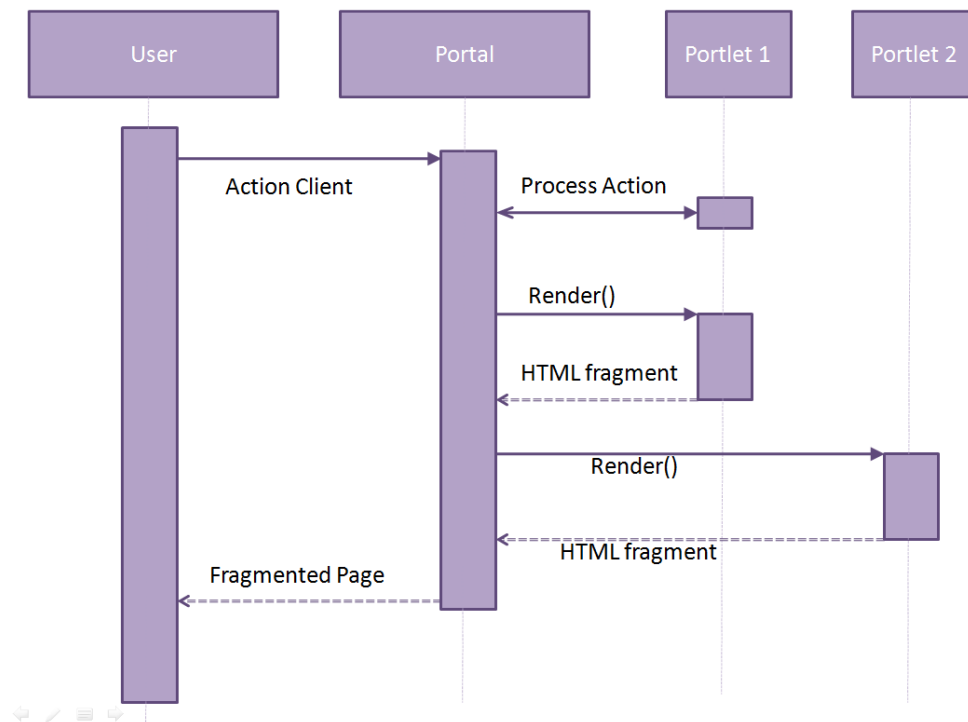


Portlet integration into a portal

Common points and differences with a servlet:

- Both are Java based.
- Both evolve in a container.
- Both manage their own content.
- Both have a life cycle controlled by the container.
- Semantics are the same.
- Servlets generate a whole page whereas a portlet only generates a page fragment.
- A portlet can't be directly called through a URL.
- Communication between client and portlet is totally managed by the portal.
- Different instances of a portlet can be included into the same page.
- Portlet can generate HTML code without including the base, body, html tags.

Here is a sequence diagram presenting the way elements communicate:

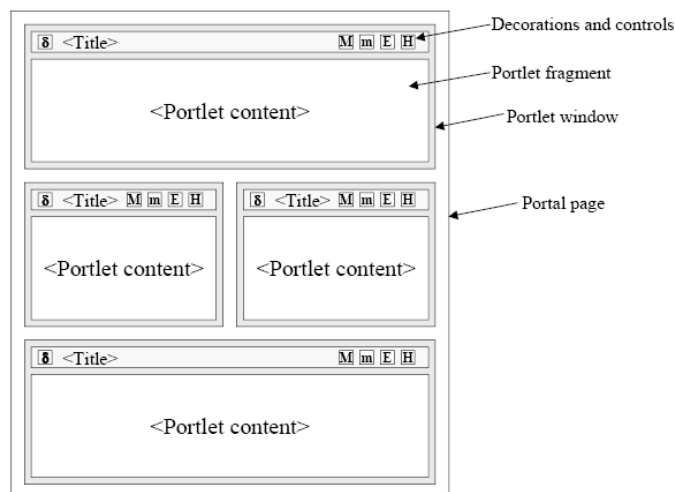


Life cycle request

2.1.3 Portlet specifications

The **Java Specification Request 168** (JSR 168) is the Java portlet specification that enables the interaction of portlets between different Web portals. To enable inter-operability into different portals, and different portlets, the Java Community Process specified how a portlet should behave according to its different aspects.

- aggregation
- personalization
- presentation
- security



Specified elements

The JSR 168 specifies the portlet API that all portals must follow. A compatibility kit of 400 tests enables to ensure the compatibility of a portal with this standard.

The Java Portlet Specification V1.0 introduces the basic portlet programming model with:

- two phases of action processing and rendering in order to support the Model-View-Controller pattern.
- portlet modes, enabling the portal to advise the portlet what task it should perform and what content it should generate
- window states, indicating the amount of portal page space that will be assigned to the content generated by the portlet
- portlet data model, allowing the portlet to store view information in the render parameters, session related information in the portlet session and per user persistent data in the portlet preferences

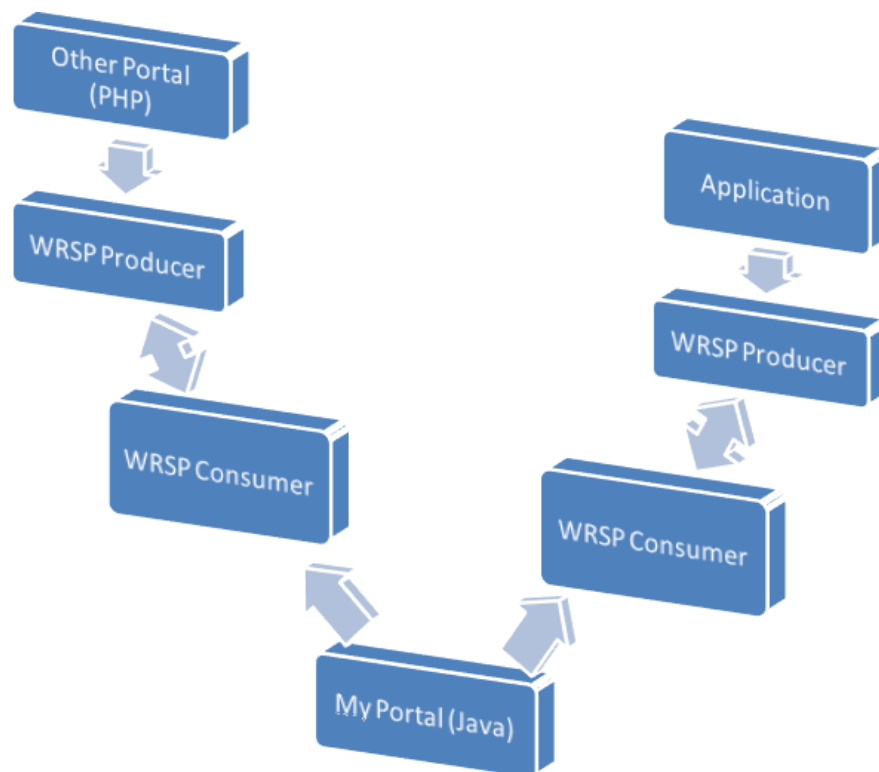
- a packaging format in order to group different portlets and other J2EE artifacts needed by these portlets into one portlet application which can be deployed on the portal server.

JSR-286 is the Java Portlet specification V2.0. It was developed to improve on the short-comings on version 1.0 of the specification. Some of its major features include:

- Inter-Portlet Communication through events and public render parameters
- Serving dynamically generated resources directly through portlets
- Serving AJAX or JSON data directly through portlets
- Introduction of portlet filters and listeners

The **Web Services for Remote Portlets** (WSRP) protocol provides web services standard to allow the 'plug-and-play' of portlets from separate sources or from different platforms. It enables a 'client' portal to interrogate portlets installed on a 'server' portal.

While the JSR 168 and 286 describes the mandatory behavior of a portlet in a Java environment, the WSRP describes the web services between two portals. The two main advantages of this system are the pooling and sharing of resources and the simplification of the communication between portals even based on different languages.



Relationship between applications

The current release, WSRP V1.0, provides a limited interoperability platform. Further versions of WSRP V1 were abandoned so that effort could be concentrated on WSRP V2. It improves the initial standard with cross-portlet coordination and access management features. This major update to the standard permit a more useful integration of multiple of content sources, regardless of whether they are local or remote, into a new web application. In addition, WSRP V2 supports some subsets of Web 2.0 technologies, such as AJAX and REST, without requiring them. WSRP V2 was approved by OASIS on April 1st, 2008.

2.2 Requirements

The two following section provides a list of the main requirements extracted from the client. The first section will provide the specification of the portal whereas the second section will focus more precisely on the functional requirement by describing the purpose of the different modules.

A the portal has a very wide range of application, the number of use cases is far too high to be described entirely in this document. As a result, no use case nor use case diagram will be provided in this section. Let just mention that a user can connect to the portal to access to one or more module. In the chapter 3, where I present more in depth my own contributions, I will introduce the use case of the concerned modules.

2.2.1 Specification

The specification of the portal, that is to say the main application, are listed bellow :

Usability

- The navigation in the portal should be intuitive and pleasing.
- The information in the portal should be as consistent as possible.
- The portal should respect the graphical chart of the region.
- The portal must be available in French (nothing in English).
- The user of the portal should not spend more than 10 minutes understanding the organization of the portal.
- The portal should be available under most common navigator (Microsoft Internet explorer and Firefox) in their two last versions.

Reliability

- The portal should be available at every moment.
- If any maintenance operation is necessary, it should be specify on the portal 2 hours in advance, specify during the maintenance operation and done in a time it will create less perturbations.
- In case the system seems to be corrupted, previous copy of the system of the system should be available.
- In case the security of the system is no more guarantee, the system should be shut down until any security expert come.

Performance

- The portal should have a time-to-respond lower than 30 seconds.
- The portal should be able to handle 25000 users without any modification of the response time.
- The portal should be able to handle 400000 users time-to-respond lower than a minute.

Supportability

- The portal should be portable, changeable, modular and extensible.
- The internal system should support remote access to perform maintenance.

As the operating part of the portal (that is to say the installation, the servicing, the update, ...) is at the responsibility of the region, we don't have to worry about the efficiency of the machine. We only have to provide application reliable and secure, efficient and user friendly. Due to the fact young person are involved a particular attention must be provided to the security aspect and the functionality must be very intuitive.

2.2.2 Available modules

Through the portal we implemented, many modules are accessible. First the functional modules will be described and then the technical modules will be introduced. They are integrated into the portal via portlet, servlet, iFrame or Web Proxy channel. All the windows containing these modules can be moved into another part of the same page of the portal.

The functional modules are the following one:

News reading Through this module, a user is able to keep itself informed of the news related to its institution.

News reading for the parents This module is similar to the previous one, except that the news are more specifically directed to the student family. These news will last longer than the simple news (like a whole year).



Directory A user can find there information about the others users of the portal. It is possible to filter the search of another user according to different criterion.

Pedagogical resources Through a Moodle, each teacher is able to publish documentation to the intention of its students.



Homework book This module takes up the prescribed principles of the paper version of the class homework book to create its digital equivalent. In a Homework Book, the teacher must detail for each sequence he does, all the session performed, the homeworks he gave, the attachment files he proposed and any information related to the session and the sequence : date, name, goal, ...



Brevet informatique A module to access to the "Brevet Informatique et Internet" (a part of the "Brevet" a French national diploma that students take at the end of secondary school, in year 10). The "Brevet Informatique et Internet" also called b2i is an attestation delivered to students to prove their ability to use both informatics tools and the Internet. It evaluates also their documentary and ethical competences. Gibii is the software evaluating the competences and delivering the certification.



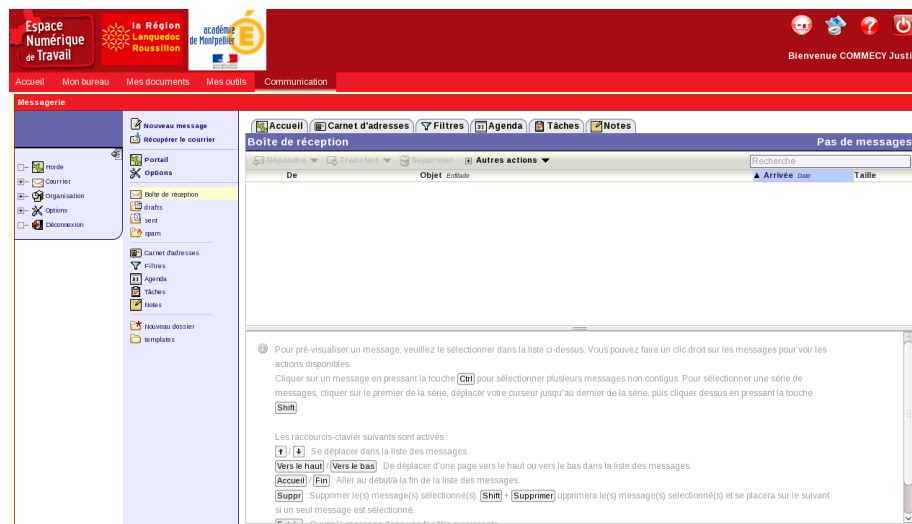
Online help Using this functionality, each user is able to get an online help through webpages and report any incident. Thus, this module will help to react very fast to any problems that should occurs.

The technical modules are the following one:

News publication This module give to some low level administrators the possibility to publish news for the News Reading module. The user able to publish news must be specifically designated by an administrator through an administration interface.

News publication for the parents This module provide same functionality than the previous one, except it is designed for the News reading for the parents module

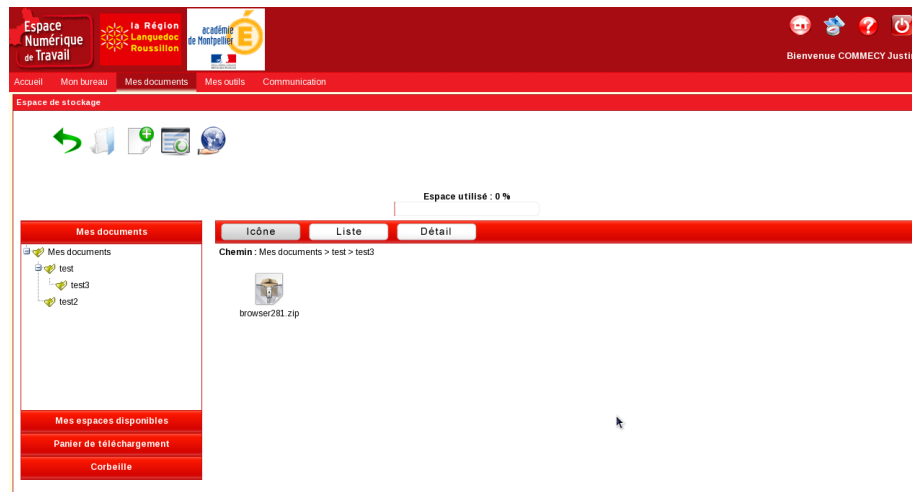
Web-mail An integrated Webmail will let the user received and send mails (only portal internal mails are allowed). The solution chosen for this Webmail is Horde a PHP-based application which provide webmail functionality.



Contact manager A contact manager is integrated with the Horde project: Turba.

Calendar manager Kronloth is the calendar manger integrated within the Horde project. The calendar can be shared among user or keep private.

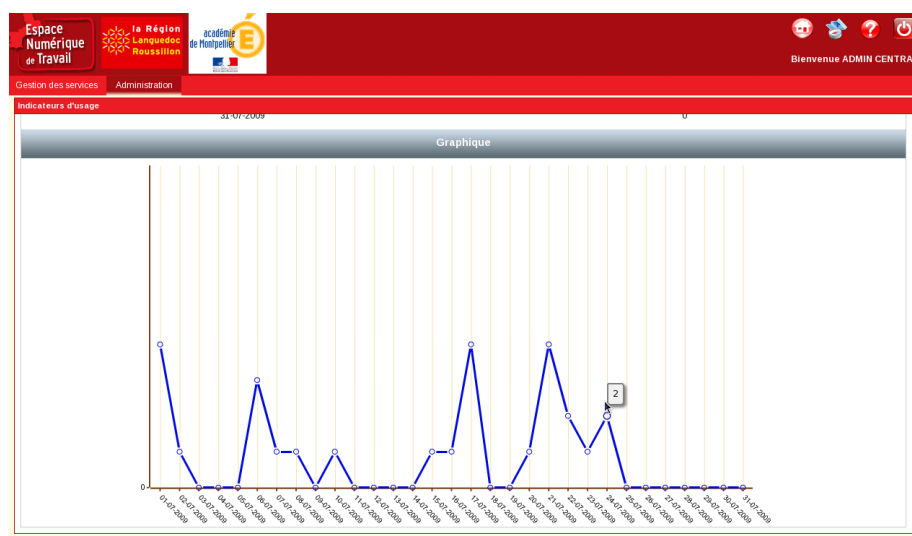
Storage Each user can access a personal storage area where he can create a tree directory and keep files. Sharing among users and web 2.0 functionality are provided.



Mark's management A module to manage marks of the students. The management of the marks is internal to each institution through the "sconet-notes" application, and thus can not be shared for all the portal. So an administrator of each institution must connect to the portal and setup the correct link to the specific application.

Absence's management A module to manage absences of the students. It is based on the same principle than the previous module. It is using the "sconet-absence" application.

Usage indicator This module let the user monitoring the use of the portal through different statistics: number of user connected, average connection time,... These figures can be sorted by day, month, institution, type of users,...



All the modules are accessible through different tabs:

- "Accueil" - Homepage
- "Mon bureau" - My desktop
- "Mes documents" - My documents
- "Mes outils" - My tools
- "Communication" - Communications
- "Gestion des services" - Services management
- "Administration" - Administration
- "Outils de test" - Tools test (will not be described - these tools were provided by the ESUP-Portal to ensure everything is running smoothly)

Another particular module is not integrated into the portal :

- *Inscription* This module allows an administrator to register a new person or a new institution in the system. It allows also the modification or the deletion of a person or an institution. Since the creation/modification/suppression are provided through a set of XML files, this module is a batch process that must be launched as often as necessary.

2.2.3 Appearance of the portal

In accordance with the different profile, a different interface is proposed. As a result, we get eight different interfaces, one for each of the following profile:

- Student
- Teacher
- Student family
- Headmaster or assistant headmaster
- School inspector
- Local administrator
- Worker in an academic institution
- Person from collectivity
- Person form academic services
- Portal administrator

A person can have different users profile in the mean time. For example, a teacher can also be a local administrator. For each profile, the available tabs and modules are outlined in appendix B.

2.3 Technical points

2.3.1 Portal comparison

Many different portals exist. Among the most well-known, we can quote:

- eXo, JetSpeed by Apache and uPortal which are Java open-source Portals,
- DotNetNuke and Rainbow which are Microsoft Technologies open Source Portal,
- and many others such as PHPNuke, Microsoft Share Point, Oracle Portal, Peoplesoft Portal, Microsoft Technologies...

We will study and compare more in detail a selection of Java open-source (which is a request from the client) : uPortal, eXo, JetSpeed and JBoss.

uPortal

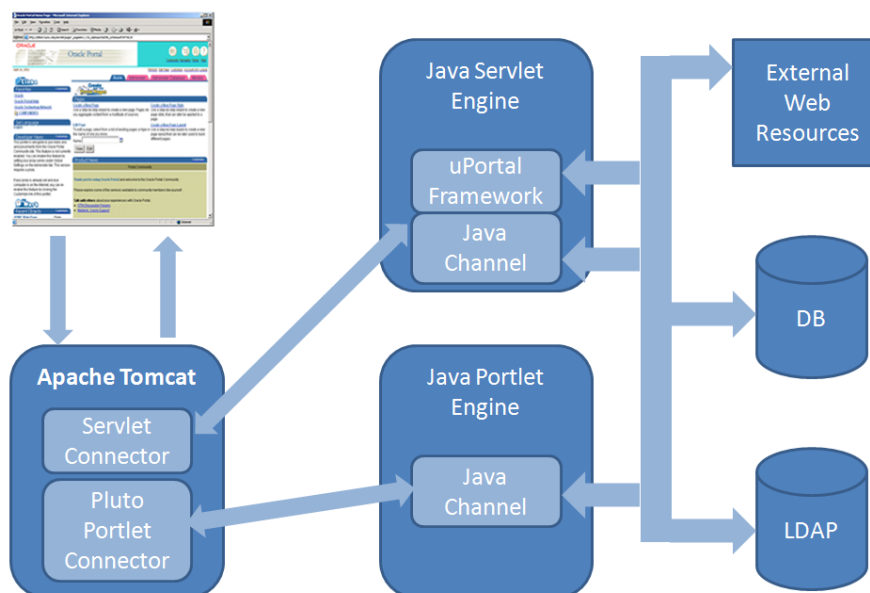
JA-SIG (Java Architecture - Special Interest Group) is an organization devoted to promoting the use of Java in US higher education. uPortal has been developed under JA-SIG by institutions including Princeton, Yale, and University of Delaware.

uPortal is currently used by many American universities as a portal for the students. The main difference between uPortal and other portals is its focus on the requirements of higher education institution.

The French ESUP community has developed its own version named ESUP-Portal (uPortal being its core element with a top layer: ESUP)

The high number of references listed on the uPortal site attests of its stability and its durability.

To integrate different applications, Yale University developed the Central Authentication Service (CAS), a SSO software which tends to become one of the standards for the integration of applications in portals.



Technology used :

Java, XML, JSP, J2EE.

Strong points

- UPortal allows to manage a community of user
- Lots of documentation of every level
- WSRP and JSR 168 compliant
- Very advanced personalization for the end user
- Openness to any application server

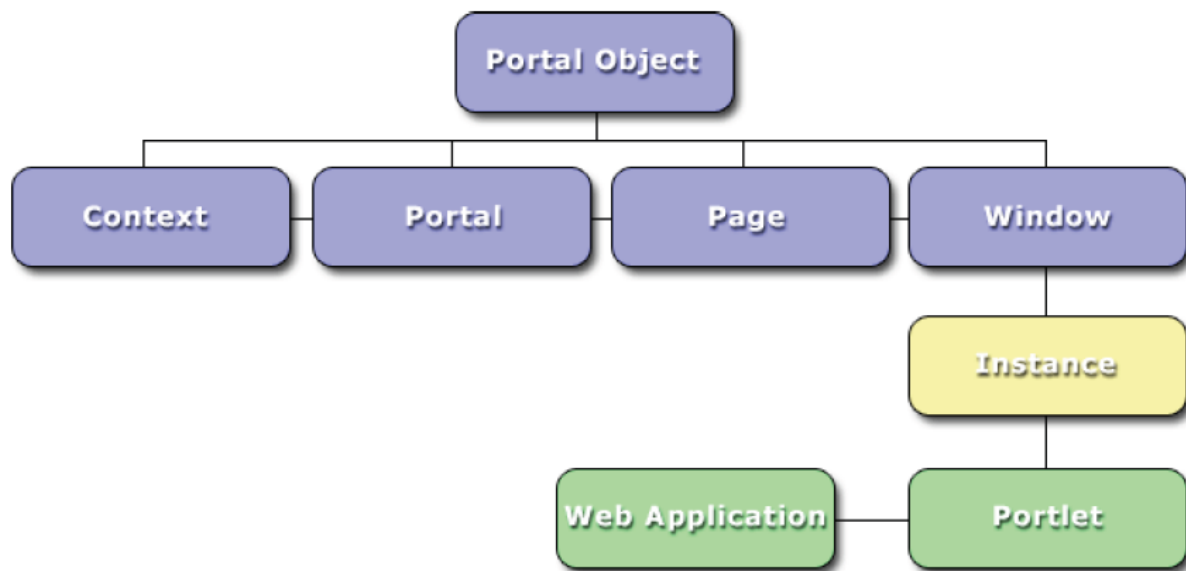
Weak points

- The main target of this portal is University. Its use in another context may lead to development of too many functionalities.
- Time **to take charge of the framework** may be high and requires highly skilled developers.
- Lack of advanced functionalities such as variation language, search function, customizable URLs and mobile device access.

JBoss Portal

JBoss Portal is an open source portal framework licensed under the Lesser General Public License (LGPL). It is 100% pure Java-based and so is platform independent. It simplifies access to application and information by providing a single source of interaction with web-based information.

It supports multiple portal instances running within one container. A portal instance can be viewed as a 'bucket', containing portal objects such as, Portal Pages, Themes, Layouts, and Portlet Windows. It also manages the relationship between the user and the portal itself. The hierarchy of portal objects is depicted as follows:



A JBoss Portal can only be deployed on a JBoss enterprise Application Platform (a J2EE-certified application platform). As a result the JBoss Portal only focuses on integration, aggregation and personalization problems without having to take care of the infrastructure.

It is compliant with many standards : JSR168, JAAS, WSRP, JSF, LDAP

Technology

Java, JBoss, Hibernate, J2EE

Strong points

- High flexibility at the API level and customization of the portal
- A scalability and performance level adapted to enterprise portal (JBoss infrastructure)
- High functional unitary test coverage

Weak points

- The user interface is not very attractive
- Documentation is incomplete
- Lack of project spaces
- Runs only on JBoss Application Server
- Not so many out-of-the-box portlets

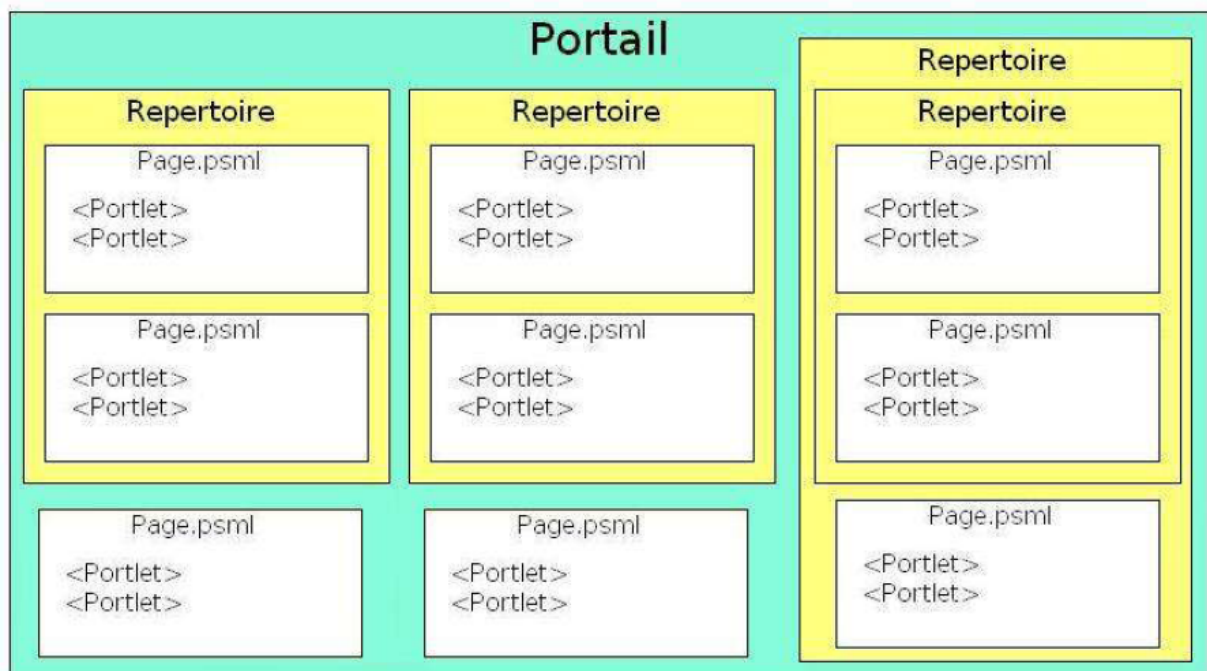
Apache Jetspeed

It is part of the Apache project. It is one of the first portals, developed in 1999, using Java and XML, based on the API portlet 1.0. It has been for a long time the J2EE portal reference because of its compliance with the JSR 168.

It is entirely based on an XML configuration and does not require a database (though its performances are enhanced using a database).

All accesses to the portal are managed through a robust portal security policy.

The **schema** bellow describes the Jetspeed organization. The portal contains pages and directories, the pages are used to organize the portlets which allows managing rights into three different levels: at the portal level, at the directory level and at the page level.



Technology

JAVA, XML

Strong points

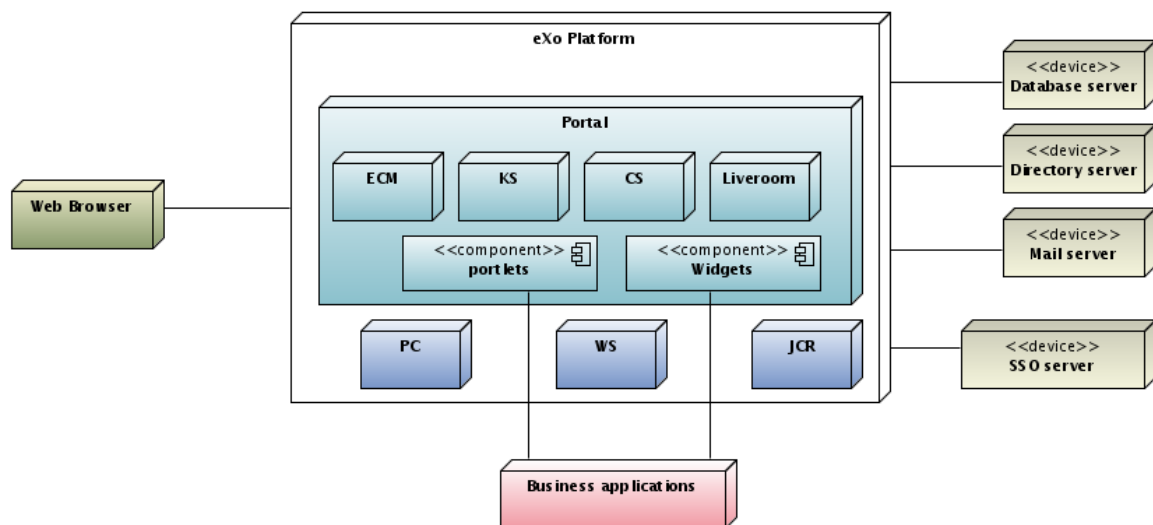
- Compliant with the JSR 168
- Personalization
- Native use of XML standard
- Many extension possibilities
- Solid architecture (multi thread)
- Openness to any application server

Weak points

- Difficult to **take in charge**
- Lack of documentation
- Some functionalities are totally outdated (especially **authentication** methods)
- Lack of project space
- Need highly skilled developers

Exo

ExoPlatform is a J2EE portal, the first one to be certified as JSR 168, using the WSRP protocol. It is based on Java Server Face (JSF) in order to manage action mapping of all portlets in a single XML file. It is marked out by the flexibility of its graphical restitution, based on dynamic layout which enables integration of personalized modules in any part of the page.



PC : the portal applications runtime container

JCR : as the main content storage

WS : as the web services integration stack

Technology

JAVA, XML, JSF, J2EE

Strong points

- Recognized for its functionality and the compliance with the standards
- Active community
- An eclipse plugin is available.
- Based on JSF
- Openness to any application server

Weak points

- the motor is not very performant
- Access to documentation is difficult
- Only most common functions are provided and only a few portlets are available
- Lack of project Space
- Need highly skilled developers

Since the June, the 10th of this year (2009), eXo and JBoss had annouced their partnership. The eXo portal code will be integraetd into JBoss portal to create JBoss eXo Portal. As a result, we can expect a portal with the performance of the JBoss motor and the rich functionality of eXo.

Comparison

All of the portals presented are JSR168 and WSRP compliant. They are all based on a JAVA technology. They take profit of the inherent vitality in open-source project. However it also implies higher development cost due to lower level of out-of-the-box functionalities.

About functional features, it may be noted that none of them supports search capability, customizable URL and only JetSpeed provides mobile device access.

They all provide user friendly interface but it may be due to the 'not so many' function provided (compared to closed source portal).

On the next page a table allows comparison of the 4 portals presented previously:

Portal	uPortal	Jetspeed	JBoss	eXo
Standard and architecture	Few databases natively supported (Oracle, MySql and SQL Server)	Supports the Java Portlet standard 2.0 (it is a reference since Apache is involved in its implementation)	Only JBoss Enterprise Application Platform supported	Works only with well known Browser
Functional Features	Very poor Available in English	Multi language content managed Statistic logging engine	Platform independent Multi language document managed Site analysis available	Only Available in English Site analysis available
Community and collaboration tools	No forum available Project space available	File sharing available No email nor forum	Real time communication Wiki Google gadget available	Real time communication Project space Wiki available
Look and Feel	Attractive interface User gets the full control of it	WYSIWYG portlet editor	Simple and attractive	Complicated and attractive
Security	CAS solution developed for uPortal Role access managed	SSO available Role access	SSO available	No SSO solution
Support and Cost	BSD license	Apache license	LGPL license, provides also on line demo	GPL license

There seems to be many clear-cut benefits of using portals rather than hand-rolling a site from scratch:

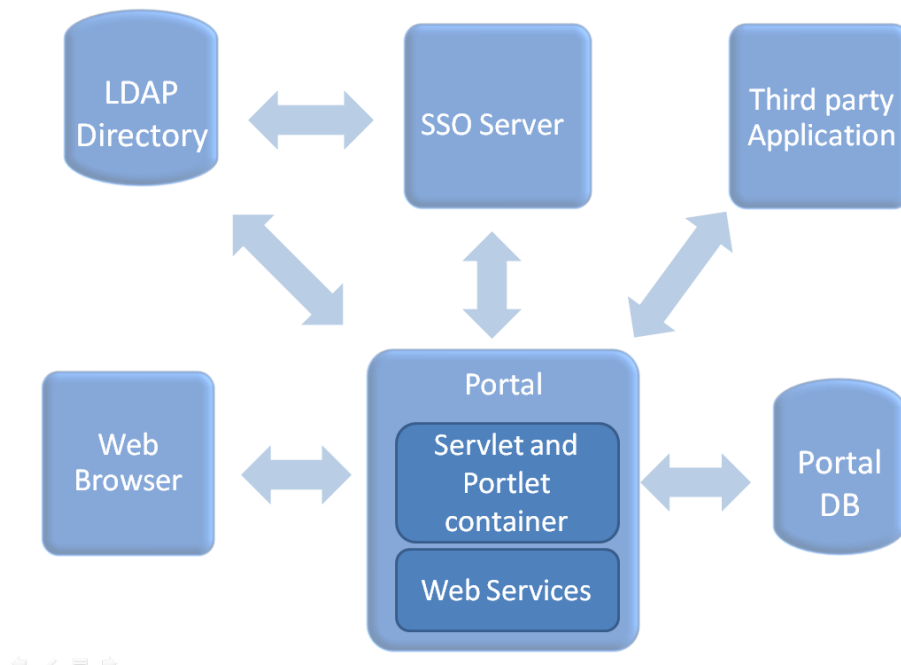
- User, group, and role management
- Authentication and authorization
- Content management
- Personalization
- ...

A portal can not be considered as the best one. The chosen portal should be in line with the targeted use of the portal.

In our case, we chose to take profit of the educational functions provided by uPortal and more specifically by the ESUP project, which is the French adaptation of uPortal. It is currently used in many French universities (it is the only digital workspace certificated by the ministry of **the national education**). Although it cannot be said that uPortal is one of the best portal, it achieves in all category, except functional features, good results.

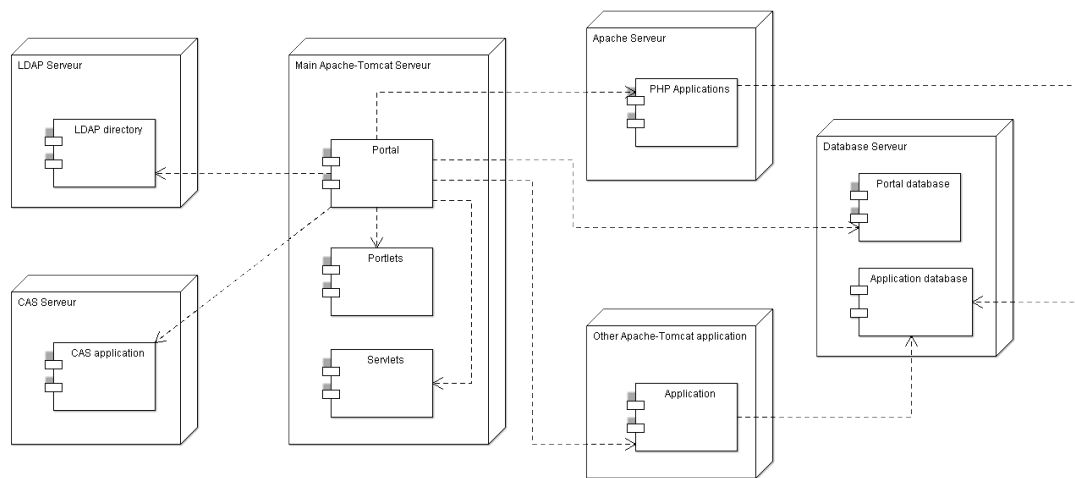
2.3.2 Architecture and organization of the portal

Here is a schema of our portal's architecture :



The portal is composed of:

- The Esup Portal at the core of the system (including the uPortal at its own core)
- A database for the portal - only SGBD accepting standard SQL, transactions and proposing JDBC native drivers can be plugged (PostgreSQL in our case)
- An LDAP directory to store user's related information
- A CAS for the authentication (SSO solution)
- A set of portlet and servlet integrated into the portal
- A set of third party applications (mainly PHP applications) integrated into the portal through Inline Frames or Web-proxy



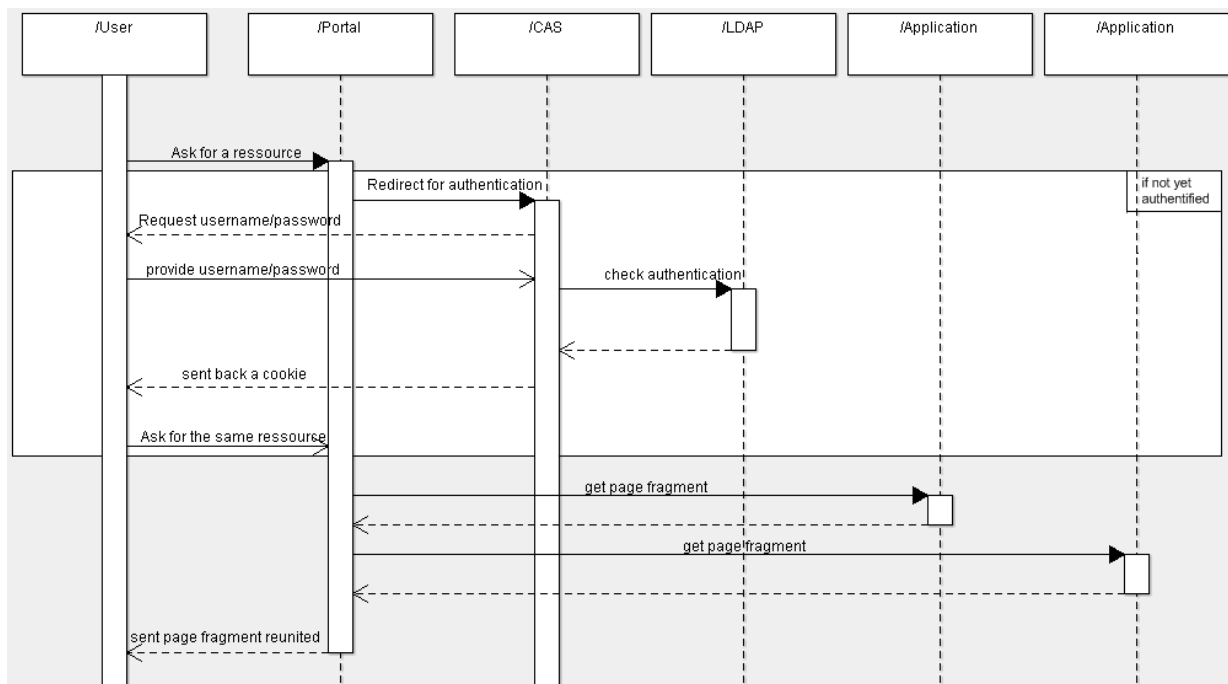
Deployment diagram

Requisite:

- JDK 1.5 for compilation purposes
- Ant 1.6.5 to automate the build process
- Tomcat as a web container to run the portlets, the servlets and the portal

Life cycle of a request:

The user, through its Web browser, accesses the Portal. The portal redirects the request to the SSO server. It ensures authentication using data from the LDAP directory. The SSO server redirects the user to the portal as an authenticated user. Then the user can access any service provided by the portal (in the limitation of his profile): servlet, portlet, web service, or third party applications. The portal may use its internal database and/or the LDAP directory to keep/retrieve information.



Sequence diagram of a request to the portal

Organization of the different packages:

```
esup-2.6-esup-2.X.Y
| Portal
|   | uPortal
| update
|   | uPortal
| custom
|   | uPortal
| logs
|   | [...]
| packages
|   | [...]
| resources
|   | [...]
| build.properties
| config.properties
| build.xml
| ant.sh
| start.sh
| stop.sh
```

- portal : contains the uPortal web application.
- update : contains the files that ESUP-Portal uses to personalize the uPortal distribution - this folder must not be modified
- custom : folder used to personalize the portal: all configuration files, source files,... that must be adapted from the uPortal project. Theses files will replace the files from the uPortal and update folders. As a consequence, the custom folder must respect the same organizations than the uPortal and update folders.
- packages: contains compressed distribution of third part products (Ant, Tomcat, ...)
- resources: contains files required by the different packages (drivers, scripts, ...)
- start.sh & stop.sh : file to start and stop the server
- build.xml : file to define the ant tasks
- build.properties and config.properties : files to configure the portal

The uPortal is organized as follow:

uPortal

```
| build
| build.xml
| build.properties
| lib
| properties
| source
| webpages
```

- build : contains the build files.
- build.xml : file to define the ant tasks such as init, deploy, ...
- build.properties : files to configure the portal
- lib : contains the lib required by the uPortal application
- properties : contains the properties files
- source: contains the java sources of the uPortal application (controllers, ...)
- webpages : contains the web part of the application : CSS, jsp files, ...

The LDAP directory

Each end-user of the portal must be registered somewhere in order to:

- ensure authentication
- keep descriptive attributes (such as gender, date of birth, ...)
- keep functional attributes (such as the institution of a person, ...)

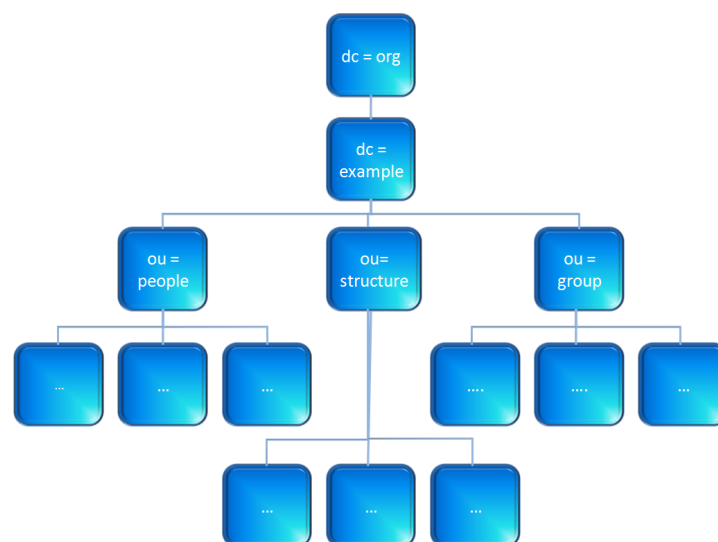
We chose to use a LDAP directory to do so, as recommended by the MENESR.

LDAP (Lightweight Directory Access Protocol) is a software protocol for enabling anyone to locate organizations, individuals, and other resources in a network. LDAP is a 'lightweight' (smaller amount of code) version of Directory Access Protocol.

An LDAP directory is organized in a simple 'tree' hierarchy consisting of the following levels:

- The root directory (the starting place or the source of the tree)
- Countries
- Organizations
- Organizational units (divisions, departments, and so forth)
- Individuals

In our case the tree is represented this way:



(where **dc** means domain component and **ou** means organizational units, although it is only a convention based on the DNS service).

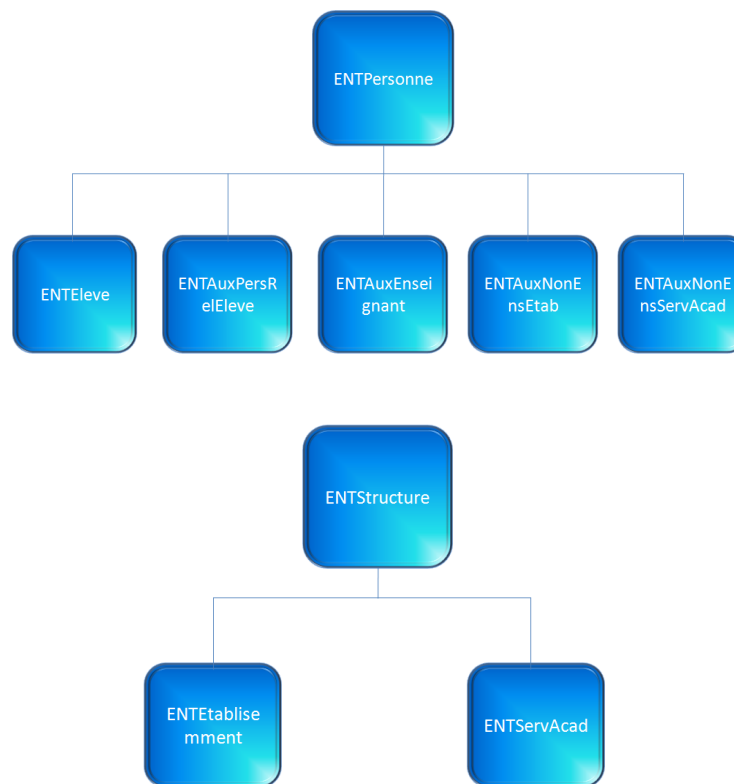
The different branches are:

- **Structure** : contains all the institutions involved: school, company related to schools, local educational authority,...
- **People** : contains all the people involved in the educational world: student, teacher, student family,... Being a member of this branch is a prerequisite to connect to the portal.
- **Groups** : contains groups of people: it can be a class, a sub-group of class, a group of teachers teaching the same class,...

For a person, a special attribute, the `objectClass`, allows to know which kind of object it is. Here is a list of all the `objectClass` we are using:

- `ENTPersonne` : People have this attribute, they all have another attribute to determine in which category they belong to.
- `ENTEleve` : Students have this attribute.
- `ENTAuxPersRelEleve` : Persons related to students (family,...) have this attribute.
- `ENTAuxEnseignant` : Teachers have this attribute.
- `ENTAuxNonEnsEtab` : Staff from an academic institution have this attribute.
- `ENTAuxNonEnsServAcac`: Staff from local education authority have this attribute.
- `ENTStructure` : Structures have this attribute, they all have another attribute to determine in which category they belong to.
- `ENTEtablissement` : Academic institutions have this attribute.
- `ENTServAcad` : Structures from the local education authority have this attribute.
- `ENTEntreprise` : Companies have this attribute.

As a result we get the following architecture:



It is possible for a person to be part of different categories. For example a teacher can be a parent. However we are not currently able to deal with this multi-profile in the portal.

Although it might seem similar, an LDAP directory is not a database. Among the main differences, let's quote:

- A database is more performant while writing than an LDAP directory and the contrary is true about reading.
- An LDAP directory does not manage constraint (only basic one such as checking if an attribute exist) nor foreign key.
- An LDAP directory does not enable sorting and join.
- An LDAP directory is set over a generic schema, whereas a database can be modified as desired.
- The use of an LDAP directory is generally restricted to user management.

The CAS - a SSO solution

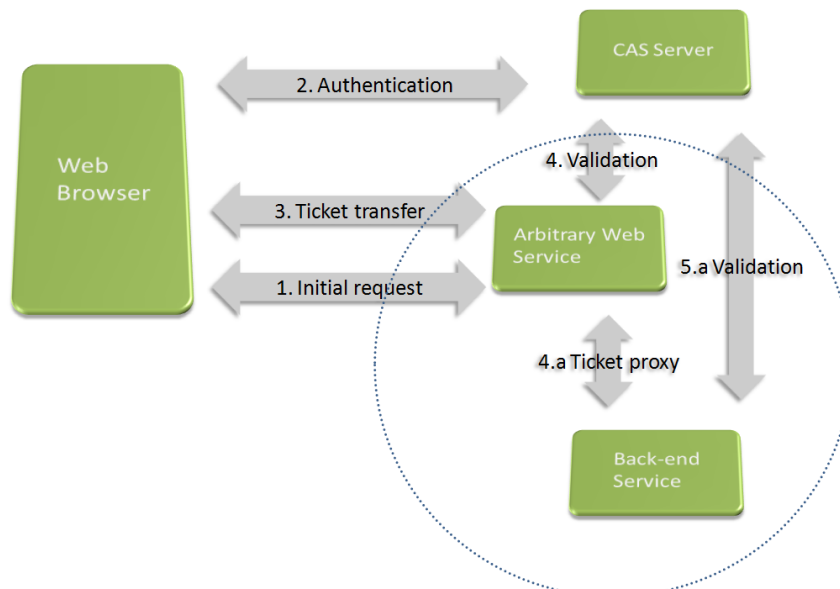
Single Sign On (SSO) is a session/user authentication process that allows a user to provide his or her credentials once and for all in order to access multiple applications.

Web Single sign on works strictly with applications accessed through a web browser. The request to access a web resource is intercepted either by a component in the web server, or by the application itself. Unauthenticated users are forwarded to an authentication service and sent back to the original address only after a successful authentication.

The Central Authentication Service project, more commonly referred to as CAS, is an SSO authentication system originally created by Yale University to provide a trusted way for an application to authenticate a user. CAS provides enterprise single sign on service:

- An open and well-documented protocol
- An open-source Java server component
- A library of clients for Java, .Net, PHP, Perl, Apache, uPortal, and others
- Community documentation and implementation support
- An extensive community of adopters

The CAS solution has been especially developed for the uPortal solution and so is perfectly well integrated with it.



Authentication process

The appendix D present more in depth how the authentication process works.

3 Data integration

Most of my task were related with the data integration process.

Among them, I will present the following one:

- How to insert and modify data in the LDAP directory
- How to provide data to other application
- How to synchronize the LDAP directory and others data storages
- How to deal with the amount of data

3.1 Amount of data

The region Languedoc-Rousillon is composed of five department : the Aude, the Gard, the Hérault, the Lozère and the Pyrenées Orientale. Each department is in charge of the middle school, whereas the region is in charge of the secondary school.

Below some figures summarize the number of institutions and persons involved.

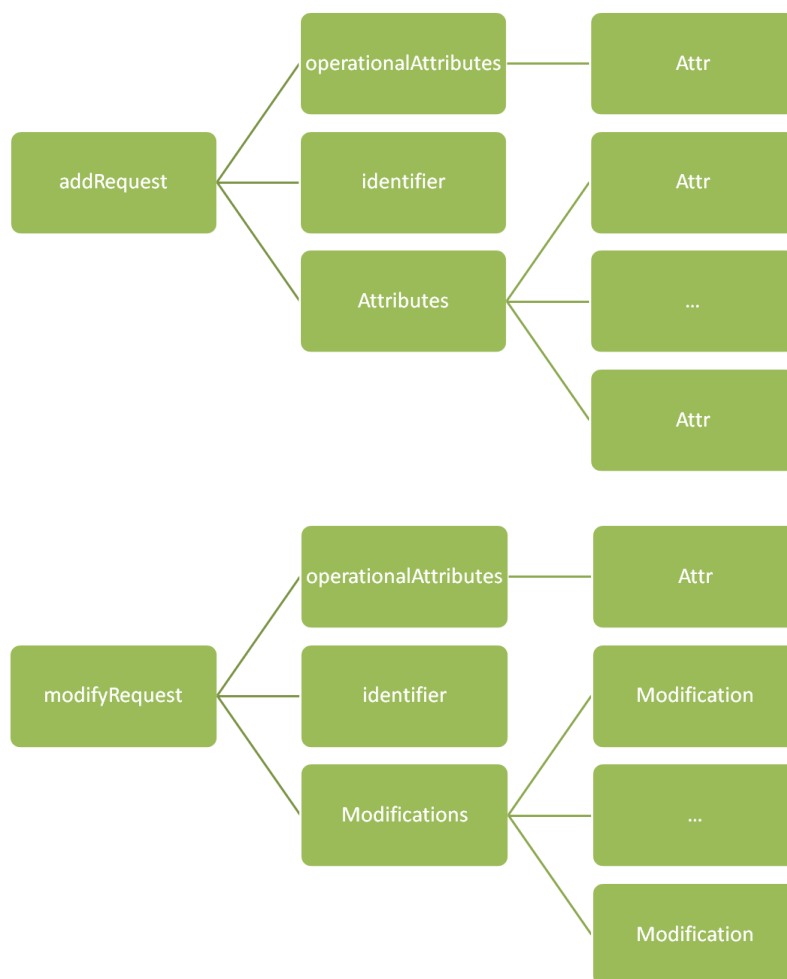
	Aude	Gard	Hérault	Lozère	Pyrenées Orientale	Region
Number of institution						
Middle school	27	53	78	13	30	201
secondary school	11	21	30	4	11	77
Total	38	74	108	17	41	278
Number of students						
Middle school	15809	34798	49940	21202	125425	
Secondary school	10050	23012	32000	2893	13405	81360
Total	25589	57810	81940	6569	34607	206785
Educational staff						
Teacher	2378	5325	7561	815	3223	19302
Other staff	136	279	466	46	180	1107
Family (student * 1,25)	32323	72262	102425	8211	42258	258481
Total	34837	77866	110452	9072	46661	278890
All persons						
Total	61202	135912	193198	15641	81578	487533

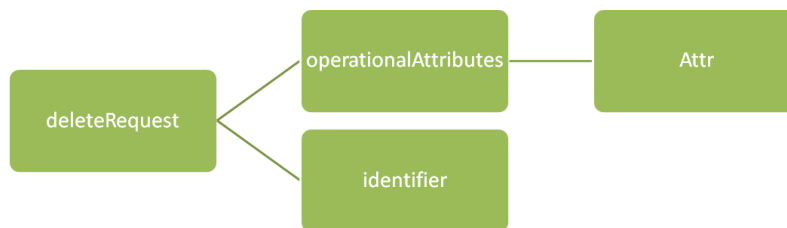
As seen, the involved person is high. As a consequence, I had to take a careful look to the performances.

3.2 Source of the data

The data are coming mainly from the Educational Offices. This organization can provide us with XML extracts from their own data that contains every necessary information from academical institutions, students, family and teachers to fill the LDAP directory. They can also provide 'delta' XML extracts that informs us of modification and suppression in the data. Other information sources are currently being integrated to insert people and structure not existing in the Educational Offices records.

For every source considered, the record must be in the XML type and follow the references DTD (available in the appendix E):





The main tag (addRequest, modifyRequest, deleteRequest) allow to differentiate through the three possibilities: add, modify or delete.

The operational attributes allow us to distinguish between different types of structures and persons.

The identifier tag allow us recognized an entity. It is a unique number equivalent to a primary key. One problem is that data are coming from different sources and thus the number might not be unique. Anyway it is a strong constraint for us: this number is considered as unique and it is the role of the client to ensure this.

The attributes and modifications tag allow us to insert and modify the different type of attributes.

In the appendix E, an example of the XML extracts can be found as well as the DTD of the XML files.

3.3 Data transfer to the LDAP directory

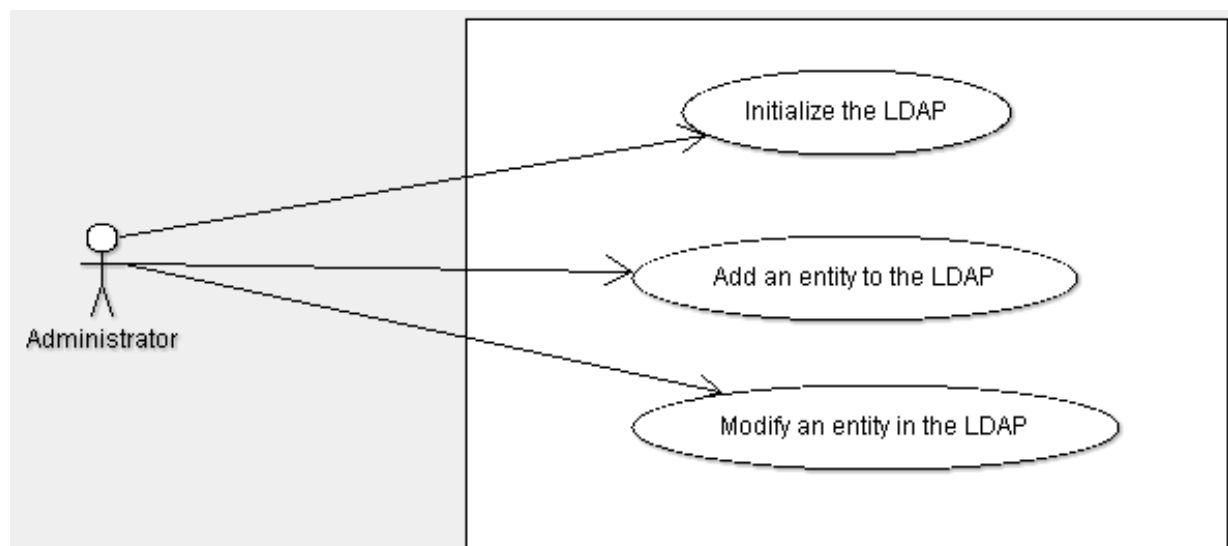
3.3.1 Functionalities

As said before, our LDAP directory must contains all necessary information to identify structures and people (login, password, address, qualifications, ...). This LDAP directory must be populated and modify as soon as new data are available.

As the amount of data is quite important, an application is needed to transfer the XML data to our LDAP directory.

Another third party functionality can also be provided such as initialization of the LDAP directory and generation of default data to help the portal to work.

This application will be used at the beginning of each scholar period to create or update the users (a student should keep data from a year to the next one) and during the school year to update the data.



use case

Use case 1

Actor : administrator *Requisite* : none

Goal : Initialize the LDAP directory

Actions :

- 1 : The administrator requests an initialization of the LDAP directory
- 2 : The system empties the LDAP directory
- 3 : The system initialize the LDAP directory

Use case 2

Actor : administrator *Requisite* : initialized LDAP directory

Goal : Add new entity in the LDAP directory

Actions :

- 1 : The administrator provide the XML data to the system
- 2 : The system transfer correct data to the LDAP directory
- 3 : The system show a summary of how many entity were inserted

Use case 3

Actor : administrator *Requisite* : initialized LDAP directory

Goal : Modify an entity in the LDAP directory

Actions :

- 1 : The administrator provide the XML data to the system
- 2 : The system transfer correct data to the LDAP directory
- 3 : The system show a summary of how many entity were modified

The main requirement for this application is the robustness. It aims to build a huge database so if an error happen in the process, it must not be ignored. Moreover, the administrator must be able to ensure everything went well so it is requested that each action are logged into different files.

3.3.2 The problematics and analysis

It seems obvious that we will first need a parser to read the information in the XML schema. Then the data will be treated to be set in forms so that it fits with the LDAP directory format.

However we have to deal with three following pitfall:

- quantity of data
- consistency of the data
- error management
- destruction of entity

Quantity of data :

As said before, the amount of data is quite important. It implies, we need an effective application and deal carefully with request to interrogate the data.

Consistency of data :

The XML documents provide us the institution and the people involved. However, we can not assume that the XML files are perfect and a number of assumption must be checked. Among them, let's quote and analyses the following one:

- *A student must be part of at least one valid class.* Class are designed through the identification number of the institution, followed by the name of the classes separated by the '\$' character. To ensure validity of the student, we must check at least one class is valid (ie the institution exist and is valid). So, how to ensure the institution is valid? The solution is to insert all the institution before the students. As a result we decided that all the institution will be inserted first and then all people will be inserted.
- *A teacher mus not teach to an class.(a class without student)* To ensure this constraints, all the students must be inserted before the teacher, so that we can know if a class contains or not student. So the teacher must be inserted after the students.
- *A parent must be related to at least one valid student.* A parent not related to a valid student has no reason to get an access to the portal. So he must not be transfered to the LDAP directory. However, the relation between a student and a parent is written only in the student files. The parents file does not contains any information about the students he may be related with. As a result, we can not insert the student and the parent independently because we need to insert the link from a parent to its child in the LDAP directory. Moreover, if children are invalid, the parent must not be inserted neither. So, how to check consistency? A solution is to insert all the parents, then all the students and update the data in the parents field and finally at the end of the insertion, check that the parents have some child. It appears we need to be able to insert, retrieve, check data. However an LDAP directory does not have the same performances than a database and constraints can not be easily checked. As a result we choose to insert a database between the LDAP directory and the XML files. Data will be transfered into the database where consistency will be checked and then will be once again transfered to the LDAP directory. By the way, it will help us to deal with the amount of data by keeping a record of all the data transfered even if they were invalid.

So it appears we must insert data in the following order: institution, parents, students, any other profile. It is also stated we need to have a database that will check every possible constraint. We can notice that the choice of a relational database will help us to materialize relation between persons and institution far easier than the LDAP directory alone.

Error management :

Due to the high number of data, it is highly possible that errors occurs. How should we deal with errors? We have decided first that whatever the error is, the process must not stop. Even if something failed, it will be registered in one of the log files and all the other data will be processed. However, we must recognize that some errors are more important than others and require us to invalidate the entity (for example, someone without name can not be registered). In the database we keep track of all the insertion or transfered error and we classify them into two categories: minor or major. The minor errors are only

logged so that they can be fixed while the major errors prevent entity to be inserted into the LDAP directory.

Destruction of entity :

A request can come to make us destroy an entity. However, it is also possible that the next day, the same entity should be re-created. It would be convenient to be able to keep all the setting of this users so that he will be able to experience the same data than before. Using our database in the middle, we only need to de-activate the user into the LDAP directory and put a flag in the database to mark it as a "removed" entity. If the user is reactivated, we change back the flag to its original value and re-transfer the person to the LDAP. That way, he is the same 'person' than before (in the sense that he has the same identifier and thus can be recognize by third party application as the same person than before).

3.3.3 The Sarapis project

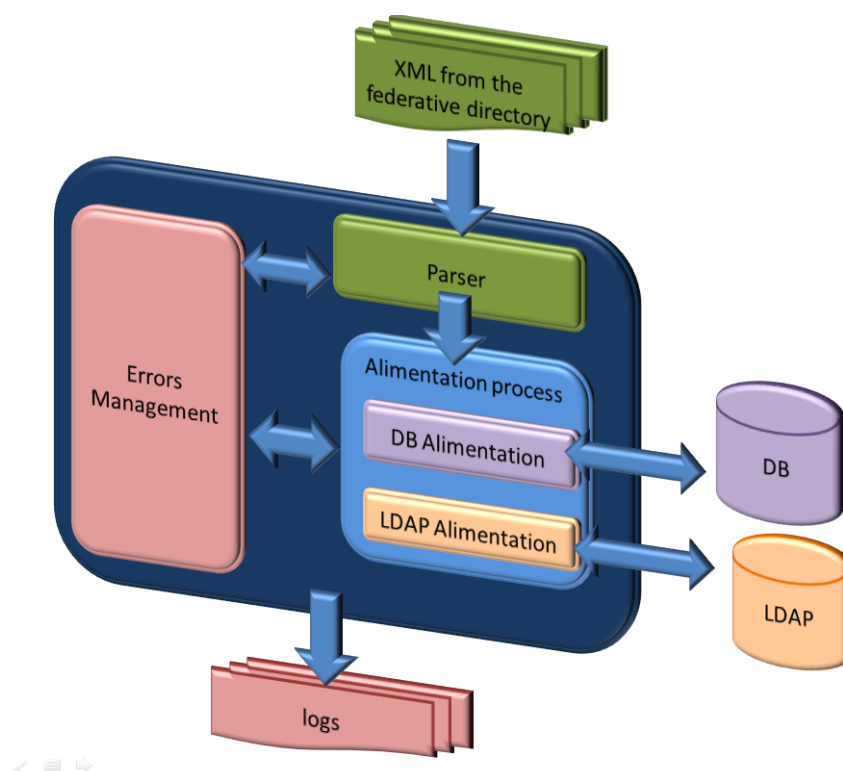
A project developed by the ESCO community is adapted to our needs. Its name is the Sarapis project. It is in charge of taking XML extracts provided by the Educational Office to transfer data into an LDAP directory. It has been first developed by RECIA and then passed to the ESCO community which kindly allows us to reuse in the context of this project.

ESCO is a community promoting the development of digital workspace for secondary school. It is a partnership of different regions, capitalizing ESUP work and articulates works on the ESUP base.

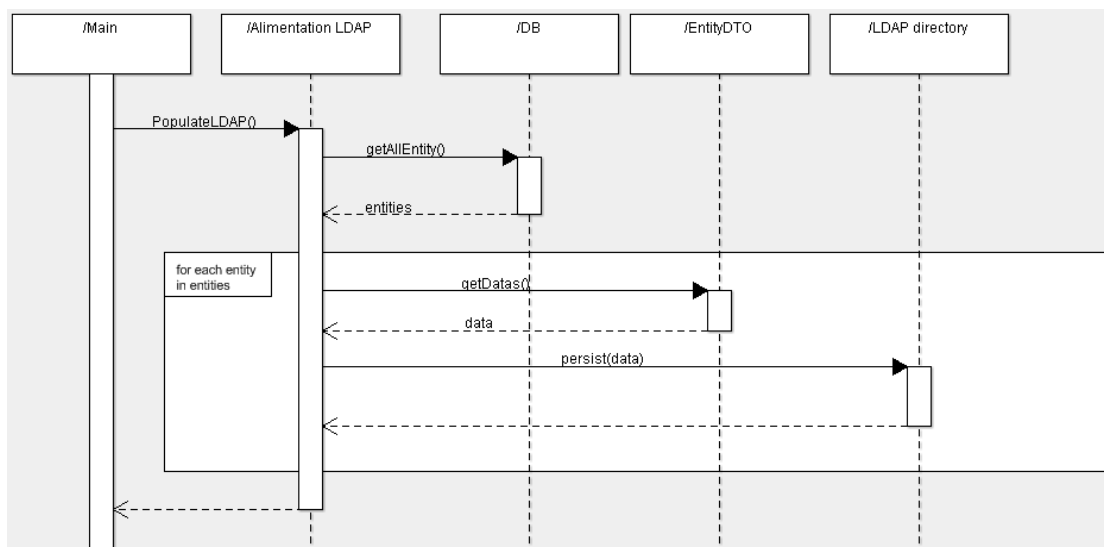
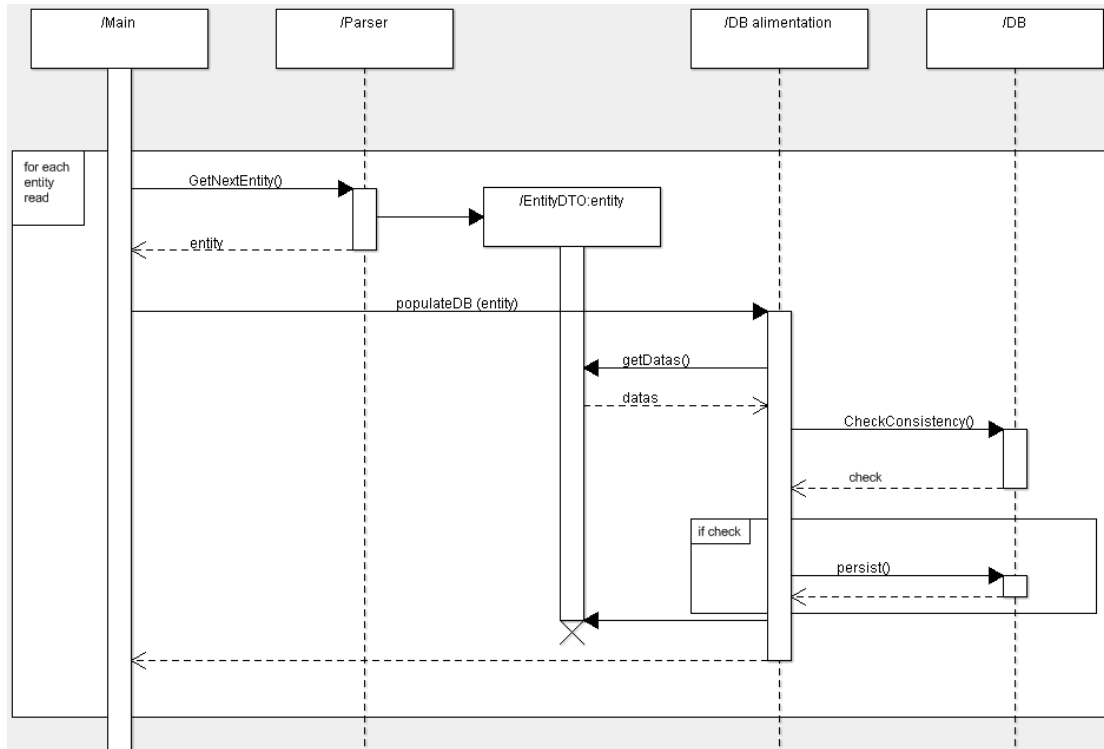
RECIA is a community which aims to promote the society of information in the region Centre, gives a technical, juridical and financial expertise, supervises the coordination of TIC projects and ensures the usage and development of services and applications related to the TIC.

Architecture

The Sarapis project works according to this schema:



The local educational authority provides the extracts from their own database which notifies us of the creation/modification/suppression of a person or a structure. Using the Sarapis project, these elements are transferred into a database. Consistency of all the information is checked and only then, data are inserted them into the LDAP directory.



Sequence diagram of the insertion process

It is developed in Java with the spring framework, using mainly hibernate to manage persistence with the database and spring LDAP to communicate with the LDAP.

The project is organize as follow:

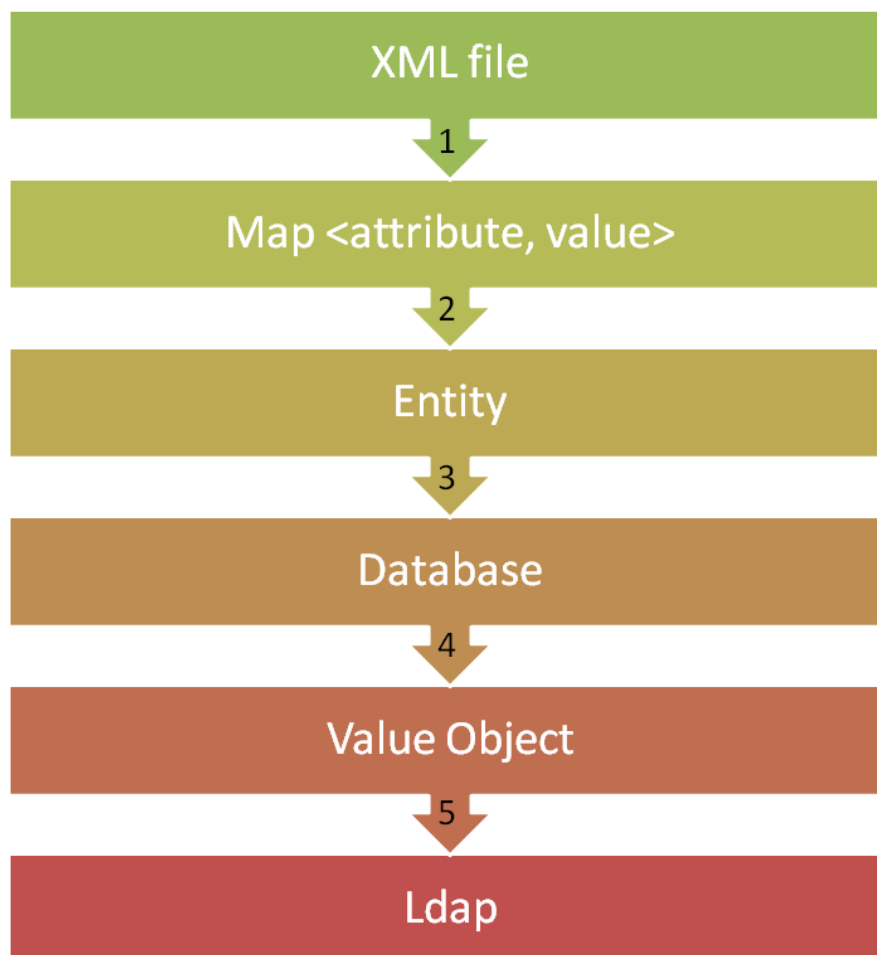
```
|Alim
| | XML files
| | DTD file
|
|Lib
| | hibernate
| | spring
| | ...
|
|Properties
| | hibernate.properties
| | ldap.properties
| | xml files to describe beans
|
|src
| | org
| | | esco
| | | | Sarapis
| | | | | Commons
| | | | | | ValueObjects
| | | | | | Exceptions
| | | | | | db
| | | | | | | TransactionManager.java
| | | | | | | DAO
| | | | | | | Domain
| | | | | | | | ToVO
| | | | | | | | ToEntity
| | | | | | | Entity
| | | | | | Error manager
| | | | | | AlimManager
| | | | | | LDAP
| | | | | | | Branches
| | | | | | | | People
| | | | | | | | Structure
| | | | | | | Domain
| | | | | | Parser
```

In the appendix C, there is a schema of the database (where the name and attributes may have been changed for security reason).

In order to insert/modify/delete data, different tasks are performed:

- Parsing the XML files
- Modification into the database
- Consistency checking
- Modification of structure into the LDAP
- Modification of people into the LDAP

In the appendix D1, there is a pseudo algorithm to explain how the process works. To summarize, the information associated to a person or a structure follows this process:



1. Parsing to transfer data into a map (Parsing folder)
2. Transformation of the map to a Java entity (toEntity folder)

3. Insertion of the entity into the database (persistence of the entity using hibernates)
4. Data retrieves from database and inserted into a Java Value Object (toVO folder)
5. Transfer of the data to the LDAP (LDAP folder) using Spring to manage connection

It may be noted that the process is mainly composed of loop. For the one loop retrieving elements from the database, it was not possible to get all the elements in one request, so a paginated iterator structure has been put in place.

Many problems may occur:

- Encoding of the file: the extracts from the local education authority must be in UTF-8. If it is not the case, any special character will lead the parsing operation to fail.
- Key duplication: Different elements are used as a key: in the local education authority database, the 'Join Key' is the primary key. In our database the 'SIREN' is used as a primary key and the UID is performing the same for people. When structures are inserted, it may happen the SIREN is not unique. This should not happen due to the SIREN nature. The UID of the persons are generated by the Sarapis project and can not be duplicated.
- Special characters: Some characters are not expected in the XML files such as \. They must be escaped or removed.
- Bad format: the XML files must follow the DTD provided and be consistent (the id attribute must be the same than the join key attribute for example).
- Elements requested not provided: according to the type of object some elements are requested (such as the name and the surname for a student). If these elements are not provided, this object can not be inserted.
- No reference found: people and structures are referring ones each others. It may happen that a reference to another entity is not found (because it has not been provided yet). The order the files are treated is designed to avoid as much as possible this problem: we first insert structures that normally are not depending on anything else than structures, then the parents that do not have references to anyone, then students that make references to their institution, parents and finally the teacher that references their institution. However it is possible an entity is referring to another one, not yet inserted, because of a lack of consistency of the data.
- Duplicated description: the XML files contain many codes to refer to the different subjects and career. It is usually followed by the description of the code. It may happen, due to a wrong insertion in the database, that the same code is associated to different descriptions (such as "History & Geography" and "History and geography" which will both have the same code).

- Modifications: while modifying or deleting data, inconsistency may appear from a previously correct entry. In that case the modification is ignored.

The first fourth errors are major errors. In case one of these happens, we consider that the data are too corrupted to continue the process and the application will stop without catching the error. The other errors are caught and do not cause the application to stop. They are lists in the logs files for review.

3.3.4 Sarapis Analysis

Sarapis is very well designed. However some problems prevented us for using directly:

- grouper modules
- profile not treated
- attributes not managed
- database design
- amount of data
- performances
- database transaction

The "Grouper" module is a part of the application with deal with propagation of groups into other application. We are not using the same kind of group than the one the "Grouper" modules were using and so we don't need this part.

We are currently dealing with eleven profiles of person whereas Sarapis was only dealing with 5 of them. It implies we add lines of code and modify the database schema.

Due to our portal, we need new attributes that did not exist previously. Theses attributes are not provided by the XML files but can be modified through an administration web page. So theses attributes must be generated with default values and then kept when doing a modification operation.

The schema of the database is very complex, with inheritance relation and a lot of abstract data. The names of the tables were not fitting with the names of the entities used in Java. As a result, it became very difficult to modify anything related to the database or simply to follow the process.

Due to the high number of data, it happened that some requests to the database returned so many results that the Java virtual machine could not handle them.

The main problem came with the performances. For the first version of this project we only had to deal with twelve institutions which represent 10,000 students. Now, let us examine the performances observed :

for one institution, ie 4225 persons we spent 1 hour

for 12 institutions, ie 25000 persons we spent 26 hour

In the latest version, we must be prepared to deal with all the institutions of the region, which means 177592 students that is to say 603812 person (according to ministry document, we have 1.4 other person for each students). An amelioration of the performances was necessary (it would cost 209 hours if we consider a linear augmentation - which is probably not the case)

It did not work properly with postgresql due to a different way to manage transaction. I had adapted the code to manage properly transaction.

3.3.5 Modification to the Sarapis project

The grouper module : I could not just 'cut' these parts since all the elements were related together and I could not just let it there for performance reasons. So I created "Mock" class to simulate the behavior of those parts.

Profile not treated and attributes not managed : I had to 'copy-paste' the code of other profile, adapted it and create the needed tables. I had to add the insertion of groups in the LDAP. This part was not done, so I had to entirely develop it, based on the model of the insertion of people and structures.

I had to finalize the development of the modification and suppression of data which was only partially done.

I had to add new constraints to ensure all the recommended consistency checking.

We decided to reforge the project to have a better visibility and improve performances.

The first act was to restructure the database model to fit better to our needs (the schema is available in the Annexe B).

As a result, we had to rewrite all the dependent code. Since the database is the middle part of our application, it meant rewriting quite everything (but having capitalized an experience with this project, it was easy). We decided to take this opportunity to change the way we managed many-to-many association: previously there were no middle tables and Hibernate was representing this by keeping sets in each tables. As a result, it went a lot easier to retrieve data using our solution.

Moreover, our schema does not require us anymore to do long and costly join between tables. All the needed data are kept in the same table or close to it.

To improve performances, I timed the different modules and discovered that in the case of 12 institutions, the process was spending 23 hours in the insertion onto the LDAP directory. After looking more precisely, I discovered that the acquitment process (ie the request that says to the database that the element has been transfered to the LDAP directory) was the time consuming process: a new connection was open to acquit each entity! It decided not to acquit each element after inserting it but do it when the paginated iterator was called. That way I was doing only one connection each 300 elements. We obtained this result:

for one institution, ie 4225 persons we spent 15 minutes
for 12 institutions, ie 25000 persons we spent 3 hours.

The improvement is enormous but it implies the introduction of coupling between classes, which did not exist before.

Moreover, from the point of view of the error management, in case a major error happen and cause the application to crash, the LDAP directory will not be in the state expected by the database: the daptabase will think some entity are not yet transmitted to the LDAP whereas they are already. In the next execution, the process will try to insert these entity again, but since theses entity already exist, the process will fail. I decided to keep the key of the element not already transmitted in the LDAP directory in a file. When starting the process again, first it will check if the file contains some key and transmit the validation to the database.

3.3.6 Test case

As explained before, it is crucial the application performs its task without any error. As a result the need for test was very important. Due to the number of cases tested, it is not possible to present them all. I will rather introduce the methodology used to ensure all needed test was done.

First of all, I created one instance of each different entity we have: a teacher, a student, a school,... Then I tested the modification by modifying each attribute. Finally I tested the deletion case. After each modification I had to ensure each relation between person was still correctly set. Thus I ensure the "normal" case was performing well.

In a second time, I had to check all the "planned errors", that is to say the case where an information is not consistent with the constraint we had. I tested again the 3 case of modification, suppression and deletion. But this time, since the same algorithm is used for each entity, I decided to test each constraint on one kind of entity.

Then, I tested the program with corrupted data: relation not well set between persons,

institution that does not exist, wrong type of data, ...

Finally, I used "real" extract provided by the region to ensure everything went well.

However, it must be said that the number of case is so huge, we can not be sure every case is tested. As a result, I only tried to follow the most probable case in the two first part and in the third part, I was creating test by following the flow of execution of the program and creating a test for each branch. One of the problem in the test case is that when a problem occurs because of a data it can be considered the problem of the program because it is not able to handle it correctly or it can be considered a problem of the data because data we have in entrance should be correct (since they are themselves coming from a program that ensure data are not too corrupted).

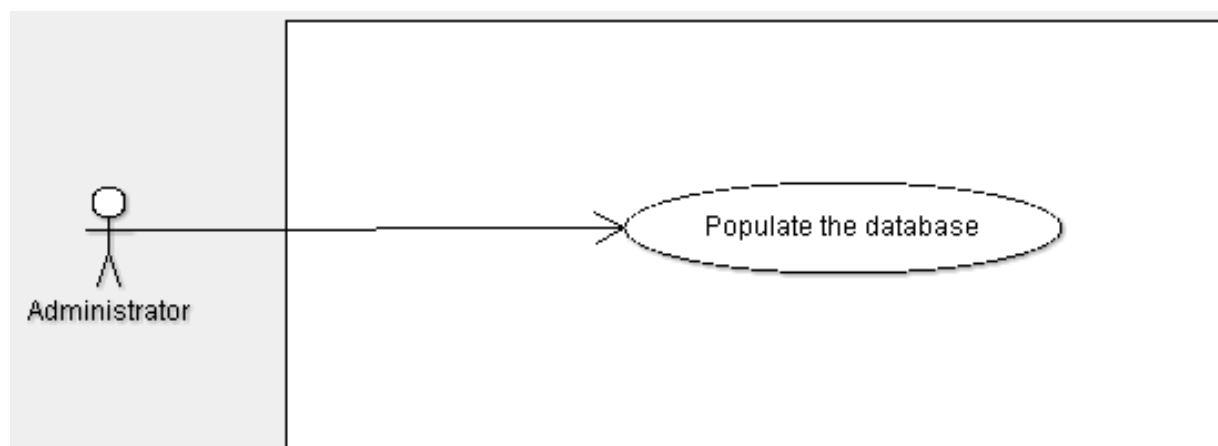
3.4 Management of external datas : Initialization

Each third-part application keep its own data. So it needs its own database. In each database, there is a copy of some data of the LDAP directory (to avoid too much connection to the LDAP directory). As a result, we must initialize the third part-application with the correct data.

We will study this problem through the Homework book application which is an application developed by the team, so that we have a clear view of the data requested and of the code.

3.4.1 Functionalities

Information related to institutions, classes, subjects is extracted as much as possible from the LDAP directory data. The non-provided information will be the object of complementary inputs. The functionality requested is that an administrator can populate a database provided by the LDAP directory. It is a batch process that is required only at the installation of the application, so it does not have to be effective nor have a graphical interface.



use case

The schema of the database is available in the appendix C.

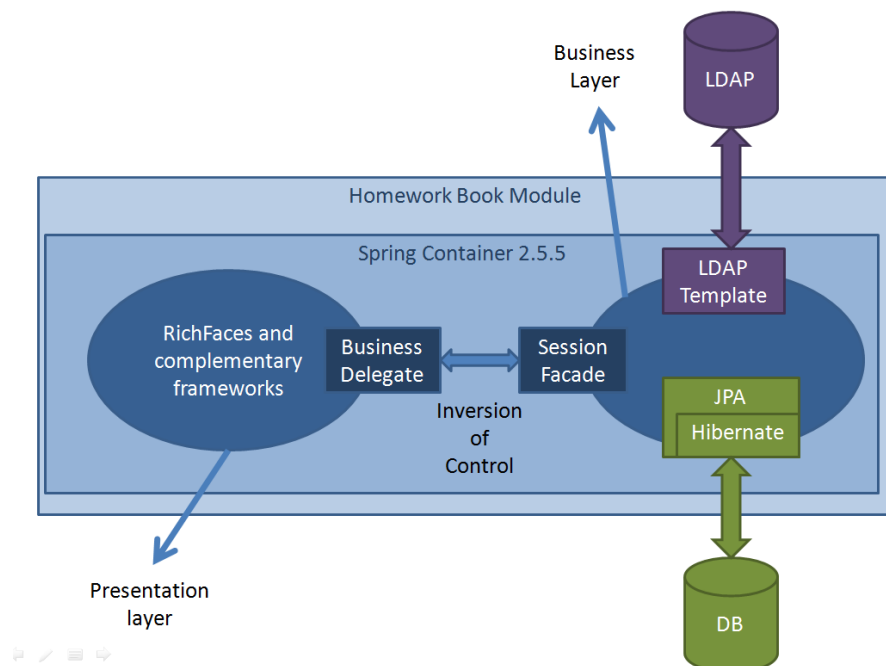
Only the following tables can be fulfilled from the data we get:

- Institution
- Student
- Teacher
- Class
- Group
- Subject
- Scholar year

Of course all the relations they hold together had also to be fulfilled.

3.4.2 Architecture

I developed this functionality in Java, using spring to manage bean and connection to the LDAP and database. As a result,I got the following architecture:

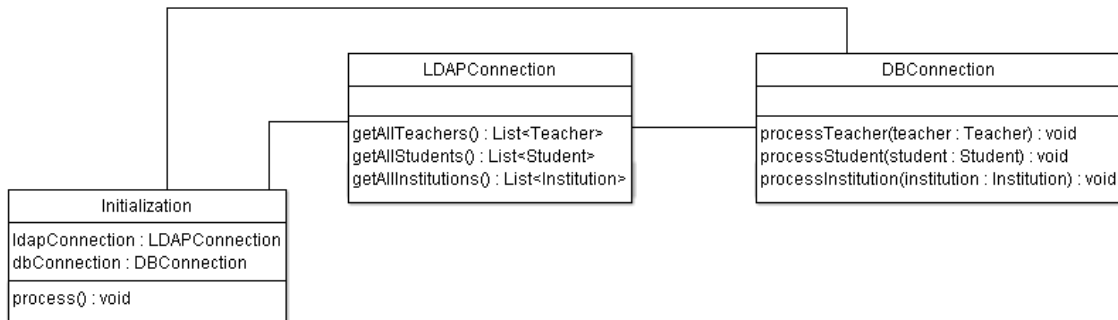


I chose four entry points into the database:

- Scholar years
- Institutions

- Teachers
- Students

The scholar year was provided as an input data. All the other information can be extracted from the LDAP directory.



3.4.3 Problems and solution

Due to the high number of data to transfer, I had to use paginated request to retrieve data from the LDAP directory: When searches return large numbers of results, it is convenient to have the server return only a certain number of results each time it is called. This is known as paged search results. Each 'page' of the result could then be treated at the time, with links to the next and previous page.

This process is called PagedResultsControl, which requests that the results of a search operation are returned by the LDAP server in pages of a specified size. The user controls the rate at which the pages are returned, simply by the rate at which the searches are called. However, the user must keep track of a cookie between the calls. The server uses this cookie to keep track of where it left off the previous time it was called with a paged results request.

It does so by providing two classes: PagedResultsRequestControl and PagedResultsCookie. The PagedResultsRequestControl class creates a PagedResultsControl with the requested page size and adds it to the LdapContext. After the search, it gets the PagedResultsResponseControl and retrieves two pieces of information from it: the estimated total result size and a cookie. This cookie is a byte array containing information that the server needs the next time it is called with a PagedResultsControl. In order to make it easy to store this cookie between searches, Spring LDAP provides the wrapper class PagedResult.

The paginated request to LDAP is a new feature of the Spring framework. It works only with the last version (Spring LDAP 1.3.0-RC1) in the special context of a not pooled connection.

So I had to find a way to make the connection close and re-open the exact same connection which implies to use a very special configuration : a MutablePoolingContextSource

but with only one connection.

3.4.4 Test cases

For this functionality the testing process was really easy. From an LDAP directory, we transfer data to a database. Basically two test were done: one with a small LDAP directory, one with a big one.

The first test let us check manually if everything were transfered correctly. Of course we had to ensure all cases existed in the LDAP directory. However there are very few number of possible case:

- A student is part of one or more classes
- A student is part of one or more groups
- A teacher teach to one or more classes
- A teacher teach to one or more groups
- A teacher tech in one or more subjects
- A person is related to only one institution

The second test let us know that the program could handle the quantity of data without problems. All we could check was that the program finished without errors and consulting the number of student, teacher, institution, class and groups created. It also let us know that with "real" data, everything went. Let also mention that with the quantity of data, we could expect any "strange" case to be covered.

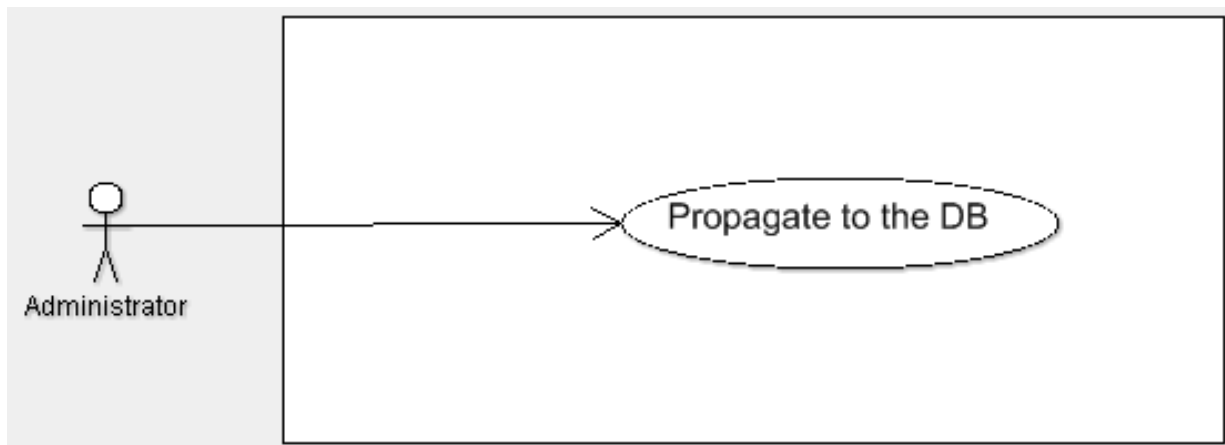
3.5 Management of external datas : Propagation

3.5.1 Functionalities

In previous chapter, we have seen the need to initialize each database. However, after each modification of the LDAP directory, databases are not consistent any more. So we need to synchronize database with the LDAP directory. Modification in the LDAP directory can come from the insertion module or from an administration module.

After each modification, all the database must re-synchronized.

This process is also a batch process, directed by an administrator. There are no requirement for a graphical interface. However, there is a strong need to ensure everything happened correctly without unplanned errors. To do so, we will need to log each actions.



3.5.2 Analysis

Two possibilities exist to synchronize databases and LDAP directories. The first one consist into propagating the modification into all the other databases, the second one consist for all the database to perform their own update. The main difference between the two option are the following: in the first case, only one process must be run whereas in the second, as many process than database must be executed. Moreover, in the first case we can factorize the code and kept in memory the read information from the LDAP directory. However in the second case, the error management is better since the effect on one database won't affect the management of another one. Finally, in the first case, we will need to manage connection to multiple database with independent transactions.

We choose to use the first solution , mainly because it will be easier for the client to use it. We will need to take a very careful look to the error management to avoid a database suffers from a problem that appear in another database.

Another difficulty was to decide how to propagate data: we could take care of synchronizing all the data or propagate only modification done.

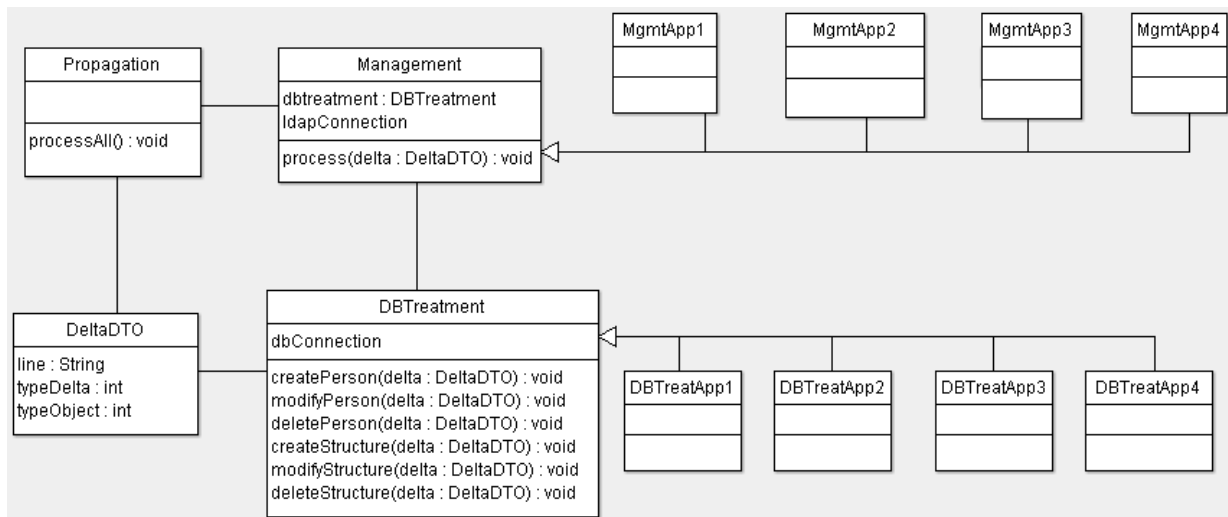
In the first case, it is very costly : we have a high number of data and it would be like re-inserting all the data at each modification. In the second case, if a problem appears, we may end up with a database not consistent with the LDAP directory without any solution to do correction.

The last solution was chosen : by logging each modification of the LDAP directory, we were able then to know which entity were modified and finally to modify only the needed record of each database. Moreover, in the log files, we declare if it is a creation, modification or a suppression and if it is a person or a structure. As a result, just by reading a file, we can decide if it is necessary to perform the synchronization or not (for example, we might not synchronize the apparition of a new structure). So just with a file we can avoid useless connection and deal with the performance problem.

3.5.3 Implementation

The propagation application has been developed in java using the spring and hibernate frameworks.

The application is in charge of reading each line of a log file. This information is stored into a DeltaDTO object containing all the information extracted from the line. For each line, it will propagate to each database. For each database a particular class is designed to decide without any connection if propagation in this particular database is requested or not. Then a call to a class in charge of modifying the database is done : one function for creating, one for deleting, one for modifying and that for each type: structure or person. Then the application need to perform a call to the LDAP directory to get the needed informations.



class diagram of the propagation application

We get the following loop:

```

for each ligne
read and analyse the ligne
propagate to database 1
propagate to database 2
...
propagate to database n

```

And each line get its own error management to avoid interaction. (Only 'planned' errors are catch) All the errors are logged in different files, one for each database.

At the end the delta file is cleaned by cutting as many lines that were treated by the application.

3.5.4 Test case

The testing of this functionality were done in parallel of the testing of the data transfer to the LDAP directory. The same data were used and transfered to appropriately set database. Since in the data transfer to the LDAP directory we think we had covered most case, I must be the same for this functionality.

3.6 Integration of pedagogical resource and Gibii

3.6.1 CAS-ifying and integrate

In order to integrate the two PHP application Moodle and Gibii in the portal, we had to cassify them.

Moodle is already designed to be used with a CAS authentication. When the Moodle works as a CAS, it operates this way:

When trying to authenticate, the user is re-directed to the CAS authentication page, where the CAS server does the authentication job. If username and password are correct, Moodle creates a new user in its database (if it does not already exist), and inherits LDAP attributes such as name, surname, email, ...

Gibii is a PHP application. CAS-ifying a PHP application is easy using the PHP-Cas library:

```
//import library
require_once('CAS/CAS.php');

//define CAS parameters
define('CASURL','xxx.xxx');
define('CASPORT','0000');
define('CASPATH','/cas');
define("CAS_VERSION_2_0",'2.0');

//code to connect
phpCAS::client(CAS_VERSION_2_0,"$CAS_SERVEUR",$CAS_PORT,"$CAS_URI",false);
phpCAS::setNoCasServerValidation();
phpCAS::forceAuthentication();

// at this step, the user has been authenticated by the CAS server
// and the user's login name can be read with phpCAS::getUser().
$loginssso=phpCAS::getUser();

//code to disconnect
$chemin_exit = /exit.php";
phpCAS::logout($chemin_exit);

?>
```

Using only these few functions the user can be automatically redirected to the CAS authentication page and, when going back from authentication, the user will be automatically recognized. In our case, the CAS parameters such as CAS address, port number,.. were stored into the database and the login page had been replaced by a page managing CAS authentication.

The integration into the portal of both application has been done through an IFrame. However, it causes some problems:

- Encoding problems: the portal is UTF-8 encoded, Gibii was Latin encoded. I created a shell script to re-encode all PHP and SQL files. The database is dynamically created by the Gibii application, so I also changed the encoding of the database.
- Disconnection: In the Gibii and Moodle application, a link is available to disconnect from the application. Since the application are integrated into the portal, these link had to disappear. However, when disconnecting from the CAS (using the disconnect button of the portal) and reconnecting as another users, does not make the CAS cookie disappears, so the new user will see the Gibii/Moodle page of the previous user. It is clearly stated in the CAS login page that, after disconnection, the user must close his browser to ensure security. If we did not want to force the user to close his browser, we could use the CAS single sign out (which is not perfectly integrated with PHP-CAS and would require a lot of adaptation) or, while trying to disconnect, calling the disconnection method for each application.
- Gibii application is using iFrame. Sometimes, by clicking on a link, the application was asking for a re-direction of the top-element, that is to say the whole page. In our case the top element was not anymore the Gibii application but the portal. So clicking caused us to get out of the portal! This problem was fixed by calling the same function on the parent Frame and no more on the top Frame.

3.6.2 Data transfer

We initialize both Gibii and Moodle database with the correct students, teachers and institution.

For the Moodle application, using a python script, we registered as a student all the person that should be able to connect to the Moodle. We created a category for each institution and registered each teacher as a course creator in the correct category. We had to face the problem Moodle is not really designed to be multi-institution and so we could not get easily separated parts.

The installation of the Gibii application requires to populate the database with data from the LDAP directory. This modularity was not done. The needed information were:

- List of institutions with their UAI (identification number of an academic institution)
- List of students with their institution and class
- List of teachers with their institution, classes and subject taught.

Although the two first sets of data were quite easy to retrieve from the LDAP directory and adapted to the Gibii format (CSV format), the last one was impossible. In the LDAP directory, we have information about the teacher and the subject they teach. In Gibii, for

each teachers, we have to provide the subject he is teaching chosen into a list of subjects. Since the designation may change between the two applications, it has been impossible to link teachers to subjects. We finally decided to provide a template file with the subject blank so that local administrators that 'know' can complete it. These listing are created using a python script.

However some problems are still unresolved :

- data transfer from previous database
- delta management

We are currently not able to get the previously installed Gibii and Moodle data to transfer them to the new version because we do not have any correspondences between the previously used login and ours.

Delta management are not take in charge (mainly because we don't know who should it and also because the delta are not really required).

4 Planning and budget

The internship has taken place from February, the first to July, the 31st. A first version of the portal had to be delivered on May, the first and a new release will be delivered September, the 15th. This special organization forced my internship to particular organization: every module should work at the 1st of May.

Since I was working in a team, the planning was not totally defined at the beginning. There were many task and as soon as someone finished one, he just started a new one according to its competences.

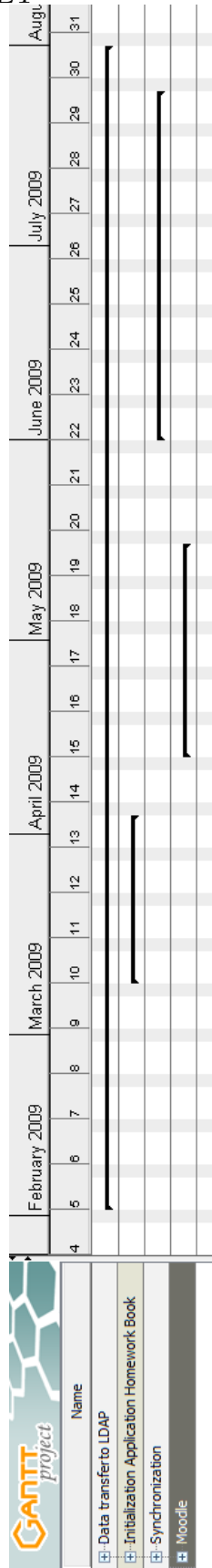
We finished in time the first version, without the need to push back the dead line, nor to suppress a module. Let us just precise that the Gibii module was, at first, not supposed to be included, but since we get enough time to do, we added it to the delivery.

As a result, I may say that the timing of the project did not differ a lot from what it was planned to be. Some modules may have take a little more or less than it was supposed to but it was not a big modification.

On the next pages, there are Gantt diagrams of the project. The five first one are those planned at the beginning of the project, the seven last one are the one done at the end of the project. You can notice there is no time assigned to the redaction of this document as I did it out of the working time.

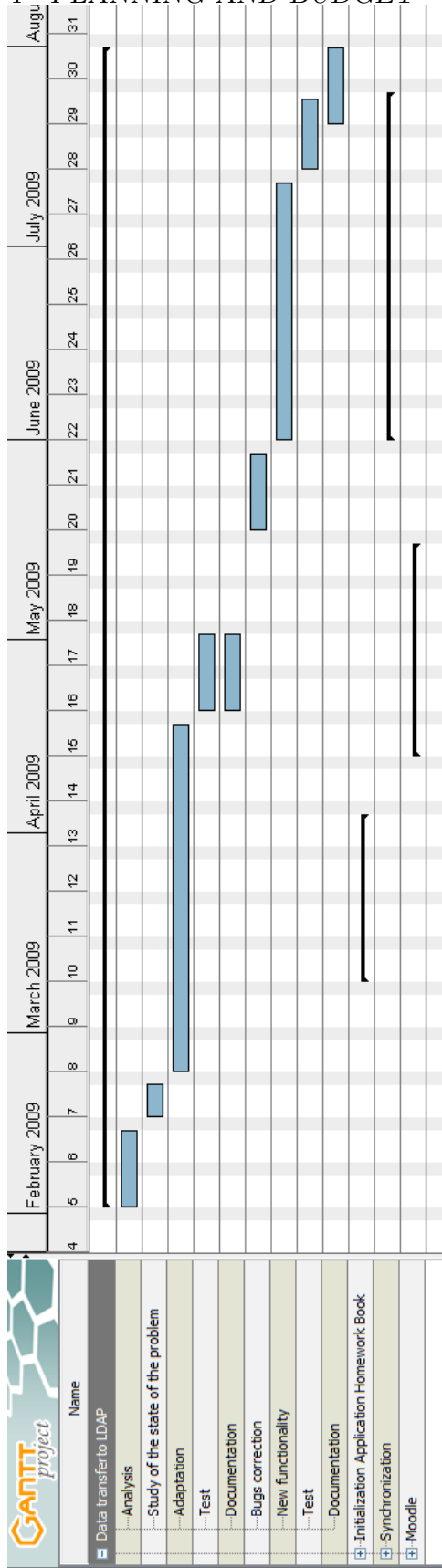
Following the Gantt diagrams, a table summarizes the estimated number of hours spend on each part and the estimated budget associated.

gantt - first version

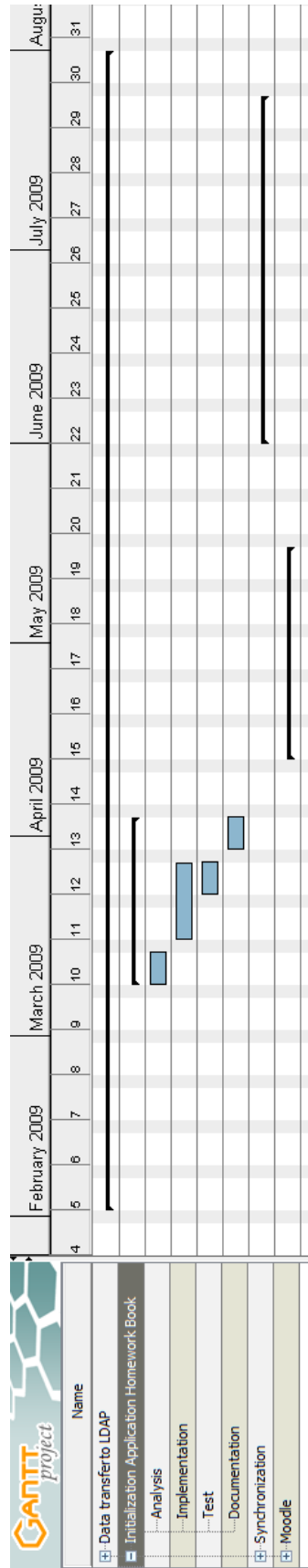


Global planning designed in February

4 PLANNING AND BUDGET

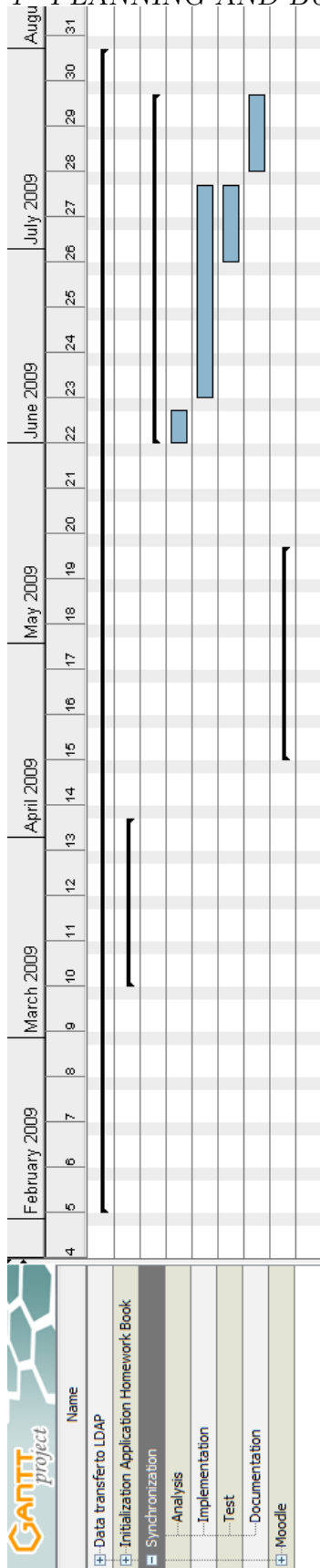


Detailed planning designed in February - part 1/4

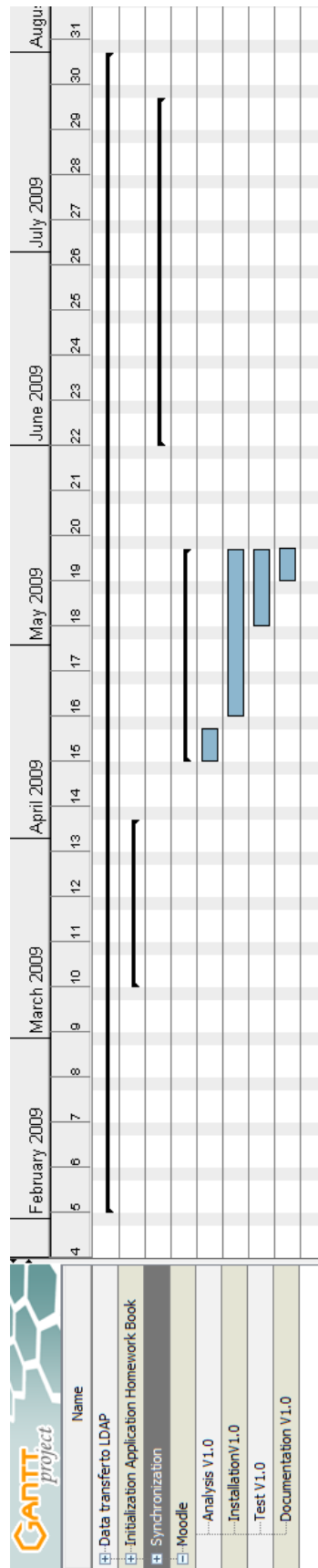


Detailed planning designed in February - part 2/4

4 PLANNING AND BUDGET

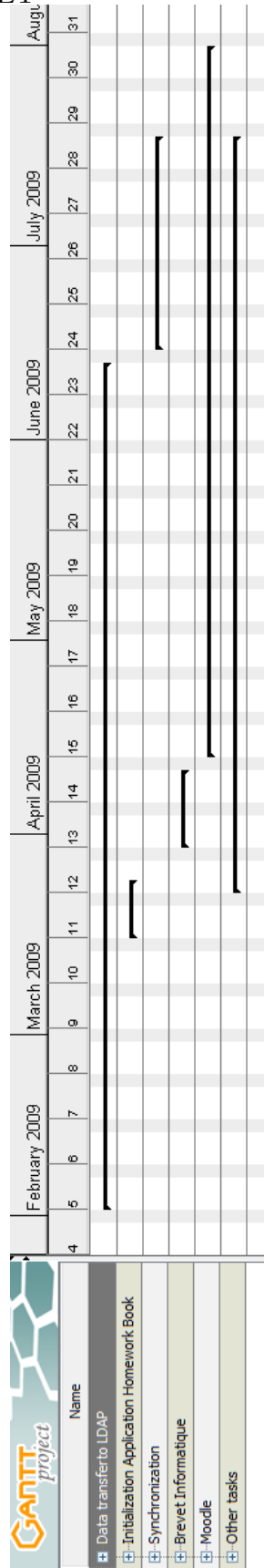


Detailed planning designed in February - part 3/4



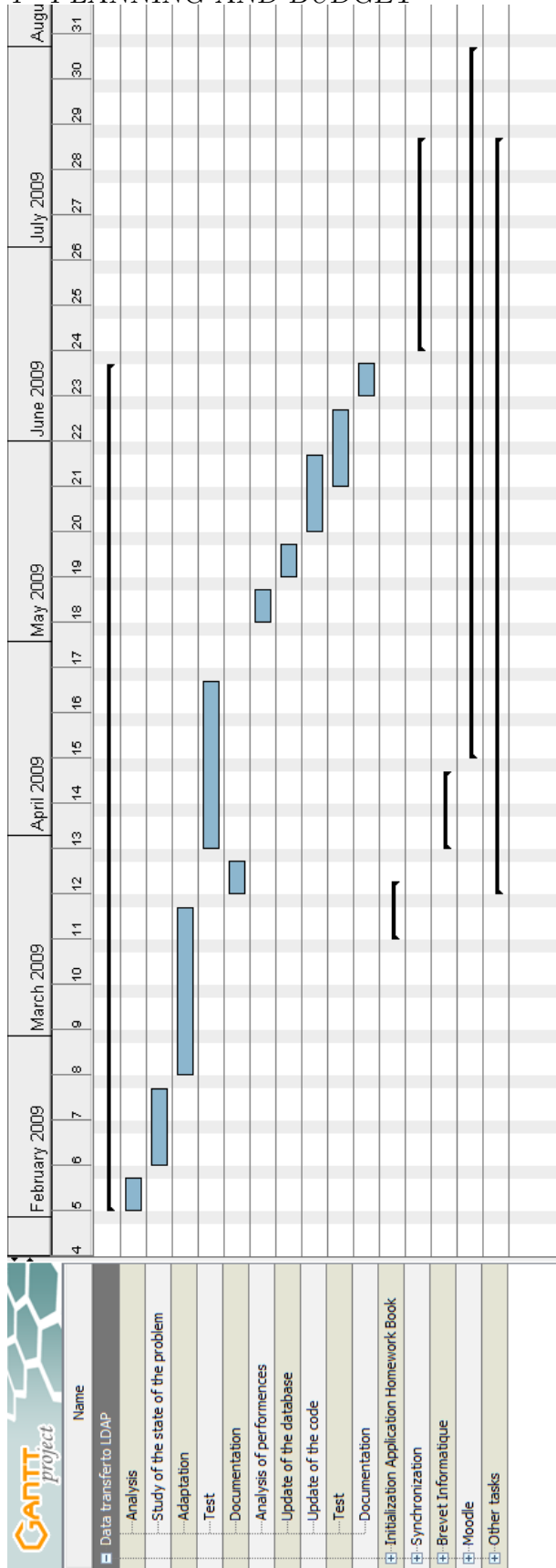
Detailed planning designed in February - part 4/4

gantt - last version

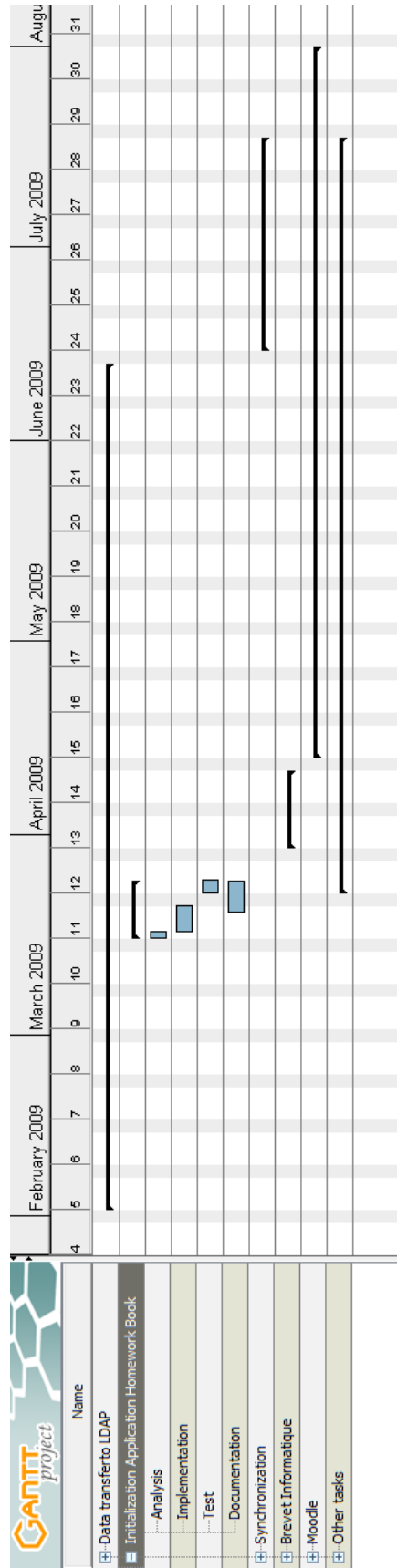


Global planning designed in August

4 PLANNING AND BUDGET

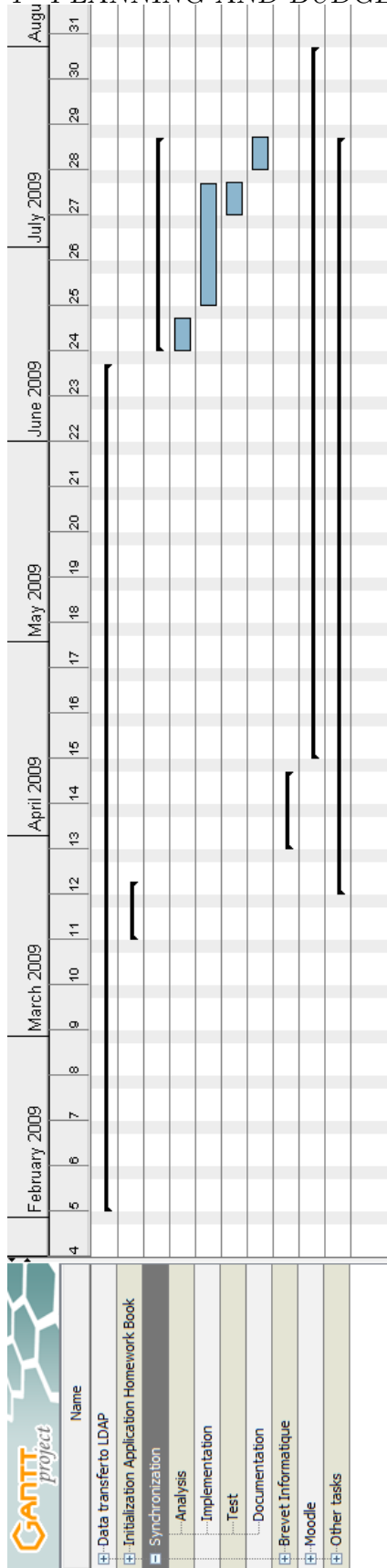


Detailed planning designed in August - part 1/6

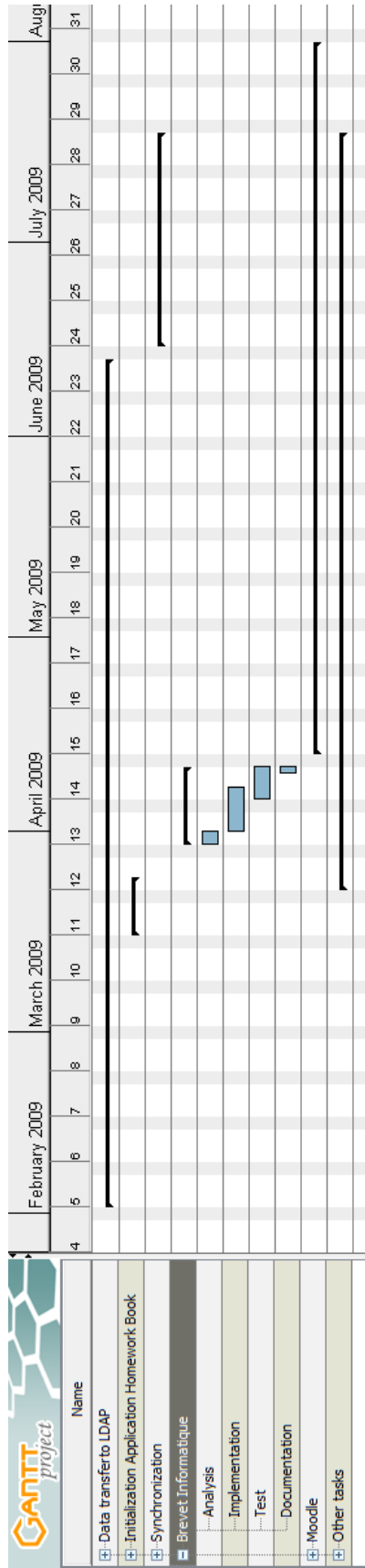


Detailed planning designed in August - part 2/6

4 PLANNING AND BUDGET

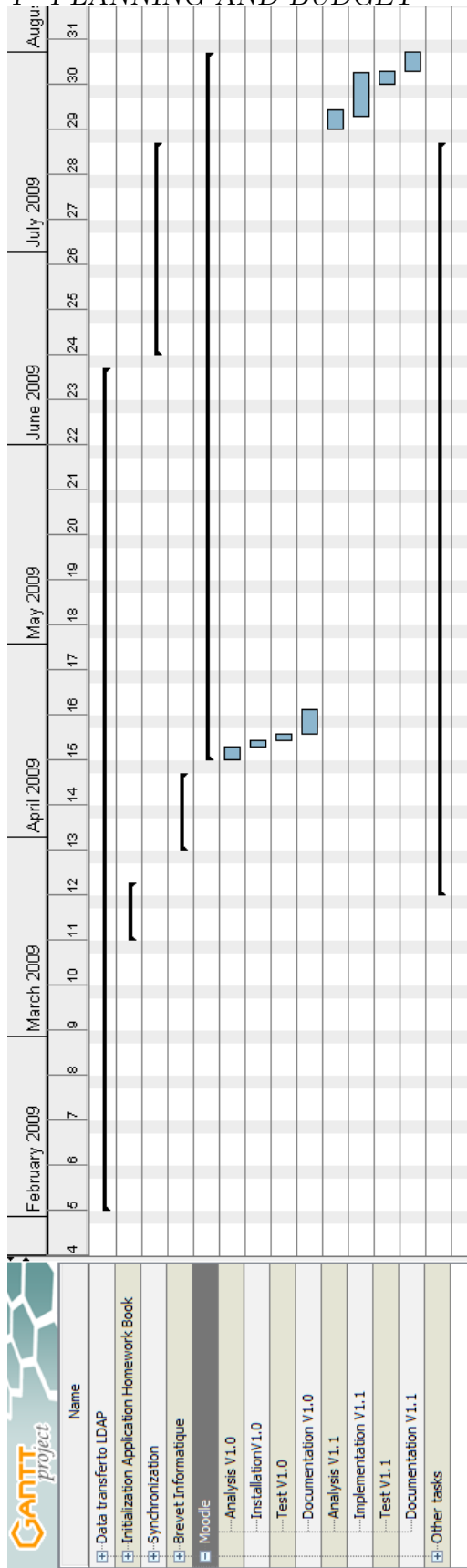


Detailed planning designed in August - part 3/6

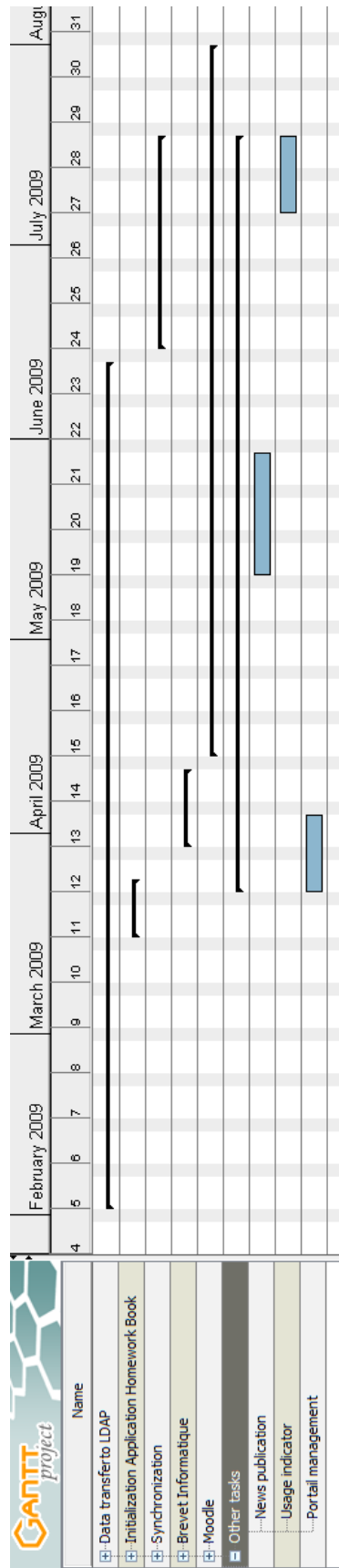


Detailed planning designed in August - part 4/6

4 PLANNING AND BUDGET



Detailed planning designed in August - part 5/6



Detailed planning designed in August - part 6/6

Task	estimated time (hours)	estimated cost in euros
Data transfer to LDAP		
Analysis	35	1925
Study of the state of the problem	70	3850
Adaptation	122	6710
Test (for V1.0)	46	2530
Documentation (for V1.0)	17,5	962
Analysis of performances	11	605
Update of the database	11	605
Update of the code	20	1100
Test (for V1.1)	20	1100
Documentation (for V1.1)	11	605
Initialization of the homework book application		
Analysis	7	385
Implementation	28	1540
Test	14	770
Documentation	14	770
Synchronization		
Analysis	11	605
Implementation	32	1760
Test	10	550
Documentation	10	550
Brevet informatique		
Analysis	14	770
Implementation	35	1925
Test	11	605
Documentation	4	220
Moodle		
Analysis V1.0	4	220
Implementation V1.0	5	275
Test V1.0	2	110
Documentation V1.0	5	275
Analysis V1.1	17	935
Implementation V1.1	28	1540
Test V1.1	7	385
Documentation V1.1	21	1155
Others		
News publication	37	2035
Usage indicator	32	1760
Portal management	32	1760
Total	743.5	40 892

5 Conclusion

I was very pleased by the web application we delivered at the beginning of May:

I mainly chose this internship because of the subject: the Technologies of Information and Communication for Education. It is quite an interesting subject where I can project myself easily. It is quite popular and, with the emergence of regulations imposing the use of TICE in education, it has a good future. In the same way, the technologies we used (Java J2EE, PHP, Hibernate, Springs ...) are very fancied and widely used.

Our implementation of the portal has been especially designed to be modular and easily changeable. We knew that the requirements would often change and that the adaptation should be done fast. As a result, we got a portal easily adaptable to any kind of situation. For example, the introduction of a new profile in the portal can be done in three steps: copy and modify a close profile for creating a new profile into the data transfer application, adapt the portal to the new profile (modify configuration files) and finally adapt each application to the new profile (only for application where the new profile has access).

The modularity of the portal allows us, now, to re-use a lot of parts in new modules we are developing for the second version. That way, our experience is capitalized and helps us to lower time to develop new modules. It helps us also to gain a graphical unity of the whole portal.

The portal we developed suffers from applications we integrated/modified and that did not fit our requirements. It imposed us limitations and inconsistencies. For example, even if we have groups into the LDAP Directory, we did not have the possibility to use them into integrated applications. It would have been better to develop from A to Z each module, except that we did not have enough time. Now, that the first version has been successfully delivered, we have more time to improve things. Some parts are currently being re-developed from zero in order to free us from the limitations they imposed. As a result, we expect a better visibility and maintainability of the code. Applications are more precisely targeted on our needs.

It may be noted that the client seemed very pleased by the demonstration of the first version we delivered. We also delivered technical and functional documentations to explain how they should install and use the portal. However, even if documentations are very detailed, they are currently facing problems to install the portal. This is mainly because their architecture differs from ours (we developed on 32 bits machines, they want to install the portal on a 64 bits architecture, we had two servers, they try load balancing and reverse proxy,...).

I would like to enforce all the strong points of my internship:

- I have integrated myself into a team and have worked under strict specifications
- I acquired experience on the J2EE and Hibernate technologies. These two technologies are very valuable nowadays and can only be an asset for my carrier.
- I discovered the Spring framework, the Python language and by the way enlarged my field of competences with new technologies.
- I saw how consequent projects are handled, what kind of methodology was applied to respect delays.
- I gain experience under the supervision of experienced developers who helped me to take correct decision when I was in needing help.

A Glossary

API : An application programming interface is a set of routines, data structures, object classes and/or protocols provided by libraries and/or operating system services in order to support the building of applications.

IT : Information technology, a broad subject concerned with aspects of managing, editing and processing information

JSR : Java Specification Requests is the formal documents that describe proposed specifications and technologies for adding to the Java platform.

LDAP : Lightweight Directory Access Protocol is an application protocol for querying and modifying directory services running over TCP/IP.

Portlet Portlets are pluggable user interface software components that are managed and displayed in a web portal.

Servlet Java programming language objects that dynamically process requests and construct responses.

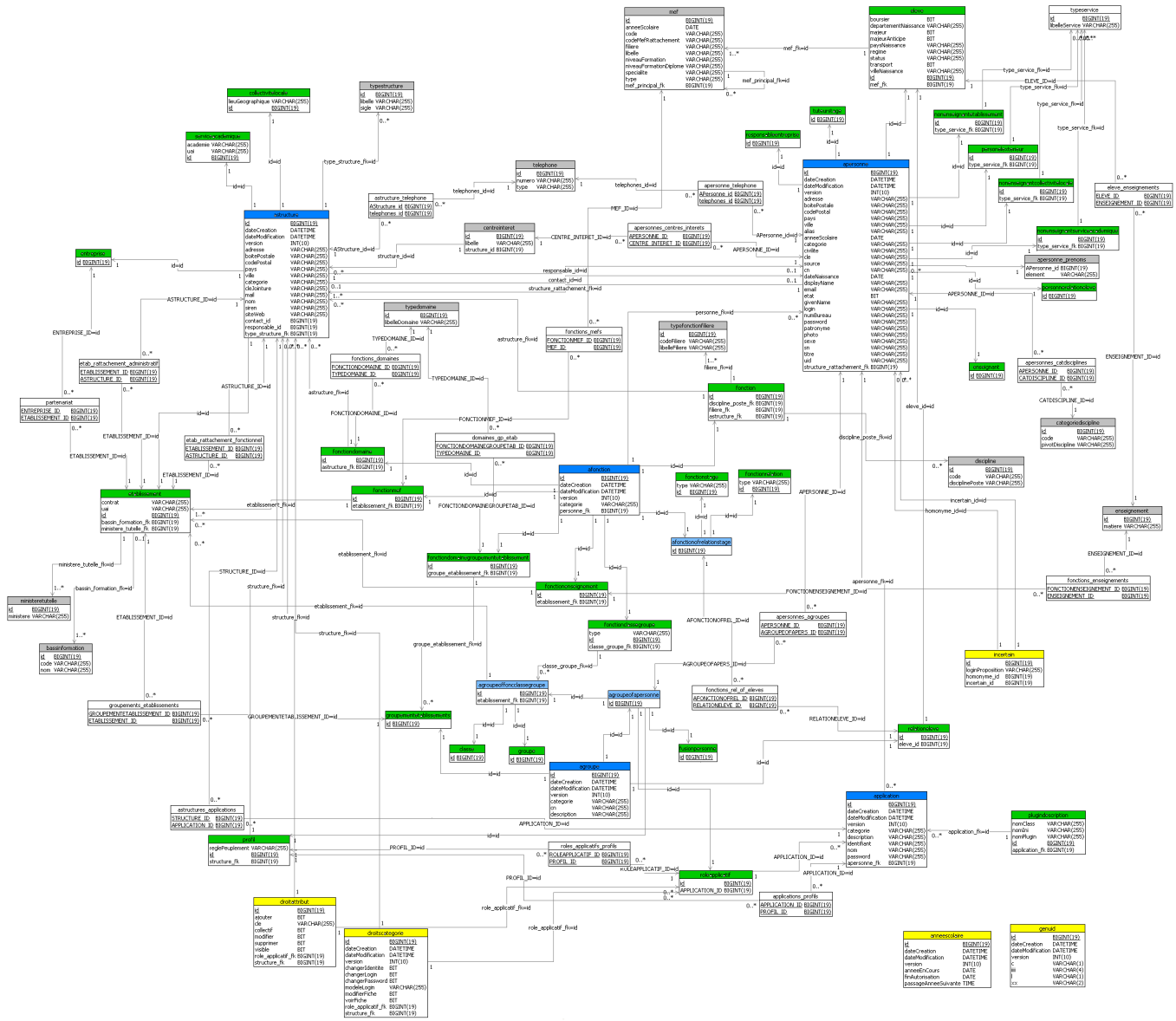
SSO : Single sign-on (SSO) is a property of access control of multiple, related, but independent software systems. With this property a user logs in once and gains access to all systems without being prompted to log in again at each of them. Single sign-off is the reverse property whereby a single action of signing out terminates access to multiple software systems.

B Accessible modules

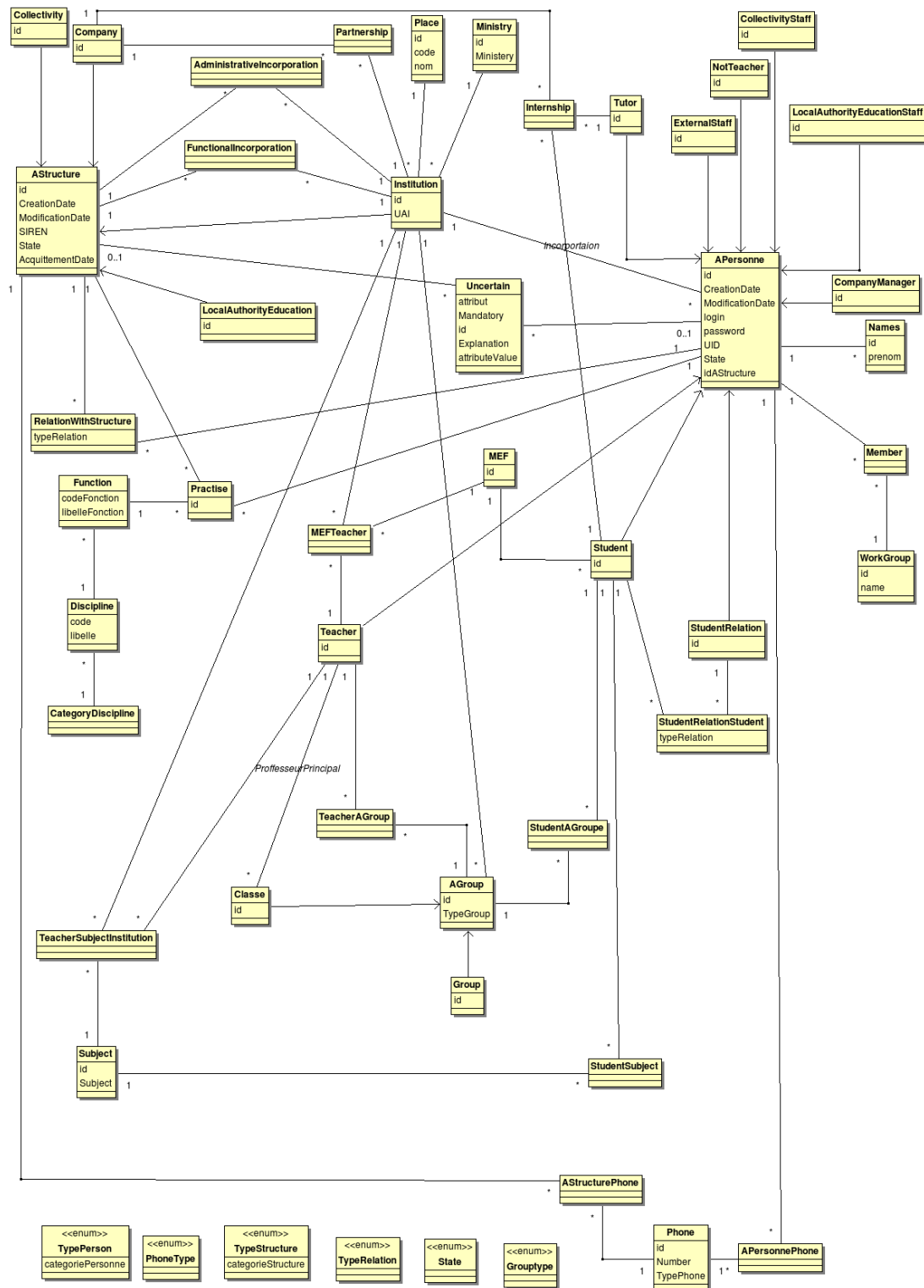
	Student	Teacher	Parent	Headmaster	Educational Inspector	Local administrator	Global administrator	Portal administrator
Tab "Acceuil"								
News reading	X	X	X	X	X			
Tab "Mon bureau"								
Homework book	X	X	X	X	X			
mark	X	X	X	X				
absences	X	X	X	X				
pedagogical ressources	X	X		X				
Tab "Mes documents"								
Storage	X	X	X	X	X			
Tab "Mes outils"								
Brevet informatique	X	X		X				
Directory	X	X	X	X	X			
Tab "Communication"								
Web-mail	X	X	X	X	X			
Address book	X	X	X	X	X			
Diary	X	X	X	X	X			
News reading for the parents	X	X	X	X				

C Databases schema

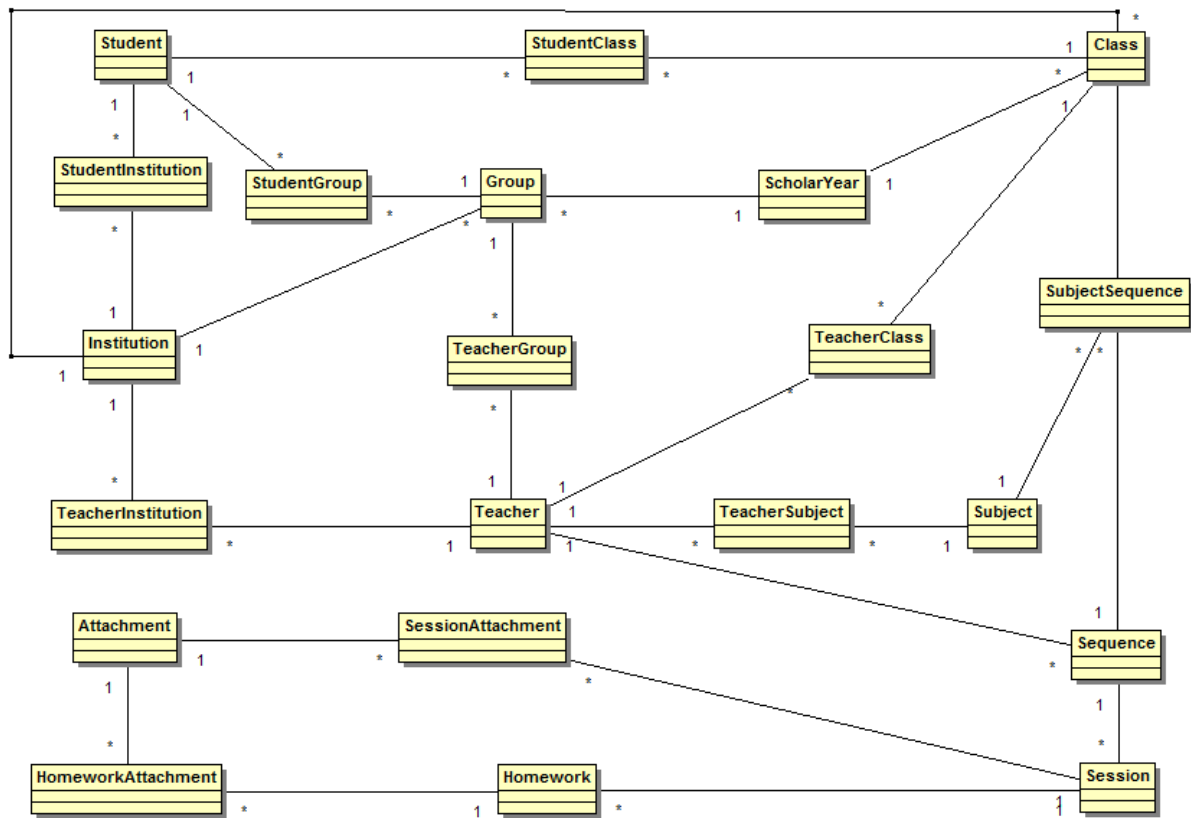
Old Sarapis database



New Sarapis database



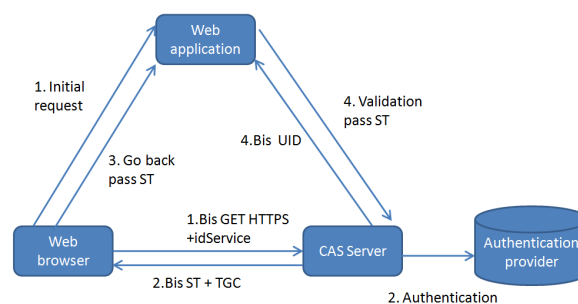
Homework book database



D CAS life-cycle

The Central Authentication Server is designed as a standalone web application. It is currently implemented as several Java servlets and runs through the HTTPS server. It is accessed through three URLs described below: the login URL, the validation URL, and the optional logout URL.

No proxy mode:



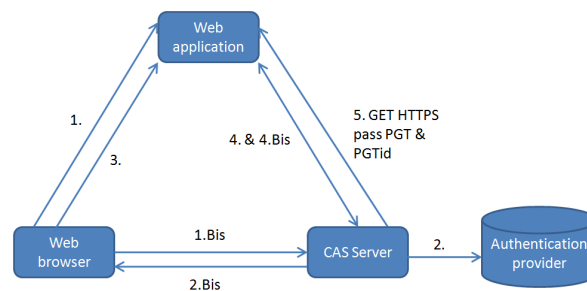
ST: Service Ticket

TGC: Ticket-Granting Cookie

The ticket generation algorithm uses the Java `SecureRandom()` function.

1. Initial request: the client accesses to a web application requesting an authentication. This application redirects the request to the login url of the CAS server (HTTPS request). The ID of the service is passed (this ID is the url of the web application) as a parameter.
2. Authentication : the CAS server prompts the user for a NetID and a password to validate it against a backing authentication provider, the LDAP directory in our case. The CAS server redirects the request to the initial application, with a cookie (if accepted by the client) called TGC and a 'Ticket', the ST. This ticket is usable only once for a given user and a given application
3. Returns to the application with the ST, as a GET parameter
4. Validation : Application accesses the CAS server (HTTPS connection) at the validation url and gives as a parameter the service ID and the ST. The CAS server ensures validity of the ticket and gives back the uid (an identifier) of the person. The resources associated with this ticket are freed in the CAS server.

Proxy mode :

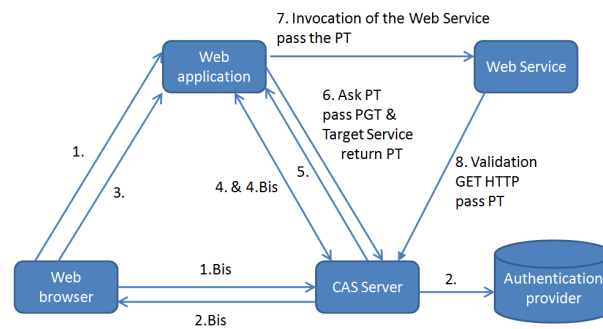


PGT: Proxy-Granting-Ticket

1. the same than previously
2. the same than previously
3. the same than previously
4. Validation: Application accesses the CAS server (HTTPS connection) at the validation url and gives as a parameter the service ID and the ST and a callback url. The CAS server ensures validity of the ticket. In case of success, it gives back the uid of the person and a PGT-id, an index to validate the PGT.
5. Send the PGT : at the same moment, the CAS server generates an HTTPS connection to the callback url of the application with the PGT and the PGT-id. The PGT is time limited. Only TGC and PGT can be reused.

Now the web proxy application has the PGT of the user. With it, the proxy CAS can ask the CAS to re-generate PT which are the equivalent of ST but for third party applications.

If the proxy CAS asks for a third party service requesting authentication :



PT: Proxy-Ticket

- Ask for a PT: the proxy CAS generates an HTTPS request to the CAS server (proxy url), passing in parameter the PGT and the 'target Service' (the url of the targeted application). The CAS returns a PT.
- PT transmission: the proxy CAS passes the PT to the third party service.
- Validation: the service uses the PT as an ST except that the CAS server may return along with the uid of the person, the list of the proxies that separate it from the user.

E XML extract

Here are the XML files for the Sarapis project

E.1 DTD associated to the XML

```

<!ELEMENT ficAlimMENESR (addRequest|modifyRequest|deleteRequest)*>
<!ELEMENT addRequest (operationalAttributes, identifier, attributes)>
<!ELEMENT modifyRequest (operationalAttributes, identifier, modifications)>
<!ELEMENT deleteRequest (operationalAttributes, identifier)>
<!ELEMENT operationalAttributes (attr)>
<!ELEMENT identifier (id)>
<!ELEMENT attributes (attr+)>
<!ELEMENT attr (value+)>
<!ELEMENT modifications (modification+)>
<!ELEMENT modification (value+)>
<!ELEMENT value (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ATTLIST attr name CDATA #REQUIRED>
<!ATTLIST modification
name CDATA #REQUIRED
operation (replace) #REQUIRED>

```

E.2 XML School example

```

<addRequest>
<operationalAttributes><attr name="categorieStructure">
<value>EtabEducNat</value></attr></operationalAttributes>
<identifier><id>1234</id></identifier>
<attributes>
<attr name="ENTStructureJointure"><value>1234</value></attr>
<attr name="ENTStructureUAI"><value>123456789A</value></attr>
<attr name="ENTetablissementUAI"><value>123456789A</value></attr>
<attr name="ENTStructureSIREN"><value>19110008000016</value></attr>
<attr name="ENTStructureNomCourant"><value>LP-JULES FIL-ac-MONTPEL.</value></attr>
<attr name="ENTStructureTypeStruct"><value>LYCEE PROFESSIONNEL</value></attr>
<attr name="ENTetablissementMinistereTutelle">
<value>MINISTERE DE L'EDUCATION NATIONALE</value></attr>
<attr name="ENTetablissementContrat"><value>PU</value></attr>
<attr name="postOfficeBox"><value>57</value></attr>
<attr name="street"><value>BOULEVARD JOLIOT CURIE</value></attr>
<attr name="postalCode"><value>11021</value></attr>

```



```

<attr name="l"><value>CARCASSONNE CEDEX</value></attr>
<attr name="telephoneNumber"><value>01 23 45 67 89</value></attr>
<attr name="facsimileTelephoneNumber"><value>01 23 45 67 89</value></attr>
<attr name="ENTetablissementStructRattachFct1"><value>3197</value></attr>
<attr name="ENTetablissementBassin"><value>11011$CARCASSONNE</value></attr>
<attr name="ENTServAcAcademie"><value>MONTPELLIER</value></attr>
</attributes>
</addRequest>

```

E.3 XML Student example

```

<addRequest>
<operationalAttributes><attr name="categoriePersonne">
<value>Eleve</value></attr></operationalAttributes>
<identifiant><id>123456</id></identifiant>
<attributes>
<attr name="ENTPersonJointure"><value>123456</value></attr>
<attr name="ENTPersonDateNaissance"><value>11/06/1991</value></attr>
<attr name="ENTPersonNomPatro"><value>ABCDEF</value></attr>
<attr name="sn"><value>ABCDEF</value></attr>
<attr name="givenName"><value>Abcdef</value></attr>
<attr name="ENTPersonAutresPrenoms"><value>Abcdef</value>
<value>Ghijk</value><value>Lmno</value></attr>
<attr name="personalTitle"><value>M.</value></attr>
<attr name="ENTEleveParents"><value/></attr>
<attr name="ENTElevePere"><value/></attr>
<attr name="ENTEleveMere"><value/></attr>
<attr name="ENTEleveAutoriteParentale"><value>123457<value></attr>
<attr name="ENTElevePersRelEleve1"><value/></attr>
<attr name="ENTEleveQualitePersRelEleve1"><value/></attr>
<attr name="ENTElevePersRelEleve2"><value/></attr>
<attr name="ENTEleveQualitePersRelEleve2"><value/></attr>
<attr name="ENTEleveBoursier"><value>N</value></attr>
<attr name="ENTEleveRegime"><value>1/2 PENSION</value></attr>
<attr name="ENTEleveTransport"><value>0</value></attr>
<attr name="ENTEleveStatutEleve"><value>SCOLAIRE</value></attr>
<attr name="ENTEleveMEF"><value>20211010110</value></attr>
<attr name="ENTEleveLibelleMEF"><value>TERMINALE SCIENTIFIQUE SVT</value></attr>
<attr name="ENTEleveNivFormation"
><value>TERMINALE GENERALE & techno YC BT</value></attr>
<attr name="ENTEleveFilier"><value>TERMINALE GENERALE</value></attr>
<attr name="ENTEleveEnseignements"><value>ARTS PLASTIQUES</value>
<value>ANGLAIS LV1</value><value>ESPAGNOL LV2</value>

```

```

<value>EDUCATION CIVIQUE JURIDIQUE ET SOCIALE</value>
<value>EDUCATION PHYSIQUE ET SPORTIVE</value>
<value>FRANCAIS</value><value>HISTOIRE ET GEOGRAPHIE</value>
<value>MATHEMATIQUES</value><value>PHYSIQUE-CHIMIE</value>
<value>SCIENCES DE LA VIE ET DE LA TERRE</value>
<value>TRAVAUX PERSONNELS ENCADRES</value>
<value>PHILOSOPHIE</value><value>MUSIQUE</value></attr>
<attr name="ENTPersonStructRattach"><value>1234</value></attr>
<attr name="ENTEleveClasses"><value>1234$TS2</value></attr>
<attr name="ENTEleveGroupes"><value/></attr>
</attributes>
</addRequest>

```

E.4 XML Teacher example

```

<addRequest>
<operationalAttributes><attr name="categoriePersonne">
<value>PersEducNat</value></attr></operationalAttributes>
<identifiant><id>123456</id></identifiant>
<attributes>
<attr name="ENTPersonJointure"><value>123456</value></attr>
<attr name="ENTPersonDateNaissance"><value>15/08/1967</value></attr>
<attr name="ENTPersonNomPatro"><value>ABCDEF</value></attr>
<attr name="sn"><value>ABCDEF</value></attr>
<attr name="givenName"><value>ABCDEF</value></attr>
<attr name="personalTitle"><value>M.</value></attr>
<attr name="mail"><value>mail@ac-montpellier.fr</value></attr>
<attr name="ENTPersonStructRattach"><value>1234</value></attr>
<attr name="ENTAuxEnsCategoDiscipline">
<value>41$GENIE MECANIQUE DE LA CONSTRUCTION</value></attr>
<attr name="ENTAuxEnsDisciplinesPoste">
<value>GENIE MECANIQUE CONSTRUCTION$41</value></attr>
<attr name="ENTAuxEnsMEF">
<value>1234$21125106110$1_STI GENIE MECANIQ: SYSTEMES MOTORISES</value>
<value>1234$31120107210$1BTS2 MECA.ET AUTOMATISMES INDUSTRIELS</value></attr>
<attr name="ENTAuxEnsMatiereEnseignEtab">
<value>1234$ETUDE DES CONSTRUCTIONS</value>
<value>1234$CONCEPTION DES PARTIES OPERATIVES</value></attr>
<attr name="ENTAuxEnsClasses"><value>1234$MAI1</value></attr>
<attr name="ENTAuxEnsGroupes"><value>1234$PRE 2 SM</value>
<value>1234$PRE 2 SA</value><value>1234$TS1 MAIA</value></attr>
<attr name="ENTAuxEnsClassesPrincipal"><value/></attr>
<attr name="ENTPersonFonctions">

```

```
<value>1234$ENS$ENSEIGNEMENT$L4100$GENIE MECANIQUE CONSTRUCTION</value></attr>  
<attr name="PersEducNatPresenceDevantEleves"><value>0</value></attr>  
</attributes>  
</addRequest>
```

References

- [BULL] **BULL** *www.bull.com*
 Company description <http://www.bull.com/about/company.html>
- [EDUCNET] **educnet** <http://www.educnet.education.fr/>
 Technical recommendation about digital environment :
<http://www.educnet.education.fr/services/ent/sdet>
 Preconization from the ministry about the directory
<http://www.educnet.education.fr/services/ent/scolaire/annuaire-ent>
- [REGION] **Region Langudoc-Roussillon** <http://www.laregion.fr/>
- [JBOS] **JBoss** <http://www.jboss.org/>
- [LIFERAY] **Liferay** <http://www.liferay.com/web/guest/home>
- [EXO] **Exo** <http://www.exoplatform.org/portal/public/en/>
- [ESUP] **ESUP-Portail** www.esup-portail.org
 Project space <http://www.esup-portail.org/display/PROJ/Projets>
- [JASIG] **JASIG** www.jasig.org
 uPortal documentation <http://www.jasig.org/uportal/documentation>
 CAS documentation <http://www.ja-sig.org/wiki/display/CAS/Home>
- [DOC1] Yen Lu Soh **A Review and Evaluation of Portal Software solutions for the northern Leadership Academy Web Portal**
<http://northernleadershipacademy.co.uk/> 6th September 2007
- [DOC2] Xiaobo Yang, Xiao Dong Wang, Rob Allan **JSR 168 and WSRP 1.0 : How mature are portal standards?**
<http://epubs.cclrc.ac.uk/bitstream/1082/paper06A.pdf>
- [DOC3] SMILE **Portails, les solutions open source**
<http://www.smile.fr/publications/livres-blancs/portails-open-source>
- [PORTLET] Fabrice SZNAJDERMAN **Portlets - JSR168**
<http://sznajderman.developpez.com/articles/portlet/jsr168> 2007
- [SSO] Chris Dunne **Build and implement a single sign-on solution**
<http://www.ibm.com/developerworks/web/library/wa-singlesign/>
- [PLUTO] **Apache pluto official website** <http://portals.apache.org/pluto/>
- [SPRING] **Spring official website** <http://www.springframework.org/>
 Documentation <http://www.springframework.org/documentation>

[HIBERNATE] **HIBERNATE - Persistence relationnelle en Java standard**
<http://docs.jboss.org/hibernate/stable/core/reference/fr/html/>

[ESCO] **ESCO-Portail** *wiki.esco-portail.org*