Revista de Sistemas de Informação da FSMA
n. 15 (2015) pp. 11-22

SISTEMAS DE INFORMAÇÃO
http://www.fsma.edu.br/si/sistemas.html

Approaches and Technologies for Systems Integration: A Case Study at the Federal University of Lavras

C. M. Garcia, R. Abilio – *Board of Management of Information Technology (DGTI)*
N. Malheiros – *Department of Computer Science (DCC)*
Federal University of Lavras (UFLA)
{cristiano.garcia,ramon.abilio}@dgti.ufla.br, neumar@dcc.ufla.br

***Abstract*—This case study aims to analyze the integration's approaches and technologies among information systems and services in an academic environment. It has been done a study on the integration history in the Federal University of Lavras. The data had been gathered via questionnaires and documentation analysis. In this analysis, 4 distinct phases were specified. Besides, the advantages and disadvantages of each phase were discussed. The main contribution of this work is the analysis of different integration approaches among information systems and other services.**

***Index Terms*—Information Systems Integration, Service-Oriented Architecture, Web Services.**

I. INTRODUCTION

Both public and private institutions, of all areas and sizes, have information systems to help manage their processes. In corporate environments it is common to find a scenario with different types of systems, both in the operational, managerial and strategic levels [23]. With that, all data is shared even among systems that are not in the same administrative level. Thus, we can say that there are two different types of integration: vertical (between systems in different levels) and horizontal (between systems at the same level).

There are several classifications for systems integration. For instance, there are those that focus on the implementation level and others that focus on the organization and its processes. "Centralized Information" [17] is an example of approach that focuses on the implementation level and *Business-to-Business* (B2B) [13] is as example of approach that focuses on the organization and its processes.

No matter the chosen approach, one of the ways to integrate systems is by using Service Oriented Architecture – SOA) [19]. Using SOA for systems integration is called Service-Oriented Integration (SOI) [12] and a way to perform this integration using SOI is by using Web Services, which represent a vision that encompasses distributed programming and resource availability strongly linked to the Internet [22]. Several technologies and standards, such as Simple Object Access Protocol (SOAP), Representational State Transfer (REST) architecture and Java API for Web Services (JAX-WS[1]), offer support for the implementation of Web Services [3].

Using Web Services for systems integration is an approach that can also be used by academic institutions. In the University of Açores (Portugal), a set of *Web Services* was developed to optimize critical tasks that involve financial and strategic information [4]. At the Federal University of Pelotas (UFPel - Brasil), a set of Web Services was also developed to maintain information consistent among systems and services, like the university restaurant, the Virtual Learning Environment and the institutional wireless network [1].

The Federal University of Lavras (UFLA), which is under study in this paper, has experienced a strong growth in the past few years. Until June/2013, nearly 16000 students were enrolled in the University and around 1600 employees worked there [25]. This number will probably grow with the creation of new graduation programs [24].

According to the Strategic Planning for Information Technology 2011/2012 [8], until 2011, UFLA had 24 information systems being used by the Board of Management of Information Technology (DGTI). Each one of those systems had specific characteristics concerning its goal (academic, administrative and support), the specific database management system, the platforms and technologies used in its development and were developed by different companies. In 2013, a "family" of integrated information systems started to be deployed at the university. These systems are developed with technologies that are different from those currently functioning at the institution and this adds new challenges for the

1 http://docs.oracle.com/javase/7/docs/technotes/guides/xml/jax-ws/

integration process.

Besides the above mentioned software systems, the university also offers network services, such as e-mail and wireless access to the whole academic community. Hence, there is also the need for integrating those services and systems to the university environment.

This heterogeneous environment makes system integration a great challenge. The main goal of this paper was to perform a case study to analyze the integration solutions for UFLA information systems, highlighting the advantages and disadvantages of each solution used in order to help fulfill this challenge in the best possible way. One of the contributions of this work is an analysis of different approaches to integrate systems which may help IT professionals choose the more adequate approach to their scenario.

This paper is organized as follows. Section 2 presents a brief introduction on information systems. Section 3 presents the methodology and is followed by Section 4 which presents the results and finally, Section 5 will offer some conclusions.

II. INFORMATION SYSTEMS INTEGRATION

Information systems have been widely used in the last years and their usage tends to grow in a meaningful way in the next years [2]. The processing power aggregated to the big organizations has also grown as computers have become smaller and less expensive [14]. Hence, public and private institutions of the most diverse areas, from larger to smaller ones, already have computers with information systems to help manage the business and automate the daily activities [18].

Using computers and information systems is more than simply automatizing operational level activities. Information technology (IT) has become a strategic tool, because the company that uses IT in an effective way, integrating IT strategies to the business strategies, is able to obtain competitive advantages [15].

There are many reasons that can make organizations hire or develop systems integration solutions, including the following [7]:

- Understanding the existing technology to flexibilize it and reduce the implementation cost of new services;
- Allowing integration with stakeholders, expanding the reach of the services;
- Integrating common information from different databases that may come from mergers, acquisitions or legacy systems.

The integration of business systems is a complicated task that involves risks, as each organization has its specific characteristics and integration needs [17]. An organization may perform the systems integration considering different *foci*, for instance, on the implementation level or on the organization and its processes.

A. *Classifications of Systems Integration*

Information systems integration can be classified, for instance, by considering the level at which it was implemented [17] or the organization and its processes [13].

Considering the implementation level, systems integration can be classified as [17]:

- *Compound applications*: applications integrated using an API[2], which works as a connector among systems;
- *Centralized Information*: this occurs when different systems access the same database, sharing the same metadata;
- *Enterprise Management Systems*: closed systems formed by independent internal modules. This integration is usually performed at the source code level.

When focusing on the organization, integration can be classified as follows [13]:

- *Data Replication*: the integration happens at the information level. Databases are distributed and kept up to date and synchronized;
- *Business-to-Business (B2B) Integration*: extrapolates the organization limits. Represents the availability of functions from different organizations. Even though other concepts may also be applied to B2B, integration using external networks can rise new issues for analysis [13];
- *Service-Oriented Architecture*: the systems offer functions as services which, in the context of computer science, means a “well-defined and universally available function”.

Even though the *foci* are different, the taxonomies found have some similarities [13][17]. For instance, the approaches “*Business-to-Business (B2B) Integration*” and “*Compound Applications*” may both use an information sharing API and, hence, perform a possible systems integration.

B. *Service Oriented Architecture*

Software architecture represents the structure that includes all software components, its externally visible properties and the relationship among them [20]. Service-Oriented Architecture (SOA) has become one of the main paradigms of distributed systems, including originating a branch of Software Engineering called Service Software Engineering [27].

SOA is a paradigm that seeks to organize and use resources that might be under control of different owners, by making available an uniform way to offer, discover, interact and use the functionalities in order to achieve consistent

2 Application Programming Interface

desired effect [19]. This paradigm can offer several benefits, such as the control of system growth, offer and usage of services in a global scale and cost reduction in cooperation between organizations [26].

Using SOA for integration, we achieve Service-Oriented Integration (SOI), whose main goal is to create an integration among multiple systems, with little or no change in its implementations [12]. This technique exposes data, functionalities and processes to be consumed by the systems that participate in the integration. There are many approaches for SOI with the focus on integrating existing systems [12], among which we can highlight the following:

- *Services*: uses a service layer between the existing services and the service consumers;
- *Process Integration*: integrates processes in a business environment, suggesting the integration on small processes inside bigger ones, with ou without human intervention;
- *Data Integration*: refers to an approach that manages model complexity in different applications.

Web Services represent a way to implement SOI, offering a service interface that allows consumers to Interact with the service providers [5]. The most common implementation of web services are Simple Object Access Protocol (SOAP) and Representional State Transfer (REST), also mentioned as solutions to integrate processes between organizations. [29].

SOAP is a protocol based on the Web Services Description Language (WSDL), which is based on XML and describes the functionalities offered by a web service [29]. SOAP offers a basic communication standard in which each operation is represented by its terminal described in the XML document sent in the requisition through a HyperText Transfer Protocol (HTTP) method, as used in REST [29].

REST is an abstraction of the constituent principles that make the *World Wide Web* (WWW) scalable [29]. It allows for the offer of services identified by an *Uniform Resource Identifier* (URI), using the HTTP 1.0: GET, POST, DELETE and PUT methods. The usage of these methods defines the operation that will be performed: GET lists records, POST inserts new records, DELETE removes records and PUT updates a record [10]. An URI is an uniform way to identify network resources and the most known type of URI is the *Uniform Resource Location* (URL) [29].

Given that REST is not a protocol as SOAP, but an architecture [10], the return value of the calls may be formatted according the application requirements. For instance, we can use *Javascript Object Notation* (JSON) instead of XML.

The study on the classification methods for systems integration allowed us to identify the approaches used to integrate information systems and network services at our institution. Besides, the concepts related to the Service-Oriented Architecture allowed us to understand the architecture used in the fourth phase of the UFLA system integration history.

III. Metodology

UFLA has a Board of Management of Information Technology (DGTI) to keep the infrastructure of computer networks. DGTI is subdivided into five groups, one of which is the Coordinating Body of Information Systems (CSI), which is responsible for the “definition, analysis, programming, deployment, maintenance and documentation of the information systems for the Institutional administrative organs”. Among the strategic goals of the DGTI aligned with the CSI responsibilities [8], we have: i) Improve management at DGTI projects; ii) Integrate the institutional information systems and iii) Improve the quality of software development and acquisition.

This work was performed at the CSI between August and October 2014, together with those professionals that are directly responsible for the integration of systems and services within the university. This work was performed in three steps:

Step 1: Identification of the university software systems, pointing out the technologies used, the software systems goals and their mutual dependencies;

Step 2: Case studies were performed through the application of questionnaires and documental research. The goal of this study was to identify the integration approaches, the motivation behind their choice and the expectations with their usage, besides the discussion of the advantages and disadvantages of each approach. The questionnaire had the following eleven questions:

1. Which were the first system integration needs at the university?
2. Which systems and services were the first to be integrated? What was the necessary information for the integration?
3. Which were the chosen approaches? What were the reasons for choosing them?
4. What were the expectations with the systems integration?
5. What were the advantages and disadvantages found in those approaches?
6. Were the expectations met?
7. Was there a change (inclusion / exclusion) in the systems that were integrated?
8. What are the systems currently integrated (2014)?
9. Why the change in the integration strategy? What was the motivation behind it?
10. What are the expectations with the new approach?
11. What are the advantages and disadvantages of the current approaches?

The questionnaire was sent through e-mail to the team responsible for the integration among systems at the university. At that time, this team included two employees. During the analysis of the answers, they were personally interviewed to enlighten further their answers. The answers were grouped so that it was possible to establish the chronological order of the facts and for the gathered information to be checked.

Step 3: Analysis of the gathered information on the evolution of the systems integration at the university. We performed a discussion on the information found and suggestions were made for possible improvement.

IV. Results and Discussion

In this section we present the software and services that are integrated and an analysis of the characteristics of this integration, discussing advantages, disadvantages and technologies used.

A. Software and Services at the University

The University had until 2011, 24 different systems used in the management of resources and services, not all of which were integrated. The reason for this lack of integration was mainly that some of those systems were only used within some departments. Hence, emphasis was given to those that were already integrated or those under integration until 2014. These systems were grouped according to their goal, as follows:

1. Academic: software systems whose main goal is to manage data on courses, subjects, professors, students, research, student assistance and extension and cultural activities;
2. Administrative: systems whose main goal is to help manage personnel and/or planning and management at the university; and
3. Support: systems used to manage users and Access control.

The academic systems are:

- CPPD: System that calculated professors promotions and which was used until 2011;
- SIG: Academic control system. It is used by several units for several purposes, including candidate convocation, student frequency control, curricular grids, teaching semesters, enrollment renewal processes, grade registration and documents emission (such as student history and declarations);
- Pergamum: Integrated library system that can be used to managed the book inventory and other activities with the university library;
- SIGAA: System that helps control research and intellectual production. Starting in 2015, it has been used also to control graduate courses and students.

The administrative systems are:

- SIGRH: System used to all activities related to personnel, such as vacations control, retirement caculation, frequency control, public selections, courses and functional evaluations;
- SIPAC: System that integrates all material and maintenance requisitions, purchase management, public purchases, properties, contract and agreements.

The support systems are:

- CIN_Cadastro: Support system that controls the photos on all personnel identification cards, emails from employees and researchers, electronic lock control and other activities;
- SCIN: System used to open and manage requistions to the IT support staff;
- RV3Acesso: System to control the Access to the university restaurant, library and student housing;
- HCS: Digital frequency control system, also helps manage fingerprints and frequency control;
- Vehicles: Control system for all the cars that could Access the campus. Used until 2010.

The graph with the need for data integration between systems can be seen in Figure 1. For instance, the HCS system needs data on Technical Employees stored in the SIGRH system.

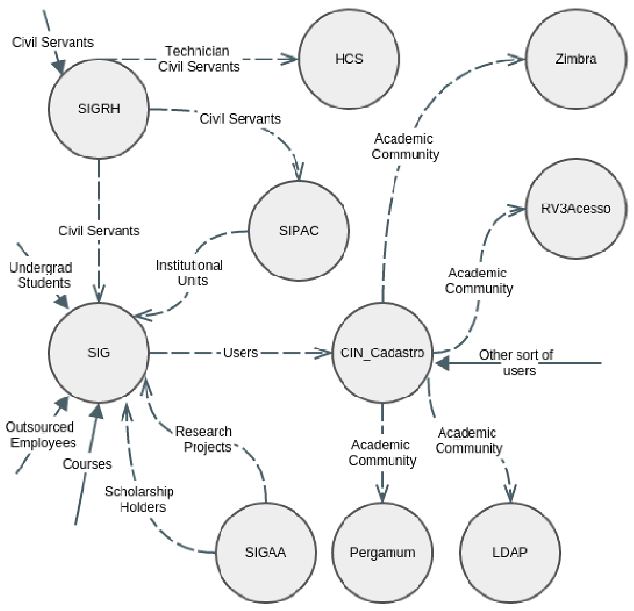

Figure. 1. Data flow between Integrated Systems

Table I presents the softwares described above with their implementation languages, the database management system (DBMS) they use and their platform. For instance, the SIG system was developed in PHP, uses MySQL as DBMS and is used through the Web. Actually, CSI developed all the systems in PHP and licenses were bought to use Pergamum, HCS and RV3Acesso.

Besides these software, the University provides network service, such as e-mail (Zimbra[3] - Message System and Collaboration Suite) and wireless network to the whole academic community. There is also a LDAP[4] institutional directory for user authentication.

The information systems and network services are installed in different servers with different operating systems (OS). For instance, the Zimbra server uses the Linux Red Hat OS and the LDAP is installed in a server with Fedora Linux. These heterogeneous environment increased the difficulty of the integration because each OS has specific and possibly different characteristics and configurations.

During this study, it was possible to identify systems in 7 different servers. Out of these servers, one of them (server1) used RedHat OS, four of them (server3, server4, server5 and server7) use Fedora Linux and two (server2 and server6) use the Windows 2003 Server OS. These servers are identified in Figures 3 and 4, together with the information systems and network servers that are installed on them.

The University systems and services share data such as user names and e-mail addresses. All the systems depend, on a different scale, on the data managed by CIN_Cadastro, SIG and SIGRH.

TABLE I
SOFTWARE SYSTEMS AND THEIR RESPECTIVE TECHNOLOGIES

System	Language	Database	Platform
CPPD	PHP	MySQL	Web
SIG	PHP	MySQL	Web
Pergamum	Delphi/Java	SQL Server	Desktop/Web
SIGAA	Java	PostgreSQL	Web
SIGRH	Java	PostgreSQL	Web
SIPAC	Java	PostgreSQL	Web
CIN_Cadastro	PHP	MySQL	Web
SCIN	PHP	MySQL	Web
RV3Acesso	-	MySQL	Desktop
HCS	-	MySQL	Desktop
Veículos	PHP	MySQL	Web

B. *Historical Analysis of the Inetgration Approaches*

After analysing the answers to the questionnaire, we could group the facts in chronological order and identify four phases, as shown in Figure 2: 1) Pre-Integration, the scenario before the first integration in 2006; 2) Phase 1: period between beginning of 2006 and beginning of 2009; 3) Phase 2: period between the beginning of 2009 and June/2014; and 4) Phase 3: period between June/2014 and October/2014. Each phase presents different approaches to integration and represents an evolution in terms of security, scalability and flexibility of the integration between systems and services.

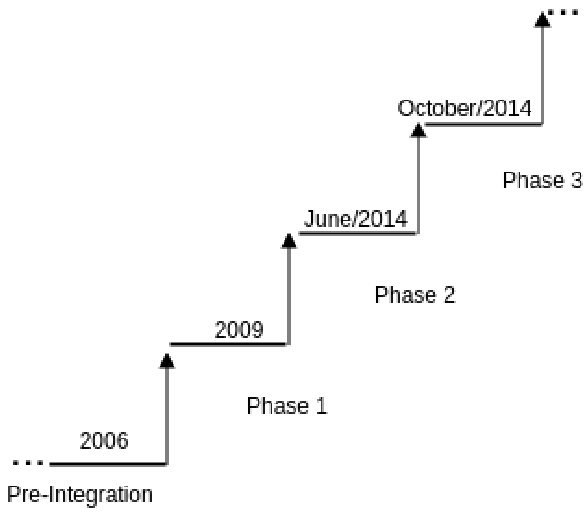

Figure 2. Phases of the Systems Integration process from 2006 to 2014.

1) *Pre-Integration*

Until the beginning of 2006, no systems were integrated in an automated way, that is, all the data transfers between systems were made manually. Even the e-mail accounts were created manually at the Zimbra server at the time, without information such as address, department, position or even the user name. Hence, it was decided that there was a need for the creation of a single repository to share this information.

Therefore, the database BD-UF was creating using the DBMS MySQL to centralize information such as login, e-mail, name, address, telephone, student courses and employees places of work. After the creation of this database, non institutional information systems (used only in a few units) start to seek this updated information in the central database.

2) *Phase 1*

The first integration step in the university began with the sharing of information among the CIN_Cadastro, SCIN, Veículos, RV3Acesso, LDAP and Zimbra systems.

The shared information were: name, address, position and place of work for all employees. The CIN_Cadastro system manages users and the data from this system were replicated in LDAP and used to create the institutional e-mail inside Zimbra.

The data for employees and students were replicated in the RV3Acesso system that controlled access to the university restaurant and the library. The SCIN and Veículos also used replicate data in the databases to authenticate and manage

[3] https://www.zimbra.com/
[4] LDAP: Open application protocol that allows for hierarchical organization of network resources (http://www.hardware.com.br/termos/ldap)

users when they called support and drove their vehicles to campus, respectively.

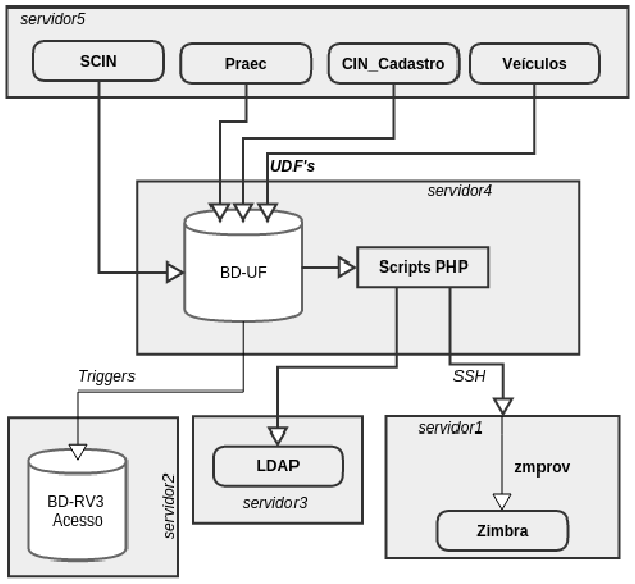

Figure 3. Initial integration among systems

The integration approach was based in the database. If both applications used MySQL, the integration was made using the tools from the DBMS itself, such as triggers, functions, procedures, views e events. For the RV3Acesso system, it was used the engine Federated[5], which shares data through the creation of virtual table that receives data from the original table, was used.

For the other systems that participated in the integration, the approach was to use User-Defined Functions[6] (UDF). UDF are functions developed in C and compiled inside the DBMS that can be used as native functions of MySQL. Using this resource, it was possible to automate the databases updates. The moment the tables were updated, the triggers related to them were automatically started and would then invoke the UDF that would call PHP scripts that performed the integration with other systems. Figure 3 shows a diagram of this initial integration.

PHP was chosen because the professionals that developed the initial integration mastered it. According to the information gathered through the questionnaire, this was the sole criterion for the adoption of PHP.

According to the approaches centered in the organization and its processes [13], this approach could be classified as *Data Replication*, because data is replicated in different bases (databases, LDAP and Zimbra) in an automatic process executed by the PHP scripts.

Integration with Zimbra was very complicated. Until its version 5.0, this system had neither a communication API nor the execution of remote commands, which caused a lot of difficulty to the integration. As illustrated by Figure 3, the script PHP invoked by the UDFs needs to establish a SSH connection with the server where Zimbra was installed in order to execute commands from the tool *Zmprov*. This tool offered the possibility of executing direct commands in Zimbra. With this chaining of operations, error detection and exception handling became a very complex task.Other systems such as Pergamum were updated manually.

Using UDFs brought the feeling that the integration was instantaneous, because the updates were executed as soon as a trigger began. Even though this approach met its goals, the UDFs were difficult to maintain because were programmed in C and compiled inside the DBMS. Using them also made it more difficult to find errors when they occurred.

3) *Phase 2*

After the first integrations, some things changed. A new management system (SIG) began to be developed and progressively aggregated functionality from older systems, which were afterwards excluded from the integration. This system, as well as Pergamum, became part of the integration. Until June/2014, the systems participating of the integration at UFLA were: SIG, CIN_Cadastro, Pergamum, RV3Acesso, Zimbra and LDAP.

With the arrival of SIG and its rapid growth, a new series of problems arose. The growing use of this system made the tables to be updated more frequently, increasing the number of times the triggers were started and invoked the UDF. The growing demand for UDF overloaded the system. At the time, one update took from 3 to 5 seconds to be propagated to Zimbra and LDAP. These systems were integrated using a chain of layers, making it more difficult to detect problems at each step, besides being more sensitive to network overloads.

The solution for this situation was to put the SIG database (BD-SIG) at a dedicated server. Once done, an asynchronous copy of this database was configured at the same server of BD-UF, and queries and reports that required further processing were directed to this copy. Hence, the production server of BD-SIG was less overloaded, decreasing the problems with the UDF that were too slow when the BD-SIG was overloaded.

Even when the problems with the UDF were decreased, an alternative for them was still sought. At that moment, Cron (a tool that allows for the scheduling the execution of tasks in UNIX-line operating systems) was deemed an interesting alternative. The option for Cron brought advantages as easier configuration, usage and maintenance. The calls to PHP scripts began to me made directly, without resorting to UDFs. The disadvantage of this approach was that data update in the other systems stopped being made instantaneously. Nevertheless, this was not considered a problem, for the systems were update in up to 10 minutes after the initial change was performed.

The approach using Cron was partially implemented.

[5] http://dev.mysql.com/doc/refman/5.5/en/federated-storage-engine.html
[6] http://dev.mysql.com/doc/refman/5.1/en/adding-udf.html

Critical systems, such as LDAP and Zimbra did not use this approach. Focusing on SIG, the integration in this phase happened as follows (and as illustrated by Figure 4):

1. A change in SIG triggers the data modification at the BD-SIG;
2. Original BD-SIG has its data replicated to a copy of the database in another server;
3. *Triggers* at the BD-SIG copy add the changes in the update buffer of the BD-UF, a table that stores the updates that must be performed, containing data such as system and records of the data that is going to be propagated to other integrated systems;
4. An event of MySQL is triggered every 10 minutes at BD-UF to generate specific records for systems update, defining which system must be updated and on which record of BD-UF this change must be based;
5. Cron executes PHP scripts, each one of them relating to a system that must be updated. The scripts get the data from the buffer table at BD-UF and propagate them to BD-Pergamum and BD-RV3Acesso;
6. PHP scripts are executed also from UDFs at BD-UF to propagate the updates to LDAP and Zimbra.

Even though the UDFs started to be less used, the Phase 2 approach still used this technology to integrate critical services, such as LDAP and Zimbra. Just like phase 1, the phase 2 approach also reached its goal, which was the integration of the university systems. Nevertheless, the addition of new systems to the integration such as it was done in this Phase still presented some complexities, such as the development of the UDFs and their maintenance.

Even with the partial substitution of UDFs by Cron, the integration was neither scalable nor easy to monitor. There was no clear policy or guideline on how to add new systems to the integration. Monitoring the integration was possible only through mechanisms programmed at the BD-UF, which sent e-mail when there was a failure such as deadlocking of the record generating procedure.

The growth of the university brought new and even bigger demands and the addition of new systems to the university environment was imminent, making the integration a task which would become progressively more complex.

1) Phase 3

In phase 3, the UDFs continued to function in spite of their limitations on performance and maintainability. The server OS update where the BD-UF was installed, as well as the version update of the MySQL DBMS made the UDFs unstable, because of the incompatibility between data types due to names modification. With this instability, the already known problems with exception handling and error treatment arose. Hence, the integration architecture was reconceived. Other integration technologies were studied to propose a new approach that would fulfill the needs of high scalability and maintainability. SOA was then a promising alternative, bringing the possibility to create a new quality integration and with mechanisms that could be monitored.

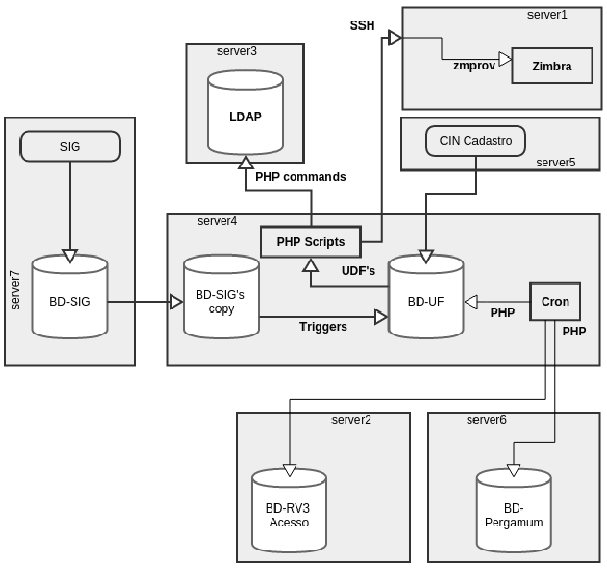

Figure 4. System integration between systems at the university at phase 2

This strategy intended to guarantee a significative advantage over the previous approach. Some decisions were taken on security, programming language, service/provider relationship, protocol and approach:

- Security: this is an item of the utmost importance, given that the integration manipulates important private data such as name, CPF (personal register id), passwords and addresses of thousands of people. Hence, we needed a security mechanism that restricts the access only to authorized persons and systems. Therefore, SOAP was chosen because the client needs to know the operations that are authorized to perform. Besides this characteristic, we also thought about an implementation that limits the access to services to authorized tokens, accessing from specific IP addresses;
- Programming Language: it was decided that the integration architecture would be developed in PHP, due to the familiarity of most of the staff with this language, besides the fact that it has native and non native modules that makes it easier to use SOAP. The language performance was not a criterion used to choose it, given that it would not overload the

database like the UDFs did;

- Relationship Provider – Service: we discussed this relationship in numerical aspects, in which the following question was asked: "would it be more interesting an approach that had a single Server with multiple services or one with many services that provided each few services?". Our option was for an architecture with many providers and few services that would make it the system more scalable, isolating the integration per specific system. This option would also make it easier for integration maintenance, because when one server was down, no other would be affected;
- Protocol: together with SOAP, we defined that the service call return would be a *Javascript Object Notation* (JSON) object made of four atributes: 1) ID: this would carry the message and the system IDs; 2) the message itself; 3) a type, like "SUCCESS", "DB ERROR"; and 4) a system attribute, that would carry the name of the requested server;
- Approach: with a vision centered on implementation, we chose the "Composed Application" approach [17], because the applications would be integrated using SOAP, which is an API. As to the strategies, given our focus on the organization and its processes, this is the "Service Oriented Architecture" [13], because the consumers (systems) of the integration were designed to consume functionalities through services, which configures a Service Oriented Integration.

Based on these decisions, we came to the architecture depicted in Figure 5.

This integration works in a similar way to the other approaches, because the initial trigger is an update in the SIG or in systems connected to the BD-UF. Based on that, Cron executes the service consumers that identify updates for their respective systems directly from the BD-UF buffer table. These consumers invoke the update services that bring their systems up to date.

All the activities performed by the service providers and the consumer are saved in a log table. This is necessary because there is a communication network between providers and consumers that may fail.

In the Phase 3, some aspects of the integration were facilitated. As an example, there was an update of the version of Zimbra that began to provide access to its functions through Web Services. Nevertheless, some complications arose, such as the addition of BD-SIPAC using the PostgreSQL DBMS, different from the other DBMS used until that moment.

With the development of services and their consumer, we came to a basic framework base on which any new service or consumer can be developed. This fact creates a high scalability, one of the essential and desired characteristics for the integration at the university.

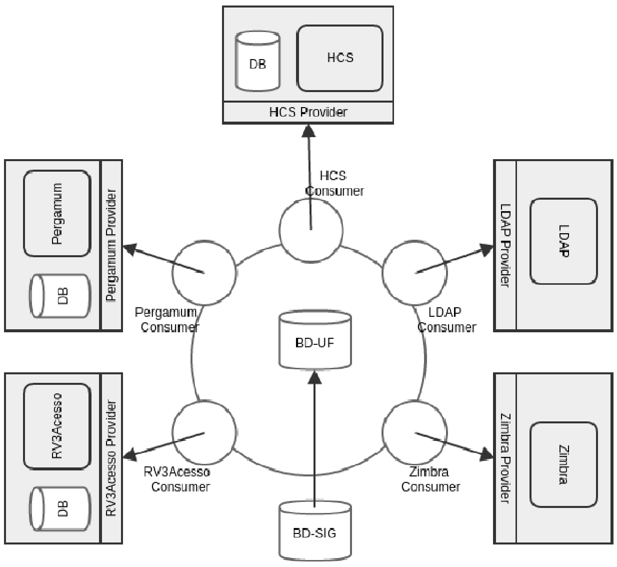

Figure 5. System integration between systems at the university at phase 3.

Besides, the interoperability brought by SOAP also helped integrate with external systems, that is, systems that use the University's web services to access specific functionalities. This interoperability is an important feature, because it allows the interconnection of software systems developed in different programming languages. For instance, a consumer built in Java can connect to services developed in PHP.

In order to organize the providers, we created two categories: public and private providers. This categorization allows higher control in terms of security because private providers are used only by the integration systems and public providers provided data for systems outside the integration, which allows us to isolate the private providers. This isolation is interesting, because consumers of public services are not aware of the existing private services and for the operational complexity is hidden from clients.

In Figure 6, we can see that the private provider is encapsulated by the public provider; the calls for the public consumer first pass through the public provider before being directed to the private provider.

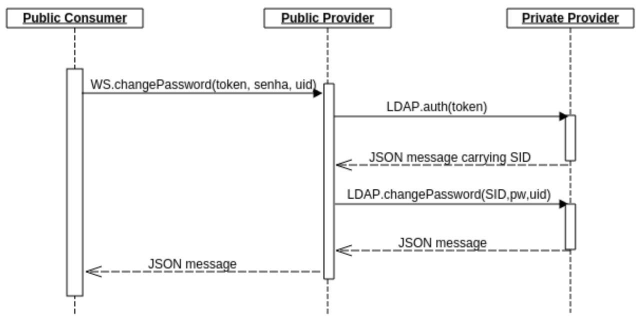
Public Consumer
Public Provider
Private Provider
WS.changePassword(token, senha, uid)
LDAP.auth(token)
JSON message carrying SID
LDAP.changePassword(SID,pw,uid)
JSON message
JSON message

Figure 6. Call flow between a consumer and a public provider.

The calls between consumers and services are made in three steps (as shown in Figure 6):

1. Consumer calls the authentication service sending its token;
2. Service verifies that the received token exists and the originating IP is in the list of acceptable IPs to access this specific token;
3. Consumer receives a message formatted in JSON. The attribute *type* defines the message. If it is “SUCCESS”, the atribute *message w*ill be a string that must be the first parameter in the next calls to the service in use. This string is a Session ID (SID). Otherwise, the attribute *message w*ill be an explanatory message related to the authentication problem.

In Figure 7, we present the call flow and the return values using the LDAP provider, exemplifying the use of the services *auth* and *changePassword* to authenticate and change the password.

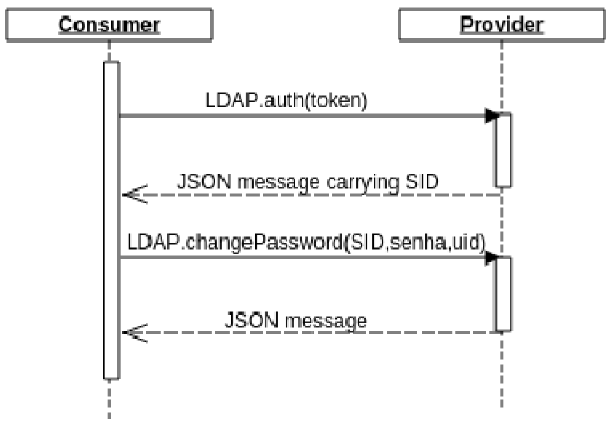

Figure 7. Call flow between consumer and provider.

The call to public services are made directly to the functionalities, similar to the calls for private services, but sending the token instead of the SID. As part of this work, we develop a management system to monitor the integration mechanisms. This monitoring system includes a record of service providers, tokens, token attachment to IPs and services and a dashboard with information on service execution in the last hours, comparison of service execution in the last two weeks and data from the last executions of each service.

Three of the graphics present at the dashboard are illustrated in Figures 8, 9 and 10. Figure 8 presented the graphic of the management system that shows the number of service calls for each system in the last 24 hours. For instance, between 18 and 19 hours of the day previous to the graphic visualization, the Pergamum provider had around 1300 service calls performed by its private consumer.

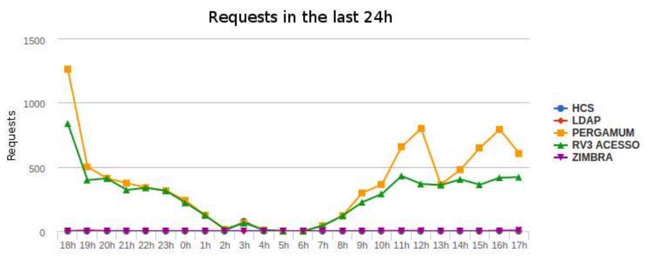

Figure 8. Number of requests responded by each service per hour.

Figure 9 presents a pie chart that allows the comparison between the number of calls from each system. For instance, the total of calls from the Pergamum provider represent 55.2% of the calls total, while the total of accesses to the RV3Acesso provider represents 43.4%. These two systems use information on employees and students at the university. Hence, whenever any data is inserted or changed in the data concentrator systems SIG-UFLA and Cin-Cadastro, it must be propagated to them. This propagation is performed with calls from the private consumer to its provider, resulting in a high number of calls.

Figure 10 shows a graphic that allows for comparison of the proportion of failure and success of web service usage for each system. For instance, provider HCS had a 97.04% success rate and a 2.86% failure rate when executing its services. The HCS system manages employee data for frequency control and these failures indicate that the provider found some inconsistency on the data sent by the consumer.

In order to illustrate the importance of the dashboard during the development of this work, the team that takes care of the integration observed in the graphics a non expected amount of service calls and could then investigate the causes of this behavior and correct it in an efficient and effective way.

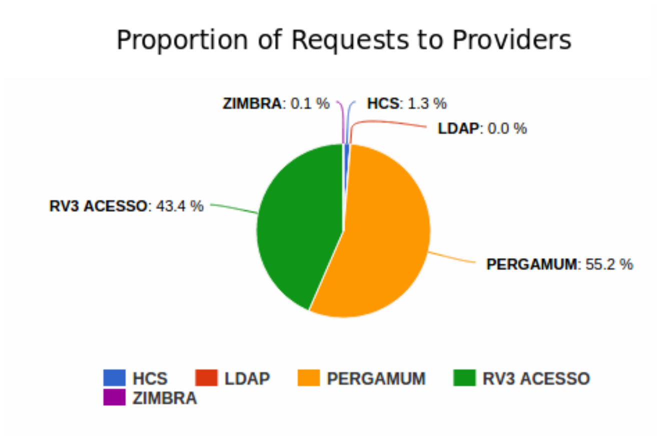

Figure 9. Global comparison between integration providers.

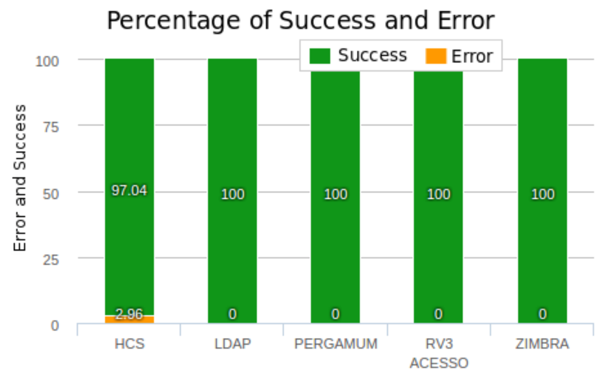

Figure 10. Graphic of a global comparison between integration providers.

C. Comparative outline

In Table II, we can see a comparative outline of the approaches used in the studied phases, showing the advantages and disadvantages of each one of them. For instance, the approach "Data replication (Triggers, Functions and Procedures)" has the advantage of its instantaneous nature, but the disadvantage of depending on a DBMS technology.

All studied approaches have advantages and disadvantages and they must be understood based on the context of the university. This means that depending on the technologies, the amount of involved systems and services, an approach may be more adequate than others. In the context of UFLA, the approach using SOAP and Web Services is the most adequate, because it allows more scalability and flexibility, given the expansion of the integrated software and network services.

TABLE II
OUTLINE OF THE APPROACHES

Approach	Advantages	Disadvantages
Data replication (*Triggers, Functions* e *Procedures*)	Feeling that it happens instantaneously	Depends on the technology of the database management system.
UDF	Feeling that it happens instantaneously and integration with Technologies outside the DBMS.	Depends on the technology of the database management system; Fail recovery; Speed.
Master/Slave	Feeling that it happens instantaneously	Depends on the technology of the database management system and of the network. May generate inconsistent records in different servers.
CRON + Scripts PHP	Maintainability.	Updates are not instantaneous.
Web Services	Scalability; Function isolation.	Depends on the external network; Depends on technology
SOAP	Well-defined services.	Requests and responses in a standard format containing too much information.

II. CONCLUSION

The IT environment in medium and large-sized organizations is complex due to the heterogeneous platforms and their size. For the consistency of the data manipulated by those systems, the ideal is for them to be integrated, working as one.

In academic environments, the scenario is similar. Several systems are used with many different goals. In this paper, we present a case study over the integration of systems and services within a University, involving 11 systems and 2 services. In this study, it was possible to identify 4 integration phases, from the moment when there was no integration, at the beginning of 2006, to the development of an integration architecture and its implantation, in October/2014.

In phase 3, a service oriented architecture was designed for the integration of the university systems. Together with this architecture, a monitoring and management interface for services and integration permissions was developed.

This study of the history of the integration at the university allowed us to understand better its evolution, its initial needs, the advantages and disadvantages of each of the adopted solutions. This understanding may help avoid adopting approaches that were proven to be inefficient in the context of information systems and network services integration at the university.

The integration of systems and services, especially in academic environments, is very interesting, because it deals with administrative systems (personnel management, purchases, public biddings, stock control), academic systems (course and student management, enrollment, document emission, student history, research projects) and network services (e-mail, centralized authentication, wireless network) that are part of the daily life at the university, besides integrating with non institutional systems.

As future work, we suggest:

- Deeper study on security aspects in Web Services;
- Identification of points common to education institutions, in order to provide a standard for integration in this sort of environment;
- Proposal for meeting the Federal Government's e-Ping[7] interoperability standards.;
- Proposal for integration with Federal Government's structuring systems, such as: SIAPE, SIGEPE, Comprasnet.

REFERENCES

[1] Andersson, V. O.; dos Santos, R. T.; Tillmann, A. L. C. & Noguez, J. H. S. *COBALTO Webservice: Solução para consistência de informações.* Resumo Publicado na VIII Workshop de Tecnologia da Informação e Comunicação das IFES (2014**).**

[2] Barros, F. *Mercado de software nacional vai crescer 400% em 10 anos.* Disponível em: <http://convergenciadigital.uol.com.br/cgi/cgilua.exe/sys/start.htm?infoid=32006>. Acessado em 10/09/2014, *(*2012).

[3] Cerami, E. *Web services essentials: distributed applications with XML-RPC, SOAP, UDDI & WSDL. O'Reilly Media, Inc.,* (2002)

[4] Costa, C., Melo, A. C., Fernandes, A., Gomes, L. M. & Guerra, H. Integração de Sistemas de Informação Universitários via Web Services. Actas da *5ª Conferencia Ibérica de Sistemas y Tecnologias de Información, p.* 290-295 (2010).

[5] Coulouris, G., Dollimore, J., Kindberg, T. & Blair, G. *Sistemas Distribuidos-: Conceitos e Projeto.* Bookman Editora. (2013)

[6] De Mello Jorge, M. H. P., Laurenti, R. & Gotlieb, S. L. D. *Avaliação dos sistemas de informação em saúde no Brasil.* Cad. Saude Colet ***18****, 07-18* (2010).

[7] Degan, J. O. C. *Integração de dados corporativos: uma proposta de arquitetura baseada em serviços de dados* Unicamp - Universidade Estadual de Campinas, (2005)

[8] DGTI. *Plano Diretor de Tecnologia da Informação 2011/2012* Universidade Federal de Lavras (2011).

[9] Do Carmo, B. & Almeida, D. Uso de Sistemas de Informação geográfica na avaliação da Microbacia do Ribeirão das Alagoas, Conceição das Alagoas, Minas Gerais *Publicatio UEPG-Ciências Exatas e da Terra, Agrárias e Engenharias, 19,* **9.** *(*2013).

[10] Fielding, R. T. *Architectural styles and the design of network-based software architectures.* University of California, Irvine (2000).

[11] Gerhardt, T. E.; Pinto, J. M.; Riquinho, D. L.; Roese, A.; Santos, D. L. d. & Lima, M. C. R. d. *Utilização de serviços de saúde de atenção básica em municipios da metade sul do Rio Grande do Sul: análise baseada em sistemas de informação.* Ciência & Saúde Coletiva, SciELO Brasil, *16*, 1221-1232 (2011).

[12] Hensle, B.; Booth, C.; Chappelle, D.; McDaniels, J.; Wilkins, M. & Bennett, S. *Oracle Reference Architecture - Service-Oriented Integration*, Release 3.0.*Oracle,* (2010)

[13] Hohpe, G. & Woolf, B. *Enterprise integration patterns: Designing, building, and deploying messaging solutions.* Addison-Wesley Professional, (2003).

[14] Intel. *Computadores ficaram 61% mais baratos nos últimos 10 anos. <Disponível em: http://newsroom.intel.com/docs/DOC-3946>. Acessado em 26/08/2014,* (2013)

[15] Laurindo, F. J. B.; Shimizu, T.; Carvalho, M. M. d. & Rabechini Jr, R. *O papel da tecnologia da informação (TI) na estratégia das organizações.* Gestão & Produção, SciELO Brasil, ***8****,* 160-179. (2001)

[16] Marin, H. d. F. Sistemas de informação em saúde: considerações gerais. Journal of Health Informatics, ***2****,* (2010)

[17] Martins, V. M. M. *Integração de Sistemas de Informação: Perspectivas, normas e abordagens.* Universidade do Minho - Guimarães - Portugal, (2005)

[18] Nascimento, A. M.; Luft, M. C. M. S.; Araujo, G. F. d. & Dacorso, A. L. R. *Implantação de Sistemas de Informação em uma secretaria estadual*

Revista Pensamento Contemporâneo em Administração, **5**, 66-82 (2011)

[19] OASIS, Organization for the advancement of structured information standards. *Service-Oriented Architecture.* Disponível em: /docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>. Acessado em 15/09/2014*, (*2006).

[20] Pressman, R. S. *Software engineering: a practitioner's approach.* McGraw-Hill, McGraw-Hill, (2001)

[21] Ramos, Célia. M. Q. *Sistemas de informação para a gestão turistica.* Revista Encontros Cientificos-Tourism & Management Studies, Escola Superior de Gestão, Hotelaria e Turismo, 107-116 *(*2010**).**

[22] Technologies, IOWA. *Web Services Definition.* <Disponível em /www.w3.org/2001/03/WSWS-popa/paper13>. Acessado em 21/10/2014., (2001)

[23] Turban, E., Leidner, D., McLean, E. & Wetherbe, J. *Tecnologia da Informação para Gestão: Transformando os Negócios na Economia Digital* 6ed. *Bookman,* (2004)

[24] UFLA. *Novos cursos de Engenharia da UFLA terão Área Básica de Ingresso.* Disponível em /www.ufla.br/ascom/2014/05/30/novos-cursos-de-engenharia-da-ufla-terao-area-basica-de-ingresso/>.Acessado em 24/08/2014 *(*2013)

[25] UFLA, ASCOM. *Números.* Disponível em: /www.ufla.br/portal/institucional/sobre/numeros/>. Acessado em 24/08/2014 *(*2014)

[26] Valipour, M. H.; Amirzafari, B.; Maleki, K. N. & Daneshpour, N. *A Brief Survey of Software Architecture Concepts and Service*. Oriented Architecture. *JASIS, (*2009)

[27] Van den Heuvel, W.-J.; Zimmermann, O.; Leymann, F.; Lago, P.; Schieferdecker, I.; Zdun, U. & Avgeriou, P. *Software service engineering: Tenets and challenges.* Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, 26-33 (2009)

[28] Vidor, A. C.; Fisher, P. D. & Bordin, R. *Utilização dos sistemas de informação em saúde em municipios gaúchos de pequeno porte.* Revista Saúde Pública, SciELO Public Health, **45**, 24-30 (2011).

[29] Zur Muehlen, M.; Nickerson, J. V. & Swenson, K. D. *Developing web services choreography standards — the case of REST vs. SOAP.* Decision Support Systems, Elsevier, **40**, 9-29 (2005)

[7] http://www.governoeletronico.gov.br/acoes-e-projetos/e-ping-padroes-de-interoperabilidade