Architecture Proposal Model for SOA in Universities Educational Software

Miguel A. González Serrano
Universidad Autónoma de Madrid
Spain
miguela.gonzalez@estudiante.uam.es

Diana Pérez-Marín
Universidad Rey Juan Carlos
Madrid, Spain
diana.perez@urjc.es

Miren Idoia Alarcón
Universidad Autónoma de Madrid
Spain
idoia.alarcon@uam.es

Abstract - **Since the first models of architecture were proposed by Zachman in the 80s, several authors have developed their own architecture models and methodologies, but most of them lack of details, and they are far away from current technology, business requirements and results needed in real projects. Particularly there are not so many real Service Oriented Architectures (SOA) in educational environments such as universities. Therefore, in this paper, the goal is to propose and validate a new complete architecture model for SOA methodologies according to current technology and business requirements that could be used in a real University environment. To do that, new types of services and the categorization model called Dimension Model are described, and the results of the proposed architecture model in 4 real-life projects. We expect that the reduction of the implementation time in 33% and testing time in 31% in business domains encourages universities to start incorporating SOA methodologies in their corporate software.**

Keywords - **SOA; E-learning; Architecture; Dimension Model; Legacy Systems**

I. INTRODUCTION

Since Zachman [1] proposed his model of business architectures, several authors have developed their own architecture models and methodologies. However, most of them lack of details, and they are far away from current technology, business requirements and results needed in real projects [2]. With the invention of web services [3], as a connecting method for distributed applications, programming oriented to services has become standard and the most common way to develop architectures [4].

Service Oriented Architecture (SOA) has been created for the treatment of complex distributed applications. Architectures that have service orientation as main implementation key require loose coupling, descriptions of interfaces independent of the software platform, and the use of standards in order to facilitate the consumption, modification, construction and distribution of services of applications in a flexible way [5-9].

However, in order to take full advantage of SOA benefits, architects should be provided with models to help them dealing with the underlying complexity related to features such as multi-layer implementation, technological diversity, specific information, security requirements, etc. This context makes the definition of new architecture components such as Content Management Systems (CMS), Document Management System (DMS), Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), or Business Process Management systems (BPM) necessary for creating an architecture based on services [10-13].

In this paper, a new architecture model is proposed to help SOA architects and complement SOA methodologies to implement, test and deliver projects into a SOA. The proposed architecture model is based on the reuse of legacy knowledge, taking into account the real business requirements adapted to the current technology. It is based on a categorization of services published in our previous work [14], and the Dimension Model [2].

It is our insight that to reach a valid and useful SOA methodology, that takes full advantage of SOA possibilities, it is necessary to provide SOA architects not only with the list of types of services as we did in [14], or the next step that was completed in [2] of classifying these services in three dimensions, namely Taxonomic, Corporate and Architecture, but to go further and integrate both classifications together in a practical architecture that can be easily implemented.

The proposed architecture model helps a SOA architect to gather and organize the information needed to implement the SOA project from the point of view of both the technical and business teams. For the technical point of view, the implementation details are integrated in the model. For the business point of view, higher abstraction level information is provided to accomplish the business requirements, and it is also integrated in the model.

The proposed architecture model has been implemented in four case-studies of real SOA projects. The implementation time has been analyzed when the model is used regarding the use of other architectures, and, it has been detected an average reduction of 33% in implementation time and 31% in testing time when the proposed SOA architecture was implemented. We believe that this is because the team finds the information and the services faster and can understand the data better and without mistakes.

However, Universities have not adopted SOA architectures for their corporate software [15-17]. Just a few American universities such as the Duke University [18] and the University of Virginia [19] are examples of early SOA adopters. Both cases show benefits of using

SOA in their education, even when a horizontal architecture is not applied.

We believe that the use of the proposed horizontal architecture with the Dimension Model can help SOA architects when working in new environments such as educational corporate software. Moreover, it is our expectation than from the detailed description of the proposed SOA architecture and its results, managers of universities worldwide may start considering adopting SOA for their corporate software.

The paper is organized as follows: Section 2 reviews the context; Section 3 describes the architecture proposed; Section 4 presents four real-life case studies to validate the architecture proposed and how it can be applied to Education; and finally, Section 5 ends the paper with the discussion of the main conclusions and lines of future work.

II. CONTEXT

Service-oriented architecture (SOA) is currently the most discussed concept for structuring enterprise IT architectures [20-23]. Many papers discuss the main issues about how to create a Service Oriented Architecture in a big enterprise, but most of them are just focused on how to use Web Services and its main technological concepts such as types, messages, operations, services and bindings all defined into the WSDL [20], with old technological paradigms such as OOAD [12] and structured programming relating classical components such as interfaces or objects operations with the new ones used by SOA [24].

However, SOA is more than just new terms for the established software engineering concepts of component-orientation [2, 22]. Table 1 gathers the main progress made in SOA architectures models.

III. ARCHITECTURE

As can be seen from the review of the state-of-the-art, up to our knowledge, there is not a clear architecture model that can be used by SOA architects to implement what business needs with current technology. However, the need for such SOA architecture model is claimed since 2005 [39-41].

Huhns et al. [20] proposed a classification centered on Web Services technology, but it does not solve the need of a business needs centered classification. In our previous work, we provided a horizontal taxonomy of services from the global needs of an organization, extending Huhns et al. classification, and from our own experience of working in real-life SOA projects. It was published in 2012 in Spanish [2, 14], and it is explained in Section 3.1, as it is necessary to understand the proposed architecture in this paper.

In our previous work, we also noticed the need of classifying the services regarding questions such as: Who gives the service? What functionality is provided by my service?; How and where is my service located?. That led us to the Dimension Model, published in Spanish [2], and explained in Section 3.2, as it is also necessary to understand the proposed architecture. Finally, Section 3.3 brings together the taxonomy [2, 14] and the Dimension Model [2] in the proposed SOA architecture.

3.1. Web Services Taxonomy

14 types of web services were identified from the review of the state-of-the-art and the real-life experience on projects working as SOA architects are the following: security, Content Management System (CMS), Document Content Management (DCM), Customer Relationship Management (CRM), presentation, mobility, data integration and applications, adaptation logic or "multification", internationalization, business rules, governance and notification, audit of applications, back-office services, support to business processes. They are described in the following paragraphs:

-Security: Currently emerging new security models for SOA [42], makes clear a need for the creation of security services for all architecture applications and that arises with the use of market security standards. A clear application of a security service should solve the functions of authentication, identification, authorization, non-repudiation, encryption, Single Sign On (SSO) and audit applications.

-Content Management Services: Industry is developing standards like CMIS OASSIS which provides a generic interface for the exchange of enterprise content without the need to show the inherent content presentation layer. That is why a content service possibly based on CMIS as a standard should be considered for a SOA platform.

-Document Management Services: The importance of these technologies is evident in all technological reports that Forrester and Gartner [43, 44] publish annually. In this study, we have received the 2010 quadrant of Gartner stating that ECM platforms tend to be used as global infrastructure for one or more organizations. Thus, providing a model very close to the kind of architectures that SOA poses. That is why we are committed to architecture with a raised corporate documents management service as a business service accessible by all corporate applications that need it. This service must not only allow consultation and download of new documents, but also functionality related to versioning and editing of documents, in combination with the security services that offer the possibility of signed and electronic certification.

-Customer Relationship Management Services: Currently the business of an organization is clearly supported by the information these systems provide, enriching areas of business such as marketing, acquisitions and sales. For this reason, and as in the case of the document management and content, is important to consider a functional block in our proposal of enterprise SOA architecture approach to solve this type of business need through a set of CRM services integrated with the rest of proposed services.

TABLE I
ADVANCES TOWARDS ARCHITECTURE MODELS FOR SOA METHODOLOGIES

Reference	**Comments**
Yourdon et al. [25] established the bases for structured software design related to programs and systems.	The main lack of this study is that an architecture as a set of systems is never named. This paper tries to go ahead with the classical way of design.
Jackson et al. [26] makes an introduction of well-known software engineering concepts such as loose coupling or workflow related to SOA.	This proposal with the services classification and architecture proposal gives SOA architects elements to implement SOA projects taking into account the main SOA principles such as loose coupling or workflow business implementation and also provides a wider vision of business.
Lack of definition of a complete methodology and their artifacts are other limitations founds on authors such as Zimmermann et al. (2004) [28] with his SOAD method. The authors named the main principles of SOA without going into detail of how to fix the main issues of creating architecture. They considered the use of UML in SOA without providing examples or relationship between major SOA and UML diagrams. They did not perform taxonomies of UML-based services.	This paper defines a complete architecture proposal with artifacts (Dimension Model and Services Classification) to empower SOA architects and business users to create a complete design of a SOA configuration a big corporation such a University.
We discover a new way for structuring the business of an enterprise where business services are the basic unit which drives the IT enterprise architecture [29-31].	The main limitation of these approaches relies on the fact that they are too close to technical aspects. Our proposal is a business centered approach taking into account the technology concepts which will be used in implementation phase.
Other approaches try to establish concrete engineering methods and guidelines for constructing a real SOA [22, 32-33] they define SOA as "a paradigm for structuring the business of an enterprise and for structuring the enterprise IT architecture accordingly".	These authors in their definition of SOA are still far away from actual and concrete business. This is why in our proposal we introduce business concrete requirements such as CMS, BPM, and mobility of security.
perfSONAR (Performance focused Service Oriented Network monitoring ARchitecture) [34,35] presents SOA in a very concrete use case, perform multi-domain measurements without being limited to specific kinds of metrics in networks communications.	These proposals are very concrete cases of business architecture from a multi-domain point of view. But one limitation that we discover is that they do not consider the option to create more domains or layer in the architecture to escalate the architecture.
Another approach to a concrete business is the one proposed [36] in their papers. A model of a collaborative service is analyzed to identify its collaborative functions, that is, another very concrete business where SOA is applied without a generic architecture framework or method. These proposals present an architecture based on SOA that shows how to increase flexibility and scalability in comparison to traditional software engineering techniques.	This reference is a good example of how SOA is needed to be used in very concrete business needs, but in the other hand it is far away from defining a global framework of SOA. Our proposal is centered on a real implementation business
Other authors are focused on service inventories, taking into account perspectives of service providers and consumers in analysis and design phases [37].	Our proposal improves the perspectives of service providers and consumers in implementation and testing phases by the definition of the Dimension Model which shows a classification method with a wider point of view. Service classification taxonomy proposes another perspective to classify concepts, focused on the horizontal coverage of the organizations and their business requirements.
S-Cube (The European Network of Excellence in Software Services and Systems) presents and approaches to SOA design using concepts such as presentation layer or security services [38].	The presented service classification supposes a much more complete services taxonomy due to the fact that our proposal is centered on more business concepts than the ones proposed (presentation layer or security services).

-Presentation: Traditionally in the structured or linear programming, the "Front-end" or presentation layer has been included in the business layer. However, in applications that follows the OOAD paradigm this separation is evident with the creation of the concept of layers, providing the presentation layer as a separate entity from the business application element, and can even be maintained and developed by teams and phases of the life cycle of a software product. To take advantage of the main benefits of orientation to services we propose that an organization with a true SOA should be equipped with services purely for the presentation of applications that allow you to submit content in different formats and styles for different technologies. Some of the studies in state-of-art use concepts similar to presentation layer such as S-Cube [38].

-Mobility: With the arrival of the so-called SmartClient Phone devices in the last decade, corporations have mobility-related requirements. This type of architecture is completely separated from the traditional enterprise architecture, whether it is SOA-based or not, since a SmartClient Phone application runs in a terminal controlling and locally managing the user interface and integrating different systems through services (data and peripheral mobility). However, it is necessary that this architecture is based on layers or corporate services for reusing components already developed in the organization, especially if it is service oriented. For all these reasons we consider it necessary to have a corporate SOA-based architecture between their dedicated services and the production of information for mobility applications.

-Data integration of applications: This type of service is evident with the advent of the need of a common model for horizontal data in the entire corporation. This common model is accessed in situations where you want to translate the business entities to provide to the enterprise architecture a single system of data for all applications. A reduction of the complexity of translation of native data models from different applications is achieved through the reuse of transformations implemented the first time it is detected, and the need for translation between two business entities in different applications. It also determines the end of point to point integrations between applications with different business languages, and promotes the extrapolated model to external organizations wishing to share information or who should receive it from our architecture.

-Logic of adaptation or "multification": It is considered a new concept within the cataloguing of services and applications. Applications are traditionally divided into presentation, business and data logic layer [45]. However, given the growing need to adapt the same logic to various types of business targeted by corporate channels that manage different business areas, it becomes necessary to define the logic of adaptation. This concept allows the adaption of the business logic services to channel presentation devices avoiding the design the logic of the business user "on the screen" likewise decouples the "variable" part of a corporate channel and the presentation of the invariable (pure business).

-Internationalization: The subtype of adaptation to the language and aspects relating to the internationalization of the business and presentation of an application are also considered within the logic of adaptation of a channel. It is considered a subtype of the internationalization adaptation service where transformations between countries such as language, the currency used or business entities that are in different formats between countries are dealt with, such as bank accounts.

-Business rules: Corporate applications are subject to fast and relentless changes in the business, the management of business rules in an independent way to other components of the applications of an organization represents a very important requirement.

- Governance and notification: Once a corporate architecture is established based on SOA, it becomes very necessary to implement Governance functions for those services. It includes all rights to make decisions for the development, deployment and management of new services and monitoring and reporting processes to capture and communicate the results of Governance.

-Audit of applications: The main requirement of the application audit is the exploitation of online (near real time) operational information from corporate applications. This information is produced by the different applications of an organization regardless of the channel they are run. Note that organizations are mostly dividing its business channels and in doing that, business data is segmented by channels, Contact Center, Internet, Intranet…

-Back-Office services: As the types of services that are traditionally considered as the OOAD applications presentation layer has been analyzed, we must consider the traditional business or back-end layer. This type of services in SOA architecture would provide purely business functionality to various parts of the architecture that needed it, without taking into account aspects included in other services such as security for access, the "multification" or the internationalization which significantly simplifies the complexity of the business logic design and implementation.

-Business process support: These services are considered to cover a wide range of functionality demanded by corporate entities today. Support configuration and execution of events business, automating the execution of processes, support to data extraction from context of business processes and the crossing with other data for other processes for obtaining cross and disjointed information between different instances of processes are among the main contributions that offer this type of processes, etc.

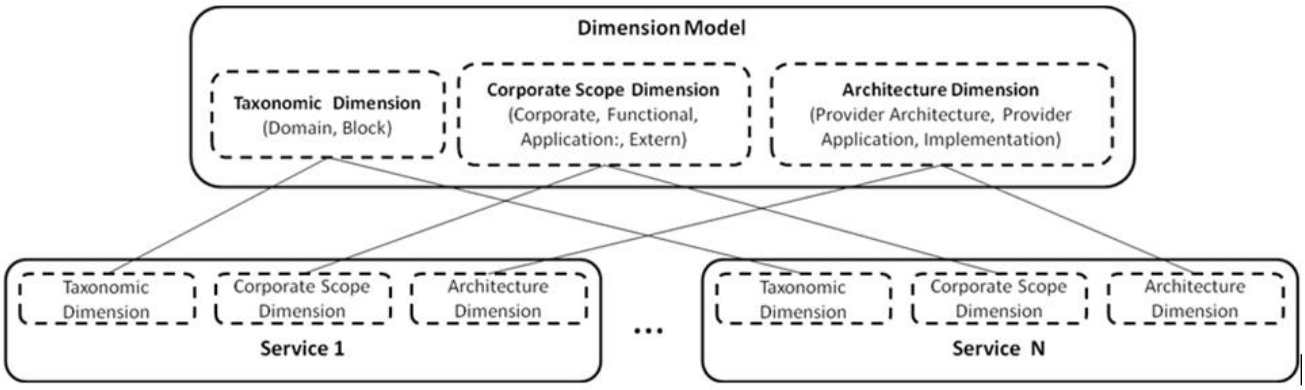

Figure 1: Dimension Model

3.2. Dimension Model

Once analyzed the types of corporate services that have been detected is necessary to achieve a taxonomy or categorization that could encompass them from different points of view [2]. A service can be seen from different perspectives such as the view of provider and consumer [37, 40]. Likewise in analysis services the following questions can help to discover the nature of a service responding to the following:

- Who gives the service?
- What functionality does my service have?
- How and where is located my service?

Approaches described by Ramollari et al. [46] and Jones [47] are different ways of analyzing the previously named points of view of a business service and its categorizations but with a low level of description. Questions analyzed by this section helps to define the model of categorization, which has been called "Dimension Model". Our model is based on dimensions, which are defined as features or quantities that serve to categorize on the basis of a set of values to a service, solving main issues detected in Jones and Ramollari works. To solve these problems it has been determined necessary for the categorization of our services the following dimensions:

- Corporate Scope Dimension.
- Taxonomic Dimension.

3.2.1 Corporate Scope Dimension

This dimension responds to the above question as "To whom do I give service?" or what is the same, who is the consumer of my service, supporting the consumer point of view [37]. The "Corporate scope" dimension indicated for a particular service, to which part of the organization that service gives coverage. With the possible values:

- Corporate: Service whose scope applies to the entire organization.
- Functional: Service that applies to a specific functional domain.
- Application: Services that give coverage to a given application.
- Extern: Service that gives or receives coverage from or to an external environment. Called virtual services are services that show a facade on the outside so it is categorized as external.

This dimension could be considered as a single attribute of a service registry in the SOA metamodel description, although with the point of view of a dimension this scope is not limited to one possible value but that the IT architects could describe it as a set of different corporative channels, areas or departments. A single service could have several corporate dimensions for its functionalities.

3.2.2. Architecture Dimension

This dimension meets the above question as "how and where is located my service?" or what is the same, who is my service provider, supporting the provider point of view [37, 43]. This is an issue that authors such Jones [47] softly consider in his approach. But from a global point of view of architecture, large organizations usually have several types of architectures (J2EE architecture.NET, architecture dedicated to integration based on an EAI architecture etc.) is necessary to specify who is serving that service. To do this, define three levels of abstraction that define the dimension.

- Provider Architecture: It indicates which corporate architecture provides the service in particular.
- Provider Application: It indicates within the framework of architecture previously defined which application is implementing the service.
- Implementation: It indicates which type of implementation of service is.

3.2.3 Taxonomic Dimension

The taxonomic dimension addresses where a service is located in the architecture and more specifically answers the question "What functionality does my service have?" The taxonomic dimension (see Figure 1) represents the different levels of organization that comprise the services which will give coverage to the applications and tools ranging from the more purely technological services to the more integrated in the business of the organization. This defines two levels of abstraction:

- Domain: First level of taxonomic categorization including what is known as block.
- Block: Second level of taxonomic categorization including what is called grouping.

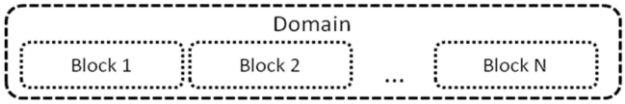

Figure 2: Taxonomic Dimension

As Corporate Dimension, taxonomic dimension could be considered as a single attribute of a service registry in the SOA metamodel description, although IT architects could describe it as a set of different domains in several Corporative dimensions (channels, areas or departments), creating a multidimensional description of a service.

3.3 Proposed SOA Architecture Model

The main goal of this SOA architecture model is to combine the services identified in Section 3.1 and locating them into the layers created by the Dimension Model (explained in Section 3.2) by performing the following steps:

- Design services for Corporative Layer.
- Design services for Business Support Layer.
- Design services Architecture Layer.
- Design services of other layers Extra of the Dimensions Model.

Each of these actions is composed by a set of steps that determines all the technical elements that should compose a service. At the beginning of this set of steps, the architects should make a revision of the existing services to reuse the existing ones. As we have seen in section 3.2, if they are not existing coincidences, candidates are passed to the design of the new services by following the next steps:

- Definition of service entities using the company data repository.
- Specification of Interfaces, messages and exceptions.
- Standardize service interface by following corporative or industry conventions.

The stop condition for these steps can be one of these two possibilities:

- The possibility of extending service design due to the arrival of new business features.
- The identification of interfaces between services and potential new features.

This services specification and categorization by locating services into layers must raise an architecture whose layers give room to all the services that we have analyzed and designed to locate them in a diagram that complies with the model of categorization. For this, it is necessary to determine the areas or categorical levels, such as the following:

- Business Domain: They are all those services that respond to directly related functionality with the core business of the organization.
- Support Business Domain: They are all those services that give coverage to the business without directly being part of it [48].
- Architecture Domain: They are all those services which are considered technicians more low level and bring uniformity to business applications from a technological level.

Note that there is room to create new layers of each customer needs, this is shown in Figure 3 by creating the Extra Domain layer. Figure 3 describes the domains that IT architects can use to organize their configuration of SOA, using blocks as a low level detail of services classification.

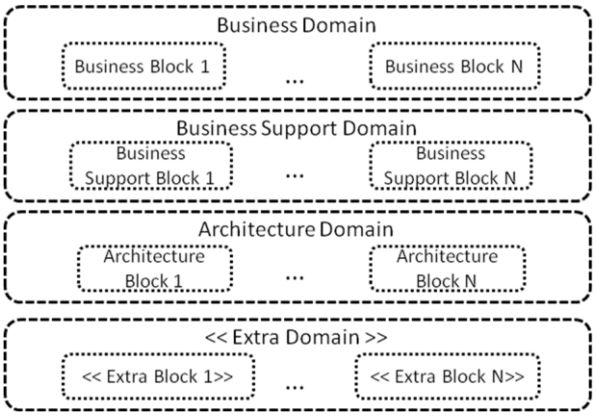

Figure 3: Architecture Domains

Figure 4 creates an important relation between taxonomic dimension and domains and its blocks of architecture areas, this relation will be used by stakeholders and IT architects for relating the principal functionalities included into the SOA configuration (services classification, section 3.3) and their point of view from the Dimension Model (section 3.4).

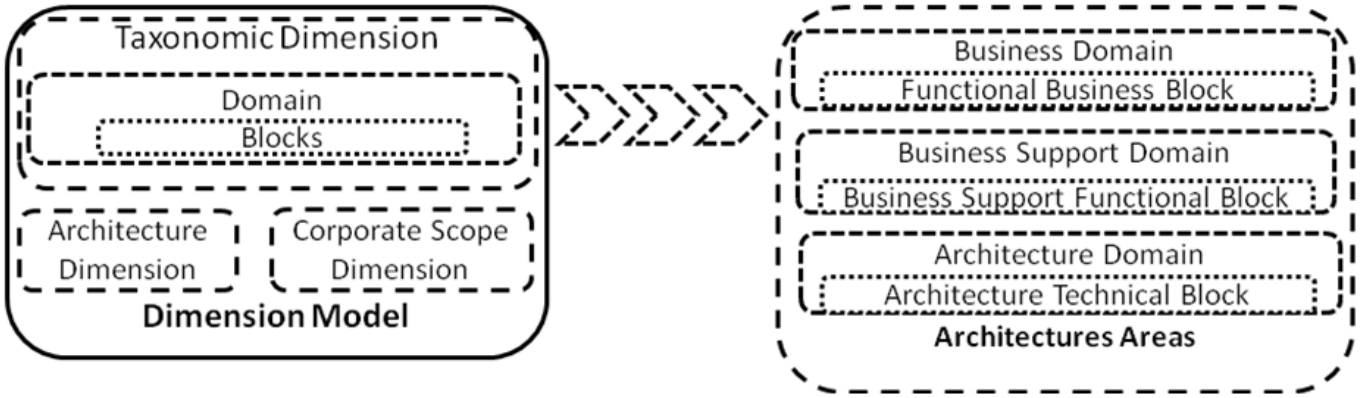

Figure 4: Relation between taxonomic dimension and architecture areas

Finally, as it is shown in Figure 5, the final purpose has been achieved, a single configuration of an architecture where all the services are well categorized according to technical and business needs. Figure 5 represents the result achieved with the execution of the lifecycles. The framework created in this method could be used by IT architects to obtain a real business needs point of view into their architectural designs. For business users this framework will suppose a representation of their functionalities from a semi-technical point of view providing them a better understandable architecture representation.

For the case of a University, following the Dimension Model shown in Figure 4, the taxonomic dimension keeps the main domains of the architecture: business domain, business support domain and architecture domain. Business Support Domain and Architecture Domain do not need to change because they are generic to any corporation. Regarding Business Domain, there are six new blocks that are introduced for education [16, 17], namely:

- **eLearning Management:** services centered on student logging on the courses, to answer questions, to take exams, to collaborate with other students, or any other activities that students can do in an e-learning setting.
- **Academic Management:** services centered on the management aspect of the courses, such as creating an appointment with the teacher to review an exam, to manage students' accounts and to publish the scores of each course and semester.
- **Course Management:** services centered on the management of the course contents, so that teachers or course designers can introduce text, multimedia activities, questions, answers, in general any learning object that can also be modified, shared, combined or deleted. Each course must have their resources associated, such as the time in which they going to be taught, the room in which they going to be taught in case that it is face-to-face, or if it is blended learning when there is no physical room available, the video conference links or no resource needed.
- **Career Management:** services centered on the management of the progress of the students after they have finished their studies, helping them with the choice of their work, the tracking of their evolution and future possibilities.
- **Study Plan Management:** services centered on the management of the study plans, convalidations, possibilities of combining them, coordination activities betweeen courses of the same study plan and with other study plans, teaching guides management, and their quality assurance progress.
- **Investment Project Management:** services centered on the research, such as to create research applications, follow their evolution, budget management, paper publishment, and résumé generation.

Figure 6 shows the proposed SOA architecture adapted to the case of a University.

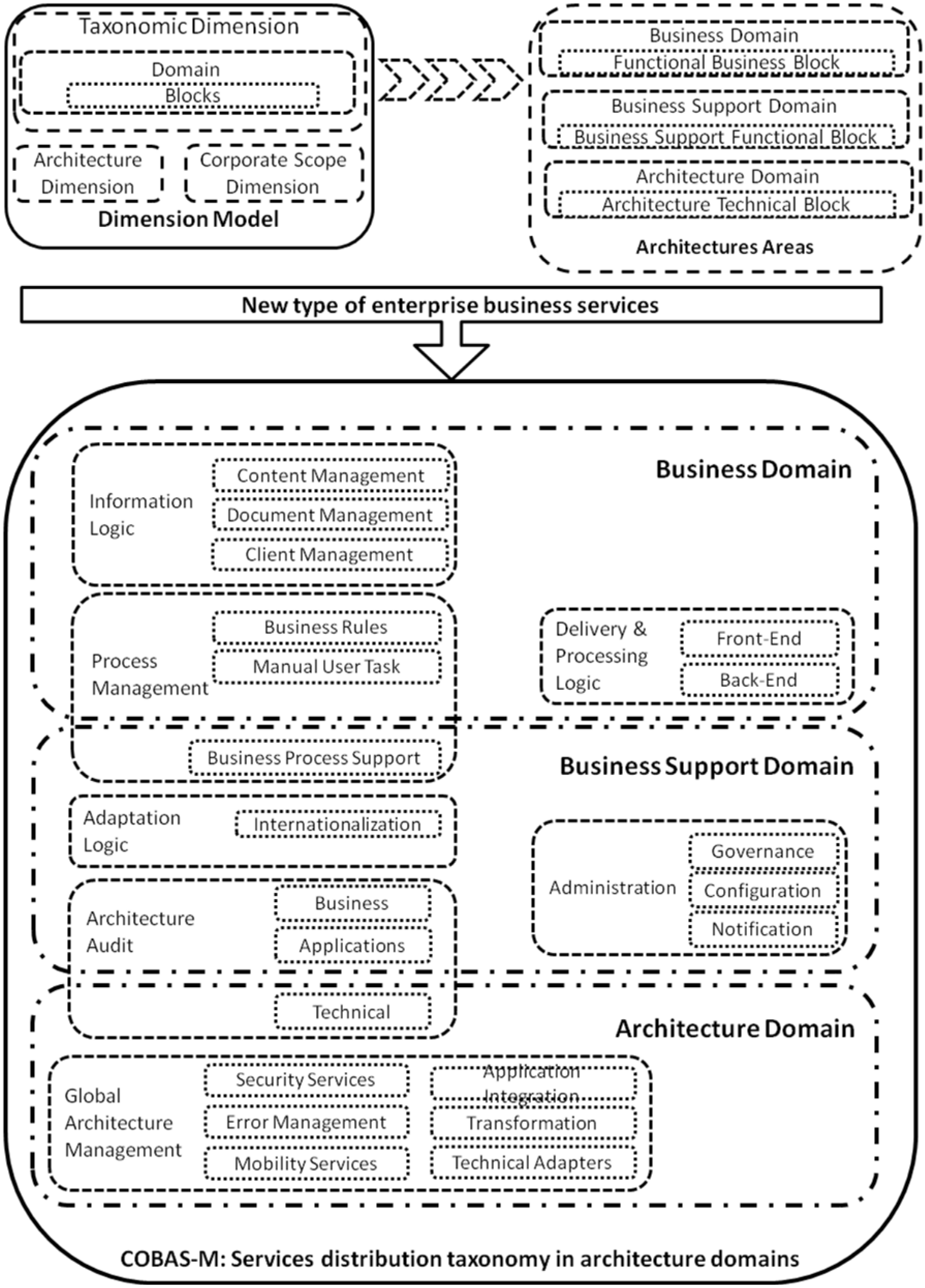

Figure 5: Architecture Proposal

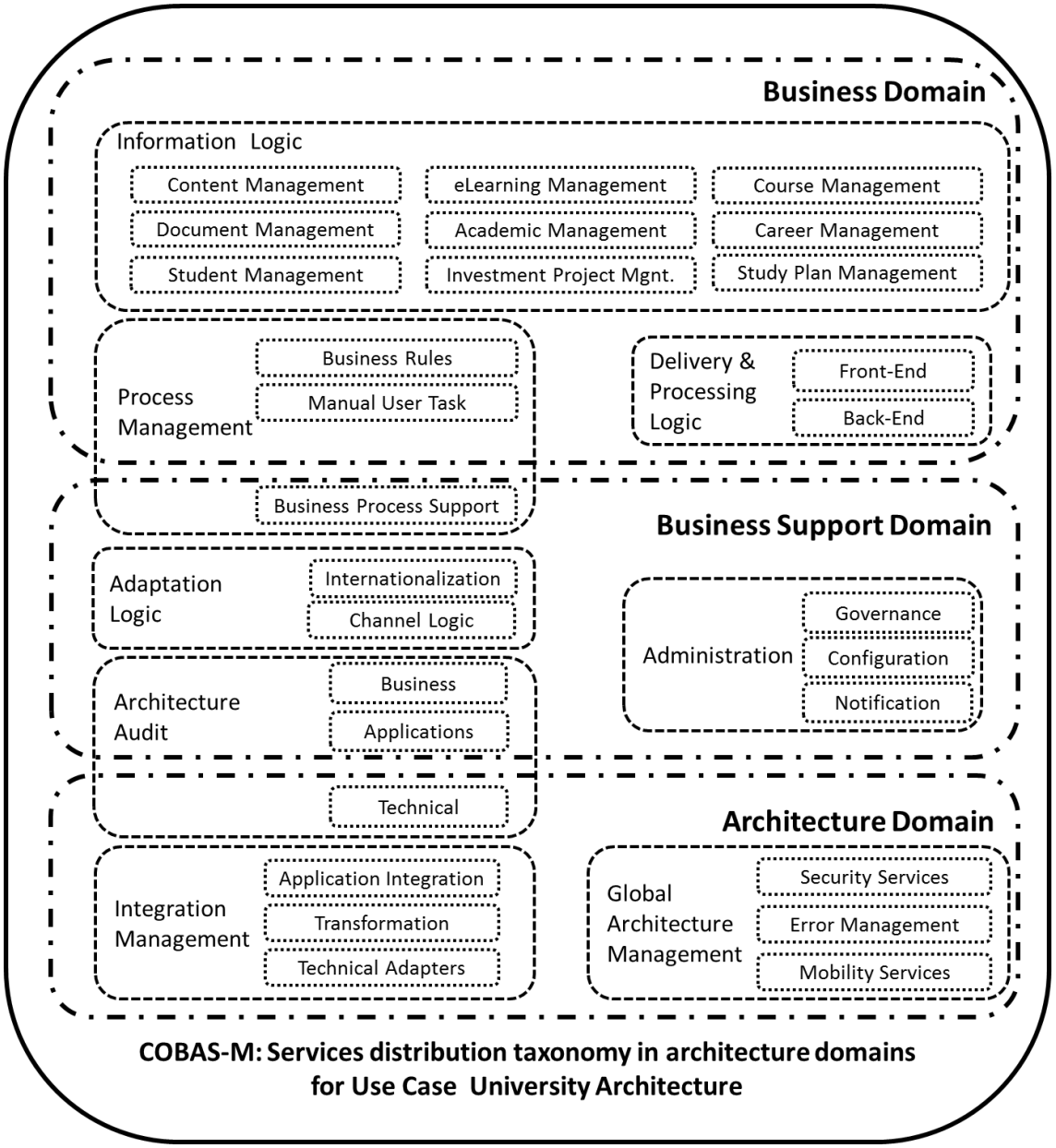

Figure 6: Architecture Proposal for a University

IV. DESIGN OF THE CASE STUDY

This section presents four case studies of real-life SOA solutions that implement the proposed architecture to validate it. All the projects have been implemented and tested under the management of the first author of the paper. However, it has not been possible to include any case of a University, because no project involving a University was carried out at the time of writing this paper. It is our expectation that the results gathered in other domains encourage managers of universities all over the world to adopt SOA for their corporate software. Table 2 shows a summary of all the projects in chronological order based on start date.

As can be seen in Table 2, all the projects have been developed in an international environment with teams located in different countries. The budgets range between 40.000 and 72.000 Euros. The number of services is between 10 and 35, and they are all quite recent, the oldest project started in January 2011, and the most recent project started in July 2011.

TABLE II
PROJECTS SUMMARY

A	**Budget**	40.000 €	**International environment**	YES
	Start Date	January 2011		
	Target Description			
	Migrate 10 services from an existing architecture and create 10 new functionalities. The technology desired will be Web Services and J2EE. All the services should be distributed in the actual architecture (Presentation and Technical/Integration).			
B	**Budget**	70.000 €	**International environment**	YES
	Start Date	July 2011		
	Target Description			
	30 services should be created from an existing .Net application layer. Web Services should be the interface between .NET existing applications and developments.			
C	**Budget**	44.000 €	**International environment**	YES
	Start Date	March 2011		
	Target Description			
	Migrate 15 services from an existing architecture and create 13 new functionalities as well as migrate legacy functionality (2 services). The technology desired will be Web Services and J2EE. Presentation and Technical/Integration layers should be considered.			
D	**Budget**	72.000 €	**International environment**	YES
	Start Date	June 2011		
	Target Description			
	35 services should be created from an existing J2EE application layer (**15 migrated and 20 new services**). Web Services should be the interface between .NET existing applications and J2EE developments.			

TABLE III
PROJECT SELECTION CRITERIA LIST

Item	Criteria
1	Timing estimated by the team in analysis phase (variability of 5 days per phase) (see Table 4)
2	Timing estimated by the team in design phase (variability of 5 days per phase) (see Table 4)
3	Project budget (variability of +- 5.000€)
4	Number of services (variability +-5 services from one project to another).
5	Similar Technology (see Table 2).
6	International environment: Several countries involved with project team members located in different places.
7	Project Kick Off. Each of the projects should start within a 6 month interval from one another.
8	Team members in each project should be the same or have at least similar profiles (see Tables 5 & 6).

TABLE IV
COMPARISON OF PROJECTS BY PAIRS ACCORDING TO TABLE III

	Implementation Estimation (days)	Testing & Delivery Estimation (days)	Budget (€)	Number Services	Technology	International environment	Project Kick Off	Team Members
A	100	70	40.000	10	Web Services and J2EE	YES	January 2011	Same as C
C	100	70	44.000	15	Web Services and J2EE	YES	March 2011	Same as A
B	250	130	70.000	30	Web Services .Net	YES	July 2011	Same as D
D	250	130	72.000	35	Web Services .Net and J2EE	YES	June 2011	Same as G

Table 3 provides a list of selection criteria identified to pair the projects of Table 4 so that each couple is similar and thus, both projects can be compared.

One project of each pair follows the Architecture Framework for SOA proposal of this article, and the other project follows the client corporate methodology. Both projects are developed in the same conditions, and at their conclusion the results are compared to analyze the influence of using the Architecture Framework for SOA in the implementation and testing, and delivery phases.

4.1 Development of the case study

Applying the indicators gathered in Table 3 to the list of projects gathered in Table 4, the pairs of projects that will be compared are: A & C and B & D as shown in Table 4, together with the values of comparison indicators of Table 3.

Once the projects were carried out, more data was collected as shown in Tables 5 and 6 (one Table for each pair of projects under study). In particular, both for the project in which the Architecture Framework for SOA proposed was used and was not used, the team members' profiles were registered. The SOA functional and technical requirements description is as follows:

- If Architecture Framework for SOA proposed was NOT used, the description specifies: Number of Services, Technology chosen to develop the solution, WS-* Standards, Architecture Layers, SOA Principles considered.
- If Architecture Framework for SOA was used, the description specifies:
  - Dimensions of Dimension Model considered in the final architectural design, Taxonomic Domains of Dimension Model created, Taxonomic Blocks of Taxonomic dimensions in Dimension Model.
  - Other SOA requirements such as Number of Services, Technology chosen to develop the solution, WS-* Standards, Architecture Layers, SOA Principles considered.

Projects A & C had to migrate 10 and 15 services respectively taking into account the presentation and technical layers native from respective client architectures, project C will use the Architecture Framework for SOA that is being proposed on this article so its final architecture diagram will follow the principles proposed by us (see Figure 6: Global Architecture Diagram). The use of some WS-* standards is a common requirements in this projects, in this case, WS-AtomicTransaction and WS-Security was mandatory to be included. The SOA principles [39] that were used in the design of services were: loose coupling, existence of a service contract, abstraction from one service to another and reutilization of legacy system cases. Table 5 presents the comparison of the development of projects A & C according to the criteria previously explained.

TABLE V
A & C PROJECTS COMPARISON

<table>
<tr><td rowspan="6">A</td><td>Architecture Framework for SOA proposed</td><td>NO</td><td>International environment</td><td>YES</td></tr>
<tr><td colspan="2">Team Description</td><td colspan="2">SOA Functional and Technical Requirements</td></tr>
<tr><td colspan="2">Project Manager
SOA Architect
Senior Funct. Analyst
1 x Programmer</td><td colspan="2">Number of Services:
10 services
Technology Used:
Web Services and J2EE
WS-* Standards:
WS-AtomicTransaction
WS-Security
Architecture Layers: Presentation and Technical layers.
SOA Principles considered: Loose coupling, Service Contract, Abstraction and Reutilization.</td></tr>
<tr><td rowspan="2">Architecture Framework for SOA proposed</td><td rowspan="2">YES</td><td>International environment</td><td>YES</td></tr>
<tr><td>Architecture Result</td><td>Included in Global Architecture (Figure 6)</td></tr>
<tr><td colspan="2">Team Description</td><td colspan="2">SOA Functional and Technical Requirements</td></tr>
<tr><td rowspan="4">C</td><td colspan="2" rowspan="4">Project Manager
SOA Architect
Senior Funct. Analyst
1 x Programmer</td><td>Dimension Model</td><td>Taxonomic Domains</td></tr>
<tr><td>Corporate, Architecture and Taxonomic</td><td>Business, Business Support and Architecture domain.</td></tr>
<tr><td>Other SOA Req.</td><td>Taxonomic Blocks</td></tr>
<tr><td>Number of Services:
15 services

Technology Used:
Web Services and J2EE

WS-* Standards:
WS-AtomicTransaction
WS-Security

SOA Principles considered:
Loose coupling, Service Contract, Abstraction and Reutilization.</td><td>Business Domain:
Front-En
Back-End
CRM
CMS

Business Support Domain:
Notification Services
Configuration services
Audit business

Architecture Domain:
Security service
Error Management
Technical Audit</td></tr>
</table>

Projects B & D had to work with a bigger set of services in this case study, from 30 to 35 new and migrated services in each project. In these projects, it should be taken into account the business, technical and data layers native from the respective client architectures. This staring point is quite useful due to it is a complete architecture desired. Project D is included as part of the Figure 6 in the Global Architecture diagram. The number of standards to be used is quite large (6 standards used), and the scope is the biggest among all the projects compared in this case study. Table 6 presents the comparison of the development of projects B & D according to the criteria previously explained.

As it has been previously described, **projects C & D use the proposed Architecture Framework** and they are included as part of the following architecture (Figure 7). This architecture is centered on provide business services (business, applications and technical level) to front end user applications without impact on the existing SOA services of the corporation applications that maintains the core business. As can be seen in Figure 7, all the services requirements of each project C & D have been fulfilled.

TABLE VI
B & D PROJECTS SUMMARY

<table>
<tr><td rowspan="5">B</td><td>Architecture Framework for SOA proposed</td><td>NO</td><td>International environment</td><td>YES</td></tr>
<tr><td colspan="2">Team Description</td><td colspan="2">SOA Functional and Technical Requirements</td></tr>
<tr><td colspan="2">Project Manager
SOA Architect
Functional Analyst
2 x Programmers</td><td colspan="2">Number of Services:
30 services
Technology Used:
Web Services, .NET
WS-* Standards:
WS-Security
WS-SecurityPolicy
WS-Policy
WS-Eventing
Architecture Layers: Business, Technical and Data Layer
SOA Principles considered: Loose coupling, Service Contract, Autonomy, Abstraction, Reutilization, Statelessness and Discovery.</td></tr>
<tr><td rowspan="2">Architecture Framework for SOA proposed</td><td rowspan="2">YES</td><td>International environment</td><td>YES</td></tr>
<tr><td>Architecture Result</td><td>Included in Global Architecture (Figure 6)</td></tr>
<tr><td rowspan="5">D</td><td colspan="2">Team Description</td><td colspan="2">SOA Functional and Technical Requirements</td></tr>
<tr><td colspan="2" rowspan="4">Project Manager
SOA Architect
Functional Analyst
2 x Programmers</td><td>Dimension Model</td><td>Taxonomic Domains</td></tr>
<tr><td>Corporate, Architecture and Taxonomic</td><td>Business, Business Support and Architecture domain.</td></tr>
<tr><td>Other SOA Req.</td><td>Taxonomic Blocks</td></tr>
<tr><td>Number of Services:
35 services (Migrate 15 and create 20)

Technology Used:
Web Services, .NET and J2EE
WS-* Standards:

WS-Security
WS-SecurityPolicy
WS-Policy
WS-Eventing

SOA Principles considered:
Loose coupling, Service Contract, Autonomy, Abstraction, Reutilization, Statelessness and Discovery.</td><td>Business Domain:
Front-End
Back-End
Business Rules
Customer Relationship Management
Content Management

Business Support Domain:
Notification Services
Internationalization
Configuration services
Mobility services
Audit business
Audit of applications
Channel adaptation logic

Architecture Domain:
Security services
Application Integration
Format Transformation
Technical Audit</td></tr>
</table>

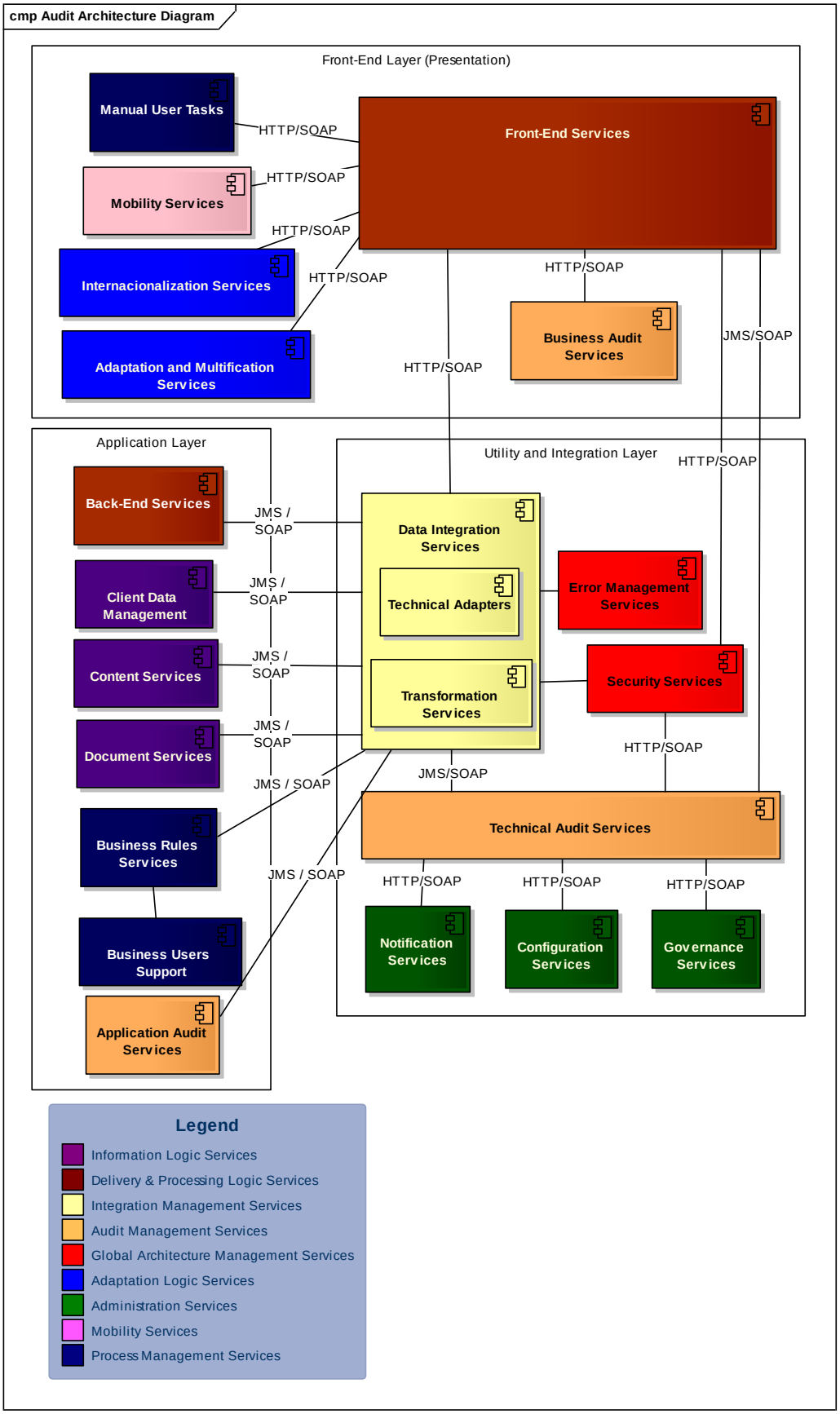

Figure 7: Proposed Global Services Oriented Architecture Diagram

TABLE VII
TABLE DATA FROM IMPLEMENTATION PHASE TIMING CONSUMING

Implementation Phase			
	WITHOUT Architecture Framework (real days)	WITH Architecture Framework (real days)	% of improvement
Project A & C	130	80	38,5
Project B & D	260	180	30,77

TABLE VIII
TABLE DATA FROM TESTING AND DELIVERY PHASE TIMING CONSUMING

Testing and Delivery Phase			
	WITHOUT Architecture Framework (real days)	WITH Architecture Framework (real days)	% of improvement
Project A & C	80	53	33,75
Project B & D	160	111	30,63

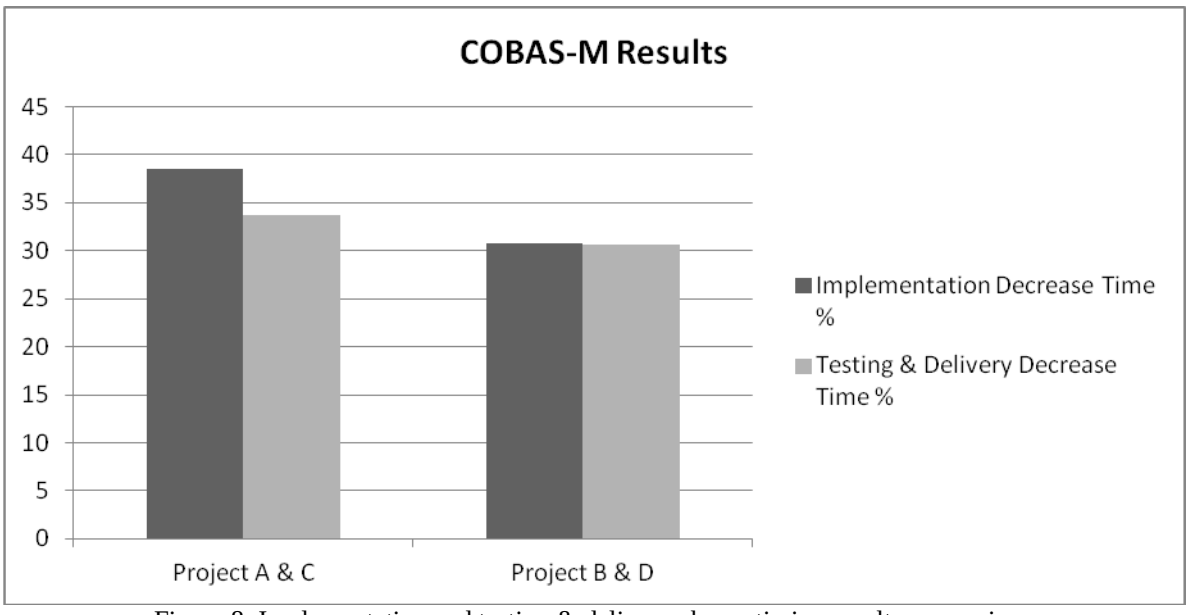

Figure 8: Implementation and testing & delivery phases timing results comparison

4.2 Results

This section provides the final results of the case study. The timing results taken from each of the projects analyzed are described. It can be seen how the proposed SOA Architecture Framework can generate an improvement of 30,6% to 38,5% in each of this lifecycle phases of implementation, and testing & delivery phases. Table 7 describes the values registered by pairs of projects in the Implementation phase. For example, projects A and C were estimated to be accomplished in 100 days, but eventually project A was finished in 130 days and C, in which the proposed SOA Architecture Framework was used, was finished in 80 days. This means an improvement of 38,5% (i.e. 50 days less than the project in the pair compared that does not used the proposed Architecture Framework).

As Table 7 shows, the projects that use the proposed SOA Architecture Framework for the implementation phase register gains, in duration, between 30,8% and 38,5%. Table 8 gathers the results registered for the testing and delivery phase.

As can be seen, using the proposed SOA Architecture Framework in the testing and delivery phase decreases the number of days estimated to complete the phase between 30,6% and 33,7%. For example, projects B and D were estimated to be accomplished in 130 days. Project B did not use the proposed SOA Architecture Framework, and its duration was 160 days (30 days more than the estimation). On the other hand, project D used the proposed SOA Architecture Framework and it registered a time of 111 days, which is an improvement of 30,6%. The best improvement was also registered for projects A & C, which a reduction of 33,7% over the project in the pair compared that does not used the proposed SOA Architecture Framework (17 days less).

Figure 8 shows a graphical comparison of the timings for implementation and testing & delivery registered for all the pairs of projects under study.

As can be seen, not only it is possible to successfully apply the proposed SOA Architecture Framework, but there is an improvement of at least 30,7% in the implementation phase, and in the best case of 38,5%. Regarding the testing and delivery phase, the improvement is between 30,6% and 33,7%.

V. CONCLUSION

In the last decades numerous advances have occurred in the IT field related to SOA architectures creation. Most of them show that integration from various methods promises to increase the adoption of architecture-related tasks by reducing the effort involved in dealing with multiple technologies, solutions and development and stakeholders teams [49].

In this paper, we have presented the benefits of a new SOA architecture proposal that comes from business analysis to architectural design. Implementation and testing time can be reduced up to 33% and 31% respectively in business domains.

Corporate University software is not usually developed taking into account SOA methodologies or architectures [15-17]. Just a few American universities such as the Duke University [18] and the University of Virginia [19] are early SOA adopters. Both cases show benefits of using SOA in education, even when a horizontal architecture is not applied.

We believe that the use of the proposed horizontal architecture can help SOA architects when working in the implementation and testing of University educational software. Moreover, it is our expectation than from the detailed description of the proposed SOA architecture and its results, managers of universities of all over the world start considering adopting SOA for their corporate software.

REFERENCES

[1] D. [1] J.A. Zachman, A framework for information systems architecture, IBM Systems Journal, Vol. 26, 3, 1987.

[2] M. A. González Serrano. Estudio conceptual y desarrollo de nuevos tipos de servicios de modelado de servicios de negocio y sus artefactos para el análisis y diseño de arquitecturas SOA, Masther Thesis, Universidad Autónoma de Madrid, 2011.

[3] E. Newcomer, G. Lomow, Understanding SOA with Web Services (Independent Technology Guides) Addison-Wesley Professional, ISBN:0321180860, 2004.

[4] M. P. Papazoglou, M. P.; W. J. van den Heuvel, "Service-oriented design and development methodology", International Journal of Web Engineering and Technology (IJWET), 2006.

[5] S. Hong, G. Feuerlicht, Web Service Interface Design for e-Business Application: A minimalistic approach, University of Technology, Sydney, Department of Information Technology, 2008

[6] D. Woods, T. Mattern, Enterprise SOA: Designing IT for Business Innovation, April 2006.

[7] R. T. Fielding, Architectural Styles and the Design of Network-based Software Architectures, University Of California, Irvine, 2000

[8] P. Allen, The service oriented process, in CBDi Journal, February 2007

[9] K. Mittal, Service Oriented Unified Process (SOUP), 2006.

[10] A. Arsanjani, Service-Oriented Modelling and Architecture (SOMA), IBM developer-Works 2004

[11] A. Arsanjani, Explicit Representation of Service Semantics: Towards Automated Composition Through a Dynamically Reconfigurable Architectural Style for On Demand Computing, International Conference on Web Services; pp 34 - 37; 2003

[12] A. Arsanjani, Service-Oriented Modelling and Architecture (SOMA), IBM developer-Works 2004

[13] R. Pérez-Castillo, I. García-Rodríguez de Guzmán, M. Piattini Alarcos Business process archeology using MARBLE, Research Group, University of Castilla-La Mancha, Ciudad Real, Spain, Information and Software Technology.

[14] M. A. González Serrano, D. Pérez-Marín, M. I. Alarcón. Clasificación de los Servicios Web de Negocio Corporativos basada en la Funcionalidad, Horizontal de las Organizaciones. M.V. de Castro, J.M. Gómez, I. Iribarne (Eds). JCIS'2012, pp. 81-87, ISBN 978-84-15487-26-5. Jornadas Sistedes'2012, 17-19 septiembre, Universidad de Almería.

[15] S. Yang & M. Joy. 2008. SOA Services in Higher Education. In Proceedings of the 7th WSEAS International Conference on Education and Educational Technology, 145-150.

[16] G. Katsionis & M. Virvou. 2008. Personalised e-learning through an educational virtual reality game using Web services. Multimed Tools Appl, 39:47–71

[17] M. Zelić, M. Rosić, & V. Glavinić. 2011. Intelligent Provision of Semantic Web Services in Educational Systems. C. Stephanidis (Ed.): Universal Access in HCI, Part IV, HCII 2011, LNCS 6768, pp. 670–679, Springer-Verlag Berlin Heidelberg.

[18] Gartner. 2007. Case Study: Duke University Health System Finds Excellent Productivity Using SOA. Industry Research Report, ID Number: G00152168. Published on-line at http://www-03.ibm.com/industries/ca/en/healthcare/files/gartner_report-duke_u_health_sys_soa_case.pdf. Last visit on April 18th, 2013.

[19] Alan Joch. 2011. SOA with Smarts. University of Virginia makes a new-school change to a venerable application. Oracle Magazine. Published on-line at http://www.oracle.com/us/corporate/profit/features/071111-uva-432602.html. Last visit on April 18th, 2013.
[20] M. Huhns, M. Singh, Service-Oriented Computing: Key Concepts and Principles, IEEE Computer Society, Febrero 2005
[21] T. Erl, SOA: Principles of Service Design, Prentice Hall, 28 de Julio de 2007, ISBN: 978-0132344821
[22] G. Engels1, A. Hess, B. Humm2, O. Juwig, M. Lohmann, J.-P. Richter, M. Voß, J. Willkomm, A method for engineering a true service-oriented architecture, sd&m Research – software design and management, Carl-Wery-Straße 42, 81739 München, Germany 1 additionally University of Paderborn, 2 additionally Darmstadt University of Applied Sciences, 2008Shannon et al. 2000
[23] Shannon, B., Hapner, M., Matena, V.: Java 2 Platform, Enterprise Edition. Addison Wesley 2000
[24] Reussner, R. Hasselbring, W.: Handbuch der Software-Architektur. dpunkt.Verlag, 2006
[25] Yourdon, E., L. Constantine, L. L.: Structured Design: Fundamentals of a Discipline of Computer Program and Systems Design. Prentice Hall, September 1986
[26] Jackson, M., Twaddle, G.: Business Process Implementation: Building Workflow Systems, Addison-Wesley 1997,
[27] Siedersleben, J.: SOA revisited: Komponentenorientierung bei Systemlandschaften. Wirtschaftsinformatik 49 (2007) Sonderheft, S. 110-117
[28] O. Zimmermann, N. Schlimm, G. Waller & M. Pestel, Analysis and Design Techniques for Service-Oriented Development and Integration, IBM Deutschland, 2004
[29] Woods, D.: Enterprise Services Architecture. SAP Press, 2004
[30] Bieberstein, N., Bose S., Fiammante, M., Jones, K., Shah, R.: Service-Oriented Architecture (SOA) Compass: Business Value, Planning, and Enterprise Roadmap. IBM Press, 2005
[31] D. Krafzig, K. Banke, D. Slama, Enterprise SOA: Service Oriented Architecture Best Practices, Prentice Hall International, November 2004
[32] Richter, J.-P., Haller, H., Schrey, P.: Serviceorientierte Architektur. Informatik-Spektrum, 28(5), Oktober 2005, 413-416
[33] Voß, M., Hess, A., Humm, B. : Towards a Framework for Large Scale Quality Architecture. In: Hofmeister, Ch., et.al. (Eds.): Perspectives in Software Quality - Short Papers of the 2nd International Conference on the Quality of Software Architectures (QoSA), Interner Bericht 2006-10, Universität Karlsruhe, Fakultät für Informatik.
[34] Hanemann, A., Boote, J. W., Boyd, E. L., Durand, J., Kudarimoti, L., Lapacz, R., Swany, D. M., Zurawski, J., Trocha, S., "PerfSONAR: A Service Oriented Architecture for Multi–Domain Network Monitoring", In "Proceedings of the Third International Conference on Service Oriented Computing", Springer Verlag, LNCS 3826, pp. 241–254, ACM Sigsoft and Sigweb, Amsterdam, The Netherlands, December, 2005
[35] perfSONAR (Performance focused Service Oriented Network monitoring ARchitecture). http://www.perfsonar.net/.
[36] I. Jørstad, S. Dustdar, D. Van Thanh, A Service Oriented Architecture Framework for Collaborative Services, Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05), June 2005
[37] P. Lago and M. Razavian, A Pragmatic Approach for Analysis and Design of Service Inventories, Department of Computer Science, VU University Amsterdam, The Netherlands, 2012.
[38] K. Pohl, S-Cube - The European Network of Excellence in Software Services and Systems, 2012.
[39] T. Erl, Service-Oriented Architecture: A Field Guide to Integrating XML and Web Services, Prentice Hall, 26 de April de 2004, ISBN: 978-0131428980
[40] T. Erl, Service-Oriented Architecture (SOA): Concepts, Technology, and Design, Prentice Hall, August de 2005, ISBN: 978-0131858589
[41] O. Zimmermann, Krogdahl & Gee, Elements of Service-Oriented Analysis and Design, 2004, IBM Zurich Research Group.
[42] A. Erradi, SOAF: An architectural framework for service definition and realization in Proceedings of the IEEE International Conference on Services Computing, pp 151-158, Chicago, USA, September 2006.
[43] M. Cantara, Common features of external service providers' SOA frameworks and offerings, Gartner, September de 2005.
[44] Gartner, Magic Quadrant for Enterprise Content Management, 16 de Noviembre de 2010
[45] A. Delgado, F. Rui, I. García-Rodríguez de Guzmán, M. Piattini, A Model-driven and Service-oriented framework for the business process improvement, Computer Science Institute, Faculty of Engineering, University of the Republica, Technologies and IS Depto., Faculty of Computer Science, University of Castilla-La Mancha, Journal of Systems Integration, 2010
[46] E. Ramollary, D. Dranidis, A. J. H. Simons, A Survey of Service Oriented Development Methodologies, South East European Research Centre, 2007
[47] S. Jones, A. Methodology for Service Architectures, Capgemini UK plc, August 2005.
[48] G. López, A. Echeverría, P. Fierro (PhD.); I. Jeder. Una propuesta de modelos de ciclo de vida (mcvs) para la integración de los procesos de negocio utilizando service oriented architecture (SOA), Laboratorio de Informática de Gestión Facultad Ingeniería - Universidad de Buenos Aires, 2007
[49] R. Weinreicha, G. Buchgeherb, Towards supporting the software architecture life cycle, The Journal of Systems and Software 85 (2012) 546– 561