Normalized Enterprises Architectures: Portugal’s Public Procurement System Application

Tiago Sampaio
Instituto Superior Técnico
tiago.sampaio@ist.utl.pt

***Abstract*—**The Normalized Systems Theory, which is designed to be applied to software architectures, provides a set of theorems, elements and rules, with the purpose of enabling evolution in Information Systems, as well as ensuring that they are ready for change. In order to make that possible, this research applies the Normalized Systems Theory to the domain of enterprise architectures, using Archimate. This application is achieved through the adaptation of the elements of this theory, making them artifacts of the modeling language. The theorems are applied through the identification of the viewpoints to be used in the architectures, as well as the transformation of the theory’s encapsulation rules into architectural rules. This way, it is possible to create normalized enterprise architectures fulfilling the needs and requirements of the business. The solution proposed was demonstrated using the Portuguese Public Procurement System. The Portuguese government aims to make this system as fair as possible, allowing every organization to have the same business opportunities. The aim is for every economic operator to have access to all public tenders, which are published in any of the six existing acquisition platforms, independently of where they are registered. In order to make this possible, we applied our proposal in the assessment of two different architectural scenarios. Apart from the two architectural scenarios, we also represent the AS-IS architecture of the Portuguese Public Procurement Systems. Our evaluation is based on a comparison between the AS-IS and the TO-BE architectures, regarding the fulfillment of the rules and theorems of the Normalized Systems Theory and a set of quality metrics.

***Keywords*—** Archimate, Architecture, Broker, Enterprise, Evolvable Systems, Interoperability, Normalized Architectures, Normalized Systems, Normalized Systems Theory, Platforms.

I. Introduction

Organizations have grown in an accelerated way. This new complexity and requirement imposed to organizations led (and leads) to high demands of the same access to information [9]. Thus, innovation and agility in redefining business plays a key role in the success of organizations [10].

In order to respond to the changing business needs, organizations are required to be in continuous change according to needs and wishes of the market where they operate. Nowadays Information Technology (IT) is a central tool for supporting change and for providing competitive advantages [10]. IT provides quick and efficient access to information, allowing the business processes to have automation and acceleration [1].

In Portugal, understanding IT benefits, a government initiative to promote the use of Information Technology in the procurement process was created. This initiative led to the development of procurement platforms by private owned companies. Public bodies need to contract a procurement platform in order to place their tenders, and suppliers need to register in that same platform in order to present their tendering offer. For the public bodies it is not exactly a problem, since they only need to contract with one procurement platform from the market, but the potential tenderers need to have as many contracts with procurement platforms as the ones that exist, since their clients are spread among all of them.

The existence of multiple Procurement Platforms led to one more functional requirement: companies should be able to make their tendering offers no matter in which one of the six platforms it was published, from the interface of the platform where they were registered. The government believes that such change will provide public bodies better tendering opportunities.

As in the case of Portugal’s Public Procurement System, IS doesn’t adequately respond to the continuous changes that organization are confronted with – which causes a misalignment between business requirements and IT and, consequently, reduces the enterprises competitive capacities, compromising their future and growth [1].

Information Systems Architectures (ISA) offer support from technology to business requirements leading to orderly growth-oriented organizations [1]. Additionally, in this paper we believed that an adaptation of Normalized Systems Theory (NST) to the business domain can bring all of their characteristics to the architecture. This extension of the Normalized System Theory to the IS domains, aims to make evolvable architectures that are prepared for change. In addition to that, it is intended to create stable architectures, i.e., any change can’t be dependent on the system’s size and should always require the same effort.

This document has the following structure (section): Introduction (1), Research Methodology (2), Problem (3), Related Work (4), Solution (5), Demonstration (6), Evaluation (7) and finally Conclusion (8).

II. Research Methodology

This work development is based in the *Design Science Research Methodology* (DSRM). This methodology aims to build and evaluate solutions that meet the businesses' needs. [2]

By analyzing this methodology, the five stages of development can be identified: problem identification and motivation, solutions' objectives, design and development, demonstration, evaluation and communication. This methodology is used in IS development, where they include content from different fields to meet the challenges and problems faced by organizations. A key criterion for the choice of this methodology was the success demonstrated in related work in this field [3].

To be coherent with our work, this document will follow the same structure as DSRM. Section 3 ("Problem") and 4 ("Related Work") identifies the work's problem and motivation. Section 5 ("Solution") represents the solution's design and development to apply in this investigation, which is demonstrated in section 6 ("Demonstration") and evaluated in section ("Evaluation"). In section 8 ("Conclusion") this work is finalized with contributions, limitations and future work.

III. Problem

With constant evolution and development in organizations at IT level comes the need to develop solutions that make enterprise architectures ready for change and agile to quickly respond to business changes.

With the business decision to create interoperability among privately owned platforms, comes the need to define and architecture that allows tenderers to consult tender announcements submitted on any of the electronic platforms and to submit their proposals as well. These systems should be able to create interoperability among all existing platforms, complying with the complex regulatory framework that dictates the conditions in which it must happen (security, authentication, ownerships) as well as for all the services that are expected, from the announce of tender, to the evaluation and granting of contract.

The Normalized Systems Theory sets the ground to integrate systems and software oriented problems, but it is not clear on how it can be the base for the operationalization of the interoperability among different platforms where it is not expected, or viable, to eliminate applications or services that provide the same tasks, since that is where the reason to exist in market (and be competitive) lies.

Therefore, the question we aim to answer is: ***how to develop an Enterprise Architecture that ensures agility and organizational evolution?*** Associated to this main question, it's important to underline other goals to be accomplished:

- Does NST allows Heterogeneous Information Systems Architectures' integration?
- How to apply NST's elements in enterprise architectures?
- How to apply NST's encapsulation rules in enterprise architectures?
- How to apply NST's theorems in enterprise architectures?

In order to solve this problem, in this research contributes for the creation of an evolvable, stable and easy to maintain architecture, integrating this new functional requirement and providing the following topics to the scientific community:

- Application of NST's theorems based on Archimate;
- Application of NST's elements based on Archimate;
- Application of NST's encapsulation rules based on Archimate;
- Architecture creation integrating all platforms;

IV. Related Work

In this section, we introduce some topics that will support the development of this research. First of all, we describe the state of the public procurement system in the EU and, more specifically in countries comparable to Portugal when it comes to how public procurement works. After that, we present the theory that will be the foundation of our work, the Normalized Systems Theory. We also cover enterprise architectures, including Archimate modeling language, some architectural principles and the work's evaluation method.

A. EU's Public Procurement System

The correct, efficient and effective application of the EU public procurement rules in the different countries poses a great challenge. The European Union considers that the retrieval, analysis and report elaboration about each country's information with respect to public procurement improves the response to politics' new challenges, namely achieving savings, using public contracts to support other political goals e reinforcing the country's administrative ability towards public contracts]18].

Every year, about one fifth of the EU's GDP is spent by the different government authorities and public entities in the acquisition of goods, services and construction. There's a great difference among EU countries with respect to the money spent on proposals [18]. The most common procedure to execute public contracts is to publish an announcement inviting every interested supplier to present their offer(s) (public procurement), which represents 73% of the market share. Then, in second place, comes the direct adjustment procedure, with 22%. The remaining 5% represent the other type of public procurement processes, for instance the simplified direct adjustment [18].

The use of electronic means to improve the public procurement process has led to a great cost reduction in all the EU member states. This solution is associated to several advantages to all the parts involves, since it eases the execution of the contracts, reduces the costs supported by the economic operators as well as the administrative charges associated with it. It also guarantees that the opportunities to access this type of contract are accessible to everyone involved [18].

The use of electronic platforms enables the support of information flow as well as the negotiation process between the interested parties. All future efforts must focus on the transition of communication to these electronic platforms, specially the electronic access to the offer's documents and respective submission, since these changes aim to increase the transparency and reinforce competition in the public procurement market [18].

B. Normalized Systems Theory (NST)

NST's goal is to make information systems evolvable and ready for change. Therefore, the software architectures must not only satisfy the current requirements but also support all possible requirements that may appear in the future [7] - [23].

As such, to enable the support for these changes, the NST mentions that an essential feature of an information system (IS) is stability. This concept, applied to IS's is dependent upon the fact that there can be no combinatory effects or change propagation effects, i.e., a change in an IS must always require the same effort, independently of the size of the system or the timing in which it is applied [8] - [23].

In order to counter these combinatory effects, we must create an architecture that follows a set of design rules. In NST, those rules are in the form of a set of four theorems, whose goal is to identify most of the combinatory effects. Essentially, these theorems identify parts of the architecture in which evolution is in danger of not occurring [8] - [23]:

- **Concern Separation**: all drivers of change or concepts must be separated. This theorem allows the isolation of each change driver;
- **Data Version Transparency**: all data must be communicated between the components in a transparent way and the changes in one component cannot impact the other associated components;
- **Action Version Transparency**: a component can be updated without affecting the call components. This principle can be achieved through systematic and adequate use;
- **States Separation**: certain actions and steps on a workflow are separated from each other, with a state after every action or step. This theorem leads to the use of asynchronous calls with the state of other components.

Apart from these theorems, the NST proposes a set of five elements to encapsulate the software builders. These elements are modular structures that adhere to the principles previously described, with the goal of reaching stability in the face of the anticipated changes: [8]

- Data Element;
- Action Element;
- Workflow Element;
- Trigger Element;
- Connector Element.

The Data Version Transparency and Action Version Transparency theorems mean that the software builders which represent actions and data, need to be encapsulated, aiming to build stable information systems. This leads to the following elements: [23]

- ***Data Encapsulate*:** the support tasks must be implemented at the data entity level. There should also be Get/Set methods, to maintain the data version transparency;
- **Action Encapsulate:** an action element must only have a functional task. The arguments and parameters must be encapsulated data elements. The workflows must be separated from the action elements and may have several versions without impacting the calls.

On the other hand, the Concern Separation and State Separation theorems, according to task aggregation, mean that the workflows must be separated from the other actions in the action element. These actions must be separated or isolated by states and the IS must be able to react to errors/mistakes. The aforementioned leads to the following additional encapsulations: [23]

- **Workflow Encapsulate:** the workflows cannot contain other functional tasks and each action needs to be stored;
- **Trigger Encapsulate:** the Trigger Elements must control different states and verify if an action element should be initiated;
- **Connector Encapsulate**: the Connector Elements must guarantee that the external systems can interact with the data elements.

1) Applying NST to IS development

NST incorporates some design, development and IS maintenance traits. The system should: [14]

- Be organized according to the five high level elements: data, actions, workflow, connectors and triggers;
- Support the four theorems: concept separation, data version transparency, action version transparency and state separation;
- Be based on the modularity concept, no ease the process of hiding information and allowing the separation of the system's concepts.

2) Applying NST at the Organizational Level

Beyond software systems, the organizations (and other system categories) can also be considered modular structures and, therefore, NST must be applicable. With the purpose of being in line with the theory's theorems, we propose that a business process works as a sequence of tasks in a single information object. This information object must be interpreted as a concrete and identifiable informational entity, with a clear business meaning and a different life cycle, represented by the business processes [28].

Furthermore, the combinatory effects defined in the previous section can also occur on the business process level, when the impact f an alteration is related to the global size of the system (or entity). As such, each change driver (concept) must be isolated and encapsulated in its own task or business process [28].

Every modulation defines different viewpoints for the system's functionality and data (action and data elements) that reflect different user goals. So, each system must have an artifact that is able of describing how the system is viewed by the users. In terms of NST elements, this artifact is the Connector Element [7].

If we aim to represent the triggering of a use case we use the Trigger Element, which on turn can set off Action Elements. The global system, or functional subsystem can be represented using Workflow Elements, which perform a series of Action Elements in a single lifecycle of a specific Data Element. The entities' lifecycle determines the order in which the Action Elements are invoked [7].

A business process can have multiple business entities and roles. We can model the Workflows that are capable of containing several functional subsystems and records through the analysis of lifecycles from different entities [7]. Although the main task of the records is to record data, we cannot describe them as the only Data Elements [7].

C. *Enterprise Architecture*

Nowadays, building an enterprise architecture is very important to ensure the success and good operation of the organization. So, architecture is the art and the science of building a system through the specification and organization of its components. It's a formal description of a system that defines the structure and properties of its components, their relationship and the behavior of the components as well as the principles necessary for its analysis, design and evolution [9].

Enterprise architecture is a group of models defined by getting a coherent and comprehensible picture of the enterprise. The main goals of EA are understanding the organization's communication strategy; Development of systems, products and services according to business goals; Optimizing operations; Optimizing organizational resources (including their people); Guaranteeing alignment between all the layers of the organization: business, data, application and technology [10]. In this thesis we will only focus on the Archimate with Business and Application Layer.

1) *Archimate*

The ArchiMate modeling language supports the complete modeling of enterprise architecture through all of this architecture's development cycle. It allows the description and visualization of different domains present in an organization and how they relate to each other. Furthermore, ArchiMate allows us to know the services, attributes, data and processes of each of the different layers. [11].

This modeling language consists of three element categories: passive structural elements (information and data), behavioral elements (processes) and active structural elements (organization and their actors). Archimate is divided in 3 layers: Business, Application and Technology [11].

Active structural elements are business actors, application components and devices that show any type of behavior. So, an active structural element is an entity able of executing its behavior [11]. Passive structural elements are the elements for which that behavior is executed. Usually, these elements are information objects but they can be represented as physical objects too [11].

Despite the differences in the elements of the three ArchiMate layers, these can be related and connect via internal links, services or interfaces. Aside from the main concepts, ArchiMate has a set of connections that establish the nature of the relationships between concepts. Most of them are based on other languages, such as UML [11].

Lastly, we can say that ArchiMate language is a good alternative to other languages because ArchiMate is easy to understand, less complex and supports interaction between Business, Application and Technologic Layers through many viewpoints [11].

2) *Architecture's Principles*

A principle is a set of long lasting rules and guiding lines inform and support an organization/entity fulfill its mission (20).

Currently, it is a general belief that architectural principles are fundamental to guarantee the efficacy of an enterprise architecture. Several authors and publications that they are the essence of an architectures and have the goal of correcting the flaws between the strategic level and the architectural building's concrete decisions. Furthermore, principles ensure that the enterprise architecture is the future of all organizations, and that they are able of orienting implementation decisions, avoiding parallel analysis, focusing mostly on its essence [20].

3) *Evaluation*

The evaluation theme in the domain of enterprise architectures is presented in this section. The main goal of an evaluation is to verify if the quality attributes we intend to include in an IS architecture are present and according to each case's needs [1].

In the previous sections we mentioned the advantages and importance of these architectures for any system we aim to build. However, using the evaluation process, we can accomplish several other benefits, such as financial benefits, risk and problem assessment [1].

Each evaluation process must be adapted according to the goals, needs and specificity of every architecture. As such, this is an area that is in constant adaptation to the rhythm in which IS investigation evolves [1].

The evaluation process defines the methodology to follow when conducting an evaluation. In the process of evaluation an ISA, we aim to analyze the existing architectural features, as defined in the quality model. So, through the adaptation of the software quality model described by the standard ISO 9126, a quality model for enterprise information systems was created. We call this our evaluation metrics [1].

V. SOLUTION

The solution proposed is based on NST. For this, it's necessary to extend the software rules for IS architectures' development. This architectures' development is accomplished using Archimate modeling language. In the following figure we provide an overview of the solution proposed.

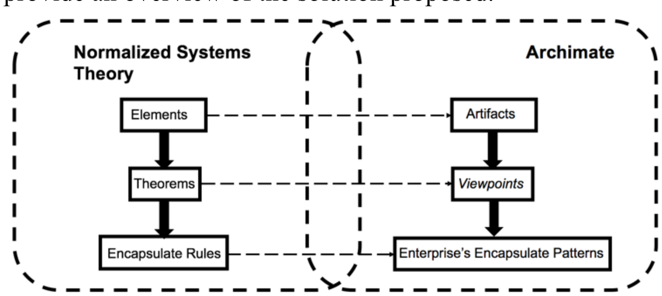
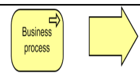


Fig. 1 Solution's Organization

A. *NST Elements' Application to Archimate*

In order to enable the application of an NST Element to Archimate, it's necessary to make a mapping between the theory's main elements and archimate's artifacts. In the following table we present an example of this mapping.

TABLE I
MAPPING BETWEEN NST'S ELEMENTS AND ARCHIMATE

NST Element	NST Description	Archimate Element	Archimate Description
Workflow Element	It constains the sequence in which a number of elements should be performed in order to meet the workflow.		A behavioral element, containing a set of sequence activities.

There are cases where it is possible to represent various Archimate's elements and therefore it is necessary to apply the corresponding element to the layer.

B. NST Theorems' Application to Archimate

In addition to the elements, it is essential to make a mapping of the four main NST's theorems. This is performed through identifying the viewpoint to be used.

1) Concerns Separation

TABLE II
CONCERNS SEPARATION

NST Theorem	NST Description	AE Description
Concerns Separtion	All concepts must be separated and, thereby, allow the isolation of the changes' impact in these same concepts	All funcionalities must be created independently, so that each change doesn't affect the rest of the system.

In addition to the application's description, we selected the following viewpoints to be used:

- Business Process Viewpoint;
- Application Structure Viewpoint;
- Application Behavior Viewpoint;
- Actor Co-Operation Viewpoint.

2) Data Version Transparency

TABLE III
DATA VERSION TRANSPARENCY

NST Theorem	NST Description	AE Description
Data Version Transparency	All data must be communicated in a transparent manner between the components and, thereby, data's modification doesn't have an impact on the system.	All data objects must be communicated through components or services and, in turn, can be communicated by interfaces.

In addition to the application's description, we selected the following viewpoints to be used:

- Application Structure Viewpoint;
- Application Behavior Viewpoint;
- Application Usage Viewpoint;

3) Data Action Transparency

TABLE IV
DATA ACTION TRANSPARENCY

NST Theorem	NST description	AE Description
Data Action Transparency	All data can be updated without affecting the other elements attached to it.	All data objects can be changed or deleted from the system without affecting the remaining structure.

In addition to the application's description, we selected the following viewpoints to be used:

- Application Structure Viewpoint;
- Application Behavior Viewpoint;
- Actor Co-Operation Viewpoint.

4) States Separation

TABLE V
STATES SEPARTION

NST Theorem	NST Description	AE Description
States Separation	All actions and steps performed in a workflow must be separated from each other, keeping only a state of each action or step.	In a business process, all updates and changes to informational entities must be independent of each other.

In addition to the application's description, we selected the following viewpoints to be used:

- Business Process Viewpoint
- Application Usage Viewpoint;

C. NST Encapsulate Rules' mapping to Archimate

In order to apply the NST in an enterprise architecture construction it is necessary to represent all elements, theorems and encapsulation rules in the business domain. Therefore, in addition to the already performed mapping for the elements and theorems, the following table represents an example of the encapsulation rules' application in Archimate.

TABLE VI
NST'S PATTERNS MAPPING FOR ENTERPRISE DOMAIN

NST Rule	NST Description	AE Description
Workflow Encapsulate	Workflow can't contain other functional tasks;	A business process should only contain one functionality.
	Each workflow action must be saved.	Each process' internal action should do an update to an informational entity

D. Encapsulate Rules' specification at Enterprise Domain

Based on the mapping performed in the previous section, it is necessary to detail these enterprise encapsulation patterns. For each pattern we will identify the corresponding information types, attributes, a pattern analysis and the implications it will have on the system. The two following patterns represent an example of this detailing.

Pattern 1 – A business process should only contain one functionality

- <u>Information Type</u>**:** Business;

- Attributes: Encapsulation;
- Analysis: with this pattern we ease changing because it ensures that a change doesn't affect the rest of the system.
- Implications: it requires that there are as many processes as functionalities to represent in the system.

Pattern 2 – Each process' internal action should do an update to an informational entity

- Information Type: Business;
- Attributes: Encapsulation;
- Analysis: with the informational entities' update, each action is saved by updating the informational entity.
- Implications: it requires an informational entity able of being updated for each action in a process.

VI. Demonstration

The practical challenges to address are related to the Portuguese Public Procurement Process. For this, we made an Achimate modeling of the public procurement system architecture using NST application in the enterprise domain in order to show the advantages and benefits of its use in heterogeneous and competitive information systems.

For a better section's organization, it is important to build an AS-IS, where we represent the current state of the public procurement system, i.e., we represent, architecturally, the platform's operations, the communications, the stakeholders and the system's existing processes. Then, we will present two TO-BE scenarios representing all AS-IS elements but integrating two different solutions to the problem. One creating a new integration platform (Message Broker) and another in which communications were made directly between all platforms without the creation of any other component. So, and with these two possible solutions, we see the changes made by NST and advantages of using this theory in an architecture.

A. AS-IS

In this subsection, will be demonstrate an architecture associated to the actual public procurement system in Portugal. Applying the mapping described in the previous section, the following figure represents an example of these applications.

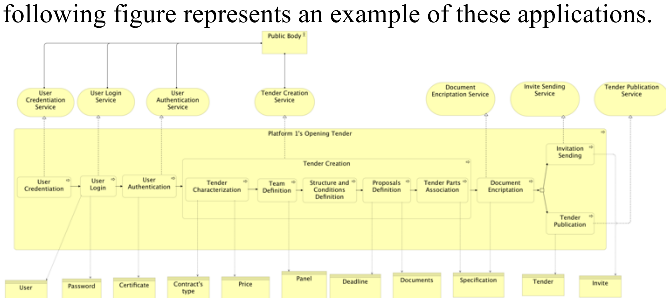


Fig. 2 Tender's Opening Business Process

The main process represents Platform 1's Opening Tender constituted by business service and business objects. In this modulation we used different workflow elements that correspond to business process. Beyond this element, two more NST's elements are used. We applied the Concerns Separation and States Separation theorems, using the Business Process Viewpoint. Both patterns are applied in this process. Besides this modeling, it is important to show the platform's independency in this process, showing the existing business problem in Portugal, where invitation sending and tender publication are executed exclusively on the platform where the public body is registered.

In the following figure, we present a public procurement system's global vision in order to enhance the independency between the platforms and the existing problem with which it is associated.

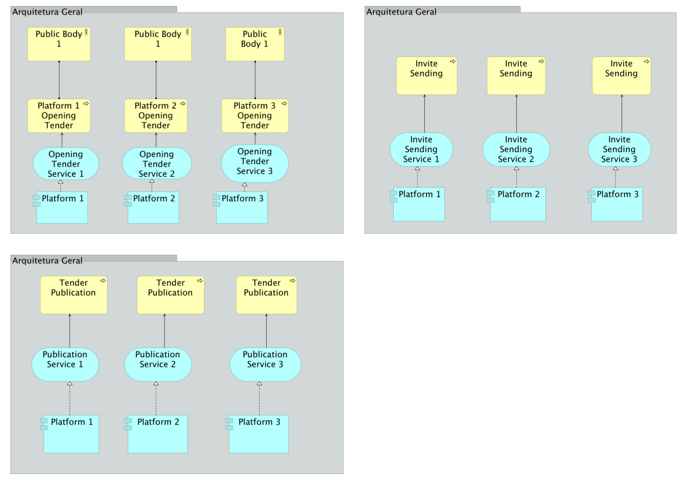


Fig. 3 Architecture's Global View

B. TO-BE A

Having finished the representation of the current state of the public procurement system (through the AS-IS architecture), in this subsection we will present a possible solution for the public procurement platforms' integration in order to solve the existing business problem in Portugal, TO-BE A. As for the AS-IS, this architecture will apply the Workflow Element mapping and the two architectural patterns created.

The following figure represents an example of such application.

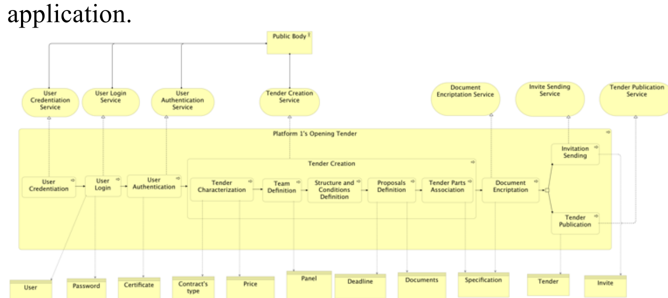


Fig. 4 Tender's Opening Business Process

The main process represents the Platform 1's Opening Tender constituted by business service and business objects. In this modulation many workflow elements that correspond to the business process are used. Beyond this element, we used two more NST's elements. We applied the Concerns Separation and States Separation theorems, through Business Process Viewpoint. Both patterns are applied in this process.

Besides this modeling, it is important to show the solution found to introduce the functional requirement desired by Portuguese government.

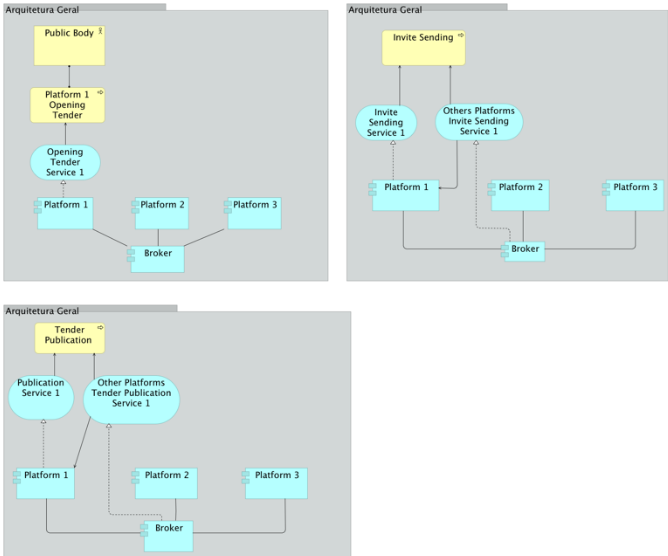


Fig. 5 Architecture Global View with Broker

C. *TO-BE B*

Alternatively to the solution presented in the previous subsection, we will present another possible solution for the public procurement platforms integration, in order to solve the existing business problem in Portugal, TO-BE B. As for AS-IS and TO-BE B, we will present the modeling of the selected business process.

Applying the mapping described for Workflow Element and the two created architectural patterns, the following figure represents an example of such applications.

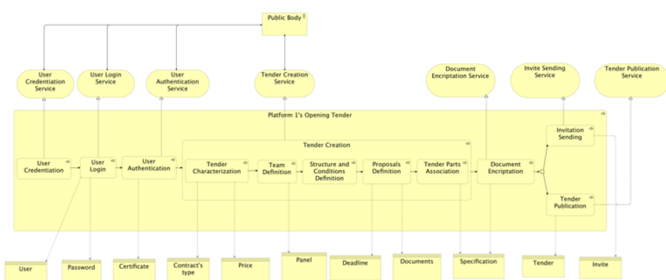


Fig. 6 Tender's Opening Business Process

The main process represents the Platform 1's Opening Tender constituted by business service and business objects. In this modulation we used many workflow elements that correspond to the business process. Aside from this element, we used two more NST's elements. We applied the Concerns Separation and States Separation theorems, using Business Process Viewpoint. Regarding enterprise standards, they are both applied in this process.

Besides this modeling, it is important to show the solution found to introduce the functional requirement desired by the Portuguese government.

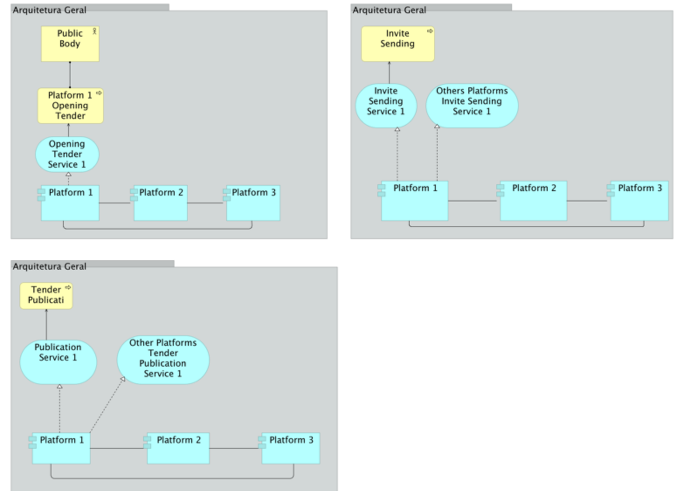


Fig. 7 Architecture Global View with directly communications

VII. EVALUATION

This section corresponds to the DSRM Methodology's evaluation section presented in this document's beginning. This activity aims to perform a validation and evaluation of the demonstration created in the previous section by comparing an architecture with a business problem and two possible solutions to solve it. It is intended to verify the NST application to the business domain and also evaluate which of this solutions can better solve the Public Procurement System problem, i.e., which exposes in a more obvious way the NST's qualities and characteristics.

A. *ISA's quality metrics evaluation*

In this subsection we will evaluate if the three develop architectures, AS-IS, TO-BE A and TO-BE B, meet the quality requirements according to the NST's characteristics. Each figure will be achieved by applying metrics, each one corresponding to a quality, thus allowing to check which one is more normalized and more suitable for responding to the problem.

As NST tells us that systems must be evolvable over time, they should be prepared for changes and also stable, it is necessary to adapt the quality metrics that we intend to evaluate in the three architectures.

- Functionality
  - Suitability
    - BSRPF – Business Service Required and Provided Factor
  - Interoperability
    - DIIEF – Different Implementation Informational Entities Factor
- Maintainability
  - Analyzability
    - SCCF – Complexity Ciclomatic Factor
  - Changeability
    - LCOISF – Cohesion Lack in Application Components

        - NOISF – Number of Operations in Application Components
    - Testability
        - RSF – Service Response Factor
- Dimension
    - NE – Informational Entities Number
    - NA – Applications Number

However, not all metrics have the same weight for this Public Procurement System problem and therefore it is necessary to perform an importance calculation of these qualities represented in the following table. The relative weight is calculated by dividing the value attributed to the importance level and other characteristics importance level sum.

TABLE VII
NST'S PATTERNS MAPPING FOR ENTERPRISE DOMAIN

Quality	Importance (0-10)	Relative Weight (0-100 %)	Subquality	Importance (0-10)	Relative Weight (0-100%)
Funcionality	7	50%	Suitability	6	42,9 %
			Interoperability	8	57,1 %
Maintainability	7	50%	Analyzability	6	27,2 %
			Changeability	10	45,6 %
			Testability	6	27,2 %

1) BSRPF

This metric measures the alignment and suitability of the services provided by IS to the business processes. This analysis is done counting all services that require process and aren't matched by business services and the processes matched by business services, that aren't implemented in any component [1]. Making the respective calculations, the results are:

- BSRPF(AS-IS) = 2/7
- BSRPF(TO-BE A) = 2/7
- BSRPF(TO-BE B) = 2/7
- BSRPF(AS-IS) = BSRPF(TO-BE A) = BSRPF (TO-BE B)

2) DIIEF

Through this metric we intended to measure the number of different implementations for each informational entity. Each informational entity may correspond to other entities that implement it [1]. Doing the math, we obtained following results:

- DIIEF(AS-IS) = 1
- DIIEF(TO-BE A) = 1
- DIIEF(TO-BE B) = 1
- DIIEF(AS-IS) = DIIEF(TO-BE A) = DIIEF(TO-BE B) = 1

Therefore, in this case, each informational entity is implemented only in a single low level informational entity.

3) SCCF

For the Software Engineering area, the larger the paths number, the more complex the control flow will be. This metric measures the ISA complexity according to the support of the available business services. This service's complexity is measured by counting the difference between the number of dependencies and applications used in support [1]. Applying this metric to the two architectures, the results are:

- SCCF(AS-IS) = 1
- SCCF(TO-BE A) = 0,89
- SCCF (TO-BE B) = 1
- SCCF(AS-IS) = SCCF(TO-BE B) > SCCF (TO-BE A)

4) LCOISF

This metric measures the correlation between application blocks and informational entities used. It is quantified by the average of number of informational entities that are used by different functionalities in the same application [1]. The results are:

- LCOISF(AS-IS) = 0,949
- LCOISF(TO-BE A) = 0,958
- LCOISF(TO-BE B) = 0,96
- LCOISF(TO-BE B) > LCOISF(TO-BE A) > LCOISF (AS-IS)

5) NOISF

The ISA's Changeability is maximized when the change impact of each operation is confined to an application block. Applying this metric [1], the results are:

- NOISF(AS-IS) = 0,5625
- NOISF(TO-BE A) = 0,59
- NOISF(TO-BE B) = 0,55
- NOISF(TO-BE A) > NOISF(AS-IS) > NOISF (TO-BE B)

6) RSF

This metric counts the number of application blocks that can be invoked in support of a particular service. According to recent researches, it is suggested that each business process must be supported by the lowest possible number of applications [1]. The results are:

- RSF(AS-IS) = 0,5
- RSF(TO-BE A) = 0,472
- RSF (TO-BE B) = 0,5
- RSF(AS-IS) = RSF(TO-BE B) > RSF (TO-BE A)

7) NE

The ISA's size will grow in direct proportion to the value of this metric. The Informational Architecture's size is a direct measure of the number of informational entities, measured by this metric. This metric also presents a strong interrelation with the ISA's maintainability [1]. The results are as follows:

- NE(AS-IS) = 11
- NE(TO-BE A) = 11

- NE(TO-BE B) = 11
- NE(AS-IS) = NE(TO-BE A) = NE(TO-BE B)

8) NA

The ISA's size will grow in direct proportion to the value of this metric. Using this metric, it is possible to estimate the ISA's size in the application level [1]. The results are:

- NA(AS-IS) = 43
- NA(TO-BE A) = 54
- NA(TO-BE B) = 49
- NA(AS-IS) < NA(TO-BE B) < NA(TO-BE A)

B. Evaluation Conclusion

In order to evaluate the qualities presented in each architecture, the AS-IS, TO-BE A and TO-BE B were subjected to a quality metrics' evaluation to determine which can be more normalized and solve this Public Procurement System problem.

As for the functionality quality, measured in terms of adequacy and interoperability, the architectures present similar results and are, therefore, adequate for this practical scenario.

For the Maintainability quality, measured by Analyzability, Changeability and Testability, there are some differences between the architectures. For Analyzability, the AS-IS and TO-BE B are similar and present the maximum value while TO-BE A presents a slightly lower value. For Changeability, TO-BE A has a slight advantage over the two other architectures. And, finally, for Testability AS-IS and TO-BE B are equal and TO-BE A presents a slightly lower value.

In dimensional accordance, the three architectures have the same number of informational entities. As for the number of applications, TO-BE A has a larger value and therefore it becomes a more complex architecture than the others. Based on relative importance of each architectural quality (indicated in table 7), the following table contains the global evaluation.

TABLE VIII
GLOBAL EVALUATION TABLE

Total	AS-IS	TO-BE A	TO-BE B
	72%	76%	72%

Thus, the TO-BE A solution has a higher total quality. However, the values are very close among the three architectures. The decisive factor for the advantage presented by TO-BE A was this architecture's superior value in Changeability, because this sub quality has a more relative weight.

VIII. CONCLUSION

Nowadays, organizations tend to focus their attention towards planning the system, prior to development. Therefore, enterprise architectures assume a leading role towards success and system reliability [9], [10].

In order to enable the construction of reliable systems that can be deployed in a way that produces value for organizations, these need to be evolvable and ready for changes that may occur on a strategic/planning level in the organization [9], [10].

With that in mind, the Normalized Systems Theory is a technique capable of guaranteeing stability for any system and it predicts the possibility of change in these systems, simplifying any modification that can happen [23].

With this work, we applied NST to the domain of enterprise architectures. So, with the mapping of the elements of the theory to the concepts of ArchiMate modelling language, we can develop normalized architectures, capable of supporting the development of new solutions for the Public Procurement System. Through the application of NST, we can integrate the existing public procurement platforms and make the access to these platforms a fair process.

With the related work section, we aimed to show how public procurement works in the EU, going into further detail in three countries whose public procurement mechanism is similar to Portugal's. We also explain the normalized systems theory, which is the foundation of our work. We describe its origin, application and functioning, when it comes to software and enterprise architectures. Furthermore, we also described the ArchiMate modelling language, in which the architecture for the new system will be modelled. Since any architecture should be built according to standards, we also approached that topic, describing the global features of these standards in the enterprise domain.

After analyzing and comprehending all topics that evolve this theme, we formulated the problem which is the focus of this work: how to develop a system architecture that can simplify the access from all economic bodies to all public tenders, independently of the platform used, applying the NST.

The proposed solution is based on the application of NST in enterprise architectures through mapping between the theory and the enterprise modelling language. First we mapped the theory's basic elements to ArchiMate, enabling the application of those concepts in the construction of an architecture with the specificity of the modelling elements to use, in order to build a solution that's based on NST. Afterwards, we mapped the theory's four theorems to the enterprise domain, by choosing the viewpoints to use in the system architecture's development. Lastly, we mapped and created architectural rules that represent guiding lines that should be followed when building an architecture.

With the purpose of demonstrating the applicability of this solution in enterprise architectures, we created two alternative architectures, which are able of providing a solution to our problem. Architecture TO-BE A has a new component, a Message Broker that handles the communication between the platforms. On the other hand, architecture TO-BE B promotes communication directly between the platforms, without any other intermediary.

To sum up, we believe that the NST can provide an answer for developing evolvable agile architectures, thus contributing to the organizational development that is essential for success in any organization. As for the practical case, it can integrate the existing platforms, building solutions capable of adding the needed functional requirement that serves the business interest of the Public Procurement System in Portugal.

REFERENCES

[1]. A. Vasconcelos. "Arquitecturas dos Sistemas de Informação: Representação e Avaliação (PhD Report)". Instituto Superior Técnico, Portugal, 2007.

[2]. A. Hevner, S. March, J. Park e S. Ram. “Design Science in Information Systems Research”. *MIS Quarterly,* Vol. 28 No.1, March 2004.

[3]. K. Peffers, T. Tuunanen, M. A. Rothenberger e S. Chatterjee. "A Design Science Research Methodology for Information Systems Research". *Journal of Management Information Systems,* Vol. 24 No.3, 2007.

[4]. A. Hevner e S. Chatterjee. “Design Research in Information Systems”. Springer, 2010.

[5]. D. D’Hooghe, A. Carton. “Public Procurement 2016”. [Online]. Available: http://www.iclg.co.uk/practice-areas/public-procurement/public-procurement-2016/belgium. [Accessed: 25-Abril-2016]

[6]. T. Biachi e V. Guidi**,** “The Comparative Survey on the National Public Procurement Systems across the PPN”, Instituto Poligrafico e Zecca Dello Stato, Roma, 2010.

[7]. E. Eessaar, “On Applying Normalized Systems Theory to the Business Architectures of Information Systems”, Tallinn University of Technology, Estonia, 2014.

[8]. P. Huysmans, D. Bellens, K. Ven e N. Dieter, “ Designing Enterprise Architectures Based On Systems Theoretic Stability”, International Conference on E-Business (ICE-B), Greece, 2010.

[9]. A. Vasconcelos, A. Caetano, P. Sinogas, R. Mendes e J. Tribolet, “Arquitectura de Sistemas de Informação: A ferramenta de Alinhamento do Negócio/Sistemas de Informação”, INESC Inovação, Portugal, 2002.

[10]. A. Vasconcelos, P. Sousa e J. Tribolet, “Information System Architecture Evaluation: From Software to Enterprise Levels Approaches”, Instituto Superior Técnico, Portugal, 2005.

[11]. The Open Group, "ArchiMate 2.1 Specification", The Open Group, 2013.

[12]. H. Mannaert e J. Verelst, **“**Normalized Systems - Re-Creating Information Technology Based on Laws for Software Evolvability,*”* Koppa, Antwerp, 2009.

[13]. UMIC**.** “Programa Nacional de Compras Electrónicas,” Ministério da Educação e Ciência, Portugal, 2003.

[14]. A. Varela, “Avaliação da Teoria de Sistemas Normalizados na Implementação de SI Evolutivos (master thesis report)”. Instituto Superior Técnico, Portugal, 2011.

[15]. R. Castelão, “An Information Architecture for the Public Administration (master thesis report)”. Instituto Superior Técnico, Portugal, 2010.

[16]. B. Gomes, “Information Architecture and Enterprise Ontologies – An Enterprise Ontology Approach for Defining the Enterprise Information Architecture (master thesis report)” Instituto Superior Técnico, Portugal, 2011.

[17]. P. Alves, “Arquitecturas de Sistemas de Informação de Referência para Atendimento Multicanal na Administração Pública (master thesis report).” Instituto Superior Técnico, Portugal, 2011.

[18]. European Commission, "Annual Public Procurement Implementation Review 2012", European Commission, 2012.

[19]. D. Van Nuffel, H. Manneart, C. De Backer, J. Verelst. “Towards a Deterministic Business Process Modelling Method based on Normalized Systems Thoery”. University of Antwerp, Belgium, 2010.

[20]. The Open Group, "TOGAF Version 9.1", The Open Group, 2011.

[21]. D. Greefhorst e E. Proper, **“**Architecture Principles – The Cornerstones of Enterprise Architecture*”*. Springer, Berlin, 2011.

[22]. S. Assar, I. Boughzala, “E-procurement platforms in the French public sector”. GET/INT Information Systems Department, France, 2006.

[23]. D. Van Nuffel, “Towards Designing Modular and Evolvable Business Process”. Department of Management Information Systems, Universiteit Antwerpen, 2011.

[24]. European Commission, “Electronic Public Procurement in EU Member States: Country Reviews”. European Comission, 2004.

[25]. European Commission, “Study on administrative capacity in the EU Lithuania Country Profile”. European Comission, 2015.

[26]. European Commission, “Study on administrative capacity in the EU Sweden Country Profile”. European Comission, 2015.

[27]. P. De Bruyn, **“**Generalizing Normalized Systems Theory: Towards a Foundational Theory for Enterprise Engineering*”*, Universitas, University of Antwerp, 2014.

[28]. P. De Bruyn, D. Van Nuffel, J. Verelst, H. Mannaert. “Towards Applying Normalized Systems Theory Implications to Enterprise Process Reference Models”. Department of Management Information Systems, University of Antwerp, 2012.