PROASIS 2.0: Collaborative Mechanisms to Update Technology Architecture

Pedro Jacinto

1 Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais, Lisboa, Portugal

Keywords: Enterprise Architecture, Update, Technology Architecture.

Abstract: This work aims to solve the misalignment between the representation of Enterprise Architecture (EA) and the reality of organization. This issue is very important, because the decision-makers of organizations use the EA to take decisions and if EA doesn’t correspond to the reality of organization these decisions can be harmful for organization. Although there are some theories about how maintain the EA updated, there are no tools to help organizational actors to maintain EA updated in a collaborative way. This work aims to contribute to the definition of a collaborative process to update the technology architecture in a continuous way. This process is supported by a tool that was developed for representing EA in the most reliable way of order update all layers of EA. This process will be applied on AMA (Agência de Modernização Administrativa), a Portuguese public organization.

1. INTRODUCTION

Presently many organizations need tools and instruments to help them to maintain competitive advantages between other organizations to survive in the enterprise world. One of these tools are Enterprise Architecture (EA), but often EA doesn’t reflect the reality of organization, mainly because is out-of-date. To solve this problem organizations need a process to update their EA in a continuous way.

This is particularly important, in organizations that use EA in their decision process and with an out-of-date EA, because these decisions can be harmful to organizations's future.

EA is used like a strategic concept and not like an initiative in short term, in other words, EA has to belong to the culture of the organization. “The process of implementing EA specifically to a particular organization is called institutionalization” (Song & Yeong-Tae, 2010).

This work follows the work done in (Castela, N.; Dias, P.; Zacarias; J. M., Tribolet, 2012) with the definition of PROASIS 1.0 which is the process to update the Business Process Architecture in a collaborative way with the usage of annotations. With this process we provide organizations with an instrument to help them to maintain their Technology Architecture updated to thereby reflect the reality of organizations. Another important goal of this work is to provide an instrument to all organizational actors take part in this process. This is important too, because in one hand, if more people participate in this process is more easily to detect misalignments between the representation and the reality, in the other hand, if people can participate in this process they feel more embedded in the organization, this could lead to better accept the EA and see how important it is for the organization.

In section 2 we describe the problem we try to solve.

In section 3 we describe the research methodology followed in this work.

In section 4 we describe the related work.

In section 5 we describe the proposal to solve the problem defined in section 2.

Finally, in section 6 we describe some results of this work and in section 7 we make some conclusions about the work done.

2. PROBLEM DEFINITION

Accordingly with the problem of out-of-date EA, the final product of this work is the definition of a
process to update the Technology Architecture, which belongs to EA, in a collaborative way. This process aims to solve some problems that automatic tools don’t solve, like the identification of some .Net Framework 2.0 applications. “Major Problems: On the occasions that some architectural component was not found (apart from "nodes" that have always been discovered), it is always due to lack of a specific monitor loaded in the autodiscovery application. In the SI shown above, there was one of those situations, particularly for .NET Framework 2.0. The more coverage in monitors, the greater is the usefulness of the approach developed. We believe, however, that we achieved a significant and relevant coverage that failed only in this case and that the production of a user-made monitor update for this application would imply a very small effort.” (Soares, 2011). This process aims to solve another issue, like someone says for example: “The organization needs to buy a new server.” In this situation, if the purchase is approved, the organizational actors can automatically update the Technological Architecture. Without this process the new server is added to the organization, but in the Technological Architecture model this new server doesn’t exist.

With the definition of this process the following questions must be answer:

Q1: Can we have a Technology Architecture constantly updated?

Q2: Can be the update process used on the Enterprise Environment?

The first question must be answered with the usage of this process and compare the results after the application of the process and before.

The second question must be answered with the application of the process in an organization. The organization is AMA (Agência de Modernização Administrativa), this is a public organization.

3. RESEARCH METHODOLOGY

The research methodology chosen to define the process is Action Research Methodology, because is a rigorous methodology to get some valid scientific results in a systematically and rigorous way. (Vasconcelos, 2001). This was also chosen because in this methodology the researcher has a near involvement with the organization that tries to solve the problem and with this research methodology the researcher gets near involvement and fast results and feedback.

In (Rapoport, 1970) the action research is described as a research methodology that tries to solve a concrete problem (action) and tries to get some knowledge with a process and increase the base of scientific knowledge (research). (Vasconcelos, 2001)

In Fig. the steps of action research methodology are shown.

4. RELATED WORK

In this section we start to describe a framework to define the Enterprise Architecture (EA), next creation process of EA, phase H of TOGAF (ADM) Framework, Enterprise Architecture Management (EAM) and the base of this work PROASIS 1.0 developed in (Castela, N.; Dias, P.; Zacarias; J. M., Tribolet, 2012).
4.1 ArchiMate Framework

In this section we make a description of ArchiMate Framework to do a contextualization about what is Enterprise Architecture (EA).

The architecture is seen by (IEEE, 2000) as “The fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.”

EA is the specification of the components of the organization (departments, divisions, etc.), the relationships between them and the relationship between organization and the environment. EA provides a lot of views of the organization depending what is the target public or a particular group of stakeholders. (Group, 2012)

ArchiMate Framework is divided in three main layers: Business Layer, Application Layer and Technology Layer. Each layer uses the components of below layer. Business uses components of Application and Application uses components of Technology. In Business Layer is defined the business process, in Application Layer is defined the applications which support the business of organization and in Technology Layer is defined the hardware where applications run, network and servers needed to the applications of organization.

Environment

Business
Application
Technology

Passive structure Behavior Active structure

Fig. 3- ArchiMate Framework. (Group, 2012)

The core elements of ArchiMate Framework that each layer has are: the active structure elements, passive structure elements and behaviour elements. The active elements are the elements which perform actions; the passive elements are the elements where active elements perform actions and behaviour elements are the actions that active elements perform. In this work ArchiMate Framework is the basis of Technology Architecture and every models use this Framework.

4.2 Creation Process to build EA

In (Land, et al., 2009) the creation process to build EA is divided in three main phases: creation phase, apply phase and maintenance phase.

In creation phase the main objectives are the identification of stakeholders and what the client/organization wants and expectations in relation to the final product.

In apply phase we need to apply the EA to the organization. In this phase is more important to make EA belongs to the culture of the organization. (Land, et al., 2009)

In maintenance phase we update the EA every time organization changes, like new applications, new business process, etc. This phase is the most important in the scope of this work. In the next section we describe with more detail this phase with the phase H of TOGAF (ADM).

4.3 TOGAF (ADM) Framework

In this section we describe the TOGAF (ADM) Framework and with more detail one of the TOGAF’s phases, phase H, because is very important in the scope of this work. TOGAF (ADM) is a Framework divided in eight main phases:

- Phase A – Architecture Vision
- Phase B – Business Architecture
- Phase C – Information Systems Architecture
- Phase D – Technology Architecture
- Phase E – Opportunities and Solutions
- Phase F – Migration Planning
- Phase G – Implementation Governance
- Phase H – Architecture Change Management

As we said before, in the scope of this work, the most important phase is Phase H. In that phase is verified if the architecture of Phase G is aligned with the strategic needs of organization, and in this phase is essential to separate big changes from small changes, because big changes implies another ADM cycle beginning in Phase A. (Group, 2011)

In Phase H the main objectives are: ensure that the architecture lifecycle is maintained, ensure that the Architecture Governance Framework is executed and ensure that the enterprise Architecture Capability meets current requirements. (Group, 2011)
4.4 PROASIS 1.0

In this section we aim to describe what PROASIS 1.0 is and why we try to extend this process to Technology Architecture.

In (Castela, N.; Dias, P.; Zacarias; J. M., Tribolet, 2012) PROASIS 1.0 is a process to update the Business Process Architecture in a collaborative way. This process can update Business Process in a collaborative way through the usage of annotations in modelling elements. In PROASIS 1.0 there are three kinds of annotations: correction, detail increase and adaptation. (Castela, N.; Dias, P.; Zacarias; J. M., Tribolet, 2012)

In PROASIS 1.0 three levels of detail are defined: organizational unit, business process and activity. The stakeholders can make annotations, at each level of detail. Where is defined who is the reviewer or reviewers of annotations. Each time one element is annotated that annotation enters in a negotiation process defined in (Castela, N.; Dias, P.; Zacarias; J. M., Tribolet, 2012). This negotiation process is the core of PROASIS 1.0, because if one annotation is accepted by the owner of the business process where that annotation is made, it contributes for the update of the Business Process Architecture. With this mechanism the organizational actors are the key agents in the update of Business Process and we can reduce the gap between the representation of Business Process Architecture and the reality of organization.

We choose to extend this process to Technology Architecture, because (Soares, 2011) identified some problems that automatic tools can’t solve, like the identification of some .Net Framework 2.0 applications and the distinction of virtual or native components.

This process is used as the basis for the definition of a process to update the Technology Architecture.

5. PROPOSAL

In this section we aim to describe one possible solution for the problem: How to maintain the Technology Architecture constantly updated? In order to solve this problem we choose to extend the model of PROASIS 1.0 and apply it to the Technology Layer of Enterprise Architecture.

The first step of this proposal consists in two assumptions:

- First we need to define the levels of detail in Technology Architecture and identify the stakeholders of each detail level;
- Second, in each level of detail identify the owners, reviewers and evaluators of annotations, these roles will be delivered to the stakeholders of each level.

After that, we use the action research cycle and in each cycle we apply the negotiation process of PROASIS 1.0 and evaluate the results, if is necessary we adapt the negotiation process to obtain better results in Technology Architecture.

With this process we aim to maintain the Technology Architecture updated and transform organizational actors in active part of this process. Other advantage that this process could bring is that’s the organization can saves money and time, because if the Technology Architecture model changes and this change isn’t reflected in the model, that model doesn’t reflects the reality of organization, to update that model, so what we need to do is another model and to do that we spend time and money more than if we do one simple change in the old model.
One scenario where this process can be useful is, for example, if one stakeholder of IT department thinks that is important for the organization to buy a new server, he can make an annotation in Technology Architecture saying that. After this, annotation enters in a negotiation process, and in the end can two things happen: one, if annotation is accepted, the organization buys a new server and the Technology Architecture model is updated in an automatic way; two, the annotation is rejected, and the Technology Architecture Model remains unchanged.

We needed to adapt the process defined in PROAIS 1.0 due to, the main difference between Business Layer and Technology Layer defined in ArchiMate Framework that states, that in model Business Process Architecture is necessary to make some analysis about what are the organization’s processes and what are their relationships and in Technology Architecture some tools can be used to automatically discover the objects of Technology Architecture.

### 5.1 AUTOMATIC TOOL (SPICEWORKS)

In this section we aim to describe the tool used to discover some elements to model Technology Architecture.

This tool is the basis to discover some objects to model Technology Architecture and create the first model of Technology Architecture. This tool uses IP range to make auto-discovery of objects and create a database Sqlite 3. With this database, we match the database tables and the meta-model (AMA’s meta-model) used to define Technology Architecture. The most important elements discovered by the tool are: devices, software and services.

The table of devices has some attributes that are very useful to build Technology Architecture: name, primary owner name, type, manufacturer, description, model, bios date, processor type, memory, operating system and serial number. To complete the information of devices on meta-model, we use information of other tables. Those tables are: physical disks and network adapters. On those tables we use information about disk capacity and about the device’s network adapters.

To complete software information, we use the information of software table. On that table we use the following fields: name and vendor. To know which device corresponds some software, we use the device id on device table and on software table, on software table is a foreign key of device table.

In order to add some elements to meta-model, we choose Virtual Machines, because in nowadays it’s a very common practice in order to reduce infrastructure costs. This is very important in public organizations. To complete the information of virtual machine we use the virtual machine table. On that table we use the following attributes: asset device id, name, vcpu count, memory, hdd capacity, mac addresses, on off, created at, updated at, guest ip addr and bios uuid. Asset device id corresponds to id of device where virtual machine is running this is a foreign key of device table.

With the usage of this automatic tool we considerably reduced the time on analysis comparatively with the traditional Technology Architecture building methods.

### 5.2 ARCHI (ARCHIMATE MODELLING)

In this section we aim to describe Archi. This tool was used to show the model of Technology Architecture.

This tool has the capacity to create objects, relations and models. It’s free and provides all ArchiMate Objects, Relations and Models. In the scope of this work we only need Technology Objects, Association Relationship to create only a model, the Technology Architecture Model. This tool is used like an accessory tool to demonstrate the As-Is Model of Technology Architecture.

### 5.3 MySQL DATABASE

MySQL is an open source database. This database is used in this work to save elements of the Technology Architecture, their relationships, annotations and revisions. Technology Architecture elements are store in the database with the representation of Meta-Model.

### 5.4 TOOL FOR MAPPING SPICEWORKS ELEMENTS AND NEGOTIATION PROCESS

In this section we describe the tool developed on this work for mapping Spiceworks elements into Meta-Model elements, their relationships and for implementings negotiation process to update Technology Architecture.

This tool has the capacity to map Spiceworks elements into meta-model elements. In this module, it uses the mapping identified on section 5.1 and makes the mapping in the automatic way without any effort for the tool user.

Another feature of this tool is the capacity to make annotations to the objects of the meta-model with the intention of update the Technology Architecture. These annotations are store in MySQL
database and are available for reviewing annotations with this tool too.

Other feature is the capacity to create models that can be read by Archi and can be also seen automatically with Archi.

With this tool we can map Spiceworks elements into meta-model elements, relations between elements, update Technology Architecture and visualize Technology Architecture. This tool keeps elements, relationships, annotations and revisions on MySQL database. The viewer used to see Technology Architecture is Archi.

5.4.1 NEGOTIATION PROCESS

In this section we describe the negotiation process developed in PROASIS 1.0. With this negotiation process we can update a model in a collaborative way, because this process use the annotation mechanism to annotate model’s elements for other users of model see what is wrong or not.

![Fig. 6- Negotiation Process](image)

This process was used to create a collaborative process to update Business Process Architecture. To use this process to update Technology Architecture we have made some adaptations, because we can use automatic tools to build Technology Architecture As-Is Model. In our case we use Spiceworks to build the As-Is Model of Technology Architecture.

To use this negotiation process to update Technology Architecture in collaborative way, is necessary to build As-Is Model of Technology Architecture. To do so we use Spiceworks to make some inspection on network to find some devices, virtual machines and other elements of Technology Architecture. After this, we use the tool developed on scope of this work to map these elements into the Meta-model used in this work and store these elements in MySQL database. After that, we can visualize the Technology Architecture model on Archi.

Then we can make changes on the model with the help of the tool developed in the scope of this work. This tool provides a mechanism to do annotations about Technology Architecture elements. Any enterprise actor can make an annotation in one element. After this step, any enterprise actor can make n revisions about that annotation. In each revision we can say if we agrees or not with that annotation. To conclude the process, the organizational actor responsible for the Technology Architecture, can accept or not the annotation. If the annotation is accepted, the Technology Architecture Model is automatically updated and it can be visualized in Archi. If the annotation is rejected, the Technology Architecture Model remains the same. At any time we can visualize the As-Is Model of Technology Architecture.

This negotiation process provides collaboration, because any organizational actor can propose improvements to As-Is Model of Technology Architecture or declare misalignments between As-Is Model of Technology Architecture and the reality of organization.

6. RESULTS

In this section we describe some results after and before the application of negotiation process. We describe the whole process, since the creation of As-Is model and after the approval and rejection of some annotations.

The first step is run Spiceworks tool to create a basis for Technology Architecture. After that Spiceworks creates a db file with the results. With the tool developed in this work, it’s possible to save these results into MySQL database.

![Fig. 7 - Main Menu of Tool Developed](image)
With model saved into a MySQL database, we can make changes with the help of negotiation process described before.

On the main menu, if we choose “Criar Novo Modelo” (Create New Model), it opens a dialog to choose a file db, in our case we choose Spiceworks results file, and it creates in the MySQL database a model with the results. After that we can visualize the model on Archi with one click on “Carregar Modelo Guardado” (Load Saved Model), and it opens the model saved into MySQL database on Archi.

After filling out all the fields we click on “Confirmar Anotação de Artefacto” (Confirm Artefact Annotation) and this action creates an annotation on our database. At any moment we can see all annotation of a specific element, its revisions and the elements of model.

![Fig. 8 - Example of a saved Model visualized on Archi](image)

After we get a model into our database we can make changes in the model. To do that we choose on main menu the option “Fazer anotação sobre o modelo guardado” (Make annotation). This option opens a page with all the possible elements of the model. In our case we choose “Artefacto” (Artefact).

![Fig. 9 - Page with all possible elements](image)

Where we choose the Artefact element it opens another page where we can choose the kind of Annotation (create new element, change existing element or delete an existing element). In our case we choose create new element, because the saved model doesn’t have Artefacts.

![Fig. 10 - Main page of Artefact Annotation](image)

After the creation of annotation we can make n revisions. When the responsible person accepts the annotation the model is automatically updated, if the annotation is rejected the model remains unchanged. To accept annotation we choose on main menu option “Confirmar Anotação” (Confirm Annotation). It opens a dialog with all objects, we choose Artefact, because we create an Artefact Annotation, choose the id of Annotation we create and then click on “Confirmar Anotação” (Confirm Annotation).

![Fig. 12- Confirm Artefact Annotation](image)

After annotation acceptance we can visualize the model again and, as we can see on Fig. 13, the model is automatic updated.

As we can see, with this process the task of keeping the Technology Architecture updated is more easy and fast than making all tasks of creation a new Technology Architecture. In this process the responsible person for acceptance of rejection of the annotations is the same for Technology Architecture and this depends of each organization. In this process every person of organizations can make annotations and revisions.

![Fig. 13 - Model updated](image)
7. CONCLUSION

In this paper is proposed a process to update Technology Architecture in collaborative way. This process begins with an automatic tool that discovers some elements of Technology Architecture and saves those elements into a SQLite3 database. We develop a tool to map these elements into a meta-model. After that we can see the Technology Architecture model discovered by the automatic tool (Spiceworks, in this case). After the automatic discovery and visualization of the model, the tool developed saves those elements into MySQL database on meta-model specification. With the elements saved into MySQL database, we can make annotations about those elements and change or not the Technology Architecture model. If that annotation is accepted the Technology Architecture is automatic changed, if not the model remains unchanged.

To conclude, as said before, does not exist a tool to change Technology Architecture in collaborative way. With this work we try to minimize and if possible eliminate this gap on organizations. The ultimate goal of this process is the involvement of organizational actors to contribute for a Technology Architecture up-date to help the decision-makers in decision making.

8. REFERENCES

Land, M. O. et al., 2009. Enterprise Architecture Creating Value by Informed Governance. s.l.:Springer.
Song, H. S. & Yeong-Tae, 2010. Enterprise Architecture Institutionalization and Assessment. s.l., s.n.