A BOTTOM-UP APPROACH FOR THE REPRESENTATION AND CONTINUOUS UPDATE OF TECHNOLOGY ARCHITECTURES IN PUBLIC ADMINISTRATION
Position Paper

João Soares, José Tribolet and André Vasconcelos
Instituto Superior Técnico, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal

Keywords: Information systems architecture, Technology architecture, Public administration, Bottom-up, Continuous update, Autodiscovery.

Abstract: In organizations where there are not any concerns about updating Information Systems Architecture (ISA) models in a strict and automatic fashion, business requirements and pressures frequently lead to an increasing gap between current Information Systems and the latest version of their ISA and, in particular, their Technology Architecture (TA) models, when they exist. One of the causes for the lack of "investment" in TA models is the considerable effort required to "discover" infrastructural features that serve as input to the development of architectures. This work is focused on devising a bottom up approach for the representation and continuous update of TAs in Public Administration (PA) using tools to automatize the discovery of architectural evidences (such as *logs* or network events) and a process based on the annotation mechanism to enable interaction contexts that allow actors to make explicit their knowledge about their activities through graphic representations.

1 INTRODUCTION

As result of a crescent globalization, competition among organizations is becoming more and more driven by it's capacity and flexibility of reorganization and redesign of business processes and business strategies (Vasconcelos, 2007).

Some of the implications that arise from the need to reorganize and redesign business processes and business strategies reflect themselves on the demand for IT able to support those new requirements.

Enterprise Architecture (EA) (and by association, ISA and TA) is considered an essential piece to the resolution of the challenges that emerge as a result from the organization's need quickly readapt themselves and their Information Systems (ISs) (Land, Martin Op 't, et al, 2008). Thus, the explicitness of an ISA has a key role in the construction of ISs that aim to contribute proactively to the alignment between business realities and IT realities, to define technology strategies, to control and manage more efficiently ISs, to ensure durable, flexible and adaptable to business needs ISs, and finally to increase the portability of applications and ease the development and alignment of ITs with (changing) business strategies.

This investigation will be focused in a more restrict problematic inside EA - the reality regarding technologies managed by organizations to support their processes. At a first glance, this work will cover tree great issues: the *capture* of that reality, the *mapping* between the results of the capture and architectural models and finally, the continuous *update* of those models.

This document is divided in 5 sections. Section 2 establishes the objectives and motivations of this work. Section 3 is about the State of the Art regarding the areas of interest that intersect with this work. In section 4 there is a description of the architecture of the solution. Finally, section 5 has conclusions over the work developed until now.

2 CLARIFICATION OF THE PROBLEM AND OBJECTIVES

Considering the previously identified issues related to the benefits of a general IT awareness in an organization, it is possible to conclude that, although there is

Soares J., Tribolet J. and Vasconcelos A..
A BOTTOM-UP APPROACH FOR THE REPRESENTATION AND CONTINUOUS UPDATE OF TECHNOLOGY ARCHITECTURES IN PUBLIC ADMINISTRATION - Position Paper.
DOI: 10.5220/0003419504670470
In *Proceedings of the 13th International Conference on Enterprise Information Systems* (ICEIS-2011), pages 467-470
ISBN: 978-989-8425-56-0

a common agreement in the advantages and potential arising out of the specification of an ISA, it is still not possible to represent that architecture at different levels and their relationship with the business model in a standardized, universal, simple and simultaneously with the richness and rigor required for subsequent inspection and/or simulation of different business scenarios and technology (Boar, B., 1999).

More specifically, in the context of PA, there is a lack of global knowledge of the current ISA, nor are there any *automatic methodologies* or processes defined to collect and identify architectural evidences. Those evidences would then be a valid input to the construction of a TA, resulting in a *bottom-up* approach, as individual base elements of the system are first specified in great detail.

Taking this context as a starting point, the fact that drives this work lies in precisely the lack of a global knowledge and collaborative update of the current ISA models in PA. To address this problem, there is one question that rises up in the first place, and constitutes the problem of this work: *"What is the TA of PA?"*. This question is clearly too much broad, so our work will be focused in a particular context of the Portuguese PA, but the techniques to be applied will be as much transversal as possible, in order to be replicated into other contexts.

The question that frame this investigation can then be decomposed into other two modular questions:

- *"What architectural evidences should be considered in order to construct a TA?"'* The answer to this question will lead to a well defined set of architectural evidences, ranging from, *e.g.*, simple *logs* produced by servers, to network events. The approach to to so, will take into account different IT contexts of the Portuguese PA.
- *"How to automatically collect and identify architectural evidences in PA?"* This is one of the challenges of this work, because it will allow an efficient, fast and ruled collection of data, which will be a major improvement regarding the creation of TAs from scratch. On the other hand, in cases where architectures are already modeled, will allow them to be much more maintainable since the process of incorporating changes will be much more agile.
- *"How to map architectural evidences into updated, trustworthy and reliable TA models?"'* The answer to this question urges because there is a recognized difficulty in maintaining models updated and aligned with the reality.

3 RELATED WORK

3.1 EA Methods and Frameworks

An enterprise architecture framework collects together tools, techniques, artifact descriptions, process models, reference models and guidance (methods) used by architects in the production of enterprise-specific architectural descriptions. A framework can also define *views* that are representations of systems from the perspective of a related set of concerns (*e.g.*, top management or operational), and have a dedicated modeling language or extend existing ones.

3.1.1 Zachman Framework

Zachman Framework was introduced by John Zachman in 1987. It was the first EA framework and it is used in numerous domains, providing a view of the subjects and models needed to develop a complete enterprise architecture. In a “big picture”, this framework is a logical structure (matrix) for classifying and organizing the descriptive representations of an enterprise: on a vertical axis it provides multiple perspectives of overall architecture, and on a horizontal axis, a classification of the various artifacts of the architecture (Zachman, 2008) (Lankhorst and *et al*., 2009a)
.

The purpose of this framework is to provide a basic structure which supports the organization, access, integration, interpretation, development, management and changing of a set of architectural representations (*artifacts*) of the organization’s ISs.

As a contribution to this work, the Zachman Framework can be quite useful as a tool for organizing and classifying relevant architectural artifacts that will be developed during this investigation. The intersection between the last two rows of the matrix (*Detailed Representations* and *Technology Model*) and the three first columns (*Data*, *Function* and *Network*) can be assumed as a repository of representations about a technical infrastructure that can serve as a baseline for the construction of an ISA.

3.1.2 TOGAF

The Open Group Architecture Framework (TOGAF), a framework developed and maintained by The Open Group Architecture Forum and its members, is a global standard for assisting in the acceptance, production, use and maintenance of architectures based on a iterative process (Josey, 2009) (The Open Group, 2009).

TOGAF covers the development of four related types of architecture, subsets of a global EA: *Business Architecture, Data Architecture, Application Architecture* and *TA*.

There are two important components of TOGAF according to the scope of this project (The Open Group, 2009):

- The **Architecture Development Method** (ADM) plays a central role in TOGAF and describes cyclically how to develop a complete enterprise architecture, providing guidance on a number of levels (Lankhorst and *et al*., 2009b):
- The **Enterprise Continuum** comprises a number of reference models and methods, such as the *Technical Reference Model*, *The Open Group's Standards Information Base* and the *Building Blocks Information Base*, for classifying architecture and solution artifacts, showing how they evolve and correlate to each other.

The Technical Reference Model (TRM), which provides a model and taxonomy of generic platform services, can be used in this work to structure a coherent description of the components that are part of the TA of an IS.

3.1.3 Framework CEO 2007

Proposed by Vasconcelos in (Vasconcelos, 2007) this framework features a taxonomy of concepts used to define a ISA (architectural primitives).

The meta-model of CEO 2007 is organized in the following architectures: Business Architecture, Organizational Architecture and Information Systems Architecture

This framework will serve as the conceptual basis for the description of TAs and ISAs during this work, because it was developed in the context of the group CODE[1], as this work is.

3.2 Application Discovery and Dependency Mapping

Application discovery is the process of automatically analyzing artifacts of a software application. *Dependency Mapping* creates visibility between applications and infrastructure dependencies.

The use (and possible extension) of software tools that implement these features could be a major value-adding element in the pursuit of the objectives of this investigation. Leveraging the power of automation, automated application discovery and subsequent dependency mapping, can capture, connect and unveil intricate relationships including the way in which applications behave and relate to the TA on which they rely.

Automated application discovery normally leverages an effective non-agent-based discovery method - *passive discovery* - to automatically discover the structural and behavioral aspects of applications. Passive discovery operates by watching the network traffic, examining all the applications as they execute, and then identifying relationships based on real business usage. Passive discovery also means that the management burden of maintaining software agents is null.

Passive discovery is just a part of the equation regarding these systems. Some products often use a hybrid approach, where *active discovery* is also used to enable the collection of deep and fundamental data about configuration details.

3.3 A Dynamic Update Process

The present section is about describing a process to continuously update an enterprise model (Castela et al., 2010). In the process explained in this section, organizational actors act as active updaters of the AS-IS model, through the comparison between the modeled activities and the ongoing real executed activities, becoming key agents in reducing the gap between the organization and its representation.. The following text expresses the essence of PROASIS (business process model dynamic updating process): "The "client" of PROASIS (corresponding role of the operational model that detects misalignment between the model and "reality") wants to update the model, so it makes an annotation (update request). This update request is received by the modeler (which is who actually update the model if the annotation is approved) and by the reviewers. When reviewers receive the annotation, they can begin the revision of the annotation (which is optional). The approval of the annotation is made based on the analysis of the annotation (update request) and reviews. If the annotation (request update) is approved, the model will be updated and delivered to the "client"". Besides the fact that PROASIS is suited to business processes, it can be adapted in this work in order to map architectural evidences into updated, trustworthy and reliable TA models. The idea of using an extended version of PROASIS in this work enables a collaborative maintenance of the TA models originally incipiently build using mechanisms of autodiscovery. On the other hand, PROASIS and the annotation mechanism could be used in an earlier stage: to "decide" within an or-

[1]Center for Organizational Design and Engineering - INESC

ganization, what would be the most commonly accepted language to represent TAs.

4 SOLUTION'S ARCHITECTURE AND METHODOLOGY

This section is about describing the steps that the solution's architecture will follow.

According to the questions that constitute the problem that drive this investigation, as well as all the related work described earlier, following, there is a description of each stage/landmark of the proposed solution facing the initial motivation to this problem:

- **Contexts.** It is the phase where different relevant contexts (entities that manage IT assets) are analyzed in order to propose a structure of concepts to describe the manifestations of IS and its TA exhibited through architectural evidences, considering terms introduced by *TRM* and *CEO 2007*.
- **Analysis & Evaluation.** In this stage, two activities take part:
  - Analysis of CEO 2007 regarding it's capability of expressing detailed TA aspects (*e.g*, regarding network connections, firewalls, *etc.*).
  - Analysis of *Application Discovery tools* regarding it's extensibility potential.
- **"Products".** This is the phase where "visible work" is done regarding tree facets:
  - *CEO 2007:* TA extensions; Mapping between the structure of concepts; Adaptation of the continuous cycle of monitoring and verification of the IS and verification of it's TA reality to the reality of automatic discover of architectural evidences and continuous collaborative update.
  - *Application discovery tools:* Necessary extensions.
  - *PROASIS:* Adaptation to TA representation and maintenance.

The architecture of the desired bottom up approach can then be described briefly in tree steps:

1. Use PROASIS prior to any representation, to agree, inside an organization, on the notation to represent the TA.
2. Perform *Autodiscovery*.
3. Use PROASIS as repository to the early representations and tool to support collaborative update.

5 CONCLUSIONS

EA (ISAs and TAs in particular) are regularly relegated to a secondary role in the context of organizations, due to factors such as *counterculture*; it is not considered as an essential piece regarding the organization's survival; it is a job that consumes a certain amount of time and requires considerable effort, and the companies, eager to get quick results, tend to see architecture as an effort too high with too many actors involved within the organization.

This last reason has particular relevance in the context of PA due to funding reasons: money from financing is much more easily allocated in areas/project that have the potential to produce "visible" impact.

This work began by the clarification of the problem together with the motivations towards the solution. Regarding the related work, several EA Methods and Frameworks were revisited according to the potential to be used in this investigation as well as *Application Discovery* tools (to quickly capture the infrastructural reality of an organization) and a *process* to enable collaborative update of TA models.

The course of this investigation will follow the steps described in ´the previous section to achieve a bottom up approach.

REFERENCES

Boar, B. (1999). *Constructing Blueprints for Enterprise IT Architecture*. John Wiley & Sons, New York.

Castela, N., Zacarias, M., and Tribolet, J. (2010). Business Process Model Dynamic Updating. *CENTERIS 2010*.

Josey, A. (2009). TOGAF™Version 9 Enterprise Edition. White Papper, The Open Group.

Land, Martin Op 't, et al (2008). *Enterprise Architecture (The Enterprise Engineering Series): Creating Value by Informed Governance*, page 5. Springer.

Lankhorst, M. and *et al*. (2009a). *Enterprise Architecture at Work*, pages 25–26. Springer, Heidelberg, second edition.

Lankhorst, M. and *et al*. (2009b). *Enterprise Architecture at Work*, pages 26 – 29. Springer, Heidelberg, second edition.

The Open Group (2009). *TOGAF™Version 9*. Van Haren Publishing.

Vasconcelos, A. (2007). *Arquitecturas dos Sistemas de Informação: Representação e Avaliação*. PhD thesis, Instituto Superior Técnico.

Zachman, J. (2008). John Zachman's Concise Definition of the Zachman Framework. *Zachman International*.