Enterprise Transformation: The Serasa Experian Case*

Pedro Sousa[1], Ricardo Gabriel[2], Gustavo Tadao[2], Rosana Carvalho[3], Pedro Miguel Sousa[3], and André Sampaio[1]

[1] Link Consulting, Avenida Duque de Ávila, nº23 1000-138 Lisboa, Portugal
[2] Serasa Experian, Alameda dos Quinimuras, 187, CEP 04068-000 São Paulo, SP, Brasil
[3] Aitec Brasil, Rua Luigi Galvani, 200, CEP 112 04575-020 São Paulo, SP, Brasil
pedro.sousa@link.pt, ricardo.gabriel@br.experian.com, gustavo.tadao@br.experian.com, rosana.carvalho@aitecbrasil.com.br, pedro.r.sousa@aitecbrasil.com.br, andre.sampaio@link.pt

Abstract. This paper presents the development of an Enterprise Architecture project at Serasa Experian. With a strong transformation plan Serasa needs tools to monitor the progress of the strategic roadmap. With this project Serasa achieved an integrated management of architectural views, projects results, and the roadmap progress. It is also brings IT Architects, Project Managers and IT Governance together, identifying concepts and a common language between them, thus facilitating the aggregation of information from these groups into a consolidated vision. Project completions are the drivers for updating the architectural views and progress of the roadmap. A knowledge base keeps the information gathered from external information sources and supports the automatic generation of architectural views, which can be browsed back and forward in time.

Keywords: Enterprise Transformation, Enterprise Architecture Case Study, Temporal Architectural Views.

1 The Situation

The Serasa Experian[1] (hereafter designated by Serasa) is the biggest credit bureau in the world outside the USA, holding the largest data bases in America Latina about consumers, companies and economics groups. With over 40 years of experience in the market, the company is involved in most credit decisions taken in the country, corresponding to 4 million queries each day, done by over 400 thousand clients.

Serasa makes available integrated solutions that cover client needs regarding credit data: Market Prospecting, Customer Management, Retention and Profitability, Procurement and Concession of Credit Portfolio Management, Credit Management and Billing Fraud and Validation.

* The statements and opinions expressed in this paper are those of the authors and not an official position of Serasa Experian.

[1] www.serasaexperian.com.br

F. Harmsen, K. Grahlmann, and E. Proper (Eds.): PRET 2011, LNBIP 89, pp. 134–145, 2011.

Serasa has a strategic plan for the entire IT that is embodied in roadmaps for the key business and technologic areas of the company. These roadmaps set the targets to be achieved for each area over the next five years. These targets are defined in terms of a maturity level measured on a 5 level scale (non-existent, initial, defined, managed, and optimized).

At the end of each year, each area submits a plan with the initiatives proposed for the coming year. If approved, these initiatives become actual projects. Then, one year after, each area evaluates the progress achieved and updates the roadmap accordingly. Projects are transformation elements of the organization, and so project achievements drive the progress of roadmaps. Roadmaps are manually maintained in numerous Excel Sheets; therefore updating the roadmaps according to projects' results is a complex and time consuming process.

Project planning requires knowledge of the actual state of the organization, nonetheless, such knowledge is scattered across numerous fonts, and there is no simple mechanism to consolidate disparate information into an integrated view of the actual state of the organization. Consequently, the project planning should include a survey of the As-Is of the project related areas. However, this is insufficient. Since there are multiple parallel projects the identification of the project impact cannot be established taking only into account the project related As-Is, it also ought to consider the foreseen future (To-Be) accomplished by parallel projects (on-going and planned) up to the day the project is set to production [6]. Therefore, the project planning should also include a survey of on-going and planned projects that may affect the project in question. Without taking the previous considerations into account the probability of finding dependencies between projects solely during the execution is indeed high, which can lead to an increase in projects costs and execution time, thus jeopardizing the roadmap's success.

In early 2011, Serasa has initiated an Enterprise Architecture project, perceived as an instrument to control and monitor the transformation of the organization achieved in each project. This project was led by the Corporate Architecture team[2], which is comprised in the Department of Information Technology. The IT areas involved in this project were: IT Governance and Systems Development, Security and Infrastructure, and Corporate Architecture.

2 The Project

Our task was to design, plan and implement an instrument that enables the integrated management of architectural views, projects results, and the roadmap progress. More specifically, this instrument should allow Serasa to:

- Have an integrated and up-to-date view of the As-Is of business and IT architecture, by consolidation of information from different sources.
- Have a view of the To-Be state of the business and IT architecture based on the foreseen results of on-going and planned projects.

[2] See Serasa "*Case empresa: Arquitetura Corporativa, uma gestão integrada do Roadmap de TI*" presented in the conference: www.congresso-ae.com.br/programacao.php.

- Have the ability to identify the progress of each roadmap updated according to the projects' progression.

The following figure presents a generic view of the instrument. Basically, it imports textual information from various information sources, generates architectural views and keeps roadmap progress up-to-date. The instrument is built upon two tools: IBM's Rational System Architect and Link Consulting's Enterprise Architecture Management System[3] (EAMS). System Architect serves primarily as the knowledge-base. Information integration and architectural views generation is achieved via EAMS.

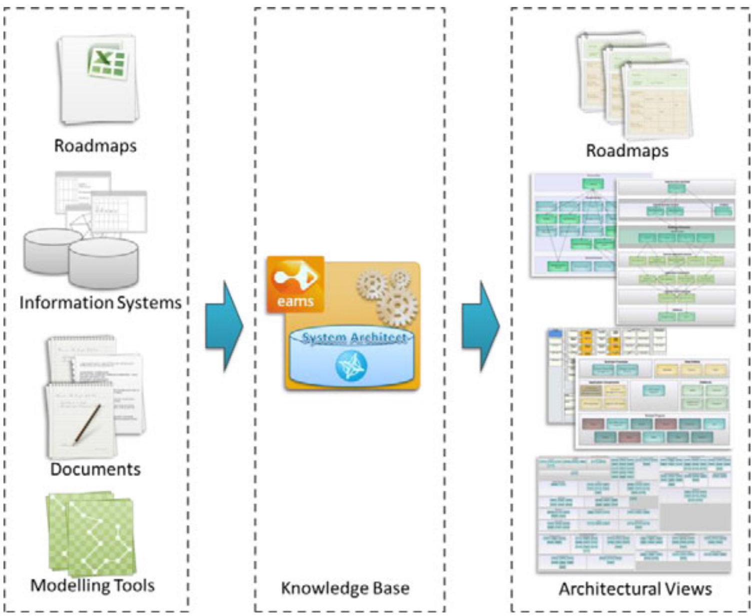

Fig. 1. Overview of the instrument to implement

2.1 Information Integration

Regarding the subject of Information Integration, our task was to import the information into the knowledge base, via integration with other information sources scattered throughout the organisation or by introducing the information directly into the knowledge base.

Whenever the organization already has the processes to maintain and update a particular source of information, one should provide the automatic import mechanism to integrate it with the knowledge base. Otherwise, one must decide between creating a new information source or entering the information directly into the knowledge base, and introducing its update process to the organization.

---

[3] www.link.pt/eams

Information sources can be highly structured such as information systems, less structured such as modelling tools, intranet pages, or even documents such as Microsoft Office (Word, Excel, PowerPoint and Visio), among others. Information integration must be made from structured sources. Therefore integrations with unstructured sources require prior structuring of the same. The roadmaps' Excel Sheets are an example where the structure is done by creating new Sheets filled with lookups and other Excel functions, creating tabular records ready to be imported.

2.2 The Knowledge Base

Regarding the subject of the knowledge base, our task was to prepare a repository that supports the evolutionary vision of architecture and its connection with projects and roadmaps. The knowledge base keeps the information that describes the organization as textual information and enables the automatic generation of architectural views [5].

A necessary characteristic of the knowledge base is that its entities have a life cycle that is associated with the project concept. We consider projects as the elements that induce the organization changes, thus being responsible for applying the changes to the organization's artefacts and relationships [6]. Such entails that every planned change results from a project. By planned change we state that the changes in the organization occur as result of activities with a planned start and end date. Unplanned changes are considered a priori in our approach, hence implying that a project has produced those changes in some past date.

Projects have a start date, an end date, and two lists with references to other artifacts. The alive list references the artifacts that will go into production (*alive*) with the project completion, and the dead list reference the artifacts that will be decommissioned (*dead*) after project completion. Using the projects start date and an end date, we tag each artifact within the knowledge base with tree time-stamps:

- *Gestation*, when it is being developed within a given project.
- *Alive*, when they are put into use within the organization as a result of a project.
- *Dead*, when they are no longer used, also as a result of some project.

These time-stamps are necessary to allow the generation of architecture views related to any time in the past, in the present or in the future.

2.3 Architectural Views and Roadmap

Here, our task is to define and implement the graphical outputs/models that represent the roadmap progress and the representation of Serasa's enterprise architecture. A key aspect is that these outputs must to be generated automatically from knowledge base contents.

As a general rule, we envisage the use of ArchiMate Viewpoints [1][3] for architectural views, but we also require a more generic approach to generate architectural views of any type of system, in the general sense.

Our approach, provided by default in EAMS, considers four types of viewpoints that can be used to analyse any set of artifacts in the knowledge base. We call them *Organic*, *Context*, *Structure* and *Integration* [6]. These views describe the elements and dependencies of a system defined by a set of elements in a graph of a larger system [5]. For example in Fig. 2 we present the system *S01* as an ellipse and in figures 3-7 we present the matching generic views.

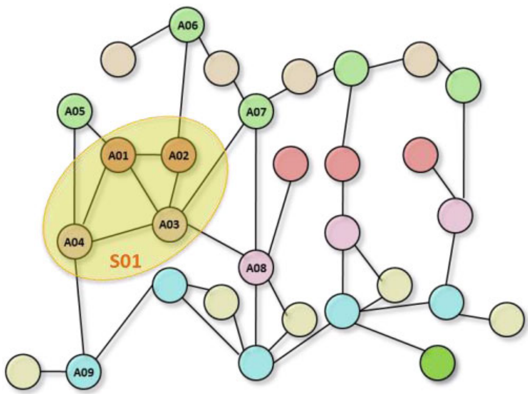

Fig. 2. System boundary of S01

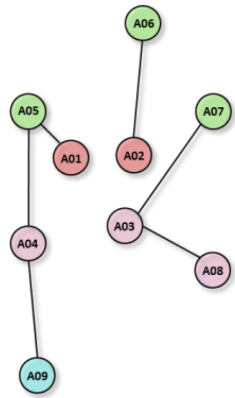

Fig. 3. Integration view of S01

The *Integration* system identifies the set of influence bonds among the members of the composition and the elements in the environment, and also the elements covered by these bounds. The *Organic* is focused in a hierarchical decomposition that identifies the artifacts in the composition of a system. The *Context* encompasses the environment of a system based on the relationship bounds established with its composition, and the *Structure* covers the artifacts in the composition and the set of structural bounds among the artifacts.

Fig. 4. Organic view of S01

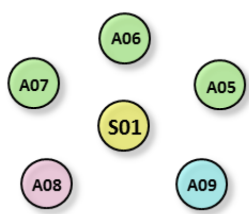

Fig. 5. Context view of S01

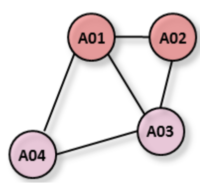

Fig. 6. Structure view of S01

In this general perspective, roadmaps correspond to a set of initiatives that relate to the artifacts to be created in each of them. Regarding the example in Fig. 2, the S01 roadmap would be complete when the initiatives A01-A04 created the artefacts A05-A09.

3 The Approach

The approach we used was designed to mitigate the main project risks, which were the following:

- Too much time spent on the definition of the knowledge base meta-model.
- Inadequacy of the meta-model's level of detail and the matching information sources. The complexity of the knowledge base meta-model should be suitable to the immediate needs and the existing information sources.
- Time spent adjusting unstructured information and manually guarantying the information quality.
- People can lose confidence in the project if they do not see short-term results.

Thus, we adopted an incremental approach by dividing the problem into several stages. The project began with an analysis and planning stage from which resulted in eight further stages.

3.1 Planning Stage

The analysis and planning stage includes the following steps:

1. Collect relevant issues from the different areas, and assess its priority.

   In this step we identified and prioritized 82 issues after workshops with each area. Some of these issues were high level statements, such as "*What are the costs of planned maintenance in the coming years?*" or detailed ones, such as "*What were the architectural artefacts produced in each project?"*.
2. Define knowledge base meta-model to hold information required to answer the identified issues. This is a quite straight forward exercise for the detailed issues. For high-level issues, one needs to unfold them into a set of detailed issues, and identify the information related to each one. For the example above, a set of more detailed questions could be: "*What is estimate regarding the resources required for the operation of each system, as disk?, mips?, bandwidth?,..?.* ".
3. Identify the best sources of information for each entity in the knowledge base meta-model.

   In a few workshops with each of the areas, 16 sources of information were selected. In parallel with this step, we did the installation and general configuration of the tools used, in particular the meta-model configuration.
4. Plan the whole project.

   The questions were clustered into eight groups, according to the related entities of the meta-model, thus resulting in eight further stages, presented next.

3.2 Further Stages

After the analysis and planning stage, one started the remaining stages, in sequence, each with having the following steps:

1. Analyse the corresponding information sources and find a way to extract the necessary information.

   The key factor is the degree of structure in each information source. For highly structured sources, the extraction is a quite straightforward task. For unstructured sources, one must first propose a structure and re-structure the data.

The structuring of information sources is based on eliminating the information in natural language and record all the information in specific tables whose cells only have predefined values. For example, the document describing the architecture of a *system A*, instead of having a natural language sentence such as "*The system A reads the information from the repository B through the middleware C*", could instead have a table named "*accessed repositories*" with the fields *Repository source* and *Middleware technology*, in which the values are filled based on pick lists with the allowed values for repositories and technologies, that in this example would be *B* and *C*.

But the structuring of an information source also involves the clarification of the semantics of each used artifact type. In the example above, one must define the semantics of the *Repository* and *Middleware* and ensure its existence as an entity in the meta-model. Thus it also becomes clear what dependencies may exist between artifacts. Continuing the previous example, it was necessary for the meta-model to hold a relationship between the entities *Repository* and *System*, and that this relationship may be qualified by the *Middleware technology*.

2. Implement a mechanism to feed the knowledge base with information residing in information sources.

   We prefer to import via editable files (like CSV or XML) because it allows a more controlled environment. One can even edit and correct wrong information before importation. The contents of each source of information were exported to conforming to a CSV format, so that responsible areas could validate the content using Microsoft Excel. The detected errors can be corrected later in origin.

   The importation mechanism must handle many situations that arise from imports, for example, repeated artifacts, artifacts with extra fields, default values, and so on. The importation mechanism must also address the rules to set the timestamps. For example, whenever a new artifact is imported the *alive* timestamp is set to the completion of associated project. If the project does not have a completion date, or if the artifact is not associated with any project, then the *alive* timestamp is set to the loading date.

   In addition, EAMS allows the definition and management of jobs and batches to handle multiple and sequential file importation thus enabling the load process to occur in well-defined moments.

3. Define and implement appropriate views to respond to relevant issues.

   As mentioned, we favour the use of well-established viewpoints, either the ones provided by ArchiMate, or the ones supported by default by EAMS. We found that in fact most answers can be found with these viewpoints.

   Another important aspect is the definition of the navigation path between representations according to the stakeholders' concerns [2], allowing different stakeholders to navigate to different graphical models after interacting with the same artifact in the same architectural view.

From the eight stages planned we are completing the first two. The time and effort depends mostly on the structuring of information sources. With clean and well-structured sources one can expect to take between 2 to 4 weeks for each stage.

4 The Results

The immediate result of this project is to provide to the various areas of IT a consolidated and updated view of the architecture of Serasa. Consolidated - because it results from the information provided from different areas. Updated - because it results from information sources maintained by the different areas with clear processes in an automated manner.

Since each group contributes with information to the other areas and receives information from others, the instrument entails an important transformation in the organization, as it homogenized languages and tools. In this regard, the knowledge base meta-model is an important asset because it identifies the concepts common to the various areas. The meta-model is illustrated in Fig. 7.

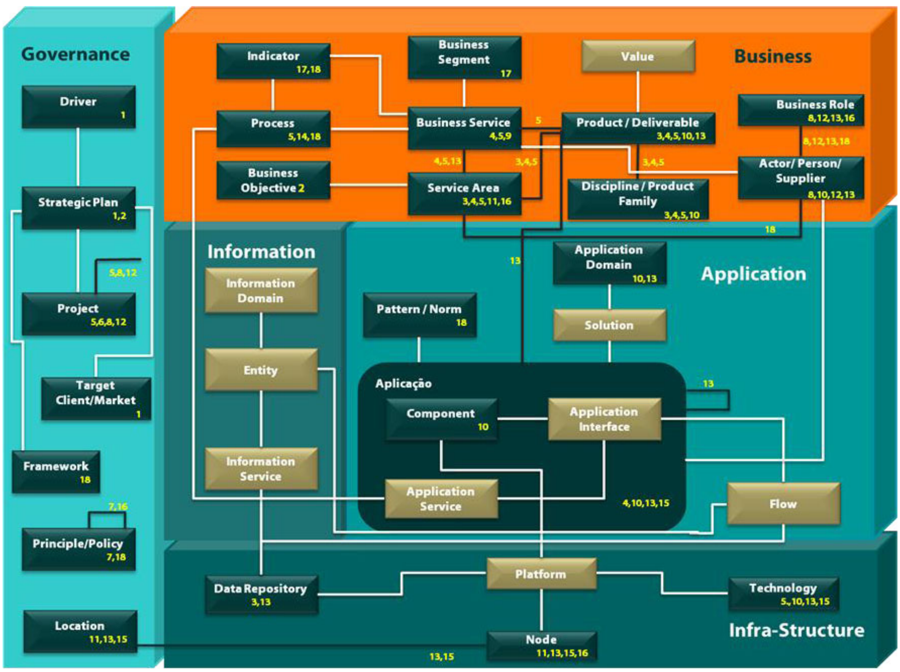

Fig. 7. Identified meta-model

The meta-model was derived from the 82 most relevant questions presented by the key areas. To answer these questions we have reached a model with 32 entities, allowing all but two of the issues to be answered. Both the entities and the relationships between them were somewhat influenced both by TOGAF [4] and the ArchiMate [3].

To populate these entities we have identified 16 sources of information, 9 of which are collections of Microsoft Office documents (PowerPoint, Excel and Word), and 7 are systems and tools (ChangePoint, SharePoint, BizAgi, Visio, and three other specific systems). After thorough analysis of potential information sources, we found that most of the information needed to populate each entity in the meta-model is in more than one source, and in different formats. The numbers depicted in Fig. 7 represent the information sources associated with each entity.

An entity without a number means that we found no information source within the company holding the necessary information. We have 9 of such cases. In these cases it is necessary to identify the processes of Serasa where this information is produced and where it should be recorded. This allows the definition of the artifact's description that will be created in the future. The information survey regarding the artefacts that already exist implies a human endeavour that should be raised incrementally, as needed by on-going projects.

We now present examples of the architectural views generated, but with fictions values in order to hide Serasa actual IT. In addition to the more common approach of static visualization the solution allows interaction with the representations. Stakeholders may interact with produced blueprints by selecting and inquiring information about artifacts and navigating between blueprints.

The blueprint in Fig. 8 presents how an application "*Customer Order Management Application*" is organized regarding its *composition* (components distributed by application layer) and identifies both the structural bounds between the artifacts and the environment and the artifacts in the environment (platforms where the components are executed).

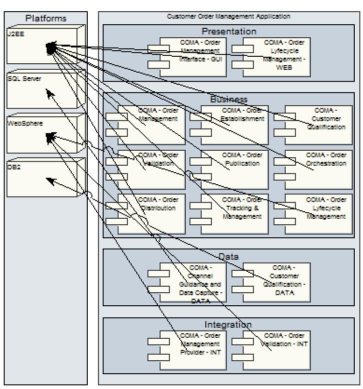

Fig. 8. Application Structure blueprint

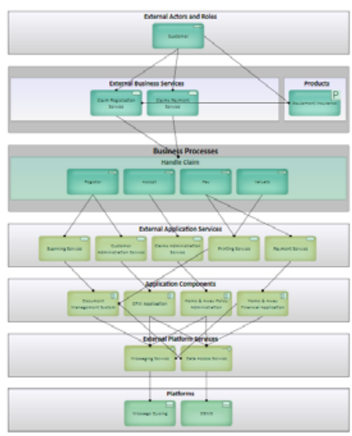

Fig. 9. Layered blueprint

Another example of an architectural view is represented in Fig. 9 this blueprint is an example of an instantiation of an ArchiMate viewpoint, the Layers Viewpoint [3].

Regarding the topic of information integration, Fig. 10 is an example of information automatically extracted from normalized documents that was provided to generate the blueprint presented in Fig. 8.

	A	B	C	D	E	F	G
1	Type	Name	Gestation Date	Start Date	End Date	Layer	Platform
2	Component	COMA - Channel Guidance and Data Capture - DATA	20-09-2010	15-12-2010		Data	SQL Server
3	Component	COMA - Customer Qualification	20-09-2010	15-12-2010		Business	J2EE
4	Component	COMA - Customer Qualification - DATA	20-09-2010	15-12-2010		Data	DB2
5	Component	COMA - Order Distribution	20-09-2010	15-12-2010		Business	J2EE
6	Component	COMA - Order Establishment	20-09-2010	29-11-2010		Business	J2EE
7	Component	COMA - Order Lyfecycle Management	20-09-2010	27-01-2011		Business	WebSphere
8	Component	COMA - Order Lyfecycle Management - WEB	20-09-2010	27-01-2011		Presentation	J2EE
9	Component	COMA - Order Management	20-09-2010	29-11-2010		Business	J2EE
10	Component	COMA - Order Management Interface - GUI	20-09-2010	29-11-2010		Presentation	J2EE
11	Component	COMA - Order Management Provider - INT	20-09-2010	29-11-2010		Integration	WebSphere
12	Component	COMA - Order Orchestration	20-09-2010	15-12-2010		Business	J2EE
13	Component	COMA - Order Publication	20-09-2010	15-12-2010		Business	J2EE
14	Component	COMA - Order Tracking & Management	20-09-2010	27-01-2011		Business	J2EE
15	Component	COMA - Order Validation	20-09-2010	15-12-2010		Business	J2EE
16	Component	COMA - Order Validation - INT WebSphere	20-09-2010	15-12-2010		Integration	WebSphere

Fig. 10. Partial view of information obtained from normalized project documents

An important result of the project is the time navigation. All architectural views have a time slider associated, in which are marked the moments in time in which there were projects that produced[4] a change in that architectural view. When you move the handle along the slider, and cross a mark, the name of the project that led to changes appears on the left and the content of the architectural view changes in accordance with those changes. An example of this is presented in the top time slider depicted in Fig. 11. If one splits the handle in two, as depicted in the bottom of the same figure, then we specify a time window and the slider presents the name of all projects that have an impact on that architectural view on that time interval, and the contents of the architectural view changes accordingly.

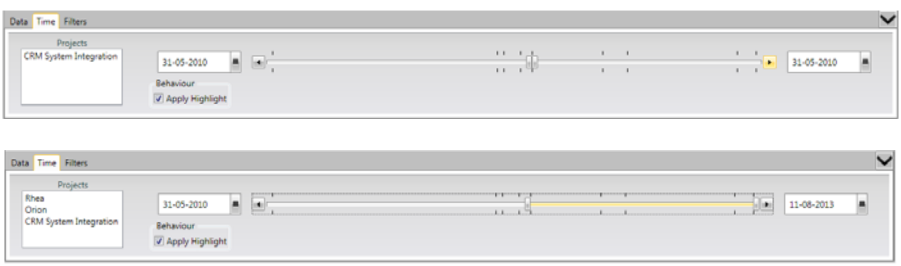

Fig. 11. The time slider specifying a point in time (top) or a time window (bottom)

In figures 12 and 13 we present a Context blueprint of the *Campaign Management Application*. This application is positioned in the centre and the artifacts in its surroundings are those having dependencies with it. In the figure, going clock-wise from top-left corner, the related artifacts types are: Processes, Informational Entities, IT Platforms, Provided Applications, Consumed Applications and Projects. The contents change from left to right-hand side based on the position of time slider, stating, for example, that additional processes the application will support on December 2011.

The blueprints for the roadmaps are not yet completed, but follow a similar philosophy. The current status of each roadmap will be displayed by the colours of

4 In fact the verb could be in the past, present or future, depending if the handle is positioned in the past, present or future.

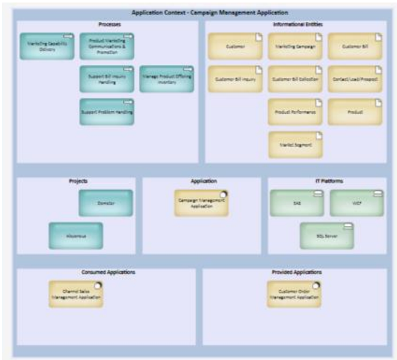

Fig. 12. Visualization for 1-06-2011

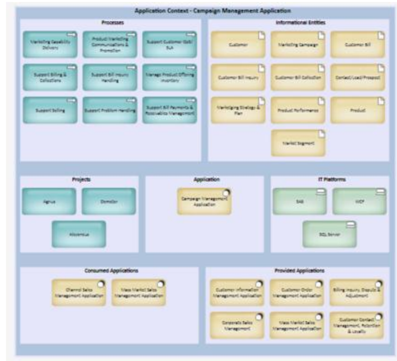

Fig. 13. Visualization for 01-12-2011

each project and, by moving the slider to a given point in time one can see the state of the roadmap at that time (actual state if in the past and foreseen state if in the future).

5 Reflection

The challenge of creating a representation of the enterprise architecture of a changing organization is not small. Making alive/updated representation of that architecture is an even greater challenge. Current practice consists on having modelling tools with which the architects manually design the architectural models of the organizations. Such is valid only for the creative phase, when one is conceiving and designing the proposed future. But the price of using this approach, the models draw by hand, is high, because when they change, someone will have to re-draw them again by hand. And this has been a well-known problem of all organizations.

Thus, in the case of Serasa, we use the opposite approach, and consider the architectural views are generated automatically because it is the only way we can keep them updated .Not wanting to argue that the previous statement is an absolute truth, we claim however that the automatic generation of architectural views is an approach much more suited to the current maturity of organizations, where "architectural chaos" is in place and where the representations are in fact the very limited.

Still, gathering all the required information needed to generate the blueprints is a huge challenge because it forces the organization to have reliable information sources, even if they are scattered and unstructured. In the case of Serasa, it was clear that about 12 of the relevant questions remain unanswered due to lack of information. Serasa will have to create these information sources and adopt practices that keep them reliable. The main difference is that now there are 12 good reasons to justify such effort. The effort associated with extracting information from existing sources of information may not be small, especially for documents and unstructured sources. But once again it is clear that the inherent value.

But the existence of the relevant issues is not only a motivation to do a corporate architectural project. It is also a necessary condition! In fact, when it comes to the design of the knowledge base meta-model, without a number of questions which are able to set a clear purpose and scope of the project, discussion of the metamodel is a process that tends to have no end, because it is no longer clear what is and what is not important to consider regarding the meta-model.

This confirms the role of enterprise architecture as a tool for organizational change. In fact, the artifacts to be included in the architecture are needed to respond to important issues by those who are responsible for the transformation of the organization. In this approach we are pursuing, we have not found many publications indeed, it may be because one may consider this "cartography" dimension of architecture of a less noble part of the overall theme of Enterprise Architecture, it is our opinion that it is certainly not less important.

Of course there are other important dimensions of enterprise architecture as an instrument of transformation, as for example the architectural principles and policies, but this is a whole new world.

References

[1] Lankhorst, M., et al.: Enterprise Architecture at Work: Modeling, Communication and Analysis. Springer, Germany (2005)
[2] Leal, R.: Navigation model between Architectural Views. DEI, IST-UTL, Portugal (2010)
[3] Open Group: ArchiMate 1.0 Specification: Technical Standard. Van Haren Publishing, The Netherlands (2009)
[4] Open Group: The Open Group Architectural Framework (TOGAF) - Version 9 Enterprise Edition. Van Haren Publishing, The Netherlands (2009)
[5] Sampaio, A.: An Approach for Creating and Managing Enterprise Blueprints. DEI, IST-UTL, Portugal (2010)
[6] Sousa, P., Lima, J., Sampaio, A., Pereira, C.: An Approach for Creating and Managing Enterprise Blueprints: A case for IT Blueprints. In: Albani, A., Barjis, J., Dietz, J.L.G. (eds.) CIAO! 2009. Lecture Notes in Business Information Processing, vol. 34, pp. 70–84. Springer, Heidelberg (2009)