A case for a Living Enterprise Architecture in a Private Bank

Pedro Sousa[1,2], André Sampaio[1], and Ricardo Leal[1]

[1] Link Consulting, Lisbon, Portugal
{pedro.sousa, andre.sampaio, ricardo.leal}@linkconsulting.com
[2] Universidade de Lisboa, Instituto Superior Técnico, Lisbon, Portugal
pedro.manuel.sousa@tecnico.ulisboa.pt

Abstract. This paper presents a case for the development of an Enterprise Architecture (EA) in a private bank. Architectural Blueprints are generated on-the-fly based on information collected from information sources. Such blueprints are integrated in the bank´s intranet and are the entry point not only for the technical documentation of the bank´s IT, but also for a knowledge database supported by wikis that development teams can update and use to improve their effectiveness. So EA is a living asset used in the day to day work of the bank´s IT teams. Currently, the bank is moving ahead further integrating EA with their Agile development process, by making sprints results (TO-BE states) also available in the EA for everyone. The EAMS tool supports such living EA, not only by supporting automatic generation of blueprints but also by providing a time travel bar allowing a seamless navigation between past, present and future states every blueprint.

Keywords: Enterprise Architecture; Enterprise Cartography; EAM visualizer; EAMS.

1 About The Client

The client referred in this case study is a private bank that operates in the Iberian Peninsula and performs investment banking activities (equities, corporate finance and private banking). The bank serves almost 2 million customers (Individuals, Companies and Institutions) through its multi-channel distribution network comprising around 650 retail branches, a network of external promoters, structures dedicated to the Corporate and Institutional Customers, telephone banking and a home-banking service. In asset management, the bank holds very relevant positions in the management of unit trust funds, pension funds and life-capitalization insurance.

2 The Context

In 2012, the bank started to plan the shape and scope of the EA project as a mean to better support business more efficiently, reducing both development costs and time to market. Amongst many concerns involved, the following are worth mentioning:

- The bank had a past experience in EA, since it went through the process of using a modeling tool to produce representations and models of its IT Architecture. The approach followed was based on a central repository holding all models, each designed manually with an EA tool. This approach required a substantial effort to keep them up-to-date, in particular if one considers the consolidation of the models produced in different projects into a single and enterprise wide view of the IT landscape. So, the maintenance effort to keep the EA representations was a key concern of this project.

- The bank was setting up a Service Oriented Architecture (SOA) infrastructure as a way to rationalize application integration and their IT architecture. Therefore, the representation of the Services and related concepts was also a major concern in the project.
- Finally, the bank was scaling up the dissemination of Agile development process. This process was initiated in 2010 (with 16 teams, and a total of 100 people) using SCRUM [1]. In 2012 the number of teams increased to 30, with 220 people, and in 2013 the bank further increased to 50 teams and 320 people working in SCRUM.

Apart from the above concerns the bank already had an EA portal on their intranet to support the registration of information regarding Application and Technology Architectures. The intranet portal included a large number of features such as a wiki, with a considerable amount of articles that described the processes, system architecture, and which established an entry point for all documentation deemed more technical.

The EA intranet portal helped the collection and presentation of architectural information about the bank, presented in the form of lists, or in the simpler cases, in the form of simple hierarchical maps generated automatically from textual information. People filled up some forms that fed the intranet portal that was then used to generate some maps.

The automation of simpler architectural views has proven to be a fundamental step in setting the bank mindset to pursuit the idea of having all architectural views generated automatically, both to reduce effort as well as to increase the level of confidence that people had, regarding such representations.

The other important aspect was to consider the EA portal as the single point of access regarding the documentation, training and consolidation of IT information.

Regardless of such a clear vision, there were also many issues that needed further discussions and clarifications, such as, which Architectures were to be considered, how to govern each one and how to ensure that they would become live entities in the organization, rather than obsolete soon after they were setup.

Thus, in early 2013 the bank had a pretty good idea of the EA project they were aiming at. However, the challenges that were presented were not easy ones.

3 The Challenges

First challenge was to improve the bank´s EA practice, by improving the EA intranet portal to support the communication and awareness of the architecture between all stakeholders. This included infrastructure teams, development teams and business areas. The IT department was determined to follow the rule “if exists it is in the intranet”. To achieve such a maturity level, the bank has to:

- Improve the overall ability of the EA portal supporting different levels of information detail and different views according to the user profile. This would enable the use of a centralized point of communication for a heterogeneous audience, from business to the different IT domains.
- Enable the sustained growth of information residing in the EA portal by reducing the effort required to create and maintain the enterprise repository information, reports and representations.
- Capture the evolution of the assets throughout their lifecycles, from the moment they were conceived up to the point when they are decommissioned.

The evolution of the Portal should uphold an integrated view while seamlessly allow users to navigate through the architectural representations, wikis, discussion forums and all other related collaborative tools and environments.

Second challenge was the need to effortlessly maintain the architectural representations. The bank stated a second rule “if an architectural view cannot be generated automatically with up-to-date information, then it cannot be presented in the EA portal”. Thus, architectural views must be generated based on consolidated and up-to-date information. This information had to be harvested from various sources, such as Microsoft SharePoint[1], Oracle Enterprise Repository[2], among others. The consolidation of the information implied the creation of a unified view of the information gathered from the various sources in a central repository that was structured according to the existing meta-model. This model had to evolve according to market best-practices and had to be able to accommodate all features, relationships and properties that characterized the concepts of the different architectures.

Third challenge was the need to keep audience’s learning curve regarding the usage of the architecture intranet portal as lean as possible. This was perceived to be a critical aspect to the success of the project. The audience, composed by different stakeholders, had to be able to easily navigate and explore content and representations. The information had to be presented in a self-explanatory and simple fashion, but also allow users to drill down on details, and get the representations as complete and detailed as necessary. This way, the portal and the architectural views would be adequate communication artefacts regardless of the level of expertise of the user in particular domains or architectural layers.

Fourth challenge was the integration of the SOA initiative in the Enterprise Architecture Initiative. SOA has been developed in Oracle Enterprise Repository (OER), with its own meta-model with over 60 entities (concepts). So, the bank had to decide to what extent these SOA concepts needed to be mapped in the Enterprise Architecture meta-model

[1] See Microsoft website (www.microsoft.com)

[2] See Oracle website (www.oracle.com)

As the main SOA repository, OER can provide views of the information out-of-the-box, but these had two major problems:

- Imply the full knowledge of the OER meta-model and thus are only suited for technical people. For example, to navigate between produced and consumed services one must go through many intermediate concepts, such as interfaces, protocols and so on.
- Were not integrated with concepts from other architectures, and thus preventing a seamless navigation for architects and analysts in the EA portal.

Finally, fifth challenge was to anticipate the conciliation of the Agile development process in practice within the bank with the EA. The challenge was to envisage a way to allow the outcomes of future sprints, to be made available to all teams before development starts. Such integration was not in the scope of the initial project, but at the time of the writing of this paper, it is something the bank is already able to do very successfully by itself using EAMS[3].

4 The Project

The project started in early 2013 and was concluded by August 2013. Link Consulting[4] provided EA consultancy services and supplied the tool set used in the project. The project involved 3 people from Link Consulting and a team of 10 people in part time from the client, namely the responsible for each Architecture (Business, Information, Solutions, Services, Platforms, Infrastructure, User Experience and Normative) as well as the EA repository manager.

The project aimed at the implementation and deployment of an EA solution fully integrated in the existing intranet, so that all but back-office housekeeping activities could be done in a seamless manner on the bank´s intranet.

The basic idea is that each architectural view should have an URL that could be placed in any page or document within the intranet. Whenever accessed, it would generate the corresponding architectural view. Navigation from architectural views to web pages and documents should also be possible. The EAMS tool was the engine for the architectural view (or blueprint) generator.

EAMS generates architectural blueprints based on information residing in the IBM-Rational System Architect[5] (IBM-SA) that was the tool selected to be the main EA repository, and holds the information regarding Information, Solutions, Platforms and Infrastructure Architectures. The bank already had a lot of information about these architectures, both in the initial intranet and in other sources, namely in Microsoft Office documents. A significant effort of the project team was dedicated to structure this information so that it could be imported into EAMS.

[3] See EAMS website (www.linkconsulting.com/eams)

[4] See Link Consulting website (www.linkconsulting.com)

[5] See “Rational System Architect” at IBM website www.ibm.com.

Below we present a view of the EAMS blueprint being displayed embedded on the intranet.

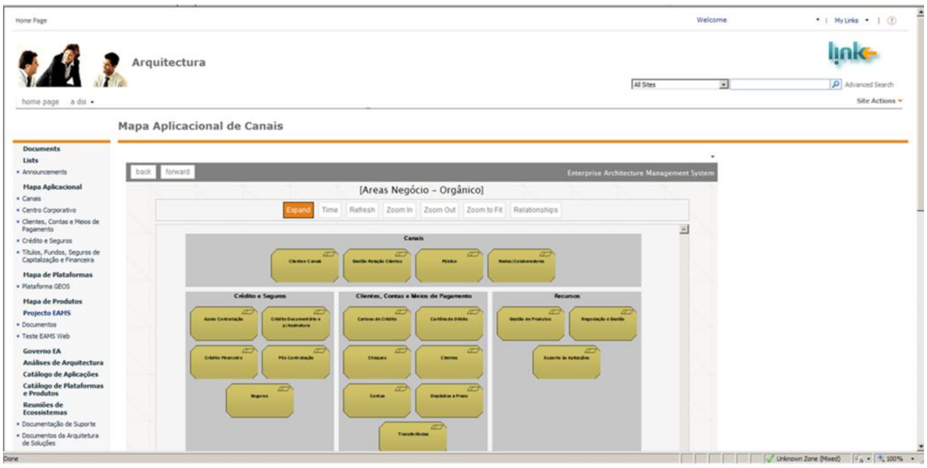

Fig. 1. A view of a blueprint embedded in the EA intranet portal

Regarding SOA, OER provides information about services, resulting from harvesting technical information from Oracle Service Bus, UDDI and directly from Service Contracts (Web Service Definition Languages). Collected Information is then transferred from OER into EA repository on a daily basis using EAMS connectors, jobs and batches.

The remaining sources of information, generically represented as XML in the figure 2, correspond mostly to less dynamic data, such as the bank organizational structure. This information is processed and imported directly to EA repository.

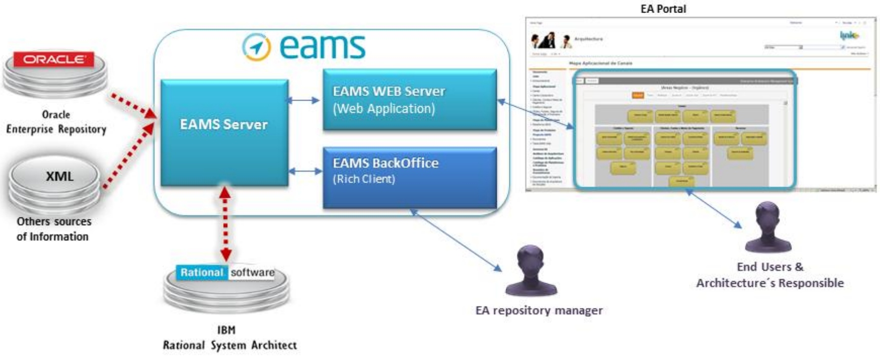

Fig. 2. Key SW components of the solution

An important issue that had to be addressed in the project was the alias problem, where the same concepts have different names in different information sources. This was not a significant amount of work, but was necessary for processing the names of classes (for example, “system” and “application component”) as well as the names of properties. This issue was addressed manually during the project, but latter it was made available to the bank an integration framework that allows automatic execution of conversions rules on each import.

The bank also wanted to address other architectures in domains they considered relevant, even though they did not had much information to start with, namely: Business, Information, User Experience and Normative Architectures.

Both Information and Business Architectures were domains that the bank planned to address in this project, starting with the definition of the meta-model, where it became clear the level of detail needed and how to best establish the relations with the remaining architectures.

Regarding the User Experience Architecture, it concerns the definition of patterns to allow the same concept (client, address, account and so on) to appear to the user with the same paradigms, regardless of the interface and application where such experience was materialized. This was perceived as an important aspect within the bank because business and operations tend to move further into electronic interfaces, and the seamless perception of concepts between different interfaces was considered a fundamental aspect to improve user experience.

Finally, the Normative Architecture structures all information regarding Architectural Principles, Rules, Best Practices and Technical Articles amongst other information. Articles represent a true knowledge base, and a wiki platform allows live interaction with end users. The definition of patterns to structure and organize this information was considered a relevant asset to facilitate the access to it.

5 The Approach

The approach we used has been successful in previous projects and was designed to mitigate the main project risks, which were the following ones:

- Too much time spent on defining the EA repository meta-model.
- Inadequacy of the meta-model’s level of detail and matching information sources. The complexity of the knowledge base meta-model should be suitable to the immediate needs and the existing information sources.
- People losing confidence in the project if they did not see short-term results.
- Ensure that users trusted the information presented in the architectural views generated.

Thus, we adopted a very pragmatic approach, focusing on designing a meta-model as close as possible to the information the bank already had at the very beginning of the project, but keeping basic coherence rules regarding layer separation as in Archimate [2], assuring that each layer of the architecture is supported by services or capabilities provided by the layer below.

This allows independence of architectural layers and simplifies bottom-up and top-down impact analyses of architectural elements. However, since not all layers have a list of the services they provide and consume, it was important to allow layers to relate themselves bypassing such services in between. Otherwise, the whole project could be stuck waiting for information, and no short-term results were perceived.

To ensure that users have confidence in the information presented in the generated architectural views, it must be kept up-to-date. To achieve such aliveness we follow an approach centered on two critical moments: (i) the information loaded during the project setup and seen as a baseline for initial exploration of architectures and; (ii) the changes that each project and maintenance activities implied.

Regarding the initial baseline, since the information that was initially loaded, refers to the architectures that the bank had already information available on the intranet, it was mostly correct. Nevertheless, it was validated again by the responsible of each architecture.

Regarding the changes resulting from ongoing projects and maintenance activities, they were integrated in the Agile development and maintenance process, where the work is planned and executed monthly. This integration was done based on two key moments: when the sprint starts and when it ends.

At the beginning of each sprint, the teams consult the architecture intranet portal to get information they need to design and plan the next sprints. Whenever a sprint results in changes in some architecture, these changes are communicated to the person responsible for the corresponding architecture, and immediately introduced in the EA repository. Such update of the EA repository could be done directly either using the IBM-SA or using the EAMS architectural scenario definition interface, or indirectly by importing excel sheets with the changes of the initial plans using EAMS import interfaces. This was the scenario in the scope of the project.

Today the bank has developed internally a simple workflow to generate XML files that match EAMS architectural scenarios based on information existing in the intranet forms (already used and kept up-to-date as part of the SCRUM development process) reducing the effort spent with these updates almost to zero. Consequently, changes to the architecture foreseen in the planning of each sprint (of each one of the 50 teams) are updated at the beginning of every month and immediately visible by all teams, so that they could have up-to-date information to planning the next sprint.

At the end of each sprint, any differences between planned and actual changes to architecture are also updated in the EA repository. Thus, from the EA repository perspective each sprint has a start date, an end date, and two lists with references to architecture elements. The “alive list” references the elements that will go into production (alive) upon sprint completion, and the “dead list” reference the elements that will be decommissioned (dead) after sprint completion. Using the sprints start date and an end date, we tag each architecture elements (and relations) within the knowledge base with tree time-stamps[6]:

- Gestating, when it is being planned, designed or produced within a given sprint. The timestamp tag is set to the sprint start date.
- Alive, when they are put into use within the organization as a result of a sprint. The timestamp tag is set to the sprint end date.
- Dead, when they are no longer used within the organization, also as a result of some sprint. The timestamp tag is set to the sprint end date.

[6] The actual model considers Retired as fourth state. See [3,4]

These timestamps are enough to track the architecture elements lifecycle and, to know the contents of the views related to any time in the past, present or future.

This means that the approach actually followed to keep architecture up-to-date is based on the uploading of TO-BE states of the organization as foreseen in the plans of each one of the 50 Agile teams working in the bank.

This approach must be supported by tools used with two major features:

- The EA Repository must be able to hold information about the future states of the architecture elements. This is achieved by associating the Gestating, Alive and Dead timestamps as default properties in each and every architecture element, allowing to establish their lifecycle.

- The blueprint generator must be able to generate blueprints representing current and future states for each architecture. This is a native feature of EAMS. All architectural views have a time slider associated, marked with the moments in time, in which there were work packages (sprints in this case) that produced a change in that architectural view. When the handle moves along the slider, and crosses a mark, the name of the work package that led to changes appears on the left and the content of the architectural view changes.

6 The Benefits

The solution has been recognized to sustain the following benefits, in the client´s words:

- Centralized point of communication.
  - Different profiles and stakeholders easily explore the information according to their needs.
- Interconnected Architectures
  - Architecture domains aligned: Business, Data, Application and Technology.
  - The ability to keep-up with all the individual components in the enterprise IT and understand all the dependencies within and between assets.
  - Analyses and representations become interconnected and more complete.
- Enrichment of the architectural representations
  - Automatically generated blueprints and reports.
  - Time-travel through the AS-WAS, the AS-IS and the TO-BE, to identify the gaps between the current and target architectures.
  - Evaluate portfolio costs and compare transformation scenarios.
- Sustained growth of Enterprise Architecture
  - Reducing the effort required to create and maintain Enterprise repository information, leading to a more populated and consolidated repository,
  - Guidance, so the client knows what it needs and how to deliver it, namely:
    - Enterprise Architecture to understand what the bank has.
    - Strategic IT planning to understand what the bank needs.

    - Information Management to know how to deliver.
- Managing Enterprise Architecture over the different development stages: Planning, Design, Construction and Deployment.

7 Reflections on the Case

The case presented is successful mostly because EA is indeed perceived as a living asset in the bank by hundreds of users in the bank. It is used daily in the bank as a basis from which decisions are made. Thus we claim that this approach is a basis to achieve a dynamic representation of organizations based on actual architectural evidences, and therefore, towards an instrument to allow steering the organizations based on the representation of their foreseen states [5,6].

We would like to stress that in the case presented, the architecture views are embedded in a seamless manner with the bank knowledge and collaboration tool, namely the intranet portal. Architectural views are the entry point to access any IT documentation and knowledge kept in wikis.

This approach enables not only the role of the architecture views in understanding the complexities of the business and the IT underneath, but also a better and more effective collaboration between the different stakeholders. This has a positive feedback on the use of the EA internet portal as a collaboration and communication platform. The living architecture plays a fundamental role on people communication and collaboration!

We now explain how our approach and tools overcome key obstacles that would otherwise prevent the achievement of a dynamic and living architecture.

7.1 Obstacle 1- Too much effort to up-to-date EA models and representations

We claim that whenever such graphical models are handmade, they must be maintained by hand. Therefore EA models and representations must be generated automatically. This is the general claim behind the idea of Enterprise Cartography [4] and requires specific tools. In EAMS, Architecture representations are generated on-the-fly based on textual information gathered from different sources.

7.2 Obstacle 2 - Transitions from AS-IS to TO-BE scenarios always require too much effort and are discontinue and unrelated worlds

We claim that the ability to have distinct maps to handle distinct moments in time, for example, a map with the AS-IS and another with the TO-BE, is not good enough to handle the complexity of real problems. The reason being that the states in between the AS-IS and the TO-BE are not known, and therefore one cannot steering the organization in between.

With EAMS, Architectural representations have a time slider and their content changes dynamically according to the date indicated on the time slider, along with the identification of the projects responsible for such changes. This allows easy access to architecture views from past dates (AS-WAS), present (AS-IS) and future dates (TO-

BE), and their associated roadmap. It also allows to know the existence of any states in between AS-IS and TO-BE states.

7.3 Obstacle 3 - EA is only for a few enlightened ones

We claim that languages used for architectural design (such as Archimate) fail to provide a universal language between all architecture stakeholders, in particular management and business staff, because it tends to be too complex for them to explore and get the answers they require. Besides the complexity of the architectural views, stakeholders also have to face the complexity of EA tools, which is adequate for conception and design, but not suitable for non-technical stakeholders. Therefore, they are mostly dependent on technical staff to provide them with answers.

With EAMS, architectural views and navigation paths between them are generated automatically according to the user profile. Furthermore, end users do not need to login in EMAS, since architectural views are embedded in bank´s intranet portal. This ensures that stakeholders face a complexity according to their needs and skills, and therefore it enables management (e.g. Business, IT Projects, Governance, SOA) to navigate and explore architectural views and get the answers on their own.

Notice that automatic generation of architectural views also ensures they are updated and kept coherent with available information. This increases the confidence on the architecture views, and increases their usage in a wider community.

7.4 Obstacle 4 - Project Planning only requires accurate knowledge of TO-BE states, not the AS-IS

In general, to plan a project that will occur on a given period in the future, one needs to know the foreseen state of the organization in that time period. As an example, let's consider an organization that intents to replace a given legacy system in 6 months' time, and wants to start the planning of that project today. The preparation and planning stages of the project must have a clear understanding of all the dependencies of that legacy system, both in what concerns integration with other systems and with the business processes it supports. But the critical aspect is that in order to plan that project, having a clear understanding of all the dependencies of the legacy system today is not good enough! One needs to know the dependencies of the legacy system in 6 months' time, because in the meantime, there may be on-going projects that will change the dependencies status of the legacy system.[7]

With EAMS, one just needs to move the time bar slider 6 months ahead in time to find out the actual dependencies of the legacy system, because EAMS generates the Architectural Blueprints based on information gathered from on-going and planned projects to be completed up to the date set by the time handler, 6 months' time in this example.

[7] See Enterprise Cartography article [4] for further analyses on this subject.

7.5 Obstacle 5 - The concept of Application is not only an IT concept

This obstacle concerns with the modeling of application concept in the EA repository, and does not relate with tool functionality. However, since it has major impact on how people communication in every day´s work, we believe it is worth mentioning in this paper.

Standards as TOGAF [7], considers two types of applications: Infrastructure and Business Applications. However this is a very simple classification schema that does not address fundamental issues in IT management. From an IT and Business alignment point of view, a spreadsheet holding data and business rules is a Business Application, as SAP ERP or any other SW package. But from a management and day to day IT operations they have little in common.

Another example comes from the fact that the business users know, and use, the names of applications in many contexts, as for example internal manuals, risk management and so on. This is a problem because it hard-wires a given software element to a set of functionalities and interfaces, removing a degree of freedom from IT management. For example, consider the case where IT decides to provide access to applications via an intranet, forcing the business to revise documents and procedures. This kind of decisions should not be made more difficult just by the naming conventions. The situation is even worse if the application is named after the platform or software packaged used to develop the application.

So, we claim that applications must be modeled in three different perspectives [8]:

- In the Business Architecture, IT is perceived as a provider of a logical aggregation of Business Functions, thus Business Functions and requirements are managed independently of the components that implements them.
- In the Solutions Architecture, the Application concept, is an artifact that implements Business Functions, thus allowing IS teams the flexibility for the best way to engineer such components.
- In the Platform Architecture, the Platform concept is an execution environment of application components and that may include different types of Technologies

8 References

1. Scrum-Guide, https://www.scrum.org/Scrum-Guide
2. Archimate, http://theopengroup.org/archimate
3. Sousa P., Lima J., Sampaio A., Pereira C An approach for creating and managing enterprise blueprints: A case for IT blueprints. In: Advances in Enterprise Engineering III, pp. 70–84 Springer (2009)
4. Tribolet J., Sousa, P., Caetano A. The Role of Enterprise Governance and Cartography in Enterprise Engineering. In International Electronic Journal - Enterprise Modelling and Information Systems Architectures. Special Issue on A Roadmap for Business Informatics, pp. 38-49, Volume 9, No. 1 June (2014)
5. Proper H. A. Enterprise Architecture - Informed steering of enterprises in motion. In Proceedings of the 15th International Conference, ICEIS 2013, Angers, France - Revised Selected Papers, Lecture Notes in Business Information Processing, Heidelberg, Germany, Springer (2014)

6. Proper H.A. Enterprise Architecture - Towards essential sensemaking. In Enterprise Modelling and Information Systems Architectures (2014).
7. The Open Group. TOGAF, the open group architecture framework, version 9 Van Haren Publishing (2009)
8. Sousa P., Martins R., Sampaio A.: A Clarification of the Application Concept: The Caixa Geral de Depósitos Case. In Lecture Notes in Business Information Processing, , pp. 1-17, Springer Berlin Heidelberg (2012)