FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Enterprise Architecture Modelling for Central HR
at Shell International B.V.

Gonçalo Salvador Ribeiro e Castro
MIEIC 2008

U.PORTO MIEIC Mestrado Integrado em Engenharia Informática e Computação

10-6-2011

Rec. 7/31

XD 125458

(D) 00253

Faculdade de Engenharia da Universidade do Porto
Mestrado Integrado em Engenharia Informática e Computação

Enterprise Architecture Modelling for Central HR at Shell International B. V.

MIEIC Project Report 2007/2008

Gonçalo Salvador Ribeiro e Castro

FEUP Supervisor: Professor Doutor António Lucas Soares

A Presidente do júri

Henrique Serpeo de Nóbrega

March 2008

CONFIDENTIAL

As agreed by the involved parties, this report is confidential.

Dedicated to my parents

Abstract

This report addresses the project that took place from 17th of September 2007 to 15th of February 2008 at the company Shell International B. V.. The main topic of the project was Enterprise Architecture, more specifically the implementation of this practice in Central HR department.

During the 1990s, the independence of operating companies around the world was gradually reduced and today virtually all of Shell's operations in all of its various businesses are much more directly managed at the headquarters of the company. Amongst other benefits, these changes enabled the company to manage better its IT resources. To support this, one of the actions was the introduction of Enterprise Architecture practice in Shell, in order that there could be a better understanding of the whole picture of the enterprise's information systems and business processes.

The project addressed in this report was part of a pilot of Enterprise Architecture for Central HR. This project was focused on Business and Application layers of Enterprise Architecture, and involved the specification of the relevant diagrams in these domains for Central HR and their respective modelling. This modelling involved also the entities that support the diagrams, and that store the detailed information concerning the elements of those diagrams.

In the Business Architecture layer, the project was focused on the modelling of business process hierarchies, which represent the different levels of detail of business processes in Central HR.

In the Application Architecture domain, the objective was to represent a good overview over the IT applications that belong to Central HR portfolio, and the interfaces they have with each other.

On the other hand, this project was concerned with testing the Meta Model and the tool adopted by Shell for Enterprise Architecture, which was performed through the mentioned modelling.

This report also addresses some recommendations about Enterprise Architecture in Shell, which resulted from the project and aim to provide input for the evolvement of the practice within the company. These recommendations are concerned with the approach to Enterprise Architecture in Shell and the tools chosen to support it, but also with the proposed actions for follow-up of this project.

The project met the expected deliverables and objectives, being an enriching experience that enabled learning at a personal and technical level.

Acknowledgements

I would like to thank everyone that, directly or not, contributed in a positive way for a successful conclusion of the project, and for the creation of the conditions that enabled it to happen in the best possible context.

To Shell, for providing me all the conditions for the realisation of the project, specifically to my supervisor, Susan Cartmel, and my mentor, Stephen Turner, that trusted in my capacities and guided me throughout this process of growth and development. I cannot forget my colleagues at HR-IT, but not only, also the fellow interns and other colleagues, that were always helpful and supportive.

To Professor António Lucas Soares, for his support during the project, and for his immediate interest in the project since my first approach concerning his availability for being my university supervisor.

To all the professors, colleagues and staff of FEUP that, during my studies, had a positive influence in my academical development.

To BEST, and to the friends that I got to know there. Being member of this association during my studies was decisive for my cultural, human and professional development.

To my family, and friends, that were always important for my personal balance, and that constitute my personal circle of trust.

Index

Figures

Tables

1 Introduction

The objective of this chapter is to introduce the reader to the context of the project: its content, as well as Shell, the company where it was performed.

1.1 Shell International B. V.

Shell International B. V. is one of the many companies owned by Royal Dutch Shell plc, the mother company of the Shell Group. It employs staff from different areas of the Group, such as Central HR, Health or Central Finance.

Shell itself is a multinational oil and energy company, being one of the largest private sector energy corporations in the world. The company's main business is the exploration, production, processing, transportation and marketing of hydrocarbons (oil and gas). Shell also has a significant petrochemicals business, and a renewable energy sector. The company has 112.000 employees and it is present in 140 countries.

The project took place in the company headquarters, in The Hague, Netherlands, in Central HR function[1], which has a global role, across the different countries where Shell is present.

More specifically, the project was performed at HR-IT: Human Resources – Information Technology. HR-IT strives to provide IT skills, solutions and services to HR in support of the Shell strategy. For this, HR-IT follows the priorities from both Central HR and IT functions, and supports systems for learning, recruitment and general HR management.

1.2 Scope of the project in the context of Shell

Traditionally, Shell was a heavily decentralised company worldwide with operating companies all over the world each of which operated with a high degree of independence. The custom and practice in Shell was that most of the business was essentially local in character and that they were best managed by local "operating companies". In the 1990s this paradigm began to change and the independence of operating companies around the world was gradually reduced. Today, virtually all of Shell's operations in all of its various businesses are much more directly managed in The Hague, at the headquarters of Shell. The autonomy of "operating companies" has been largely removed as more "global businesses" have been created in all sectors.

Along many other benefits, these changes enabled the company to move towards the standardisation and simplification of the IT infrastructure landscape and governance, in order to control it better. To support this, one of the actions was the introduction of Enterprise Architecture in Shell, in order that there could be a better understanding of the whole picture of the enterprise's information systems and business processes. Centralising all this information enables a better overview of it, thus providing valuable input for strategic decisions.

---

[1] Function is an organisational unit within Shell, a portfolio of activities and responsibilities operating according to common objectives and strategies with formally delegated organisational mandates. It could be called department in a simplified terminology.

During 2007, it was created the basis for the rollout of Enterprise Architecture in Shell: the creation of an Enterprise Architecture Meta Model[2], a reference for "enterprise architects language", and the adoption of a tool to support the practice: Telelogic System Architect. After this, the different departments within the company started their own work in Enterprise Architecture, incorporating it in their specific contexts, and therefore assessing the Meta Model and the tool in the department's environment. This was the main purpose of the project: to participate in a pilot of Enterprise Architecture within Central HR. A pilot in which Enterprise Architecture could be introduced in Central HR, proofing added value to the function. This pilot project took place from 17th of September 2007 to 15th of February 2008.

1.3 Structure of the Report

After this introductory chapter, Enterprise Architecture is introduced in the following chapter in order to give a good overview of the domain of the project.

The project and its objectives are analysed in chapter 3, as well as the approach taken, that is detailed the next two chapters with the specification of the modelling done and an overview of the modelling itself.

Chapter 6 presents the main recommendations resulting from the project, and the last chapter is dedicated to the conclusions of the project.

It is relevant to mention the usefulness of Appendix A, that includes definitions of some expressions that are used throughout this report, being those definitions useful for a complete understanding of the document.

[2] A Meta Model is a precise definition of the constructs and rules needed for creating semantic models. In the case of this project it specifies a common language for Enterprise Architecture practice within Shell.

2 State of the Art in Enterprise Architecture

This chapter provides an introduction to Enterprise Architecture, in order that the further reading of this report can be supported by good background knowledge on the area of the project.

2.1 Enterprise Architecture

Enterprise Architecture is the practice of describing an organisation´s processes and information systems current and/or future state, aligned with the organisation strategy and goals [1]. In order to improve performance and effectiveness, this modelling practice is implemented by several companies and government institutions, whose complexity in terms of business processes and IT systems demands such integrated solution [2].

Enterprise Architecture involves developing an architecture framework to describe a set of architectures, describing the current, the intermediate and the target reference architectures [1]. These architectures are typically grouped in Business, Data, Application and Technical domains and are described through artifacts, such as written documents or diagrams. These artifacts link the four architectural layers, thus documenting the processes of the organisation and the associated resources [3] (Figure 1).

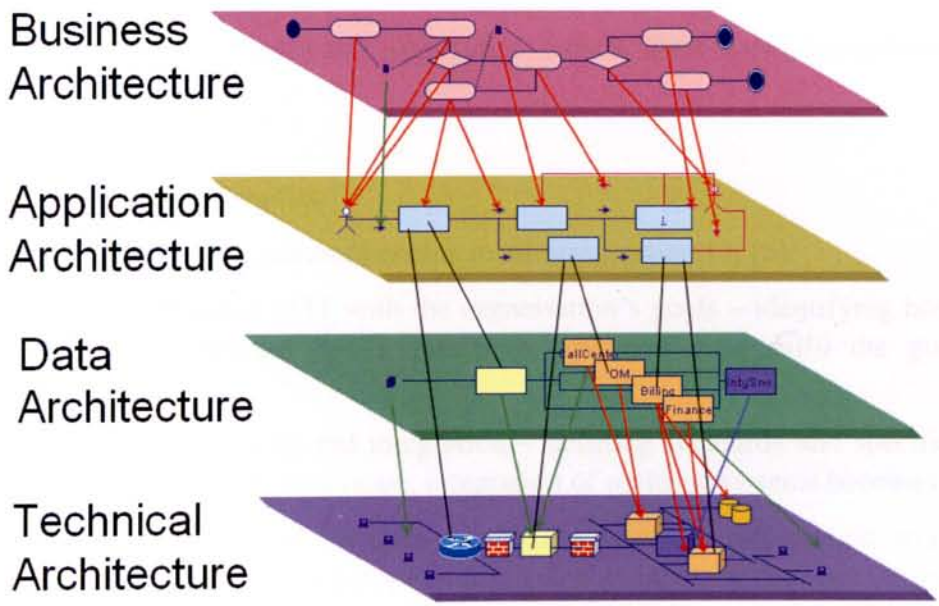

Figure 1 - Overview of Enterprise Architecture Layers

These architectural descriptions (that can have different levels of detail) provide input for decision makers to take informed decisions on where to invest, where to realign organisational goals and processes and what policies and procedures will support core missions or business functions.

Enterprise Architecture can help to make clear if the current architecture supports and adds value to the organisation, and how can it be changed to add more value. On the other hand it can also clarify if the current architecture can support the goals that the organisation wants to accomplish in the future [4].

Within the different layers of Enterprise Architecture, there are different components that can be included in the architecture description [3] [5]:

- Business Architecture:
  - Strategy maps, goals, corporate policies;
  - Functional decompositions, capabilities and organisational models;
  - Business process models: hierarchies, workflows.
- Application Architecture:
  - Application software inventories and diagrams;
  - Interfaces between applications.
- Information Architecture:
  - Metadata;
  - Conceptual, logical, and physical data models.
- Technical Architecture:
  - Hardware, platforms, and hosting;
  - Local and wide area networks;
  - Middleware;
  - Infrastructure software: Application servers, Data Base Management Systems, etc.

Enterprise Architecture Benefits

Enterprise Architecture offers several benefits to an organisation [4] [6]:

- Enables the alignment of IT with the organisation's goals – identifying how business processes are supported by IT resources, and how they fulfil the goals of the organisation;
- Improves interoperability and integration – defining standards and specifications for how IT systems will communicate, integration of multiple systems becomes easier;
- Enables reporting on IT systems – making accurate information available and whenever and wherever needed, providing valuable input for decision makers;
- Enables agility – when it is needed to quickly respond to some sort of change in the environment, Enterprise Architecture provides a reference that tells the impact that change will have on each of the components within the Architecture, and how to ensure that the components continue to operate smoothly through change management. Enterprise Architecture also enables faster design of new systems and extensions to existing systems by pre-defining ground rules and standards;
- Reduces costs – taking less time to implement systems, this implementation costs less. Through the definition of standards, a less complex environment is established (due to technical homogeneity), being easier to support and resulting in faster repairs;

- Improves security - through the development of security standards, with which all systems comply, the risk of intrusion, loss, or system downtime are reduced;
- Reduces technical risk – An organisation's Enterprise Architecture can reflect a technology infrastructure that is based on industry standard solutions. Doing so it increases the availability of support services. It also ensures that the organisation maintains a pace of technology currency that is consistent with its business context.

2.2 Enterprise Architecture Frameworks

As mentioned, Enterprise Architecture involves developing an Architecture Framework, that can play a very important role in IT governance. This framework defines a methodology that enables the description of an Enterprise Architecture as a set of building blocks, and shows how these blocks fit together. On the other hand, it provides a common vocabulary to approach Enterprise Architecture [6][7]. However, in a complex enterprise environment, Enterprise Architecture Frameworks need to be supported by a Meta Model which defines the architectural elements and the relations between them, in order to complement the models defined in the Framework [8].

Typically, three kinds of organisations create Enterprise Architecture Frameworks:

- Research Foundations, whose maximum exponent is Zachman International, founded by John Zachman following his work in IBM in the 1980s that lead to Zachman Framework [9]. This is the world's most widely used Enterprise Architecture Framework [10].
- Companies, typically:
    - Big multinational corporations for their own use (such as Shell [11]);
    - IT consultancy companies that implement their own Framework in their costumers. Example: Capgemini's Integrated Architecture Framework [12]
    - Consortiums of companies, which best example is The Open Group, that created TOGAF [3];
- Governamental, divided in:
    - Frameworks used within Governments, example: Federal Enterprise Architecture, used within the United States Government [13].
    - Frameworks compulsory used by Defense Industry providers when documenting the products sold to governments. Examples:
        - DODAF - the US Department of Defense Architecture Framework [14];
        - MODAF - the UK Ministry of Defence Architecture Framework [15].

Thirty two percent of organisations that practice Enterprise Architecture create their own Framework, while the rest invests in legacy ones, provided by research foundations or companies. The most widely used Frameworks are presented next: TOGAF and Zachman Framework [10].

TOGAF

The Open Group Architecture Framework has been developed by the Architecture Forum of The Open Group and continuously evolved since the mid nineties. TOGAF provides a comprehensive approach to the design, planning, implementation, and governance of Enterprise Architecture. It follows the typical division in four architectures of Enterprise Architecture (Business, Application, Data and Technical), that enable the architecture team to envision the current and future state of the architecture. [3]

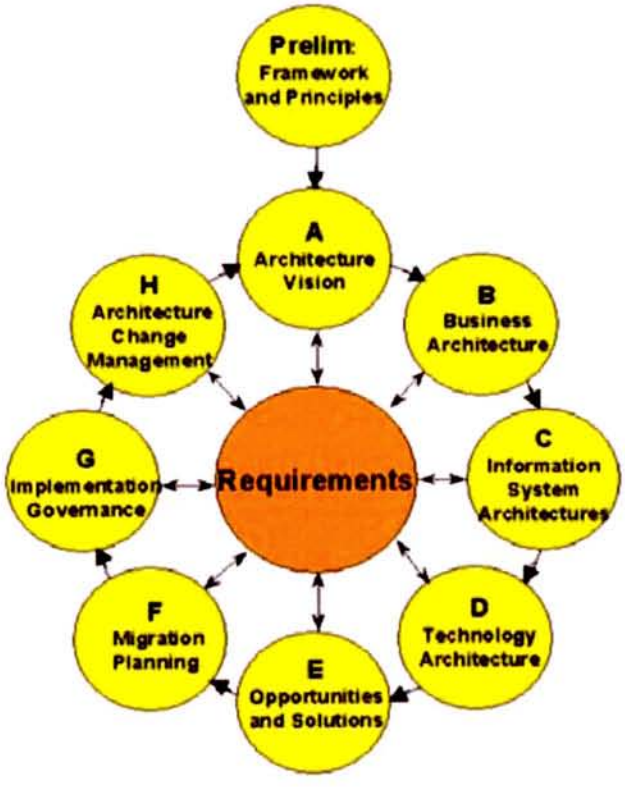

Figure 2 - TOGAF 8 Architecture Development Method (source: The Open Group)

TOGAF consists of three main parts [3]:

- Architecture Development Method, which specifies how to develop an Enterprise Architecture that meets the organisation's needs. It provides different steps that can be seen in Figure 2.
- Enterprise Continuum (Figure 3), which is a virtual repository of all the architectural assets existing in the organisation and the IT industry, which the enterprise considers itself to have available for the development of architectures. These include architectural models, architectural patterns, architecture descriptions, and other artifacts. The Enterprise Continuum divides itself in Architecture Continuum and the Solutions Continuum. The first is related to the current state of the architecture, and the second to the solutions proposed for future states.
- TOGAF Resource Base, which is a set of resources (guidelines, templates, background information, etc.) to help the architect in the use of the ADM.

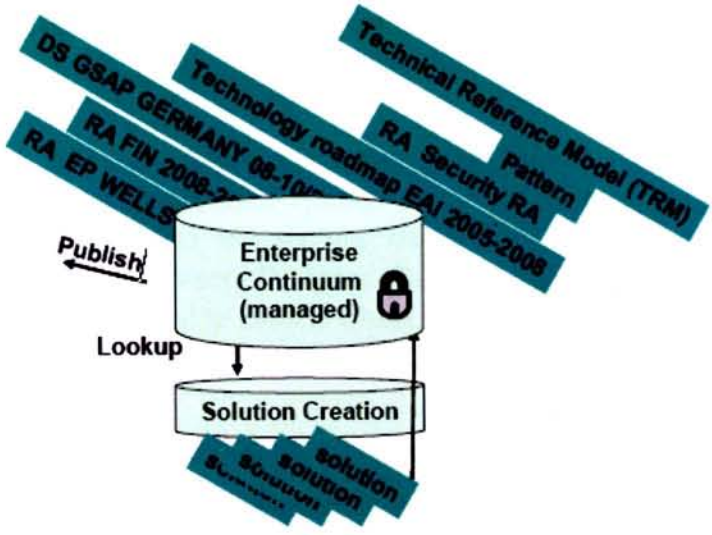

Figure 3 - Enterprise Continuum (source: The Open Group)

Zachman Framework

Originally conceived by John Zachman at IBM in the 1980s, Zachman Framework is now a world standard for expressing the elements of Enterprise Architecture.

Table 1 - Zachman Framework (source: Zachman International)

	DATA What	FUNCTION How	NETWORK Where	PEOPLE Who	TIME When	MOTIVATION Why
SCOPE (CONTEXTUAL) Planner	List of Things Important to the Business.	List of Processes the Business Performs	List of Locations in which the Business Operates	List of organisations important to the business	List of Events/Cycles Significant to the Business	List of Business Goals/Strategies
BUSINESS MODEL (CONCEPTUAL) Owner	e.g. Semantic Model	e.g. Business Process Model	e.g. Business Logistics System	e.g. Workflow Model	e.g. Master Schedule	e.g. Business Plan
SYSTEM MODEL (LOGICAL) Designer	e.g. Logical Data Model	e.g. Application Architecture	e.g. Distributed System Architecture	e.g. Human Interface Architecture	e.g. Processing Structure	e.g., Business Rule Model
TECHNOLOGY MODEL (PHYSICAL) Builder	e.g. Physical Data Model	e.g. System Design	e.g. Technology Architecture	e.g. Presentation Architecture	e.g. Control Structure	e.g. Rule Design
DETAILED REPRESENTATION (OUT-OF-CONTEXT) Sub-Contractor	e.g. Data Definition	e.g. Program	e.g. Network Architecture	e.g. Security Architecture	e.g. Timing Definition	e.g. Rule Specification
FUNCTIONING ENTERPRISE	e.g. DATA	e.g. FUNCTION	e.g. NETWORK	e.g. ORGANISATION	e.g. SCHEDULE	e.g. STRATEGY

In opposition to TOGAF, Zachman Framework is focused on describing which kinds of models are relevant for the practice of Enterprise Architecture, rather than specifying any methodology.

Zachman Framework does not follow the typical four-layered framework, it uses a two dimensional classification schema (Table 1) based on the six basic interrogatives (What, How, Where, Who, When, and Why) intersecting six distinct perspectives, which relate to stakeholder groups (Planner, Owner, Designer, Builder, Sub-Contractor). The intersecting cells of the Framework correspond to models which, if documented, provide a holistic view of the enterprise [9].

2.3 Enterprise Architecture Tools

There are several tools that support Enterprise Architecture practice (example in Figure 4 – Telelogic System Architect) available in the market. As this practice is becoming more and more holistic, there is the need to use comprehensive modelling tools that address the different pieces of a framework. Typically, the tools in the market are compliant with specific Enterprise Architecture Frameworks [16], however, the suppliers also build customise their solutions to support clients' own Frameworks [11].

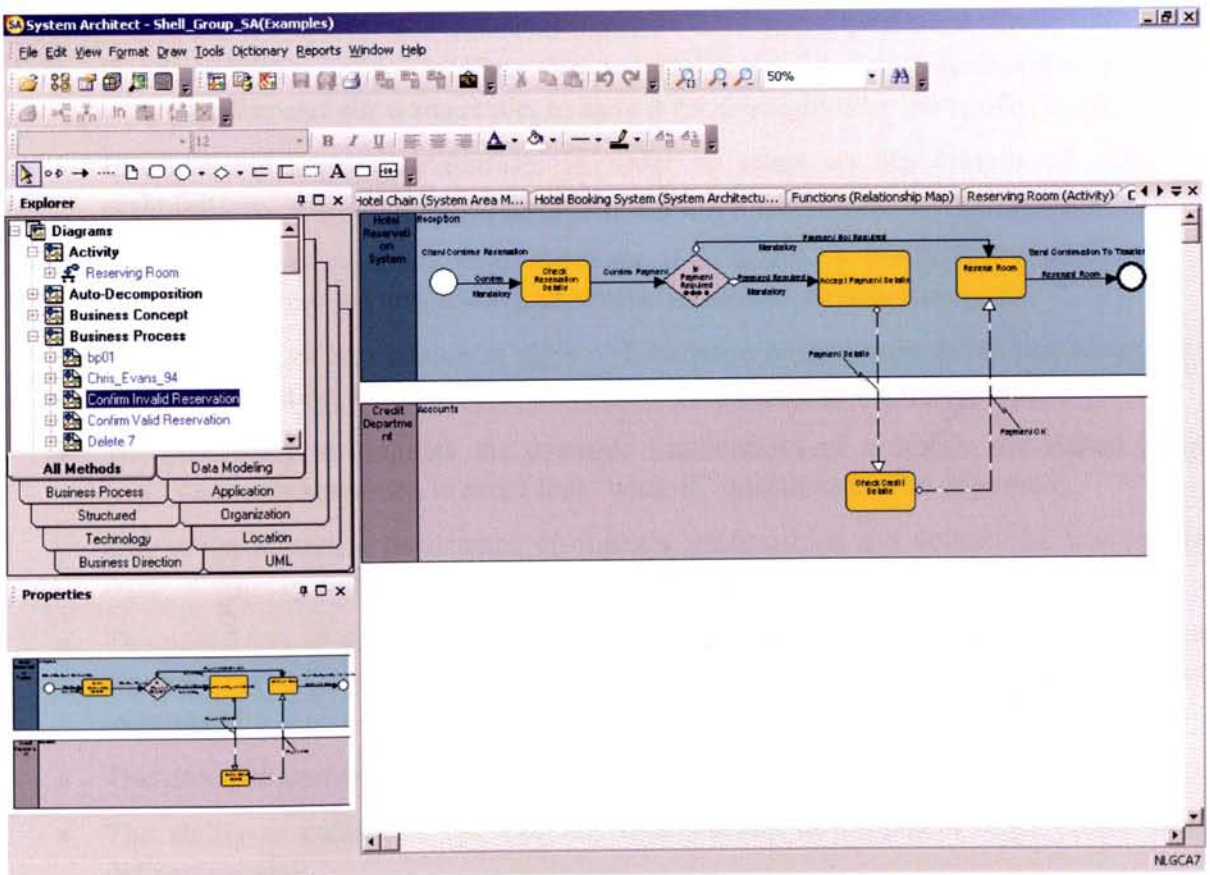

Figure 4 - Telelogic System Architect screen shot

A list with the tools with biggest market share and respective providers is available in Table 2 [16][17]. This table also addresses the Frameworks that those tools are compliant with.

Table 2 - Main players in Enterprise Architecture tools market

Provider	Tool	Frameworks that the tool is Compliant with
Casewise	Corporate Modeler Enterprise Edition	Casewise Framework, Zachman Framework, FEAF, TEAF, eTOM, DoDAF, etc.
IDS Sheer	ARIS Process Platform	ARIS Framework.
MEGA International	Mega (Process, Architect, Designer)	Zachman Framework.
Proforma	Provision Modelling Suite	Zachman Framework, DoDAF.
Telelogic	System Architect Family	Zachman Framework, TOGAF 8, DoDAF, MoDAF.

There are several parameters that are relevant to assess an Enterprise Architecture tool [18][19]:

- The ability to support all the enterprise architecture components, in the different domains, as well as the desired types of Frameworks, methodologies and models;
- Good modelling design interface, that can enable the Enterprise Architecture practice to be pleasant and allow the reader to have a good insight over the models displayed;
- Extendability and customisation, in order to adapt to the context of different organisations, which can be supported by a Meta Model definition capability;
- The capacity of analysis (e.g. checking if a model is correct or complete) and manipulation (e.g. possibility to join several models in an overview one);
- The capability of automation of parts of Enterprise Architecture development activity, such as generating visual models as result of data stored in the repository;
- The possibility to simulate the dynamic implications of a design and linked items (e.g.: load on a network), in order that "what-if" questions can be answered;
- The ability to secure persistence of object's relationships and definitions, that enable reuse of already created objects through instances;
- The possibility to understand and manage the life cycle of a design object (whether it was recently implemented or it was already decommissioned, to mention a couple of examples);
- The ability to perform searches, easily navigating through architectural information;
- The ability to customise the tool interface without development (e.g.: create user-defined menus);
- The possession of a repository that enables the users to collaborate (e.g. concurrent access) and provides data management capabilities (such as versioning of models, access control to models, or the ability to roll back to old models);
- The ability to share ideas and work together on developing new solutions, through collaboration tools, such as portals or forums;

- The ability to import and export information, through files such as XML[3], as well as word processors or spreadsheets, making the information available outside the tool environment, and usable for different purposes;
- The capability to create customised reports that suit the requirements of users, and that can provide decision makers with the information available in the repository;
- The possibility to manage documentation detailing the defined objects, models and architectures;
- The costs of the tool, and support provided by the supplier.

[3] XML – eXtended Markup Language. Document language that enables the exchange of structured information.

3 Project Analysis and Objectives

During the first half of 2007 a project was undertaken to create a Meta Model defining governance, standards, terminology and procedures for Enterprise Architecture in Shell. This Meta Model was delivered in July 2007, and reflects ETAP, Shell's Enterprise Architecture Framework. Meanwhile, the company adopted Telelogic System Architect as the official tool to support Enterprise Architecture practice, together with Metastorm ProVision for Business Process modelling. After these initial steps, it was time for the different departments within the company to start implementing Enterprise Architecture on their own [11].

The project addressed in this report constituted a five months pilot of Enterprise Architecture in Central HR, introducing this practice to the function. The project was intended to create initial models, using System Architect, across domains of application and business architectures, in their current state, in order to demonstrate benefits of Enterprise Architecture modelling for Central HR, and to assess the Meta Model. Therefore, the following objectives were defined:

- To demonstrate benefits of Enterprise Architecture modelling for Central HR;
- To test that the Meta Model and configuration of System Architect are fit for purpose for Central HR;
- To test interfaces from Metastorm ProVision[4] and E-PIMS[5] to System Architect;
- To determine way forward for Enterprise Architecture in Central HR.

The project was focused on creating visual models and related entities[6]. These entities and models are present in the Meta Model, which defines all the entities in Shell's Enterprise Architecture, and the relations between them (example: one IT application is used to perform one or more business processes), as well as the relevant diagrams for the company. In the end, this project was about defining which were the relevant diagrams, and related entities, for Central HR, and modelling them. This could lead to assess the way Enterprise Architecture is being approached in Shell, and, on the other hand, create a basis for further Enterprise Architecture work within Central HR.

Project Phases

The project had three main phases (Figure 5): Context adaptation, Specification and Development.

---

[4] As Metastorm Provision was adopted as official tool for Business Process Modelling, there was the need to create an interface to System Architect, which could feed this tool with the Business Processes.

[5] E-PIMS: Enterprise Portfolio Information Management System – System where all the information concerning Shell's IT applications is stored. Examples: locations where the application is used; people that support the tool; etc. In this context, the creation of an interface that could feed System Architect with IT applications related information was planned.

[6] Enterprise Architecture consists of entities and models. An entity can have attributes which indicate its specific behaviour and relations with other entities. Entities can be grouped into models to support a required architecture viewpoint. Examples of entities: Application Component (that represents an IT application), Application Component Interface (interface between IT applications), Process (Business Process), etc.

The first phase was concerned with learning about Enterprise Architecture, the tool used to support it, System Architect, and the context of Central HR. This was an important phase for getting a good initial basis of knowledge.

The second phase was concerned with the specification of what was desired from the project, according to the requirements of the main stakeholders. In this phase it was agreed which should be the deliverables of the project:

- Comparison between System Architect and E-PIMS, in terms of metadata[7] and capabilities;
- Visual models on the Business and Application Architectures, according to the desired scope: Architecture As-Is[8];
- Standards for these types of models (which information should be included in each diagram and related entities);
- Manual on how to use System Architect according to the mentioned standards;
- Recommendations document, proposing improvements to the Meta Model, System Architect and Enterprise Architecture in Shell and Central HR.

The last phase, Development, was about producing the deliverables agreed in the Specification Phase. This last phase involved close contact with other Shell colleagues in order to source information (example: details about interfaces between IT applications), but also for feedback gathering on the work performed, in order to improve it.

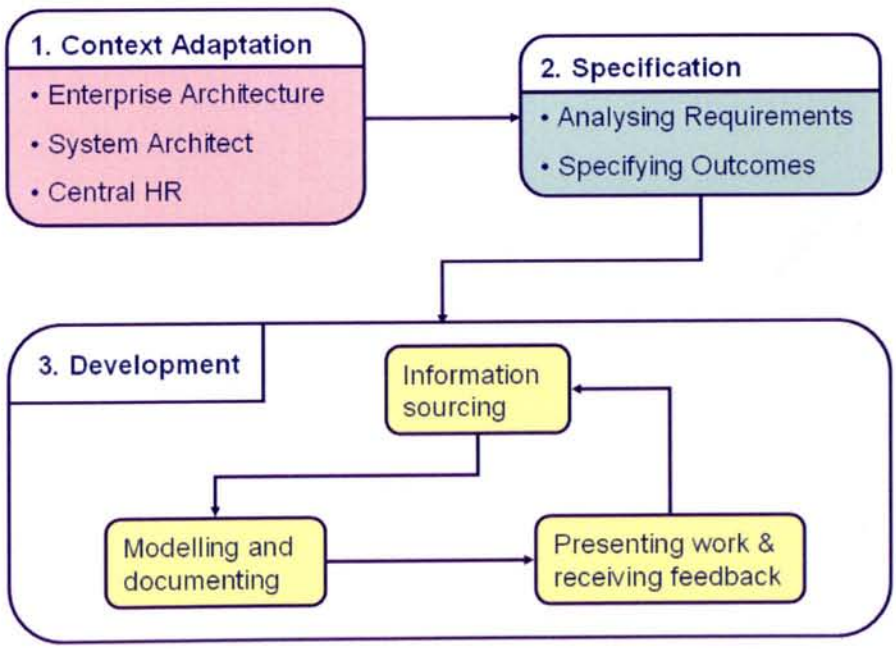

Figure 5 - Project Phases

[7] In this case metadata refers to the property fields of IT applications in both systems.

[8] As-Is architecture is the current architecture implemented in the organisation. Other types of architectures are the "To-be", architecture that will be implemented in the future, and "Intermediate" that is a passing point between "As-Is" and "To-be" architectures.

It was quite an interesting challenge to seek the information needed for the project as it was quite dispersed, in different documents, systems and there were even cases when it was not documented, and only possible to source through interviewing colleagues.

Besides the tasks mentioned in the phasing of the project there were other activities that were related to the context where the project took place, rather than with the project itself. Therefore, such things as team workshops, and other kinds of department meetings, as well as learning events with senior company staff, also took place in the period at the company. This kind of activities contributed for a more complete experience in the context of the daily work in a multinational company, and made the project experience more interesting and complete.

4 HR Modelling Requirements

During the specification phase of the project, besides specifying the desired deliverables, a detailed analysis was taken on which models should be drawn and which type of information should those models include. These models and entities were selected amongst the ones defined in the Meta Model, according to the expressed requirements from the main stakeholders of the project.

In Figure 6 it is possible to have an overview of the types of diagrams and related entities that were specified as desired outputs of this project, in both Business and Application Architectures. The entities inside diagram's boxes are represented in those diagrams; the other ones are not visually represented in any diagram, but are also important in defining the diagram's entities. The connections between entities specify their relations.

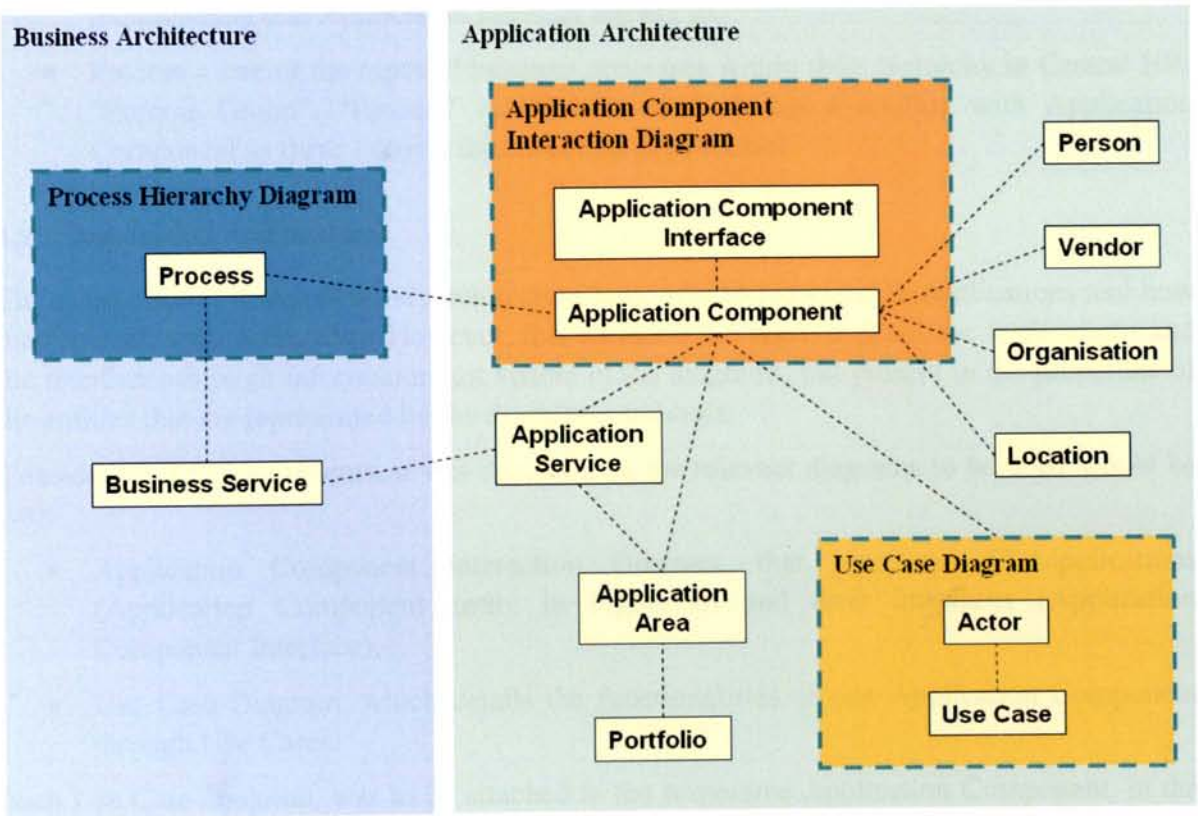

Figure 6 - Diagrams and Entities within each Architecture and their relations

4.1 Business Architecture

As mentioned, Metastorm ProVision was chosen as the official tool in Shell for Business Process Modelling. This means that part of Enterprise Architecture is performed through that tool, especially process hierarchies and workflow diagrams, which detail the different business processes.

It was important to integrate ProVision with System Architect, in order to have a consistent Enterprise Architecture practice. Therefore, Shell developed an interface that feeds System Architect with the hierarchies of business processes (that in Central HR are divided in a

hierarchy constituted by "Process Groups", "Processes" and "Sub-Processes") from ProVision, but not with the Workflow diagrams, as the main interest were the different processes (rather than how they are performed, through Workflow diagrams), in order to relate them to entities in the other layers of Enterprise Architecture.

Considering that at the time of this project, Central HR was not performing Business Process modelling in ProVision yet, there was the need to manually create the Business Process entities in System Architect, and the respective diagrams that specify the hierarchy between them. This means that in the Business Architecture layer, the expected diagrams created as result of this project were Process Hierarchy diagrams, as these are the ones that will be used after System Architect rollout in Central HR.

Within Business Architecture, two entities used needed to support the created diagrams:

- Business Service – represents services that are requested to support a Business Process, and that are owned by the users of Application Components, representing the requirements that Application Services answer to.
- Process – one of the types of business processes within their hierarchy in Central HR: "Process Group", "Process" or "Sub-Process". It has a relation with Application Component as these support the execution of processes.

4.2 Application Architecture

The main interest in Application Architecture layer was to represent IT applications and how they interact with each other. However, this included the need to detail the applications and the interfaces through information not visible in the diagrams, but present in the properties of the entities that are represented by the diagram's symbols.

Considering the requirements, it was decided that the relevant diagrams to be draw would be two:

- Application Component Interaction Diagram, that represents IT applications (Application Component entity in Figure 6) and their interfaces (Application Component Interface);
- Use Case Diagram, which details the functionalities of one Application Component through Use Cases.

Each Use Case Diagram, was to be attached to the respective Application Component, in the Application Component Interaction Diagrams, that it details. This is visible in Figure 6 through the connection between Application Component and Use Case Diagram.

Within Application Architecture, the following entities were needed to support the created diagrams:

- Application Area – grouping of IT applications through areas within Central HR application landscape. Example: "Pay People";
- Application Service – answer from IT side to the needs of the business expressed in Business Services. Application Services are provided by Application Components and belong to Application Areas;
- Location – location where an Application Component is used;
- Organisation – organisation within Shell that uses a certain Application Component;

- Person – this entity is used to represent persons within Shell that performs technical support to an Application Component or persons that own[9] that application;
- Portfolio – grouping of Application Areas. Example: Application Area "Pay People" is part of Portfolio "HR Services". These groupings exist in order to define portfolio management ownership[10] within Central HR;
- Vendor – company that sold the Application Component to Shell.
- Within Application Component Interaction Diagram:
    - Application Component – IT application, or component within an IT application;
    - Application Component Interface – interface between Application components.
- Within Use Case Diagram:
    - Actor – someone that interacts with the IT application that the diagram refers to;
    - Use Case – functionality of the IT application.

[9] Owner, in this context, is the person that uses, or leads a team that uses, an IT application.

[10] Ownership, in this case, refers to the distribution of portfolio management responsibilities to different people.

5 Modelling

As mentioned, the project was focused on Business and Application Architecture layers of Enterprise Architecture. This chapter aims to present the diagrams that were modelled, and established as Central HR standards in a document for that purpose.

5.1 Business Architecture

Process Hierarchy Diagrams were drawn in the context of Business Architecture. This diagrams show the hierarchies of processes through levels "Process Group", "Process" and "Sub-Process", as it is possible to see in one diagram extract in Figure 7, where the different colours express those levels.

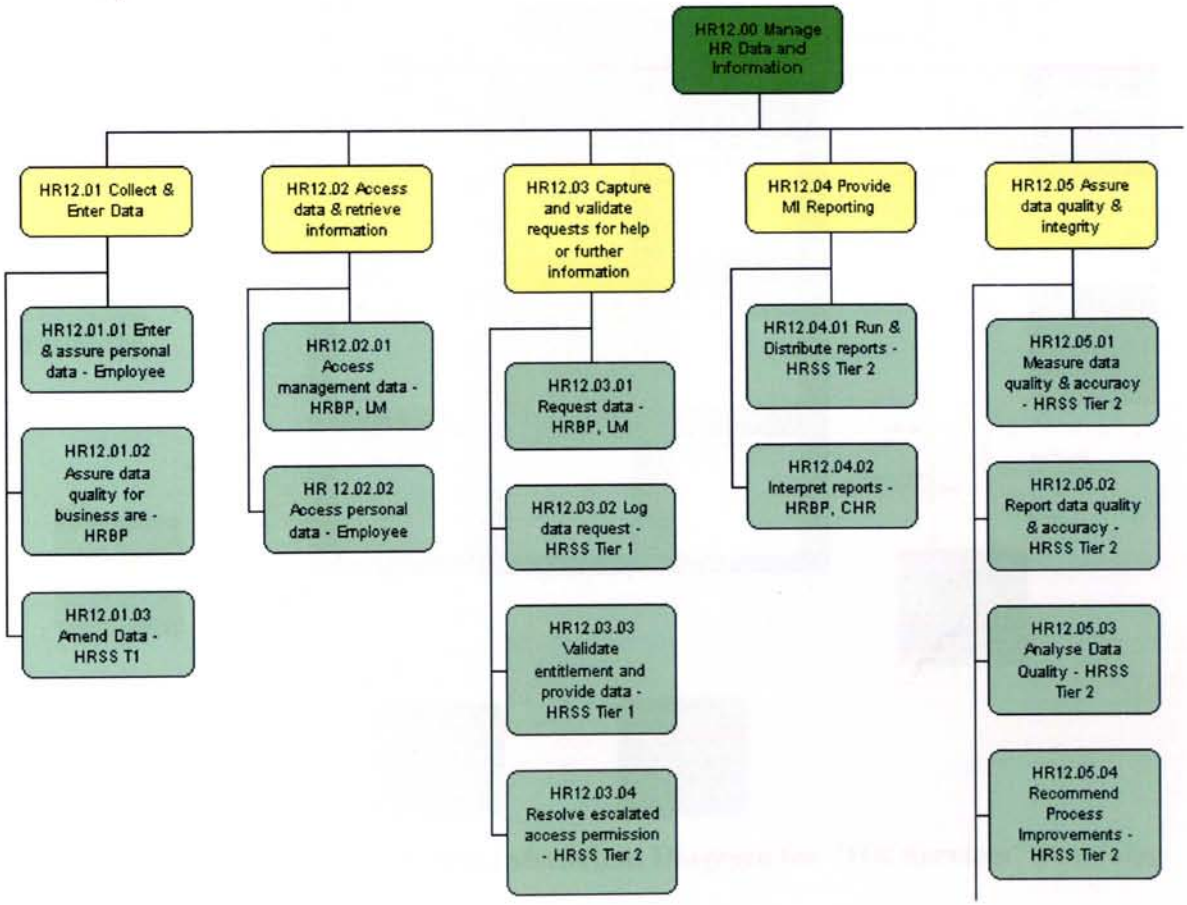

Figure 7 - Extract of a Process Hierarchy Diagram

The hierarchy present in this type of diagrams expresses that a Process Group (dark green) includes several Processes (yellow), that include several Sub-Processes on their own (light green). This is the way Processes are represented within Central HR, so there was the need to adapt it to Enterprise Architecture.

5.2 Application Architecture

Before defining which information should be represented in this Architecture, it was performed a study comparing E-PIMS and System Architect in terms of IT applications (Application Component entity) metadata, reporting capabilitie,sstrengths and weaknesses.

The objective of this report was to provide input on whether System Architect could be a substitute of E-PIMS, as main tool supporting Application Portfolio Management[11] activities.

After this, the two relevant types of diagrams for this architecture were modelled: Application Component Interaction Diagram and Use Cases Diagram.

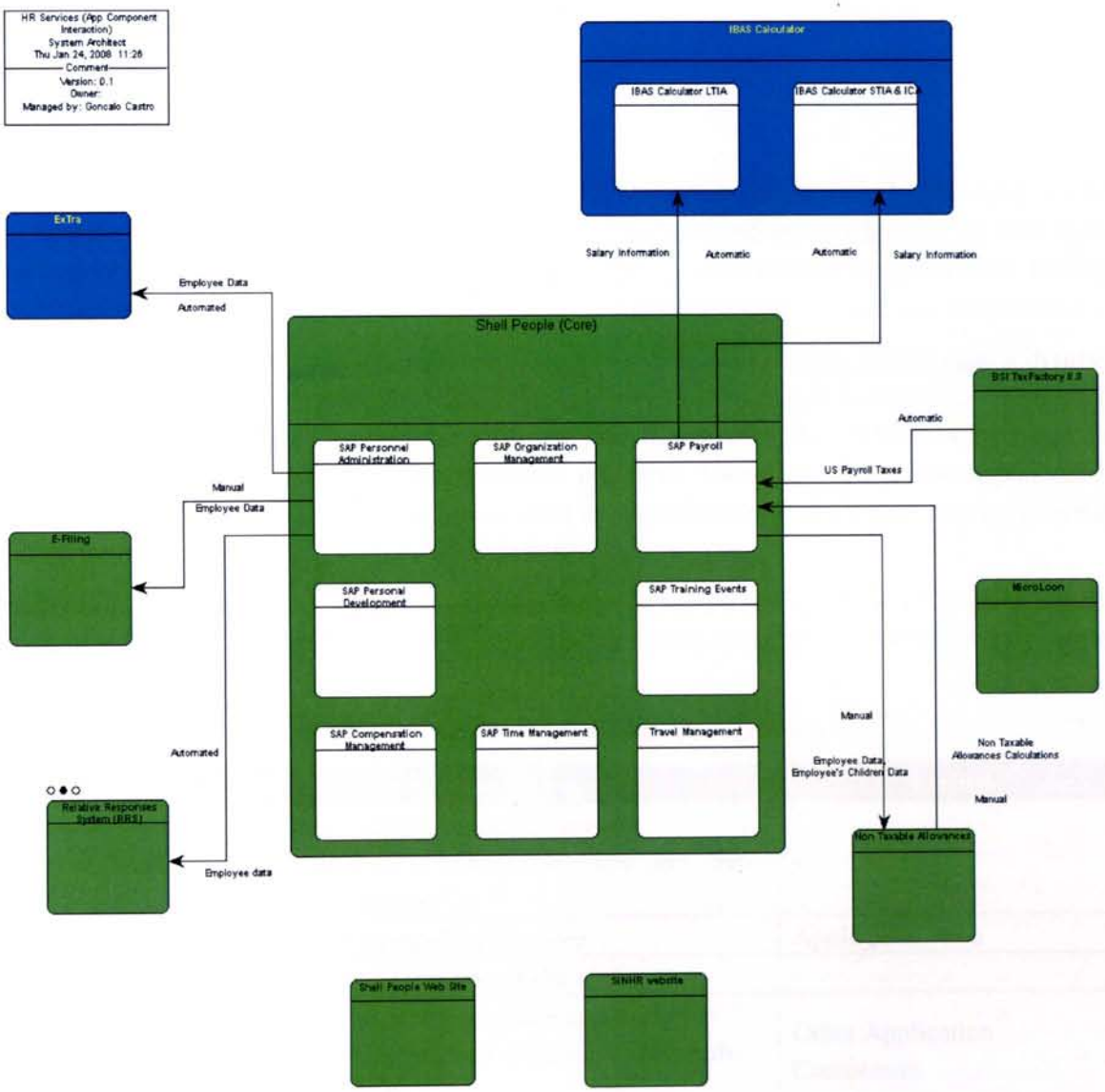

Figure 8 - Application Component Interaction Diagram for "HR Services" Portfolio

E-PIMS – System Architect Comparison

It was possible to conclude that System Architect is not a substitute for E-PIMS. System Architect has different kinds of capabilities that do not replace E-PIMS ones. The two applications have their own functionalities that complement each other.

In order that System Architect can replace E-PIMS, it needs to improve its reporting system (it does not manage to create most of the reports that E-PIMS does) and add a lot of property

[11] Application Portfolio Management – management of application portfolios (large groupings of applications) towards rationalisation, while answering the organisation's needs. It justifies and measures the financial benefits of each application in comparison to the costs of the application's maintenance and operations.

fields to Applications. On the other hand, System Architect can provide graphical overviews of portfolios, including interfaces, which are not possible with E-PIMS, and provide added value to Application Portfolio Management.

This kind of study was important to demonstrate benefits of Enterprise Architecture that are complementary to the core ones, and that support different areas of the organisation, in this case, Application Porfolio Management.

Application Component Interaction Diagrams

The aim of these diagrams is to represent an overview over the applications belonging to each Application Portfolio (example: portfolio for "Learning" related applications[12]), as well as the interfaces between those applications and what type of information is transferred through those interfaces.

In Figure 8 it is presented one example of this type of diagrams, being, in this case a diagram concerning "HR Services" portfolio. The different boxes in the diagram represent different IT applications (Application Component entities), and the arrows the interfaces between the applications (Application Component Interface entities). There are also sub-components of the applications represented in white boxes (that are Application Component entities as well), such as in the case of the big box (ERP[13] system),.

Application Component entities possesses several information detailing the IT application that they represent. This information is shown in Table 3, being some of the fields filled by related entities.

Table 3 - Application Component data

Property	Description	Entity
Level of detail	Enables the differentiation between "Components", "Sub-Components" and "Sub-Sub-Components".	-
Grouped By Area	Grouping by Application Area.	Application Area
Description	Functional description of the application	-
Sub-Application Components	Enables the detailing of one application in Sub-Components, and of these in "Sub-Sub-Components".	Other Application Component
Used By	Business using the Application.	Organisation
Used in Locations	Shell countries where the application is used.	Location
Commission Date	Date when the Application was Commissioned.	-
Decommission Date	Date planned for Decommission.	-

[12] This portfolio includes all the applications that support internal training in Shell.

[13] ERP - Enterprise Resource Planning, system that supports all areas within an organisation, integrating all of its data and processes.

Requires Technology	Technology that supports the Application.	Entity from Technical Architecture layer, not defined yet.
Owned by	Persons that own the Application.	Person
Supported by	Persons that support the Application.	Person
Vendor	Company that sold the Applications	Vendor
Security Classification	Classification of the information that the Application handles.	-
Reference Documents	Links to document management system with further information about the Application.	-

Also Application Component Interface entities include several data detailing the interfaces they represent. This data is shown in Table 4.

Table 4 - Application Component Interface data

Property	Description	Entity
Interface Type	Describes how is the interface performed (automatically or manually).	-
Description	Description of how does the interface work.	-
From and To Application Components	Application Components connected through the interface.	Application Component
Sub-Interfaces	Enables detailing an interface through Sub-Interfaces.	Other Application Component Interface
Response Data	Represents an explicit data view. Indicates the data flowing back - the response of the interface.	Entity from Data Architecture layer, not defined yet.
Request Data	Indicates the data required for the Interface.	Entity from Data Architecture layer, not defined yet.
Availability Status	Availability of the interface.	-
Interface Technology	Technology that supports the interface.	Entity from Technical Architecture layer, not defined yet.
Data - Structure Type	Notes on whether the interface data is structured or unstructured.	-
Data - Size	The anticipated quantity of data passed per interface.	-
Data - Number per Period	Expectations of the number of interface data flows per time period. Used with Size to indicate the data volumes anticipated for the interface.	-
Data - Currency	Considerations about the timing of the interface. Example: overnight batch.	-
Data - Volatility	An indication of the volatility of the Interface data.	-
Data - Persistence	Persistence requirement of the Interface data.	-

Reference Documents	Links to document management system with further information about the Application.	-

Use Case Diagrams

Each one of these diagrams details the functionalities of one Application Component. In Figure 9 it is presented one example of this type of diagram. The objective of these diagrams is to enable a better overview over the applications in Central HR Portfolio, detailing them beyond text descriptions.

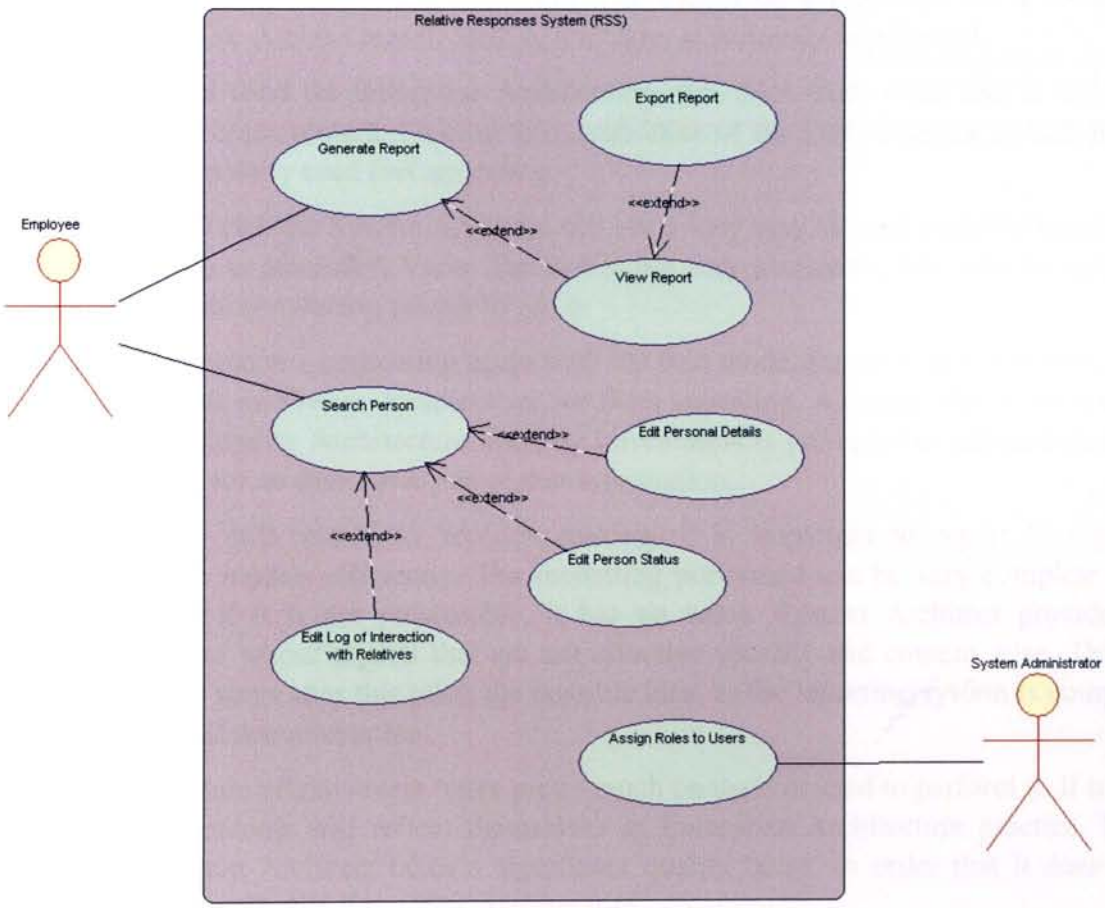

Figure 9 - Example of an Use Cases Diagram

6 Recommendations

As conclusion of the project, some recommendations can be suggested in order to move forward with Enterprise Architecture in Central HR and Shell as a whole.

Looking at the way Shell is approaching Enterprise Architecture, it seems that it can be improved through more communication between the different teams working on it in the different departments. At this point the teams are working mostly isolated from each other, taking different and non consistent approaches to Enterprise Architecture. On the other hand, this also happens because the Meta Model is not up to date, and the tool, System Architect is not reflecting it. This causes that a document that should be a reference for a common approach to Enterprise Architecture in Shell is not taken as seriously as it should.

Concerning the tool used for Enterprise Architecture, this pilot made clear that it still has plenty of room for improvement. Despite the capabilities of the tool, it seems to lack basic elements that make a daily used tool appealing:

- Usability – Telelogic System Architect still has a long way to meet usability standards of tools such as Microsoft Visio. The tool is not very pleasant to use, which can be a problem when convincing people to use it;
- Visual appearance – comparing again with the best modelling tools in the market, the visual models of System Architect are for from appealing. Although this is not a core issue for Enterprise Architecture, the way information is presented to decision makers is important for an easier analysis of that information;
- Reporting – still related to decision making, it is important to report Enterprise Architecture models efficiently. The modelling performed can be very complete and correct, but if it is not presentable, it has no value. System Architect provides a reporting tool whose reports that are not effective visually and content wise. This is the idea that stays after this pilot, the possible idea, as the reporting system is complex and still lacks documentation.

Enterprise Architecture effectiveness relies pretty much on the tool used to perform it. If it has problems, these problems will reflect themselves in Enterprise Architecture practice. It is important that System Architect takes a significant quality boost, in order that it does not seriously compromise Shell's Enterprise Architecture.

In the context of Central HR, it seems that there are some improvement points that being fulfilled can enhance Enterprise Architecture practice. Central HR should start using Metastorm ProVision tool for Business Process Modelling, being coherent with Shell standards. On the other hand, it can be interesting to start looking at Enterprise Architecture as a useful resource for different activities that already take place within HR-IT. As an example, it can be useful for Application Portfolio Reviews[14], as it can provide an integrated overview of the different portfolios, embedded in the context of the enterprise. This is the kind of benefit that is specific to Central HR, and that is complementary to the core benefits of Enterprise Architecture.

[14] Application Portfolio Reviews – Analysis on the state of the IT applications portfolio: costs, applications in the portfolio, rationalisation opportunities, etc.

After this project, it is important that the pilot is concluded with the Data and Technical layers, in order that it gives a good overview of possibilities for Enterprise Architecture best practices within Central HR. From that point, it will be possible to define standards and governance procedures of Enterprise Architecture for the function, creating structures to rollout the practice in Central HR.

7 Conclusions

In order to correctly assess the project, it seems important to look to the initial objectives and deliverables that are explained in Chapter 3. Concerning the deliverables, they were all presented in the end of the project, leaving the fulfilment of the objectives of the project to be analysed:

- To demonstrate benefits of Enterprise Architecture modelling for Central HR – from the work done, it is possible to highlight some areas where Enterprise Architecture can improve HR-IT work. Areas such as Application Portfolio Management, Business Process Management, or Infrastructure Architecture can truly benefit from a practice that integrates them, therefore improving their individual and global performance in the context of HR-IT;
- To test that the Meta Model and configuration of System Architect are fit for purpose for Central HR – part of the recommendations presented to Shell address this testing, that was a result of the analysis and modelling performed;
- To test interfaces from Metastorm ProVision and E-PIMS to System Architect – the later was not ready during the project, therefore not being possible to test. Concerning the interface from ProVision, as mentioned, Central HR is not using this tool yet, therefore it was not possible to test that interface neither;
- To determine way forward for Enterprise Architecture in Central HR – without a fully integrated and global view provided by a pilot including all the layers of Enterprise Architecture, as it was predicted to be done during the project, it was not possible to totally determine way forward for Enterprise Architecture. Data and Technical layers work was still on hold at the conclusion of this project (they were not in the scope of this project, although they were in the scope of the pilot, being other colleagues responsible for those pieces of work), and must be concluded to enable decisions on the next steps for Enterprise Architecture practice within Central HR.

In general, the fact that the project deliverables and (possible) objectives were met raises satisfaction and feeling of positive impact. This project was the start of Enterprise Architecture in Central HR, creating some basis for this important practice within the function.

On a personal perspective, Shell met the initial expectations as a truly diverse and challenging company where learning could be an important outcome of the project. It seems interesting to highlight the learning points resulting from the project:

- In a technical perspective:
    - Enterprise Architecture as main subject of the project;
    - IT as support of a big organisation, through the context where the project took place.
- In a personal perspective:
    - Adaptability and flexibility, performing the project in a truly diverse company, and abroad, having to interact with colleagues from different backgrounds and cultures. On the other hand, considering the early stage where Enterprise

Architecture is at the moment in Shell, it was important to be flexible enough to adapt to context changes that are normal when something is new;

- Analytical skills, participating in a project mostly analytical, where there was the need to analyse big quantities of data;
- Self-confidence and initiative, through the degree of autonomy received for approaching the project;
- Networking, as a needed driver to achieve the objectives of a project that demanded a lot of contact and discussion with other colleagues for sourcing information, as well as asking for feedback and ideas;
- More about myself, as it is normal in a project that requires a wide range of personal skills to be put in practice.

It is pleasant to be able to state that Shell provided all the needed conditions for the project, such as a good integration process, through several formal and informal events, such as team or Shell Student Society activities, as well as all the training and material conditions needed for a successful experience.

All these conclusions lead to a final one: it was a great experience to join Shell for this project!

Bibliography

[1] Dennis A Stevenson – Positioning Enterprise Architecture, 1997, http://users.iafrica.com/o/om/omisditd/denniss/text/eapositn.html.

[2] Institute for Enterprise Architecture Developments, 2008, www.enterprise-architecture.info.

[3] Open Group – The Open Group Architecture Framework, 2008, http://www.opengroup.org/architecture/togaf8-doc/arch/.

[4] U.S. Government Accountability Office – A Practical Guide to Federal Enterprise Architecture, 2001, http://www.gao.gov/bestpractices/bpeaguide.pdf.

[5] Wikipedia – Enterprise Architecture, 2008, http://en.wikipedia.org/wiki/Enterprise_architecture.

[6] National Institutes of Health – Enterprise Architecture, 2008, http://enterprisearchitecture.nih.gov.

[7] Architecture Framework Forum – FAQ, 2008, http://architectureframework.com/faq/.

[8] Wikipedia – Enterprise Architecture Framework, 2008, http://en.wikipedia.org/wiki/Enterprise_Architecture_framework.

[9] Zachman International – Zachman Framework, 2008, http://www.zachmaninternational.com/2/Zachman_Framework.asp.

[10] Jaap Schekkerman – Comparative Survey of EA Frameworks, 2004, ww.enterprise-architecture.info/Images/Documents/Comparative_Survey_of_EA_Frameworks.pps.

[11] Several Shell confidential documents and intranet websites.

[12] Capgemini – Integrated Architecture Framework, 2008, http://www.capgemini.com/services/soa/ent_architecture/iaf/.

[13] United States Office of Management and Budget – Federal Enterprise Architecture, 2008, http://www.whitehouse.gov/omb/egov/a-1-fea.html.

[14] Architecture Framework Forum – DoDAF, 2008, http://architectureframework.com/dodaf/.

[15] Architecture Framework Forum – MODAF, 2008, http://architectureframework.com/modaf/.

[16] Institute for Enterprise Architecture Developments – Enterprise Architecture Tools Overview, 2008, http://www.enterprise-architecture.info/EA_Tools.htm.

[17] Forrester Research, Inc. – Enterprise Architecture Tools, Q2 2007, 2007, http://www.casewise.com/NR/rdonlyres/0E9706B3-DD0E-4F4E-BA44-E26346048D2B/0/EAToolsWave.pdf.

[18] Institute for Enterprise Architecture Developments – Enterprise Architecture Tool Selection Guide, 2007, http://www.enterprise-architecture.info/Images/EA%20Tools/Enterprise%20Architecture%20Tool%20Selection%20Guide%20v4.2.pdf.

[19] Jurgens Pieterse – Be smart in choosing enterprise architecture tools, 2005, http://blogs.ittoolbox.com/eai/practices/archives/be-smart-in-choosing-enterprise-architecture-tools-6714.

APPENDIX A: Definitions

Application Portfolio Management – management of application portfolios (large groupings of applications) towards rationalisation, while answering the organisation's needs. It justifies and measures the financial benefits of each application in comparison to the costs of the application's maintenance and operations.

Application Portfolio Reviews – Analysis on the state of the IT applications portfolio: costs, applications in the portfolio, rationalisation opportunities, etc.

As-Is architecture – current architecture implemented in the organisation. Other types of architectures are the "To-be", architecture that will be implemented in the future, and "Intermediate" that is a passing point between "As-Is" and "To-be" architectures.

Entity – Enterprise Architecture consists of entities and models. An entity can have attributes which indicate its specific behaviour and relations with other entities. Entities can be grouped into models to support a required architecture viewpoint.

ERP – Enterprise Resource Planning, system that supports all areas within an organisation, integrating all of its data and processes.

E-PIMS – Enterprise Portfolio Information Management System – System where all the information concerning Shell's IT applications is stored. Examples: locations where the application is used; people that support the tool; etc. In this context, it was proposed the creation of an interface that could feed System Architect with IT applications related information.

Function – organisational unit within Shell, a portfolio of activities and responsibilities operating according to common objectives and strategies with formally delegated organisational mandates. It could be called department in a simplified terminology. Examples of functions are: Central HR, Central Finance, etc.

IT – Information Technology.

Meta Model – precise definition of the constructs and rules needed for creating semantic models. In the case of this project it specifies a common language for Enterprise Architecture practice within Shell.

XML – eXtended Markup Language. Document language that enables the exchange of structured information.

Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias, s/n 4200-465 Porto PORTUGAL
www.fe.up.pt