Chapter submitted to the book "Enterprise Modeling and Computing with UML", IRM Press.
Revised version. November 2005.

Enterprise Architecture Modeling with the Unified Modeling Language

Pedro Sousa[1,2,3], Artur Caetano[1,2] André Vasconcelos[1,2], Carla Pereira[3], José Tribolet[1,2]

[1] Department of Information Systems and Computer Science,
Instituto Superior Técnico, Technical University of Lisbon.

[2] Organizational Engineering Center, INESC.

[3] Link Consulting, S.A.

Surface mail address: INESC, Rua Alves Redol 9, 1000-029 Lisboa, Portugal.
Contact e-mail: artur.caetano@dei.ist.utl.pt

Abstract

This chapter describes the key concepts for modeling the organization's enterprise architecture using the Unified Modeling Language (UML). Enterprise architecture consists on defining and understanding the different elements that shape the organization and how these elements are inter-related with the purpose of understand and facilitate organizational evolution and change. It separates core organizational concerns as different architectural views; the authors argue that modeling the multidimensional aspects of the enterprise should be organized into five architectural components: Organization, Business, Information, Application and Technological architectures. These five components are supported in a small set of fundamental concepts described using UML 2.0. Furthermore the authors argue that any organization model may be abstracted to three elements: Activity, Role and Entity. The authors also propose a set of rules for assessing the alignment between the enterprise architectural elements.

Keywords: Enterprise Modeling, Business Process Diagram, Alignment of IS Plans with Business Plans, Unified Modeling Language, Business Engineering.

1 Introduction

Organizations are complex entities that deal with contrasting concepts such as people, control and value chains, business processes and information systems and technology. Representing the knowledge about an organization proves to be a challenging task since it requires all of these aspects to be represented in a coherent and integrated way and not as a set of unrelated and independent elements. Failing to deliver such an integrated representation contributes to the materialization of heterogeneous and misaligned views on the organization that would hinder the detection of problems and improvements as well the ability to assess the overall organization.

For an organization to change it must be self-aware, meaning that if the knowledge on the organizational aspects is not comprehensively shared and understood there will be a mismatch between the actual state of affairs and the state as perceived by the different stakeholders. This gap will hold back the definition and implementation of the changes that are required for an organization to evolve. In addition, with the ubiquitous proliferation of information systems and technology, the above problems are accentuated as, on the one hand, the pressure to change grows and, on the other, the systems facilitate information sharing and process automation, regardless of the quality of the information and how the processes are actually aligned with the organization goals. Despite the investments made on the research and development of systems and technology, most organizations still do not have adequate tools or methodologies that enable the management and coordination of these systems in such a way as to support planning, changing, decision making, controlling and, especially, as a means to use these systems to explicitly leverage competitive advantage.

Identifying the architecture of the enterprise should therefore be considered as a fundamental step for any organization that renders important to be ready to act rather than react and to be able understand whether its elements are aligned. The **enterprise architecture** results from the continuous process of representing and keeping aligned the elements that are required for management of the organization. In this paper, the term architecture stands for the fundamental arrangement of the components within any kind of socio-technical system, as well as their relationships to each other and the environment, and the design rules for developing and structuring the system (IEEE 2000). The components are depicted in the form of a model while reducing insignificant and redundant aspects. The design rules, on the other hand, describe stipulations for the development and structuring of the model, which specify the types of components, the types of relationships and consistency conditions for the use of components or their relationships.

Therefore, and set in the context of an organization, the definition of the **enterprise architecture** strategically aims at:

- Modeling the role of information systems and technology in the enterprise in order to control its life cycle.
- Assessing the **alignment** between enterprise-wide concepts so that suitable corrective actions can be defined.
- Aligning information systems with business processes and information, thus establishing a reference for efficient resource management.
- Planning sustainable changes.
- Providing the means to generate enterprise knowledge to assist management decisions in an agile and competitive environment.

1.1 Target Audience

Since an enterprise architecture allows perspectives with different levels of detail to be created, the concepts addressed in this paper are suitable to a wide target audience, including managers, modelers and architects, and software designers and developers. Managers can make use of the enterprise architecture to understand how the multiple aspects of the organization are interconnected, namely, how strategy is realized in **business processes**, and how processes depend on and are supported by the organizational service infrastructure. Modelers and architects use the architecture as the organizational *lingua franca* to represent and communicate strategy, **business processes**, information and systems and to assess and evaluate the alignment between them. Finally, system and software designers exploit the architecture as a means to identify business-driven service requirements and to explore opportunities for service reuse.

1.2 Chapter Scope and Contributions

An organization is a complex man made system that comprises a formal group of people, which share one or more goals (Scott 1997). A system is a collection of interrelated elements within a unified whole. The system concept is often used to describe a set of entities which interact, and for which a model can often be constructed. There exist multiple ways to describe complex systems, including system dynamics (Forrester 1961) and systems thinking (Senge 1994). System thinking makes use of techniques to study systems in a holistic way rather than in a reductionist terms. It aims to gain insights into the whole by understanding the interactions and processes between the elements that comprise the whole system. Systems thinking can help avoid the silo effect, where a lack of organizational communication can cause a change in one area of a system to adversely affect another area of the system.

The first step into defining an enterprise architecture model is establishing the properties that the model stands for and the architectural perspectives or views that must exist so that these properties can be successfully architected. It is a requirement that the enterprise concepts and models should be simple enough so that they may be used as a communication, analysis and discussion tool. The enterprise models need to be defined with simple common sense rules, but also be sound and coherent. This means the models must provide a high-level view that abstracts technical details and interconnections while keeping a complex structure traceable and coherent.

An architecture must be designed bearing in mind its expected usage. It is about specifying how the things on the enterprise should operate and interact having a means specified. It should allow, on the one hand, describing organizations, business processes, business information and systems, and, on the other, assessing the alignment between these concepts. This is accomplished by defining the semantics of a set of modeling concepts and a set of five traceable architectural views to represent and relate these concepts in the context of enterprise architecture. These artifacts will be expressed using the Unified Modeling Language (OMG 2004), confining and extending its generic modeling mechanisms using the standard profiling mechanisms to adapt it to the enterprise-modeling domain.

The five architectural views on the enterprise focus individually on Organizational, Business, Information, Applications and Technological concerns. The organizational view focuses on the things that are defined independently of the business and are shared by the whole enterprise from the management's standpoint. This view addresses concepts related with the enterprise vision and strategic goals. The business view deals with the issues related to a specific business, including specifying the value chain and its business processes and other concepts related to the operational behavior of the business. The information view concerns the management of the enterprise information that is required to support all strategic and operational processes and decisions. The application view focuses on the specification of a technological independent architecture of services that are put in place to support the business. Finally, the technological view concerns the specification and description of the information, computational and other technology that is required to support the organization.

The key contribution of this chapter is expressing the components and semantics of an architectural model for process-oriented organizations using five views that separately address different concerns related to the organization, business, information, application and technology. These aspects are then coherently integrated in the enterprise architecture model that aims fulfilling the goals previously discussed. The proposed framework relies on a metamodel that defines the fundamental concepts and their relationships. It also makes use of the object-oriented paradigm, exploiting mechanisms such as specialization and aggregation, with the goal of maximizing reusability and facilitating the discussion and communication of the models, thus promoting understandability. The representation of the models relies on the syntax and semantics of the Unified Modeling Language 2.0. In brief, the proposed framework will address the following questions:

- How to model business activities and business processes?
- How to model the resources consumed, modified and produced that result of executing an activity?
- How to model business services, i.e. the operations an activity requires to be executed?
- How to model system services, i.e. the services provided by systems supporting business services?
- How to model the architecture of the business support systems?
- How to express the **alignment** between business processes, business activities, business information and support systems?
- How to express the **alignment** between the requirements of an activity and the services provided by business support systems?

However, we will not address a method to evaluate an enterprise architecture or model. Also out of scope is a detailed representation of the organization’s strategy. Finally, the alignment between the organization strategy, strategic and operational goals and indicators and business processes is not addressed here.

1.3 Chapter Structure

This chapter is structured as follows: the next section reviews the concept of enterprise architecture and the major work in this area. Section 3 presents the five components of the proposed enterprise architecture model and its UML representation. These architectural components detail the organizational, business, information, application and technological aspects of the organization. The fundamental modeling concepts used in the enterprise architecture model and the corresponding UML representation are then described in Section 4. This section also synthesizes the overall enterprise architecture model, describing its structural and dynamic aspects. Section 5 provides a set of rules to analyze and assess the alignment between the architecture elements and views. Finally, section 6 provides concluding remarks.

2 Enterprise Architecture

Enterprise architecture is a concept that has been around for almost twenty years. In the meantime, areas such as Business Process Management, Business Process Reengineering and Total Quality Management, just to name a few, shed light on the relationship between process-oriented management and business process support through information technology. However, most approaches that stemmed from these fundamental studies do not provide holistic models on the enterprise’s components as put forward by orthodox enterprise architecture. This holistic vision is shared by many references on enterprise architecture found in mainstream literature:

- **Enterprise architecture** consists of defining and understanding the different elements that make up the enterprise and how those elements are inter-related (Open Group 2003).

- **Enterprise architecture** is the holistic expression of an organization's key business, information, application and technology strategies and their impact on business functions and processes. The approach looks business processes, the structure of the organization, and what type of technology is used to conduct these business processes (Meta Group 2005).
- **Enterprise architecture** is a relatively simple and straightforward model, framework, or template that can be used by everyone within your enterprise to assess how things are going, to facilitate their work, and to design new projects (Egan 1988).
- **Enterprise architecture** is the set of representations required to describe a system or enterprise regarding its construction, maintenance and evolution (Zachman 1997).
- **Enterprise architecture** is a strategic information asset base, which defines the business mission, the information necessary to perform the mission, the technologies necessary to perform the mission, and the transitional processes for implementing new technologies in response to the changing mission needs (FEAPMO 2003).
- **Enterprise architecture** is a complete expression of the enterprise; a master plan which "acts as a collaboration force" between aspects of business planning such as goals, visions, strategies and governance principles; aspects of business operations such as business terms, organization structures, processes and data; aspects of automation such as information systems and databases; and the enabling technological infrastructure of the business such as computers, operating systems and networks (Schekkerman 2004).

The preceding definitions share a common concern: **enterprise architecture** is about the structure of the things of relevance in the enterprise, their components, and how these components fit and work together to fulfill a specific purpose. In the remainder of this section, we will review a set of significant approaches to enterprise architecture.

The ANSA project (ANSA 1989, Herbert 1994, http://www.ansa.co.uk/) focused on developing a basic understanding of distributed architectures. ANSA recommends a set of components, rules, recipes, and guidelines to help designers make design decisions. This project was most likely the first to propose specific projections that were claimed to provide complete coverage of information processing systems. The projections on enterprise, information, computation, engineering and technology views, were later taken up in the open distributed processing standards, such as CORBA. This concept enables separating the multiple concerns of a complex system in such a way that they can be individually addressed and later composed in a global representation of the system. Thus, the concept of enterprise projection shares a common goal with other approaches to enterprise architecture, such as Zachman's framework and the one proposed in this chapter. One of the first standards to embrace enterprise projections was ISO's RM-ODP. The Reference Model for Open Distributed Processing (ISO 1995, Farooqi 1995, Schurmann 1995) aimed at integrating a wide range of distributed-systems standards and maintaining consistency across such systems. To do so, RM-ODP includes descriptive as well as prescriptive elements. The descriptive elements provide a common vocabulary while the five prescriptive elements, known as viewpoints, constrain what can be built as required by a system. Specifically, it defines the enterprise viewpoint for system boundaries, policies, and purpose; the information viewpoint to represent distributed information; the computational viewpoint for decomposition of system into distributable units; the engineering viewpoint for description of components needed and, finally, the technology viewpoint for describing the implementation details of components.

The Zachman Framework for Enterprise Architecture (Zachman 1987, Sowa 1992, O'Rourke 2003, http://www.zifa.com/) is amongst the recognized works on enterprise architecture from both modeling and management perspectives. It provides a view of the subjects and models needed for developing and documenting a complete enterprise architecture and is described in a matrix which provides on the vertical axis multiple perspectives of the overall architecture and on the horizontal axis a classification of the various artifacts of the architecture. The framework is structured around the perspectives related to the user roles involved in planning, designing, building and maintaining enterprise information systems. These perspectives are:

- Scope (planner's perspective). Concerns the strategic aspects of the organization, the context of its environment and scope.
- Enterprise Model (owner's perspective). Concerns the business perspective of the organization, how it delivers value and how it will be used.
- System Model (designer's perspective). Concerns the systems of the organization, ensuring they fulfill the owner's expectations.

- Technology Model (builder's perspective). Concerns the technology used to support the systems and the business in the organization.
- Detailed Representations (subcontractor's perspective). Concerns the builder's specifications of the system components to be subcontracted internally or to third parties.

The six columns focus on separating different domain concerns:

- Data (what). Concerns the definition and understanding of the organization's information.
- Function (how). Describes the process of translating the mission of the organization into the business and into successively more detailed definitions of its operations.
- Network (where). Concerns the geographical distribution of the organization's activities and artifacts, and how they relate with each perspective of the organization.
- People (who). Describes who is related with each artifact of the organization, namely business processes, information and IT. At higher-level cells, the "who" refers to organizational units, whereas in lower cells it refers to roles and system users.
- Time (when). Describes how each artifact of the organizations is organized in time.
- Motivation (why). Describes the translation of goals in each row into actions and objectives in lower rows.

The Zachman Framework contains suggested specification documents for each cell of the matrix (Zachman 1987). For example, it suggests using ER technique for modeling the data description in the owners view business model or using functional flow diagrams for modeling the owners view process description. However, it does not define a metamodel to integrate the information within each cell nor does it describe how to trace such information (Frankel 2003). Nevertheless, the framework is independent of specific methodologies. In addition, there are no specific techniques described to create the suggested specification documents in each cell of the framework.

Cap Gemini Ernst & Young has developed an approach to the analysis and development of enterprise and project-level architectures known as the Integrated Architecture Framework (Goedvolk 1999). It can be considered a design tool, aiming at the development of mutually aligned business and IT systems through a unified architecture. IAF breaks down the overall problem into a number of areas covering business (people and processes), information (including knowledge), information systems, and technology infrastructure. There are two special areas addressing the Governance and Security aspects across all the other areas. Analysis of each of these areas is structured into four levels: contextual, conceptual, logical and physical, representing phases of the design process and not different levels of management attention. The contextual view justifies the organization and describes its contextual environment. It corresponds largely to Zachman´s planner's perspective row. The conceptual view describes what the requirements are and what the vision for the solution is. The logical view describes how these requirements and vision are met. Finally, the physical view describes the artifacts of the solution. These views are not related to Zachman´s perspectives since in IAF, business, information, information systems and technological infrastructure are the artifacts of the architecture whereas in Zachman, business, information systems and technology are perspectives.

TOGAF, The Open Group Architecture Framework (The Open Group 2005), is an industry standard architecture framework used to develop enterprise architecture descriptions. It enables designing, evaluating, and building the architecture for an organization. The key to TOGAF is the Architecture Development Method (ADM) an approach for developing enterprise architecture descriptions that meets the needs of the specific business. TOGAF mainly consist of three parts: the ADM, the Enterprise Continuum and the Resource Base. None of the three parts delivers a metamodel to assure a consistent reuse of components during an iterative use of the procedure. The ADM is iterative, over the whole development process, between phases, and within phases (architecture development cycle). The cycle consists of eight phases: architecture vision, business architecture, information system architectures, technology architecture, opportunities and solutions, migration planning, implementation governance, and architecture change management. Each of these eight phases contains further detailed steps. The use of reference models (which are provided by the TOGAF Enterprise Continuum) and guidelines (which are provided by the TOGAF Resource Base) can be regarded as other important activities in the developing process. TOGAF uses a set of activities that build a detailed procedure model for developing enterprise architecture descriptions. Even though TOGAF ADM describes the different inputs and outputs for each phase of the architecture development cycle, there are no specification documents that describe the output and no instructions that clearly describe in which phase of the development cycle that specification documents have to be generated. From that point of view, specification documents only exit in part within the TOGAF framework.

In the next sections, we propose an enterprise architecture model that naturally follows the approach and several goals of the work here described. However, it emphasizes the traceability, **alignment** and assessment between enterprise components, facilitating their reuse and independent co-development. The enterprise architecture model, graphically expressed in the UML, relies on a metamodel to define its elements and corresponding relationships.

3 Enterprise Architecture Views

We propose modeling the multi-dimensional aspects of the **enterprise architecture**, and, based on that, defining and evaluating the **alignment** between business processes, business information and the corresponding support systems and technology. The first step in this direction is identifying a minimal set of components able to represent the required organizational concepts while ensuring that the **alignment** between these concepts can be assessed.

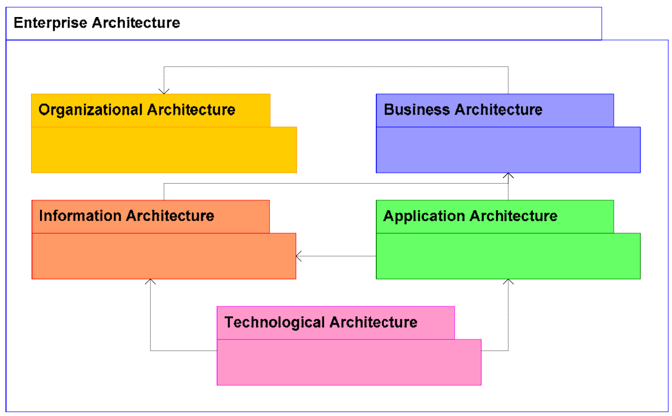

Figure 1. The five enterprise architecture components.

The enterprise architecture model comprises five architectural components: **Organizational Architecture**, **Business Architecture**, **Information Architecture**, **Application Architecture**, and **Technological Architecture**. Each of these sub-architectures is individually represented and organized as a UML package as depicted in Figure 1. Each package owns its model elements and its elements cannot be owned by more than one package. The relationships, depicted as dotted arrows, represent the dependencies of each package. The following subsections detail each of the architecture components.

3.1 Organizational Architecture

The **organizational architecture** deals with the aspects directly related with the organization that are not related with the specific business it conducts nor with the mechanisms used to accomplish the creation of value. It therefore includes concepts such as the enterprise mission, vision and strategy and the definition of organizational units.

The **enterprise mission** and **vision** state what the enterprise is and does, defining why the organization exists. The mission is a concise and internally focused statement of the reason for the organization's existence, defining the purpose toward which its activities are directed and the values that guide employee's activities. The mission also describes how the organization expects to compete and deliver value to customers. On its turn, the vision points out where the enterprise aims to be in the future through the enterprise strategy, defining the mid to long-term goals of the organization. The vision statement should be market-oriented and made publicly available. It describes, often in visionary terms, how the organization wants to be perceived by the world (Kaplan 2004).

The **enterprise strategy** states the key decisions and actions the enterprise is willing to do in order to accomplish its vision and describes how it intends to create value for its shareholders, and customers. Strategy is about selecting the set of processes in which an organization will excel to create a sustainable difference in the marketplace.

The **organizational architecture** includes other concepts such as organizational policies, organizational units, as well as human resource issues like people roles, carrier, goals and so on.

3.2 Business Architecture

The **business architecture** results from the implementation of business strategies and the definition of processes. The functional requirements of business process support systems, i.e. the information systems that will operationally support the business, are derived from this architecture.

The core concept within the business architecture is the business process. A **business process** is a set of value adding activities that operates over input entities producing output entities. The activities comprised in the process are coordinated, meaning they are either orchestrated by a central controlling entity or choreographed. The actual coordination mechanism is only relevant when detailing how the process is enacted. What distinguishes an arbitrary set of coordinated activities from a business process is the fact that the process must add value to some customer, whether internal or external to the organization. Thus, although an organization always comprises multiple sets of coordinated activities, each may or may not be classified as an actual business processes. Business processes are orthogonal to the organization's units. In fact, they frequently cross the boundaries of several units. Macroscopically, business processes are abstracted as value chains (Porter 1985).

An **activity** describes the business roles required for its operation. These roles are played by the organization entities and usually include actor role, resource **role** and observable state role. An activity requires one actor or a combination or team of actors to be executed. The **actor** represents a person, a machine or device, or an information system. An actor provides the services require for fulfilling the business role required by the activity. A resource is used as input or output of an activity during its operation. A resource is usually created, used, transformed or consumed during the operation of the activity. An observable state is specific resource role that is used as a means to observe the status of an activity. An activity is performed during a specific period. As a precondition for its enactment, all of the business roles must be fulfilled by specific entities. These entities will be engaged in playing their roles for the duration of the activity. The activity post condition is that all of the roles will have finished playing their part.

3.3 Information Architecture

The **information architecture** describes what the organization needs to know to run its processes and operations as described in the business architecture. It defines a view on the business information that is system and technology independent. It is an abstraction of the information requirements of the organization and provides a high-level logical representation of all the key information elements that are used in the business as well as the relationship between them (Inmon 1999, Gilchrist 2004).

Business information is structured as a collection of **informational entities**. An entity can result from the composition or specialization of other entities in the object-oriented sense. **Information entities** are classes, meaning they can be typified. Entities describes of most artifacts of the enterprise, namely those resources required by processes, including business, support and management processes. As such, they have an identifier, defined from a business perspective, and a set of attributes. Attributes are related to the roles the entities play. Therefore, each role integrates its set of attributes into the entity. The overall set of attributes results from merging each individual set of attributes derived from each role the entity is able to play.

3.4 Application Architecture

The application architecture fulfills two goals: (1) supporting the business requirements and (2) allowing efficient management of the organization's entities. To satisfy these goals, the application architecture should be derived top-down from the analysis of the business and information architectures.

The application architecture defines the applications needed for data management and business support, regardless of the actual software used to implement systems (DeBoever 1997). It functionally defines what application services are required to ensure processes and entities are supported in acceptable time, format and cost (Spewak 1992). According to the International Enterprise Architecture Center (2005) it should describe the characteristics, styles and interactions among multiple applications.

Thus, the application architecture defines the applications required to enable the business architecture. This includes identifying how the applications interact with each other, how they will interact with other business integration elements and how the application and data will be distributed throughout the organization. It typically includes descriptions of automated services that support the business processes and of the interaction and interdependencies between organization's application systems, plans for developing new applications and revision of old applications based on the enterprises objectives.

The architecture of a business process support system is described using a structure of **information system block**, IS block for short, which depict information system or application building blocks. An **IS block** is then defined as an organized collection of services, mechanisms and operations designed to handle organization information (Spewak 1992). We extend this definition to allow an IS block to handle not only information but also the coordination and other mechanisms required to support business process. Each block may state several attributes, such as availability, scalability (ability to scale up performance) and profile-based access (ability to identify who does what).

3.5 Technological Architecture

The **technological architecture** represents the technologies behind application implementation as well as the infrastructure and environment required for the deployment of the business process support systems.

The technological architecture addresses a large number of concepts since it must cope simultaneously with continuous technological evolutions and the need to provide different specialized technological perspectives, such as those centered on security and hardware. These concepts are abstracted as an information technology block. An **IT block** is the infrastructure, application platform and technological or software component that realizes or implements a set of IS blocks. It encompasses three parts:

- IT Infrastructure Block. Represents the physical and infra-structural concepts existing in the information systems architecture: the computational nodes (e.g. servers, personal computers or mobile devices) and the non-computational nodes (e.g. printers, network components) that support application platforms.
- IT Platform Block. Describes the implementation of the services used in the IT application deployment, such as the Operation System, the web platform and the EAI platform.
- IT Application Block. Is the technological implementation of an IS block. It classifies the type of implementation, such as presentation, logic, data or coordination block, as well as the technological concepts used in the implementation, such as object or component-oriented architecture and types of modules. An IT application block makes use of services provided by the IT platform.

Two other concepts are also important in the description of the information system architecture:

- Operation. An abstract description of an action supported by a service (W3C 2002). An operation is the finer grain concept in the information technology architecture
- **Service**. The aggregation of a set of operations provided by an architectural block. It can be seen as a generalization of the concept of web service notion (W3C 2002). We consider three distinct services:
  - Business Service. A set of operations provided by IS blocks supporting business processes.
  - IS Service. A set of operations provided by an IS block to others IS blocks. This is used to aggregate multiple IS blocks.
  - IT Service. A set of technological services provided by the specific application platforms.

4 The Enterprise Architecture Model

The architectural views describe and relate the fundamental concepts that, as a whole, describe the enterprise architecture. Each is represented as a class within a specific package, as depicted in Figure 2. This section details the fundamental concepts and their relationships that are required to represent the enterprise architecture according to the five views that were defined in section 3.

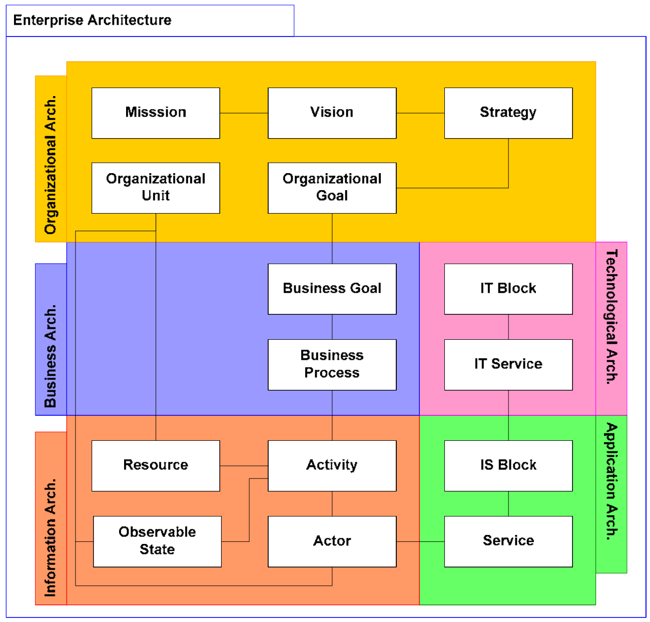

Figure 2. The fundamental concepts within each of the enterprise architecture views.

4.1 Fundamental Concepts

An organization can be modeled as a collection of business nouns that interact as described by a number of verbs. The nouns represent things within the organization that are of interest regarding the purpose of the model. The verbs stand for the enterprise activities that define how work is done and how value is added, thus describing its business processes and activities. Here we define the fundamental concept of **entity** and **activity** and that of **role**. These three concepts allow complex interactions of entities to be abstracted. The relationships between these three elements are depicted in the next Figure.

Figure 3. Relationships between Activity, Role and Entity.

4.1.1 Entity

An organization is composed of entities. Entities are nouns that have a distinct, separate existence, though it need not be of material existence. There is also no presumption that an **entity** is animate. An animate entity is able to exhibit active behavior. In enterprise modeling, an entity can be a person, place, machine, concept or event that has meaning in the context of the business, and about which some information may be stored because it is relevant for the purpose of the model.

Entities can be classified according to its attributes and methods. Entities may relate structurally to other entities, as in the case an entity is composed by other entities (e.g. an inventory is composed of products). An **entity** may also be specialized to restrict the features of a more general entity.

An **entity** is characterized by its attributes and methods. These features can be either intrinsic or extrinsic. Intrinsic features describe the entity in isolation, while extrinsic features arise from the relationships with other entities. For example, the entity `Person` has intrinsic features such as `Age` and `Sex`, and extrinsic features such as `Job Position` and `Salary`, which derive from a transitory relationship between the `Person` and the `Organization`. The state of the intrinsic features may change over time (e.g. `Age`) but always characterize the object. Extrinsic features only manifest themselves while a relationship is valid and may become unsuitable when the relationship is no longer valid. For example, the `Job Position` or `Salary` properties are not appropriate to characterize an unemployed person. This means an entity's extrinsic features are directly constrained by the potential relationships it may have in a given context, such as its whole lifespan, a bounded business collaboration or specific time interval.

When entities interact with other entities, they do so through **roles**. In this case, we say the entities are collaborating and that each of the entities is playing a set of roles in the context of a specific business **activity**.

UML representation

An **entity** is a UML Class. The features of this class represent the intrinsic attributes and methods of the entity. Intrinsic means that the features exist per se, regardless of its collaborations. An entity may relate to other entities via aggregation or generalization. An entity may only relate structurally to other entities. The collaboration or interaction between activities is mediated by roles.

Figure 4 shows a class diagram depicting three entity classes along with the corresponding attributes. In this example, the Inventory entity aggregates Products.

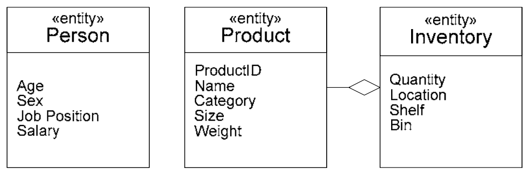

Figure 4. Entity class.

4.1.2 Role

A **role** is the observable behavioral of an entity in the scope of a specific collaboration context. Hence, a role represents the external visible features of that entity when it collaborates with a set of other entities in the context of some activity. An entity relates to zero or more role classes through the stereotyped «play» relationship. Each role represents a subset of its external or extrinsic features in the context of a specific collaboration defined in a role model.

Roles aim at separating the different concerns that arise from the collaborations between the entities fulfilling an activity. A role may be bound to multiple entities via the «play» relationship. Binding a role to an entity means that a specific instance of that entity is able to express the behavior defined by the role. It also means that the attributes and method of the role will be part of the entity's feature set.

A role is also a type and may be classified according to its attributes and methods. Therefore, it can be generalized and aggregated as a regular class.

Roles are described in role models. A role model describes how roles are structured and they collaborate in order to fulfill a task. The role model may also specify constraints.

UML Representation

Roles are described as UML Classes. A role may be specialized to restrict its behavior and may be composed of other roles. A composed role is able to put into play the behavior of each of the roles it comprises. The structural relationships between roles are shown on class diagrams.

Roles relate to entities through the «play» classifier relationship stereotype. Actually, entity instances play role instances. Role models are Packages that comprise a class diagram to describe the role structure and a UML dynamic diagram to describe its collaborations.

The class diagram depicts the roles and the role associations required to fulfill a task. It also describes any constraints or business rules that govern the role associations. Constraints can be expressed in OCL or in natural language, depending on the level of formality that is required.

Figure 5 shows the structural dependencies between two roles, Employee and Employer, both defined in the Works For role model. It also depict the binding between two entities, Person and Organization, and the two roles Employee and Employer.

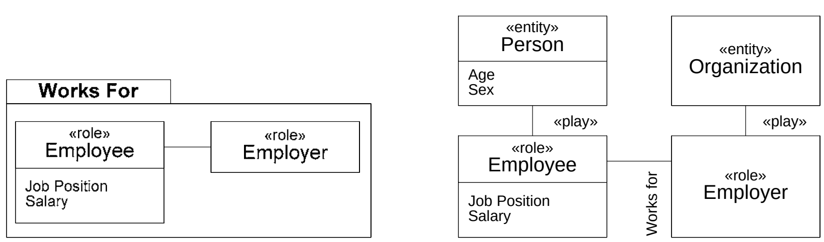

Figure 5. The Works For role model showing the dependencies between two related roles (left) and binding roles to entities (right).

It is important to contrast the definition of the Person class in Figure 5 and Figure 4. Figure 4 depicts the attributes of the Person without making clear what attributes derive from what collaborations. This mixes its intrinsic attributes (age and sex) with the external attributes that are only relevant when the person behaves as an employee (job position and salary). In opposition, the diagram in Figure 5 uses roles to separate the Person's external attributes from its intrinsic attributes. It can be observed that the job position and salary are extrinsic attributes and are dependent of the specific role Employee. Moreover, the role model makes clear that the Employee role relates with the Employer role, in the context of the Works For collaboration. Separating the intrinsic from extrinsic features allows for a more efficient informational architecture and application, since entities may be designed so that they are independent of the specific activities that use them, not only improving the reusability of the entities but also the ability to understand why a specific feature exist in a business entity.

4.1.3 Activity

An **activity** is an abstraction representing how a number of entities collaborate through roles in order to produce a specific outcome. Similarly to an algorithm, an activity aims accomplishing some task which, given an initial state, will always end in finite time and in a recognizable end-state. An activity may also be functionally decomposed into a finite set of further activities, thus add detail to the specification.

An activity specifies what entities are required to realize a task. As seen earlier, roles are used to separate the description of the actual entity features from the features required by the collaboration in context of the activity. In this way, activities and entities are described separately, and roles may be reused in different activities.

UML Representation

An activity is described by a number of role collaborations as shown earlier in the previous section. To improve readability and use a notation closer to that of BPMN (BPMI 2004) and IDEF-0 (ICAM 1981), we can alternatively represent an activity as in Figure 6, using UML's action or send signal action icons.

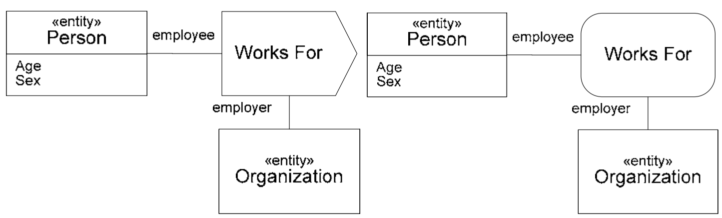

Figure 6. Role-typed entity association mediated by an activity classifier.

An activity often results from a number of interacting entities playing a set of roles specialized from four generic roles: resource, actor, observable state and business goal (v. Figure 7). The resource role is played by the entities that are used as input to the activity operation. These resource entities are handled by a number of **actors** to generate another set of output resource entities. An **entity** plays an actor role whenever is performing active behavior, i.e., putting into action its skills or capabilities. Actors may be played by entities modeling people, mechanical devices or information systems. During these operations, actors may or may not contribute to the achievement of business goals. These goals are themselves entities.

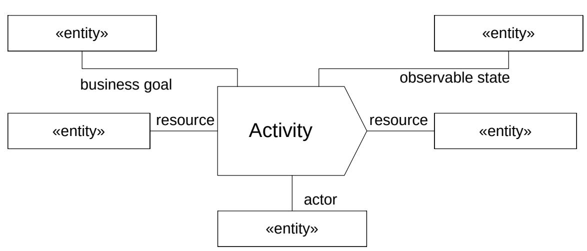

Figure 7. Common generic roles played by entities in the course of an activity.

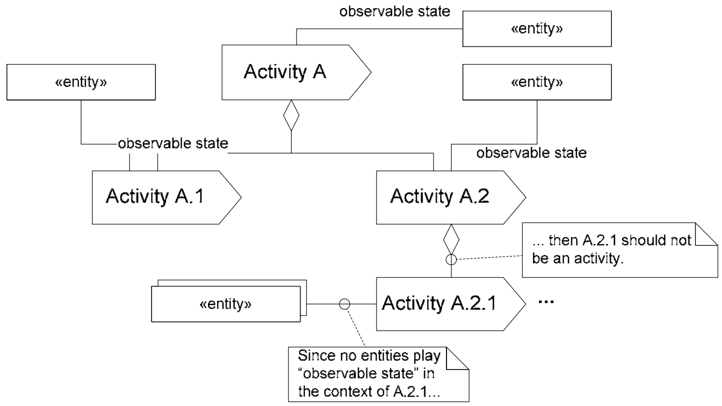

Figure 8. Activity decomposition methodology according to observable states.

Finally, and from a methodological viewpoint, activities must relate to at least one entity playing the role of observable state. An observable state models a state of affairs that is of interest to a stakeholder in the context of the enterprise architecture. It can be seen as an indicator that results from performing the activity. If there is no observable state related to a collaboration, this means the collaboration should not be modeled as an activity. This criterion can be used as a stop condition when deciding whether to decompose an activity any further. If the decomposition results in at least one activity that produces no observable state, then the decomposition should either be deemed invalid or else be rearranged so that every decomposed activity produces at least one observable state (v. Figure 8). It is noteworthy that the set of observable states depends on the purpose of the enterprise architecture. For instance, the set of observable states in an architecture that will be used to identify system requirements will probably be much more detailed than the set used to describe the core activities of an organization from a strategic perspective. Likewise, observable states are completely detached from how activities are coordinated. Until now, we have only discussed activities and activity decomposition at a structural level and have not mentioned how to coordinate activities through the specification of activity flows or other mechanism. This will be the subject of next section.

4.2 Business Processes and Activity Coordination

Coordination means integrating or linking together different parts of a system to accomplish a collective set of tasks. In the case of activity coordination, it means describing how activities are linked together so that they define a **business process**. Several definitions of business process can be found in the literature, such as:

- The set of internal activities performed to serve a customer. The purpose of each business process is to offer each customer the right product or service, with a high degree of performance measures against cost, longevity, service and quality (Jacobson 1995).
- A set of coherent activities that creates a result with some value for an internal or external customer; it is a meaningful whole of value-adding activities (Verharen 1997).
- The manner in which work is organized, coordinated, and focused to produce a valuable product or service (Laudon 2000).
- A collection of activities that takes one or more kinds of inputs and creates an output that is of value to the customer (Hammer 2001).

A common factor in these definitions is that a **business process** is a coordinated set of activities that is able to add value to the customer and to achieve business goals. This definition means that only the coordinated activities that fulfill these requirements can be classified as a business process. In this sense, classification of coordinated activities as a business process is an assessment that can be made *a posteriori*.

Coordination may mean either orchestration or choreography. Orchestration occurs when activities are coordinated by a centralized element that holds the coordination script. Choreography corresponds to autonomous coordination in the sense every activity decides its own actions according to a set of rules shared by every participant. In process modeling, orchestration and choreography are only relevant when the deciding application and technological architectures. While describing the business architecture, it is possible to describe the activity coordination in different ways, such as using explicit control or data flow between activities or using events or pre-conditions.

4.2.1 UML Representation

Coordination is represented using any of UML's dynamic diagrams. Figure 9 shows a UML activity diagram depicting a process by making explicit the control flow between the activities and the data flow between the data objects. The structural part of the role models is depicted in Figure 10. For instance, `Beat Eggs` is an activity where a `Person` acts as a `Beater Operator` while using a `Beater` and a `Vessel` to change the state of a `Resource` to `beaten`. The activity diagram shows the actual entities playing these roles while the role model describes the role relationships.

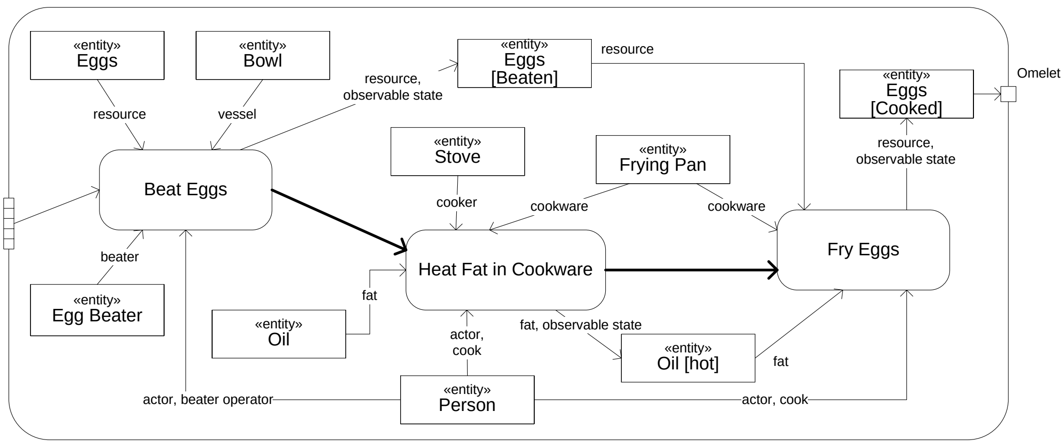

Figure 9. Activity diagram representing the "frying an omelet" process.

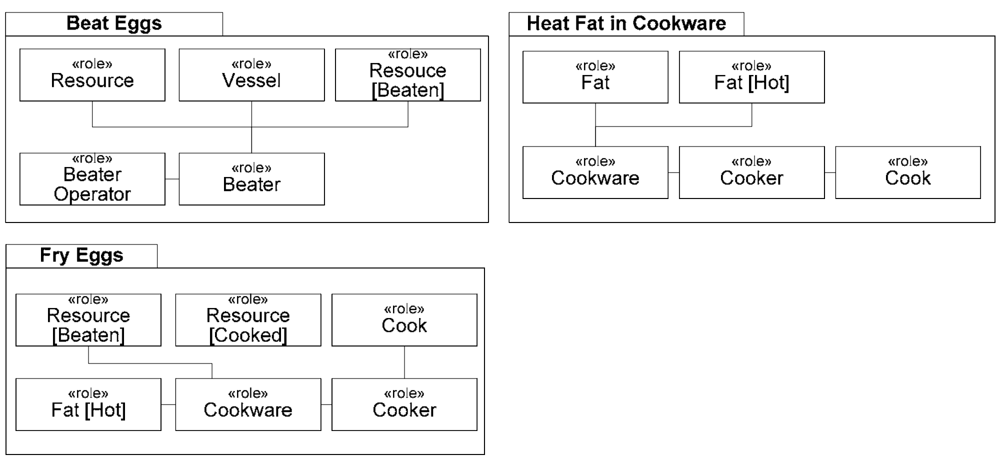

Figure 10. Role models depicting the relationships between the roles.

4.3 Role Types

Business entities are able to play a number of different roles during its lifetime. The basic **roles** we require to describe the enterprise architecture as earlier described (v. Figure 2 in page 9) correspond to the business nouns considered in the organizational, business information, application and technological architecture views. This subsection describes each of these roles.

4.3.1 Organizational Unit

An organizational unit includes information about the organizational units that make up an organization, the human resources that belong to those organizational units, as well as the structure and relationships that connect them all together (OMG 1998).

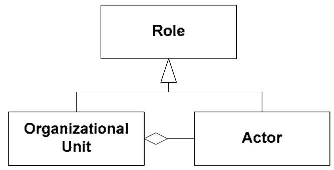

Figure 11. Organizational Unit.

Another important concept is that of chain of command, which refers to an interconnected and unbroken set of reporting relationships extending from top of the organization to the bottom. Each level in the structure from the bottom-up is accountable to a superior (Hampton 1986). The chain of command is modeled relating actors with a «supervisor» and «supervised by» association.

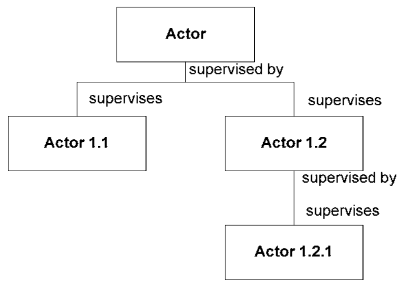

Figure 12. Chain of command.

4.3.2 Business Goal

A business goal represents a measurable state that the organization intends to achieve. Goals are achieved by the entities involved in performing activities.

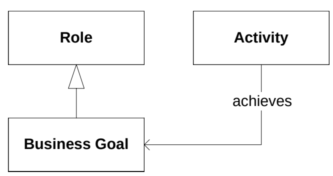

Figure 13. Business Goal.

4.3.3 Resource

A resource is the role of an entity that models capacity to be used and produced by business processes. The capacity may be consumed, incorporated, monopolized, or accessed (Taylor 1995).

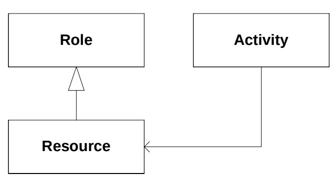

Figure 14, Resource.

4.3.4 Observable State

An observable state models a state of affairs that is of interest to a stakeholder in the context of the enterprise architecture. Observable states can guide the task of functional decomposing an activity as discussed earlier in section 4.1.3.

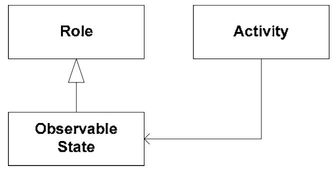

Figure 15. Observable State.

4.3.5 Actor

An **actor** is an animate entity capable of exhibiting active behavior. Actors model people, computer systems, mechanical tools or any other devices used to perform the operations required by an **activity**.

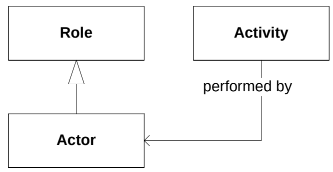

Figure 16. Actor.

Since entities only collaborate through roles, classifying an **entity** as an actor depends on the roles the entity is able to play, i.e., on the type of collaborations it participates in. This means that some entities may be potential actors but in a specific organizational case, they are just inanimate entities. Moreover, the status of actor is transient and context dependent, meaning that the same entity could be an actor in the context of a process and a resource in the context of other. For example, in a social security benefits process, the entity that represent a pensioner, although modeling a person, would not be modeled as an actor since the roles this entity plays are not related to executing any activity. However, if the same person works for the social security and is involved in playing some operational roles in that same process, then she would be regarded as an actor in that context. This means that the criteria for deciding whether an entity is an actor are the roles it is able to play.

Actors are able to perform the set of services required to play a role. This means an actor is then responsible for providing such services. In case of people, these services are correlated to the skills, capabilities and other attributes pertaining to the person that are relevant to assign her to a role in the scope of an activity. In case of computerized systems or machines, the services represent the operations and functions that these devices put into play during the role assignment. This topic will be further discussed in section 4.4.

4.3.6 Mission, Vision, Strategy, Organizational Goal

The **mission** states purpose of existence of the enterprise. Having the mission as a motto, the **vision** is the way to transform it in something possible to achieve in a near future. On its turn, **strategy** is a high-level business process that describes how to accomplish the vision. The goals that this strategic process achieves are called organizational goals. This is depicted in Figure 17.

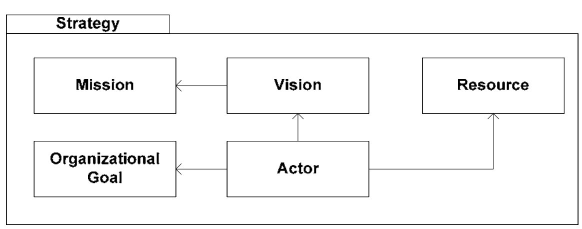

Figure 17. The strategy role model.

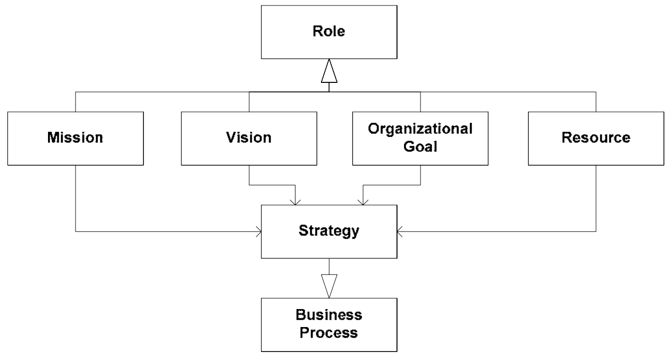

Figure 18. The strategy business process.

As seen earlier in sections 4.1.3 and 4.2, a business process is performed by actors that act upon resources, thus achieving goals. Figure 18 shows how these roles relate in the context of the strategy process. This process is a means of coordinating the enterprise. It explicitly defines who the actor responsible for conducting the strategic process is. This actor uses the vision and a set of resources to produce organizational goals.

4.4 Roles and Entities

The **business architecture** defines the **business processes** of the organization. To do so, it makes use of the repository of activities and entities specified in the organization's **information architecture**. Activities describe how the entities collaborate through roles in order to produce a specific outcome. This outcome results from actors performing operations or **services** over the other collaborating entities.

This means we can model this interaction as a marketplace where activities are the demand and actors (i.e. active entities able to express behavior) are the offer. The **activity** describes what roles are required for its operation. Entities are able to play these roles, thus providing the required service.

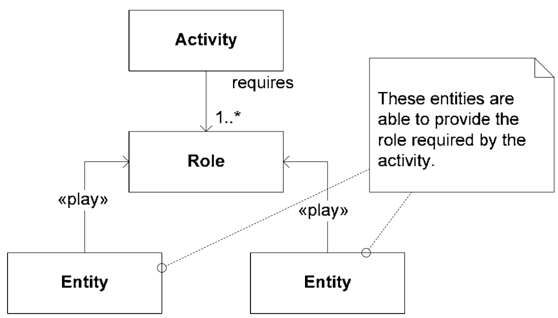

Figure 19. Roles provided by entities and required by an activity.

The scheduling process results in binding a set of entities to a specific instance of an activity. To do so, it applies scheduling criteria to select the entities able to perform the activity. In case of human actors, this can be accomplished by selecting the available actors that are able to provide the required roles to perform the activity. In this case, a role must be translated to the skills of the human actor. However, in this paper we will not focus on the task provided by people but on the services provided by information systems and technology as described in the application and technological architectures.

We will next define the concepts of service, business service and IS and IT block to conclude the definition of the concepts within the enterprise architecture.

4.4.1 Business Services and IT Services

A **service** is an aggregation of a set of operations provided by an architectural block. The operations provided by a service are implemented in other architectural blocks, such as IS Block and IT Block. It is represented in UML as an Interface element, restricting it to the interfaces provided by the IS Block and the IT Block.

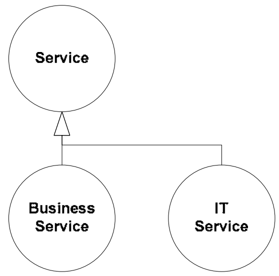

Figure 20. Services.

A business service is a collection of operations provided by **IS Blocks** that support business processes. This is the key concept in Service Oriented Architectures; the Business Service aggregates the set of operations used by business processes and, thus, provides the interface between business and information systems.

The IT Service is an interface provided by an **IT Block** to other IT Blocks. This is the lower level concept of service, which includes software services (implemented for example in a web service), the technological services provided by application platforms (e.g., operation system services, security services, data services, integration services) and the infrastructure services (e.g., the services provided by the network).

4.4.2 Application Block and IT Block

An **application block** or **IS block** denotes an application that aggregates an organized collection of mechanisms and operations that are able to manipulate organization data. It is represented as a UML Component. An IS block provides business services to roles or to other IS Blocks.

The **IT Block** represent the infrastructure, platform, technological or software component that realizes an IS Block. It is a UML class.

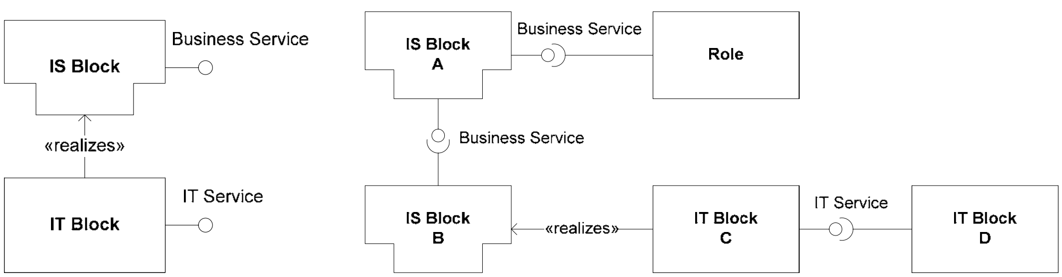

Figure 21. Definition of IS and IT blocks (left) and example showing a IS Block A providing a business service to a role and to another IS Block B that is realized by an IT Block (right)

5 Assessing the Alignment

This section outlines the rules to assess the **alignment** between the architecture views and its elements. Figure 22 shows the dependencies between the views. The next sections present an example of such rules between business and information architecture, between business and application architecture and between information and application architecture. The final subsection summarizes integrity rules that deal with the relationships between specific architecture components.

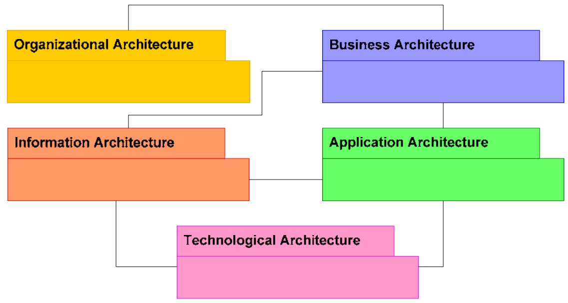

Figure 22. Dependencies between the architecture views while assessing the alignment.

5.1 Business and Information Architectures

Information and business architectures are aligned when business people have the information they need to run the business. This means accurate, on time, information, with the right level of detail. The impact of misalignments between these architectures is mostly the inability of getting the information relevant for the business. For instance, a manager asks for a report where sales figures need to be decomposed by service type. Assuming the report has either actual or foreseen business relevance, the ability to produce such report is an evidence of the alignment between information and business architectures. To produce the report the organization must possess the adequate data and applications. Common rules to assess the alignment between the business and information architectures are:

- Business activities create, update or delete at least one information entity.
- The (attributes of) entities are read at least by one business activity.
- Entities must be classified and named only within the information architecture.
- Entities have an identifier that is clearly understood by business people.
- Entities must have a means of being communicated to the appropriate audience using enterprise-standard applications and tools.
- Entities must be owned by someone responsible for its coherency, accuracy and relevance for the business.

5.2 Business and Application Architectures

Misalignments often drive people to engage in other tasks than those required by the activities being performed. The need for extra tasks is an evidence of misalignment and is measured in extra time requited by business people to conduct and fulfill the business. Common rules required to assess the alignment between the business and application architectures are:

- Business data is introduced only once.
- Business activities related to information processing should be automated as business services.
- Each business service must support at least one activity.
- For each application service, if we consider its removal, there would be at least one business activity that would no longer be supported. This means that there are no redundant business services. This is a drive too keep applications as simple as possible.
- Information required for critical processes should be supported by services with high availability.

5.3 Information and Application Architectures

Misalignments between information and application architectures result in extra time of IT department in ensure applications have the right data for processing. This means that IT people either spend time in keeping **information entity** replicas coherent, or spend time in integration projects that serve no other purpose than assuring information replicas coherency. In both cases, the extra time and money are an evidence of misalignment between information and application architectures.

There are others evidences of misalignment between information and application architectures:

- The need to keep replicas of the same **information entity** and to keep such replicas coherent because entity ownership is not specified and entities are managed by multiple independent application services.
- The need to assure the consistency of information entities used in transactions that cross application boundaries.
- Retrieving information from unrelated services and applications to produce a view on the organization's business information that has no clear owner.
- Transforming entities at business or application level when data is changed within technological applications.

To mitigate the above issues, views of information and application architectures must be consistently updated. The fundamental rules required to assess the alignment between the information and application architectures are:

- An **information entity** is managed by a single application. Business services that update the same **information entity** must be supported by the same application. The business service that manage an **information entity** must provide the means to share and distribute it across the organization using agreed-on protocols and formats as defined by the business.
- Exporting and distributing information entities across organization applications should be made imposing the minimum dependencies between application as possible. Normally, the usage of a common data store is preferable to a point-to-point application integration. Applications managing a given **information entity** should export its contents to the data store when its contents have changed. Applications requiring a given information entity should inquire the data store for up-to-date information.

5.4 Integrity Rules

The following table summarizes a set of integrity rules that apply between pairs of enterprise concepts. The list is far from being complete. The concept names are qualified and preceded by OA, BA, IA, AA, TA, standing for Organizational, Business, Information, Application and Technological Architecture, respectively. These integrity rules should be observed while creating and assessing the enterprise architecture model.

Concept Relationship		Integrity Rule
OA::Mission	OA::Vision	Every organization should have a mission defined. For that mission there should be a vision.
OA:;Vision	OA::Strategy	The vision should be accomplished by one or more strategies. A strategy is only defined for a single vision.
	OA::Organizational Goal	A vision should have one or more goals to achieve. A goal is only defined for one vision.
OA::Strategy	OA::Organizational Goal	A strategy should contribute to one or more goals. A goal can be supported by one or more strategies.
OA::Organizational Goal	BA::Business Goal	A goal can be decomposed in one or more business goals. A business goal is only related to one organizational goal.
BA::Business Goal	BA::Business Process	A business process can have one or more business goals to achieve. A business goal can be supported by one or more business process.
IA::Activity	OA::Strategy	A business process can have one or more strategies to achieve. A strategy can be supported by one or more business process
	OA::Organizational Unit	A business process can be associated to one or more organizational units. An organizational unit can handle one or more business process.
	IA::Resource	A business process can relate to multiple resources. An resource can be related to one or more business process.
	IA::Observable State	An activity must have at least one observable state associated to it, thus justifying its functional decomposition.
	IA::Actor	An activity must be owned and enacted by at least one actor. The actor is not necessarily the same.
	AA::Business Service	An activity must be supported by at least one business service.
IA::Actor	OA::Organizational Unit	An organizational unit comprises one or more actors. An actor reports only to a single organizational unit.
	AA::Business Service	An automated actor must provide one ore more business services.
AA:Business Service	AA::IS Block	A business service must be provided by at least one IS block.
TA::IT Block	TA::IT Service	An IT service must be provided by at least one IT block.
	AA::IS Block	Each IS block must be implemented in at least one IT Bbock.

6 Conclusions

Enterprise architecture consists of defining and understanding the different elements that shape an organization and how those elements are inter-related. In this chapter, we have proposed a set of concepts and their relationships to describe an organization with the purpose of understand and facilitate its evolution. These concepts are part of an enterprise architecture that is decomposed in five architectural views, each focusing on separate concerns within the enterprise.

Enterprise architecture defines the concepts that allow an organization to be described at multiple levels of detail allowing multiple dimensions of analysis. In architecture and civil engineering, for instance, the concepts and computer-aided tools already exist, allowing the design and analysis of a structure from different perspectives, ranging from and electrical wire details to its macro-structural properties, and continually assessing the coherence and alignment of such perspectives. Representing the enterprise architecture with the UML, which has already a wide tool support, is a step towards achieving a similar goal. Such concepts and tools can ultimately allow an organization to be always assessed and controlled so that alignment becomes the process of continuously guiding the enterprise resources to exploit opportunities and cope with environmental changes.

7 References

ANSA (1989). ANSA Reference Manual.: Cambridge.

BPMI (2004). BPMN 1.0. Retrieved December, 15, 2005, from http://www.bpmn.org/Documents/BPMN V1-May 3 2004.pdf.

DeBoever, L. (1997). Enterprise Architecture Boot Camp & Best Practices: A Workshop.

Egan, G. (Ed.) (1988). Change-Agent Skills A: Assessing and Designing Excellence. San Diego: University Associates.

FEAPMO (2003). Business reference model version 2.0. Retrieved December, 15 2005, from http://www.cio.gov/documents/secure/BRM_v2_Comment_Response.pdf.

Farooqi, K., Loggripo, L. , & Loggripo, J. (1995). The ISO Reference Model for Open Distributed Processing: An Introduction. Computer Networks and ISDN Networks, 27(8), 1215-1229.

Forrester, J. (1961). Industrial Dynamics. Waltham, MA: Pegasus Communications.

Frankel, D., Harmon, P. & Mukerji, J. (2003). The Zachman Framework and the OMG´s Model Driven Architecture.: Business Process Trends.

Gilchrist , A., Mahon, B. (2003). Information Architecture: Designing Information Environments for Purpose. London: Facet Publishing.

Frankel, D., Harmon, P. & Mukerji, J. (2003). The Zachman Framework and the OMG´s Model Driven Architecture.: Business Process Trends.

Gilchrist , A., Mahon, B. (2003). Information Architecture: Designing Information Environments for Purpose. London: Facet Publishing.

Goedvolk, J., Bruin, H. & Rijsenbrij, D. (1999). Integrated Architectural Design of Business and Information Systems.

Hammer, M., & Champy, J (2001). Reengineering the Corporation: A Manifesto for Business Revolution. London: Nicholas Brealey Publishing.

Hampton, D. (1986). Management. Singapore: McGraw-Hill.

Herbert, A. (1994). An Overview of ANSA.: IEEE Network.

ICAM (1981). Architecture Part II-Volume IV - Function Modeling Manual (IDEF0).: Air Force Wright Aeronautical Laboratories.

IEEE (2000). IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.

ISO (1995). ISO/IEC 10746 ODP Reference Model.: International Standards Organization.

Inmon, W. (1999). Data Architecture - The Information Paradigm. Wellesley, Mass.: QED Technical Publishing Group.

International Enterprise Architecture Center (2005). Retrieved December, 15 2005, from http://www.ieac.org/about/index.htm

Kaplan, R, & Norton, D. (2004). Strategy Maps: Converting Intangible Assets into Tangible Outcomes. Boston, MA: Harvard Business School Press.

Labovitz , G. & Rosansky, V. (1997). Power of Alignment: How Great Companies Stay Centered and Accomplish Extraordinary Things. New York: John Wiley & Sons Inc.

Laudon, K. & Laudon, J. (2000). Management Information Systems. New Jersey: Prentice Hall.

Luftman , J. (1996). Competing in the Information Age: Strategic Alignment in Practice. New York: Oxford University Press, Inc.

Meta Group (2005). Retrieved December, 15 2005, from http://www.metagroup.com/products/insights/eas_1_sc.htm

O'Rourke, C., Fishman, N., & Selkow. W. (2003). Enterprise Architecture Using the Zachman Framework. Boston, MA: Course Technology, Inc.

OMG (2004). Unified Modeling Language: Superstructure, version 2.0.: Object Management Group. Retrieved December, 15 2005, from http://www.omg.org/cgi-bin/doc?ptc/2004-10-02

Open Group (2003). The Open Group Architectural Framework (TOGAF) - Version 8.1: The Open Group.

Papp, R. (2001). Strategic Information Technology: Opportunities for Competitive Advantage. Hershey, PA: Idea Group Publishing.

Porter, M. (1985). Competitive Advantage. New York: The Free Press.

Schekkerman, J. (2004). How to Survive in the Jungle of Enterprise Architecture Framework: Creating or Choosing an Enterprise Architecture Framework. Victoria, Canada: Trafford Publishing.

Schurmann, G. (1995). The Evolution from Open Systems Interconnection (OSI) to Open Distributed Processing (ODP). Computer Standards and Interfaces, 17, 107-113. Scott, R. (1997). Organizations: Rational, Natural, and Open Systems. New Jersey: Prentice Hall.

Senge, P. (1990). The Fifth Discipline:The Art & Practice of The Learning Organization. New York: Currency Doubleday.

Sowa, J. & Zachman, J. (1992). Extending and formalizing the framework for information systems architecture. IBM Systems Journal, 31, 590-616.

Spewak , S. & Hill, S. (1992). Enterprise Architecture Planning: Developing a Blueprint for Data, Applications and Technology. New Jersey: Wiley-QED Publication.

Taylor, D. (1995). Business Engineering with Object Technology. New York: John Wiley & Sons.

W3C (2002). Web Services: World Wide Web Consortium. Retrieved December, 15 2005, from http://www.w3.org/2002/ws/

Zachman, J. (1987). A Framework for Information Systems Architecture. IBM Systems Journal, 26(3), 276-292.