Analysis and Improvement of the Alignment between Business and Information System for Telecom Services

Jacques Simonin[1], Emmanuel Bertin[2], Yves Le Traon[3], Jean-Marc Jézéquel[4], Noël Crespi[5]

[1] Institut Télécom, Télécom Bretagne, Lab-STICC UMR CNRS 3192, UEB, Technopole Brest-Iroise, 29238 Brest, France

[2] Orange Labs, 42 rue des Coutures, 14066 Caen, France

[3] Université du Luxembourg, 6, rue Richard Coudenhove-Kalergi, 1359 Luxembourg, Luxembourg

[4] INRIA and Rennes University, Campus Universitaire de Beaulieu, 35042 Rennes, France

[5] Institut Télécom, Télécom SudParis, 9 Rue Charles Fourier, 91011 Evry, France

jacques.simonin@telecom-bretagne.eu, emmanuel.bertin@orange-ftgroup.com, yves.letraon@uni.lu, jezequel@irisa.fr, noel.crespi@it-sudparis.eu

***Abstract* - The main aim of Enterprise Architecture (EA) is to master the development and the evolutions of Information Systems (IS). The EA process consists of designing the IS target architecture from several views, according to the company strategy. The business view represents the target organization of the particular company. The functional view focuses on the target functional architecture of that company's IS. In this paper, we propose a new formal solution to analyze and to improve the consistency between the target functional view and the target business view of telecom services. This solution is based on the definition of a strategic alignment of the target functional view with the target business view. Alignment is validated with a real case study implemented and deployed at Orange--France Telecom on their messaging service. An alignment measure completing this analysis provides an estimation of the gap between a target functional view and a target business view.**

Keywords - information system, enterprise architecture, business view, functional view, alignment, measure, function typing.

I. Introduction

A. *Context and Motivation*

Enterprise Architecture (EA) aims to simplify the Information Systems (IS) of a company, and to reduce the cost of IS development and evolution. This simplification of IS should be driven by the strategy of the company. For telecommunication service providers, the strategy mainly consists of providing new services (designed by marketing to fit user's needs) that rely as much as possible on existing infrastructures [1].

EA frameworks (such as that of Zachman [2]) define various points of view (business, system, technology, etc.) in order to take into account all the aspects of these strategic objectives. This paper relies on the four classic EA views (as defined in [3]): the business view defining 'why', the functional view defining 'what', the technical view defining 'with what', and the applicative view defining 'how'. The relationships between the functional view, the technical view, and the applicative view are deduced from the iterative development cycle, which relies on the Unified Process (UP) [4]. The business view should be an input for both the functional and the technical views.

This paper is focused on the strategic alignment of a company's functional view with its business view. A good alignment highlights the consistency within the organization of the company and its IS [5] and indicates that the business strategy and the IS strategy are synchronized.

The target business architecture and the target functional architecture must both fulfil the strategy of a company. However, the strategy guiding the business organization (business view) and that of the IS functions (functional view) are different and are not defined by the same people. Business and functional views evolve independently, following the business and the marketing evolutions. Moreover, the evolution of a company's organization is seldom synchronous with the evolution of its IS.

We therefore propose an innovative formal approach that allows a functional Enterprise Architecture to analyze the misalignment between the target functional architecture and the target business architecture. We also propose a metric for this alignment. The objective is to define an assessment in order to improve the alignment between the functional view and the business view.

B. *Outline*

This paper is organized as follows. Section II depicts the state of the art and Section III introduces the EA process and defines what is meant by the alignment of the functional view with the business view. Section IV describes the alignment measure of the functional view with that of the business view. The example in Section III and in Section IV is based on an Orange messaging service. Section V describes a solution to improve the alignment measure of the functional view in comparison with the business view, by typing functions according to business processes. Section VI depicts the first experimentation of the alignment measure at the Orange laboratories.

II. Related Work

In practice, most telecommunication companies directly map their business view with their IS applicative, aiming for an alignment between their core business and their IS. A company may decide that a given email platform (for example, Microsoft Exchange server) will manage the entire service business process of communicating by email. For telecom services, this method has one major shortcoming: it implies a tight coupling between the business view and the applicative view. The business analysis is then distorted by applicative considerations. For example, the messaging business may evolve so that it is driven by the evolutions of the selected platform, and no longer by the company strategy. The specification of a target business architecture that differs from the current applicative view may become virtually impossible.

In the literature, the alignment problem involving EA is mainly considered to be between the business view of a company and its IS [6]. Alignment may also be considered between the business view of a company and its objectives, as in the Business Motivation Model [7], or between an analysis model and a design model of the functional view of a telecom service [8]. The parameters related to the quality of the alignment are specific for each company [9]. For this type of alignment, heuristics may be defined to provide warnings in case of misalignment [5]. A measurement method would allow the evaluation of architectures in business terms (cost, benefit, risk). However, the measures relevant to business terms do not take IS concepts into account.

One significant contribution of this paper is a method to take into account the effects of a company's strategy on its functional view. The alignment perspective between the business and functional views is shown in Figure 1.

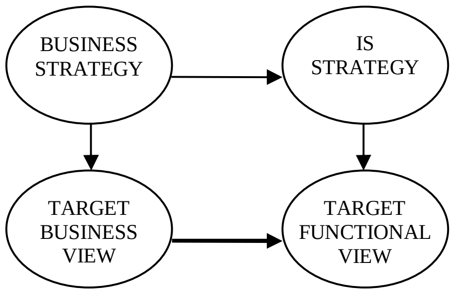

Figure 1. Alignment perspective between the target business view deduced from a business strategy and the IS target functional view deduced from an IS strategy [6].

The functional view choice is justified because an IS functional view is easier to align with than a business view. Functional, meaning comprehensible and practical, is indeed helpful to match business description. Moreover, the IS applicative view is largely the facet that implements the functional view. Alignment of the applicative view with the business view is thus dependent on the alignment of the functional view with the business view.

Moreover, many object-oriented measures exist outside the scope of EA. To estimate model alignment, coupling measures [10] are the most appropriate means, since relationships between models are the main characteristics of the solution proposed in this paper.

III. Alignment of the Functional View with the Business View Definition

We focus here on the alignment--or the misalignment of the functional view with the business view. As demonstrated in the previous section, this topic has not been studied in much detail.

A. *Alignment Definition Home Domain Analogy*

Let us introduce the alignment definition with a "home domain" analogy. In a "home domain", the house customer is responsible for the "home processes" of a home domain business view. One is the *Have a meal at home* process and its activities *Cooking at home* and *Eating at home*. The house occupant defines that *Eating at home* comes after *Cooking at home*. "Home IS" is designed by house architects, who define models of homes based on the home domain functional view. House architects design models of homes from "home functions". Three components composed of "home functions" and their dependencies support the *Have a meal at home* process. The so-restricted "Home IS" is represented by a home model in a Unified Modelling Language (UML) [11] component diagram given in Figure 2. Figure 2 illustrates three dependencies between "home functions": *Have breakfast* on *Cook at home*, *Have diner* on *Cook at home* and *Have lunch* on *Cook at home*.

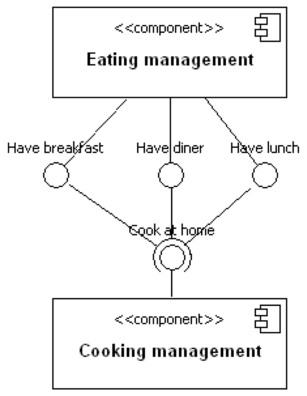

Figure 2. "Home domain" analogy with "Home IS", which owns two functional components: *Eating management* and *Cooking management*.

Alignment of the functional view with the business view is easily illustrated with this "home domain":

- *Have breakfast*, *Have diner*, *Have lunch* are "home functions" aligned with an *Eating at home* "home activity";
- *Cook at home* and *Get food* are "home functions" aligned with a *Cooking at home* "home activity",

- *Have breakfast* on *Cook at home*, *Have diner* on *Cook at home* and *Have lunch* on *Cook at home* are dependencies between “home functions” aligned with *Cook at home*, and come before any *Eating at home* activity succession.

With this home analogy, we can highlight that there is a reversal between the dependency and the succession relationships. The eating-on-cooking dependency means that cooking must come before eating.

B. EA and Target Architecture

The EA process has two main goals:

- to depict existing IT architecture, in order to describe what functions are implemented on each IT system, how each IT system is deployed and which process is supported by each IT system; and
- to design several target architecture views to separate the concerns of the various stakeholders in the enterprise.

Even if the company strategy is constant during the design of all the target architectures of these views, the required skills are different: on one side, the company’s core business experts elaborate the business view; on the other side, enterprising functional architects design the functional view. This independency is particularly significant for the evolution of each view because their lifecycles are different. A complete synchronization of a company’s organization evolutions and IS evolutions is very difficult to achieve for a large company. This is especially true for telecom service operators whose markets and technologies are very dynamic.

A company usually elaborates its target business architecture following a process analysis, which provides for descriptions of the business processes that belong to the core business of a company. The business view has the **activity** as its main concept**,** which is a part of a business process and which is under the responsibility of an organizational role. Concepts are modelled with UML [11]. A UML activity diagram can be used to capture a procedure designed in the target business architecture. Within Orange--France Telecom, the usage of telecom services is specified with approximately 10 roles and several tens of activities.

For illustration, let us consider a messaging service, limited to the message receipt. When a new requirement appears in the telecom operator strategy, such as the need to protect children from inappropriate electronic messages, the access control must evolve. In this example, the operator chooses to implement its strategy by creating a new *Child protection provider* role. Furthermore, the *Messaging service provider* role will depend on the new *Child protection provide*r role in the new target organizational infrastructure. So, to achieve the messaging receipt activity, the *Messaging service provider* role needs the intervention of the *Child protection provider* role.

The procedure deduced from the messaging service process is therefore easily captured using an activity diagram such as the one in Figure 3.

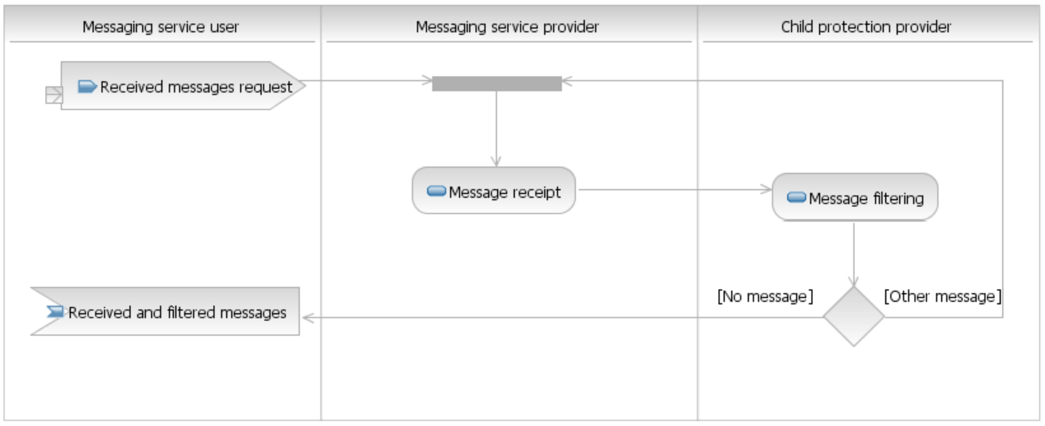
Messaging service user
Messaging service provider
Child protection provider
Received messages request
Message receipt
Message filtering
[No message]
[Other message]
Received and filtered messages

Figure 3. Sample activity diagram of the messaging service..

The IS target functional architecture contains functional elements implemented by IS systems. Functional architectures design the target functional architecture according to the company strategy. The main concept of the functional view meta-model holds that the **function** defines the functional component. Functional view concepts such as "Functional component" and "Dependency between functional components" are also close to UML concepts. The target functional view may be represented by a component diagram. The target functional view of our messaging illustration is represented by the component diagram in Figure 4.

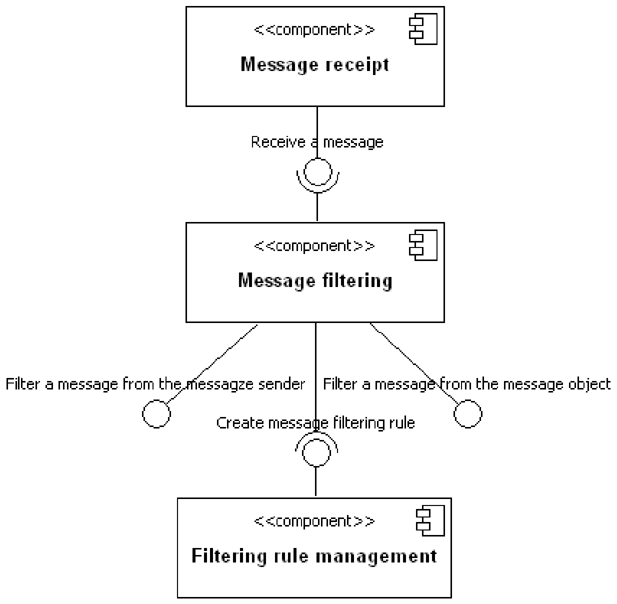
<<component>>
Message receipt
Receive a message
<<component>>
Message filtering
Filter a message from the messagze sender
Filter a message from the message object
Create message filtering rule
<<component>>
Filtering rule management

Figure 4. Sample target functional architecture of the messaging service.

C. *Alignment of the Functional View with the Business View Definition*

Alignment criteria are required to define the alignment between models. Our innovative criterion is based on the associations between concepts of the business view and of the functional view. Enterprise Architecture chooses business view concepts consistent with functional view concepts, while taking the alignment definition into account. The consistency between these concepts is known as the alignment value between a business model and a functional model.

We have considered two relevant types of possible associations:

- associations between business data manipulated by business activities and functional data manipulated by functions; and
- associations between business activities and functions.

An approach based on the first type (business data and functional data) can be viewed as static because it does not take into account the evolution of the states of data.

We have therefore chosen an innovative approach by considering the second type (business activities and functions). We have qualified this approach as dynamic, because it relies on the UML dynamic diagrams (activity diagram, sequence diagram) that show the live comportment of a system.

The idea of a dynamic approach (as opposed to an approach based on data) is to base the alignment on service usage scenarios instead of basing it on data models. For development methods in relation to the entity relationship model [12], the methodological complexity is a consequence of the simultaneous modelling of data and treatments. To resolve this complexity, our approach is based on the dynamic perspective because it allows functional reusability, a useful improvement. This reusability involves service components called enablers, as defined by the OMA (Open Mobile Alliance) [13]. Moreover, the alignment between business data and functional data can be deduced from the alignment between business activities and functions, as business data are produced by business activities and functional data by functions.

The business view, as illustrated in Figure 3, instantiates dynamic concepts. A procedure is indeed described by an activity sequence instead of a business data model. Concerning the functional view, the design of an interaction sequence carrying out a telecom service usage scenario does precede the data modelling. This chaining is feasible because each data is produced or used by a function during a scenario. With this dynamic approach, a dependency between functional components corresponds to an interaction between two functional component instances. The equivalence between an interaction sequence and a telecom service usage scenario denotes the dynamic aspect of this approach (see Figure 5).

A "request" type dependency of the functional view is therefore an information request. A functional dependency has a "resource" type if it represents an answer to an information request.

The association completing the alignment criterion is between a succession relationship of two business activities and a dependency between functions. We define the following links:

- Succession relationships are between two business activities if the end of one precedes the beginning of the other in a UML activity diagram capturing a business process (for example, in Figure 3, the succession relationship is the one from the business activity Message receipt to the business activity Message filtering);
- Dependency between two functions occurs
    - o if they are associated to two interactions between functional components, which have either the "request" type or the "resource" type,
    - o and if the end of one of these interactions precedes the beginning of the other interaction in a UML sequence diagram

  (an example in Figure 5 is the dependency between the function *Filter a message from the message sender* on the function *Create message filtering rule*).

The alignment of the functional view with the business view can thus be defined from these alignment criteria:

- a function is aligned with the business view
    - o if the function has a common meaning with at least one activity of the business view,
    - o and if each business activity aligned with the function has at least one succession relationship with another business activity aligned with the function ; and

- a dependency between two functions F1 and F2, such as F1 depends on F2, is aligned with the business view
    - if there is at least one business activity A1 aligned with F1,
    - if there is at least one business activity A2 aligned with F2, and
    - if A1 follows A2 in an activity diagram (succession relationship).

The alignment definitions are illustrated in Figure 3 for the business activity and the activity succession relationship concepts, in Figure 4 for the function concept, and in Figure 5 for the function dependency concept:

- *Receive a message* function is aligned with
    - *Message receipt* business activity;
- *Filter a message from the message sender* and *Filter a message from the message object* functions are aligned with
    - *Message filtering* business activity; and
- the dependency relationship from *Filter a message from the message sender* function on *Receive a message* function is aligned with
    - the succession relationship from the *Message receipt* business activity to the *Message filtering* business activity.

In Figure 5, the *Create message filtering rule* function and the dependency relationship of the *Filter a message from the message sender* function on the *Create message filtering rule* function are not aligned with the business view. No business activity in Figure 3 has a meaning in common with *Create message filtering rule* function.

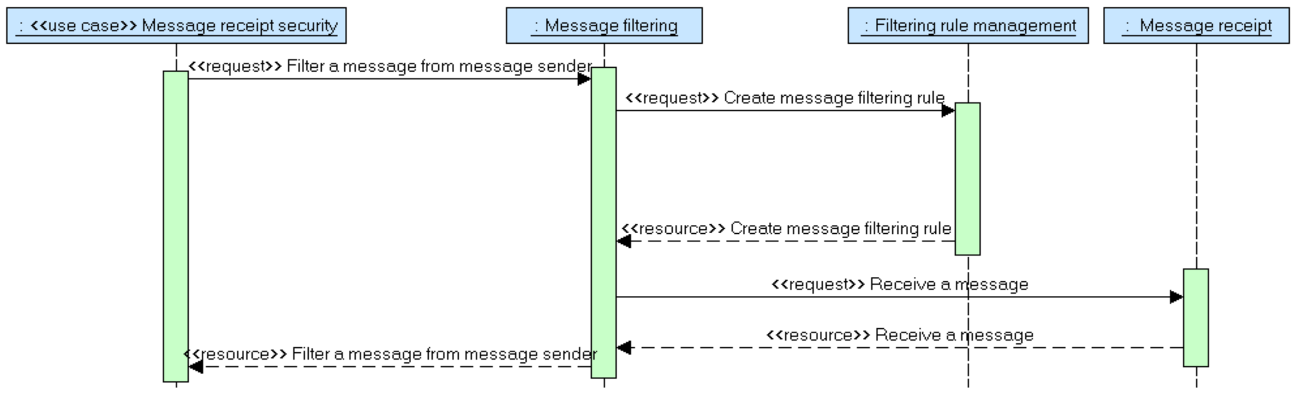
: <<use case>> Message receipt security
: Message filtering
: Filtering rule management
: Message receipt
<<request>> Filter a message from message sender
<<request>> Create message filtering rule
<<resource>> Create message filtering rule
<<request>> Receive a message
<<resource>> Receive a message
<<resource>> Filter a message from message sender

Figure 5. Functional view sequence of messaging service example.

IV. ALIGNMENT MEASURE OF THE FUNCTIONAL VIEW WITH THE BUSINESS VIEW

Axiomatization allows the intuitive properties of the alignment of a functional view to be specified in comparison to a business view description [14]. We propose an alignment measure according to these axioms.

A. Alignment Measure Home Domain Analogy

Alignment measure could be illustrated with a “home domain” analogy. We assume in this section that “home function” *Get food* is offered by the *Cooking management* component. This “home function” is added to our “Home IS” designed in Section III. Figure 6 represents our new “Home IS” with a UML component diagram.

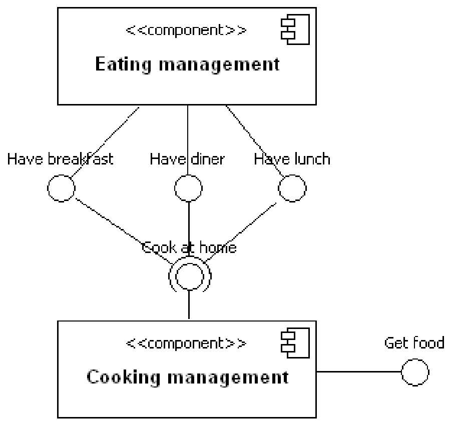

Figure 6. New “Home IS”, which owns a new “home function”: *Get food*.

This new “home function” *Get food* is not aligned with the “home activity” of *Eating at home* process because no activity of “Home processes” restricted to *Have a meal at home* shares a sense with food management, which is the closest repository management activity. We can say that the alignment of the new “Home IS” with “Home process” is worse than the alignment of the “Home IS” designed in Section III with “Home process”.

B. *Alignment of the Functional View with the Business View Axiomatization*

An axiom is an expected and understandable property of an alignment measurement that has also a meaning in the mathematical model. The following BFA axioms define this intuitive behaviour. Axioms are parameterized by functional view concepts affected by the alignment.

BFA1 – Function addition. The alignment resulting from the addition of a function in the functional view is either:

- worse than or identical to the previous alignment if the function has no business meaning in common with at least one activity of the business view,
- better than the previous alignment if the function has a business meaning in common with at least one activity of the business view.

BFA2 – Function dependency addition. The alignment resulting from the addition of a dependency between functions in the functional view is either:

- worse than or identical to the previous alignment if there is no business activity time succession of the business view that is aligned with the function dependency,
- better than the previous alignment if there is at least one business activity time succession of the business view that is aligned with the function dependency.

BFA3 – Function deletion. The alignment resulting from the deletion of a function in the functional view is either:

- worse than the previous alignment if the function shares a common business meaning with at least one activity of the business view,
- better than or identical to the previous alignment if the function has no common business meaning with even one activity of the business view.

BFA4 – Function dependency deletion. The alignment resulting from the deletion of a dependency between functions in the functional view is either:

- worse than the previous alignment if there is at least one business activity time succession of the business view that is aligned with the function dependency,
- better than or identical to the previous alignment if there is no business activity time succession of the business view aligned with the function dependency.

C. *Alignment of the Functional View with the Business View Measure*

An alignment measure depends on the alignment concepts defined in Section III. The number of relationships captured in a diagram is a well-known parameter for data model estimation [15]. The dependencies from the target functional view are the parameters of a proposed alignment measure, named **BFAM**, of the functional view with the business view. These dependencies, which are or are not aligned with the business view, enable estimation of the alignment of the functional view with the business view.

$$BFAM(FV) = \left(\frac{N\_f(FV) - N\_\{naf\}(FV)}{N\_f(FV)}\right) * \left(\frac{N\_d(FV) - N\_\{nad\}(FV)}{N\_d(FV)}\right) \quad (1)$$

where, for a functional view FV:

- N_f(FV) is the number of functions,
- N_{naf}(FV) is the number of functions that are not aligned with business activities,
- N_d(FV) is the number of dependencies between functions,
- N_{nad}(FV) is the number of dependencies between functions that are not aligned with a business activity time succession of the business view.

The BFAM value is a real number which value is between 0 (no function and no dependency relationship between functions are aligned with the business view) and 1 for a perfect alignment (all functions and all dependency relationships between functions are aligned with the business view).

BFAM measures complies with axioms **BFA1**, **BFA2**, **BFA3**, **BFA4** (see Section IV) of the alignment of the functional view with the business view.

The compliance for the **BFA$_1$** axiom is detailed as follows:

- Let F a function added to the functional view FV,

$$BFAM(FV \cup \{F\}) = \left(\frac{N\_f(FV \cup \{F\}) - N\_\{naf\}(FV \cup \{F\})}{N\_f(FV \cup \{F\})}\right) * \left(\frac{N\_d(FV \cup \{F\}) - N\_\{nad\}(FV \cup \{F\})}{N\_d(FV \cup \{F\})}\right) =$$

$$\left(\frac{N\_f(FV)+1}{N\_f(FV)+1} - \frac{N\_\{naf\}(FV) + N\_\{naf\}(\{F\})}{N\_f(FV)+1}\right) * \left(\frac{N\_d(FV) - N\_\{nad\}(FV)}{N\_d(FV)}\right)$$

because the number of dependency relationships between functions is unchanged. And then

$$\Rightarrow BFAM(FV \cup \{F\}) - BFAM(FV) =$$
$$\left(\begin{array}{l}\left(1-\frac{N\_\{naf\}(FV)+N\_\{naf\}(\{F\})}{N\_f(FV)+1}\right)- \\ \left(1-\frac{N\_\{naf\}(FV)}{N\_f(FV)}\right)\end{array}\right)*$$
$$\left(\frac{N\_d(FV)-N\_\{nad\}(FV)}{N\_d(FV)}\right)=$$
$$\left(\begin{array}{l}\frac{N\_\{naf\}(FV)}{N\_f(FV)}- \\ \frac{N\_\{naf\}(FV)+N\_\{naf\}(\{F\})}{N\_f(FV)+1}\end{array}\right)*$$
$$\left(\frac{N\_d(FV)-N\_\{nad\}(FV)}{N\_d(FV)}\right)$$

- If F has no common meaning with at least one business activity, then $N\_\{naf\}(\{F\})=1$
$$\Rightarrow BFAM(FV \cup \{F\}) - BFAM(FV) =$$
$$\left(\frac{N\_\{naf\}(FV)}{N\_f(FV)}-\frac{N\_\{naf\}(FV)+1}{N\_f(FV)+1}\right)*$$
$$\left(\frac{N\_d(FV)-N\_\{nad\}(FV)}{N\_d(FV)}\right)$$
and so,
$$\frac{N\_\{naf\}(FV)}{N\_f(FV)}-\frac{N\_\{naf\}(FV)+1}{N\_f(FV)+1}=$$
$$\frac{N\_\{naf\}(FV)-N\_f(FV)}{N\_f(FV)*(N\_f(FV)+1)}\leq 0$$
$$\Rightarrow BFAM(FV \cup \{F\}) \leq BFAM(FV)$$
because the number of misaligned functions is lower than or identical to the complete number of functions.
The BFAM measure is identical to the previous alignment measure when
$$N\_f(FV)=N\_\{naf\}(FV)$$
i.e., when no function of the functional view FV is aligned with the business view.
- If F has a common meaning with at least one activity of the business view, then $N\_\{naf\}(\{F\})=0$
$$\Rightarrow BFAM(FV \cup \{F\}) - BFAM(FV) =$$
$$\left(\frac{N\_\{naf\}(FV)}{N\_f(FV)}-\frac{N\_\{naf\}(FV)}{N\_f(FV)+1}\right)*$$
$$\left(\frac{N\_d(FV)-N\_\{nad\}(FV)}{N\_d(FV)}\right)$$
and so,
$$\frac{N\_\{naf\}(FV)}{N\_f(FV)}-\frac{N\_\{naf\}(FV)}{N\_f(FV)+1}=$$
$$\frac{N\_\{naf\}(FV)}{N\_f(FV)*(N\_f(FV)+1)}\geq 0$$
$$\Rightarrow BFAM(FV \cup \{F\}) \geq BFAM(FV)$$
The BFAM measure is identical to the previous alignment measure when $N\_\{naf\}(FV)=0$, i.e., when all functions of the functional view are aligned with the business view.

- The proof is similar for the **BFA$_2$**, **BFA$_3$** and **BFA$_4$** axioms.

The alignment measure **BFAM** may be the stop criterion of an iterative development process. A higher estimation of the alignment implies a better consistency between the target business and target functional views.

V. Improving the Alignment of the Functional View with the Business View

Aligning the functional view with the business view is one of the most complex activities of EA, since it consists of integrating in an abstract view the very concrete strategy of the enterprise. To help perform this alignment, we propose to type functions with respect to duration of its instance compared to business process of the enterprise.

A. Alignment Improvement Home Domain Analogy

A first alignment improvement of "Home IS", designed in Section IV with "Home processes" could be deduced from updating the "Home processes". For example, a new *Taking out food from the freezer* activity could be added to *Have a meal at home* "Home process". *Taking out food from the freezer* should come before *Cooking at home* because the second activity needs the result of the first. Alignment between the *Get food* "Home function" and the *Taking out food from the freezer* activity indeed improves the alignment of "Home IS" compared to "Home processes". A second improvement of this alignment takes into consideration the functional dependency of *Cook at home* on *Get food,* which could be aligned with an activity succession of the *Taking out food from the freezer* activity, which in turn comes before the *Cooking at home* activity. Function dependency aligned with this succession is represented inside a UML component diagram in Figure 7.

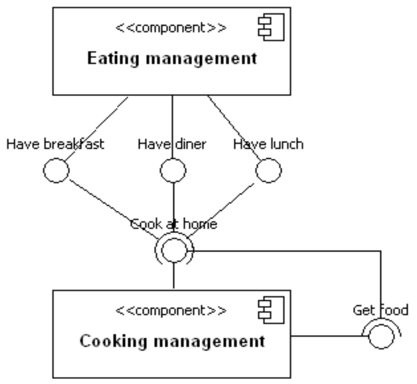

Figure 7. New "Home IS", which owns a new "home function dependency: *Cook at home* on *Get food*.

A new dependency of *Cook at home* on *Get food* allows an alignment with the activity succession (*Taking out food from the freezer* activity comes before *Cooking at home* activity).

Alignment improvement is the result of functional dependency design, which is consistent with activity succession. In the following section, we propose rules, named **BFRs**, that provide for alignment improvement from functional views in comparison with business views.

B. Typing of functions in relation to the Business View

The principle of typing functions comes from the architectural layer concept from the OSI model (Open Systems Interconnect) [16]. Typing each element allows to locate it on a specific layer. This principle is applied to the design of functions and of functions' dependencies.

A function owns types, which distinguishes it from the business activities aligned with it. The following types are consistent with a networking system [17]:

- *Command*, if this function is aligned with a business activity, then it has a life duration depending on the instances of business processes containing it (for example, *Give a telecommunication service order* function is aligned with a business activity, which has a life duration depending on the *Order capture* process (defined in [18])),
- *Data*, if this function is aligned with a business activity, then it has life duration that is independent of the instances of business processes containing it (for example, *Have access to telecommunication service catalogue* function is aligned with business activity, which has a life duration independent of the *Order capture* or *Billing* processes containing catalogue management activity).

Let F be the set of functions and T be the set of function types, then a type *t* of function *f* is the power set of T:

$$\boxed{\begin{array}{l} t : F \rightarrow P(T) \\ f \mapsto t; t \subset P(\{Command, Data\}) \end{array}} \quad (2)$$

So, a function can be of type {command, data} if it is aligned with two activities, such as life duration depending on the processes containing it and life duration independent of the processes containing it.

The functional architecture represented in Figure 4 shows four functions:

- *Create message filtering rule* function, which is of both *Command* and *Data* types. An alignment improvement would identify business activities that share a sense in common with this rule creation function. Two activities must be inserted in the messaging telecom service use process: *Filtering rule creation* and *Filtering rule consultation*. On the one hand, *Filtering rule creation* activity has a life duration, which depends on the use process, while *Filtering rule consultation* does not depend on it. We may indeed consult a filtering rule, while other messaging telecom service use process instances do not;
- *Filter a message from the message sender* and *Filter a message from the message object* functions own *Command* type. Their life duration depends on messaging telecom service process instances; and
- *Receive a message* function owns *Command* type for the same reason.

From definition (2), the typing can be expressed as:

$$\begin{array}{l} t(\text{Create message filtering rule}) = \{Command, Data\} \\ t(\text{Filter a message from the message sender}) = \{Command\} \\ t(\text{Filter a message from the message object}) = \{Command\} \\ t(\text{Receive a message}) = \{Command\} \end{array}$$

In our messaging telecom service, the *Create message filtering rule* function has both *Command* and *Data* types. To improve alignment with the business view, function typing should associate each function to only one type (*Command* or *Data*). These functions, *Create message filtering rule* and *Consult message filtering rule*, are illustrated in a UML sequence diagram in Figure 8.

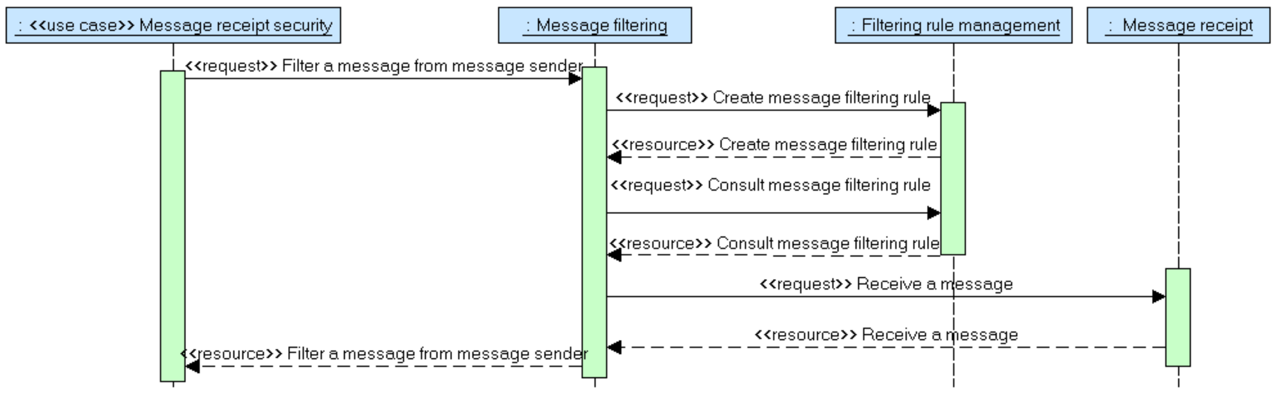
: <<use case>> Message receipt security
: Message filtering
: Filtering rule management
: Message receipt
<<request>> Filter a message from message sender
<<request>> Create message filtering rule
<<resource>> Create message filtering rule
<<request>> Consult message filtering rule
<<resource>> Consult message filtering rule
<<request>> Receive a message
<<resource>> Receive a message
<<resource>> Filter a message from message sender

Figure 8. Functional view sequence of messaging service with a multi-typed function splitting example.

We propose then the first two rules, **BFR1 and BFR2,** for improved alignment with the business view:

BFR1 – Multi-typed function splitting. Multi-type functions must be split in sub-functions so that each sub-function is only of one type.

BFR1 improves alignment because activities are under the responsibility of specific roles. Activities with life durations dependent on processes are indeed under the responsibility of the enterprise's front-office, and activities with life durations independent of processes are mainly under the responsibility of an enterprise's back-office.

On one hand, the *Filtering rule creation* activity is under the responsibility of the *Child protection provider*, which is a telecom service back-office role. On the other hand, *Filtering rule consultation*, *Filter a message from the message sender* and *Filter a message from the message object* are activities under the responsibility of the *Messaging service provider* role, which is a telecom service front-office role.

Type pattern [19] may complete function typing, defining relationships between mono-typed functions. This pattern must be consistent with type definitions in relation to the business view. Pattern motivation enables alignment improvement of the functional view with the business view. A pattern may, for example, be associated with types as defined in (2). The UML class diagram in Figure 9 illustrates that a *Command* function depends on the *Data* function. An activity, which has life duration dependent on the instance of a business process, comes before an activity that has life duration independent of the instances of business processes.

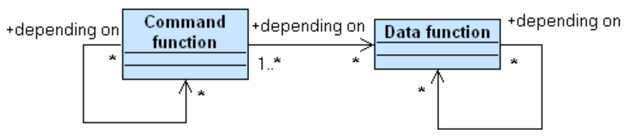

Figure 9. Type pattern applied to {Command, Data} function typing.

The motivation for this pattern is that each process provides one end-result for its "process customer" mapping with at least one activity, whose life duration depends on the instance of this process [20]. This activity may then need the end results of activities that have life durations independent of the instance of this business process. For example, in order process, order activity comes after a product reference consultation activity.

We can complete this type pattern with the following rules:

- *Command* function could depend on a *Data* function;
- *Command* function could depend on another *Command* function;
- *Data* function could depend on another *Data* function; and
- *Data* function never depends on *Command* function.

BFR2 – Multi-typed function sub-functions model. Sub-functions deduced from one multi-type function must satisfy a type pattern.

Considering the *t* function defined in (2), the type pattern motivated in Figure 9 is applied to sub-functions in Figure 10.

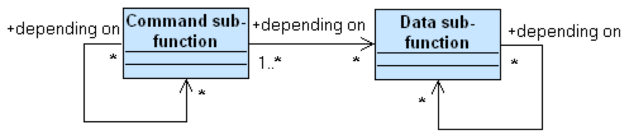

Figure 10. Type pattern applied to sub-functions typed by {Command, Data}

Alignment of the functional view with the business view may be improved because of type pattern consistency in relation to business view design.

C. Typing of Functional Component Dependency in relation to the Business View

Type pattern consistency with the business view may also be applied to function dependency.

BFR3 – Mono-typed functions model. A dependency between two mono-typed functions must respect type pattern for dependency existence and dependency orientation.

To decide the orientation of this dependency, let us note that each process has to be supported first by a command-typed function.

Messaging dynamic illustrations in Figures 5, 8 and 11 all show that message receipt process is supported by *Filter a message from message sender*, which is a *Command* typed function. A dependency of the *Consult message filtering rule* on the *Create message filtering rule* is designed in the messaging sequence diagram. This dependency does not satisfy BFR3 because the *Consult message filtering rule* is a *Data* typed function and the *Create message filtering rule* is a *Command* typed function. The business interpretation of this dependency is that one activity of message rule consulting is not useful in message rule creation.

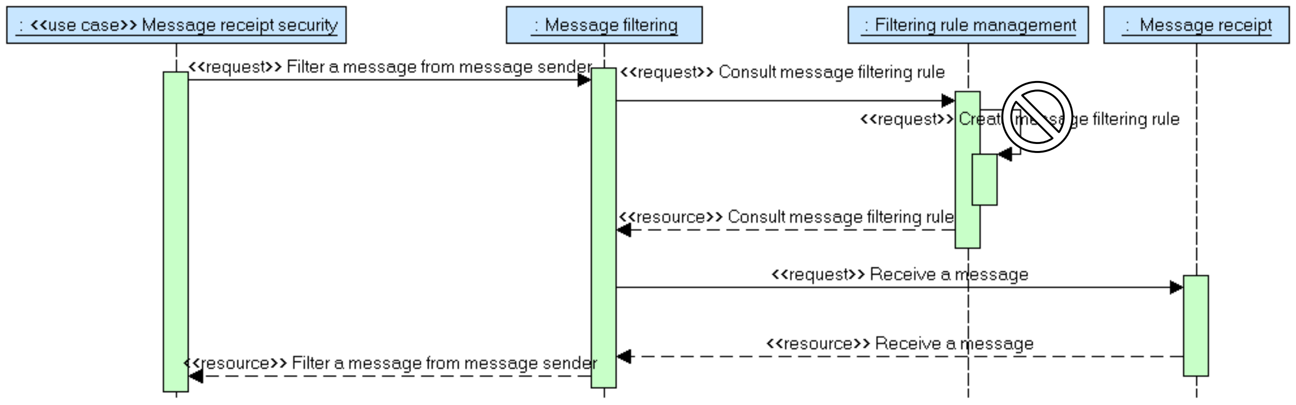
: <<use case>> Message receipt security
: Message filtering
: Filtering rule management
: Message receipt
<<request>> Filter a message from message sender
<<request>> Consult message filtering rule
<<request>> Create message filtering rule
<<resource>> Consult message filtering rule
<<request>> Receive a message
<<resource>> Receive a message
<<resource>> Filter a message from message sender

Figure 11. Functional view sequence of messaging service with mono-typed functions dependency example.

This dependency can, moreover, be analyzed at the data production level and with a data access security point of view. A command-typed function's need for a data-typed function to enable a process to be achieved is illustrated by the dependency of data characterizing the reception of an email, such as its reception date, toward data characterizing a filtering rule. This data dependency provides the opportunity to secure data produced by data-typed function. It is then possible to realize a data-typed function with high security requirements (in the present example: access control depending on the data and on the role of the user: administrator or developer of the mail service), but the realization of the command-typed block is associated with weaker security requirements (in the example: user authentication in the mail service).

VI. Case Study

The alignment measure of the functional view with the business view is a tool for functional architects to compare the business alignment of various functional domains (messaging, IPTV, telephony, etc.) and so to prioritize their actions and improve the alignment. Such action can be guided by an assessment of the alignment. A case study with telecom messaging functions was conducted within the Orange Labs. This domain contains 8 functional components, 12 functions, and 16 function dependencies between functional components for six scenarios. The associated business view contains three activities and two activity time successions.

The alignment measure, **BFAM,** of the messaging domain functional view with the business view of telecom service usage is estimated (see formula (1)):

$$\boxed{BFAM\left(\text{Messaging}\right)=\frac{7}{8}}$$

Let us illustrate with a simple case how to improve the alignment. The assessment for messaging is the following: the alignment of the functional view of the Messaging domain with the business process of message sending would be perfect (i.e., with a measure estimated as 1) if the dependency relationship

- from the *Transmit a message* function defining *Message exchange* functional component
- on the *Store a message* function defining *Message storage* functional component

could be reversed (see Figure 12).

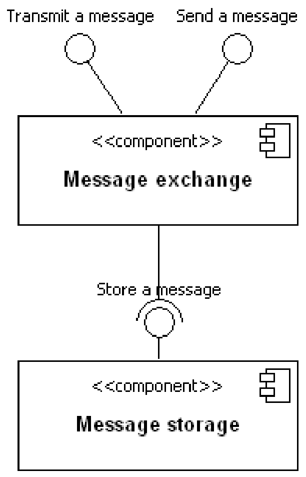

Figure 12. Messaging domain problem.

To maintain the aligned dependency from *Send a message* on *Store a message*, this reversal should require the *Message exchange* functional component to be split as follows into a:

- *Message sending* functional component defined by the *Send a message* function, and a
- *Message transmission* functional component defined by the *Transmit a message* function.

From definition (2), typing can be expressed as:

$$t(\text{Send a message}) = \{Command\}$$
$$t(\text{Transmit a message}) = \{Data\}$$
$$t(\text{Store a message}) = \{Command\}$$

Message transmission is indeed a network function within the telecommunication services IS. This function has a life duration independent of the instances of telecommunication service processes. Moreover, *Send a message* and *Store a message* are dependent on the telecommunication service process because the telecommunication service user has an access to data provided by these two functions (message sending date and message storage date, for example).

A dependency relationship of the *Message transmission* functional component on the *Message storage* results from the previous target functional architecture.

One dependency relationship represented in Figure 13 is consistent with the type pattern in Figure 9:

- a dependency relationship of the *Message storage* functional component on *Message transmission*, which improves the alignment.

Another dependency relationship represented in Figure 13 is consistent with the type pattern in Figure 10:

- a dependency relationship of the *Message sending* functional component on *Message transmission*, deduced from the *Message exchange* splitting.

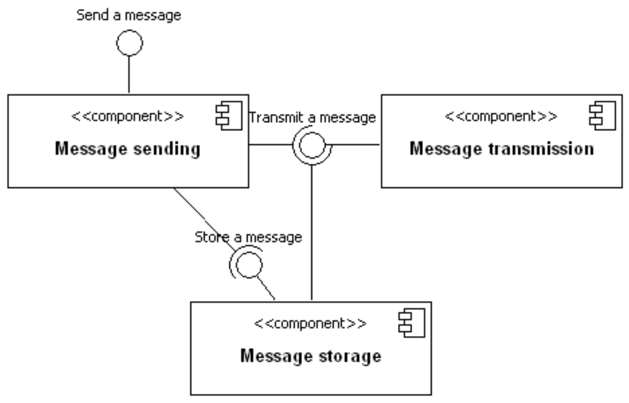

Figure 13. Messaging domain solution.

Using type patterns thus provides a perfect alignment: $BFAM(\text{Messaging}) = 1$ (see (1)).

Functional architects may use this assessment as a tool to improve the business alignment, by checking if the suggested modifications conform to the enterprise strategy.

VII. Conclusion

The modelling process described in this paper enables a representation of the alignment of an IS functional view with the business view of the IS company. The alignment definition is consistent with the meta-modelling used to instantiated both the business and the functional models. An alignment measure is proposed, which provides estimation of the synchronization of a company's strategic integration of business and functional views.

Finally, a good alignment of the target functional view with the business target view induces a good alignment of the applicative view, which in turn implements the target functional view with the target business view. We illustrate this applicative alignment in Figure 14 with our "home domain". An applicative view that implements the functional view is represented in Figure 7.

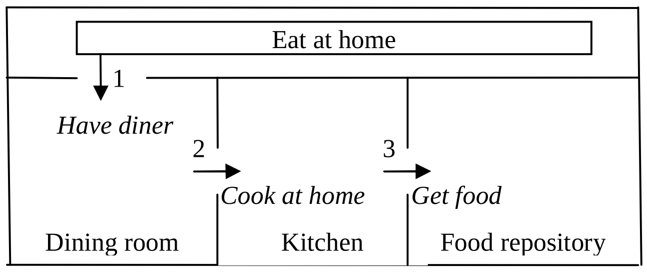

Figure 14. "Home IS" applicative view implementing the *Eat at home* "Home process".

Functional dependency is implemented in the applicative view by only one call from the "Home bus". Function dependency implementation is deployed on the doors between rooms. "Home bus" traffic is thereby lowered. This property, deduced from the alignment improvement of the functional view in compared to the business view, may be significant in Service Oriented Architecture (SOA) [21] design.

The applicative view may contain several hundred applications, which can hardly be directly mapped with the company business processes. In our approach, the target functional view is used as a link between the business and applicative views. This indirect mapping allows an efficient tool to govern IT evolution according to company strategy.

REFERENCES

[1] J. Simonin, E. Bertin, Y. Le Traon, J.-M Jézéquel, and N. Crespi, "Business and Information System Alignment: a Formal Solution for Telecom Services," International Conference on Software Engineering Advances (ICSEA 2010), Nice, France, 2010.

[2] J.A. Zachman, "A Framework for Information Systems Architecture," IBM Systems Journal 26, no. 3, 1987, pp. 276–292.

[3] C. Longépé, "The Enterprise Architecture IT project – The Urbanisation paradigm," Kogan Page, 2003.

[4] I. Jacobson, G. Booch, and J. Rumbaugh, "The Unified Software Development Process," Addison-Wesley, 1999.

[5] C.M. Pereira and P. Sousa, "Getting into the Misalignment between Businesss and Information Systems," 10th European Conference on Information Technology Evaluation, Madrid, Spain, 2003.

[6] J.C. Henderson and N. Venkatraman, "Strategic Alignment: Leveraging Information Technology for Transforming Organizations," IBM Systems Journal, vol. 32, no. 1, 1993, pp. 4–16.

[7] The Business Rules Group, "The Business Motivation Model," http://www.businessrulesgroup.org/bmm.shtml, 2007.

[8] J. Simonin, Y. Le Traon, and J.-M.Jézéquel, "An Enterprise Architecture Alignment measure for Telecom Service development," 11th IEEE International Enterprise Distributed Object Computing Conference, Annapolis, Maryland U.S.A., 2007.

[9] J.N. Luftman, R. Papp, and T. Brier, "Enablers and Inhibitors of Buisness-IT Alignment," Communications of the Association for Information Systems, vol. 1, article 11, 1999.

[10] S.R. Chidamber and C.F. Kemerer, "Towards a Metrics Suite for Object Oriented Design," IEEE Transactions on Software Engineering, vol. 20, no. 6, 1994, pp. 476–493.

[11] G. Booch, J. Rumbaugh, and I. Jacobson, "The Unified Modeling Language – User Guide," Addison – Wesley, 1999.

[12] P.P.S. Chen, "The entity relationship model - Towards a unified view of data," ACM Transactions on Database Systems, vol. 1, no. 1, 1976, pp. 9–36.

[13] Open Mobile Alliance, "OMA Service Environment," OMA-Service-Environment-V1_0_2-20050803-A, 2005.

[14] M.J. Shepperd and D. Ince, "Derivation and Validation of Software Metrics," Oxford University Press, 1993.

[15] S.G. MacDonell, M.J. Shepperd, and P.J. Sallis, "Metrics for Database Systems: An Empirical Study," Software Metrics Symposium, 1997, pp. 99–107.

[16] H. Zimmermann, "OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection," IEEE Transactions on Communications COM-28, no. 4, 1980.

[17] M. Farsi, K. Ratcliff, and M. Barbosa, "An overview of controller area network," Computing & Control Engineering Journal, vol. 10, n°3, 1999, pp. 113–120.

[18] TM Forum, "enhanced Telecom Operations Map," http://www.tmforum.org/browse.aspx?catID=1648

[19] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design Patterns – Elements of Reusable Objetc-Oriented Software," Addison Wesley, 1995.

[20] M. Hammer, J. Champy, "Reengineering the Corporation: a Manifesto for Business Revolution," Harper – Collins, 1993.

[21] R.T.Burlton, "Business Process Management – Profiting from Process," Sams Publishing, 2001.