An Enterprise Architecture Alignment Measure for Telecom Service Development

Jacques Simonin[1], Yves Le Traon[2] and Jean-Marc Jézéquel[3]
[1]*France Telecom R&D, 2 avenue P. Marzin 22300 Lannion, France,*
[2]*ENST Bretagne, 2 rue de la Châtaigneraie, 35576 Cesson-Sévigné Cedex, France,*
[3]*IRISA, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France*
jacques.simonin@orange-ftgroup.com, yves.letraon@enst-bretagne.fr, jezequel@irisa.fr

Abstract

The increasing complexity of modern Information Services (IS) makes necessary to carry on review activities. For many companies, these reviews take place within the framework of the Enterprise Architecture, which aims at sharing components among applications within the IS. These reviews can for instance help the architects in detecting alignment problems between an analysis model and a design model constrained by the enterprise data model. We propose an automatic approach, relying on Model Driven Engineering (MDE), to measure such an alignment. We propose a set of measures which are formally defined using a classical axiomatization approach. We describe how these measures can be implemented with MDE techniques. We finally illustrate their use for the France-Telecom IS of telecom services.

Keywords: telecom service development, Enterprise Architecture, Model Driven Architecture/Engineering, alignment measure.

1. Introduction

1.1 Context

An Enterprise Architecture (EA) has for main objective to plan the development of the Information System (IS) of a company. This planning activity takes the objectives of the company as input. For a telecommunication company, such as France Télécom, the IS concerns the telecom service applications, and an objective may cover the definition of a new telecom service as well as the evolution of networks.

The EA frameworks (like Zachman [1]) define various **points of view** (business, system, etc.) in order to take into account all the aspects of these objectives.

This paper focuses on the system EA point of view, here the telecom service EA point of view. It is taken into account by telecom service architecture designers. Within this point of view, the functional EA describes the functional environment in which every new telecom service should be inserted. The telecom services that we consider here are telecom services supplied to a user, e.g. an electronic messaging service.

During a development based on Unified Process (UP) [2], the design of a telecom service is iterative. The task is very complex and was done manually. It led to inconsistencies, which also had to be corrected manually in an iterative loop. The use of MDA [3] helps reducing the effort to be put in this process. We propose to introduce after each UP iteration a measure of the alignment between the functional requirements analysis and the data architecture design. This measure, consistent with the MDA approach [4], enables a design optimization [5].

We consider the following mapping of the EA models into the MDA context. The analysis model of the functional requirements of a telecom service is a Platform-Independent Model (PIM). The model of the EA functional view is considered, in a rather unusual way, as the Platform Definition Model (PDM) because it constrains the PIM transformation into a model which is the target Platform-Specific Model (PSM).

Section 2 presents how architects deal with the functional EA during the development of a telecom service. This section clarifies the MDA use as well as the characterization of a functional EA. Section 3 concerns the axiomatization [6] of the behaviour and the definition of measures for alignment. Section 4 provides examples of model transformations and alignment measures. In the paper, the chosen modeling language is Unified Modeling Language (UML) [7].

1.2 Related work

The problem of the alignment involving EA is usually considered between the business view and the IS of a company. It may be also considered between the business view of an enterprise and its objectives as in the Business Motivation Model [8]. For this kind of alignment, heuristics have been defined to provide warnings in case of misalignment [9]. Measurement method concerning EA allows especially evaluation of this architecture in business terms (cost, benefit, risk).

Without taking in account of EA, many object-oriented measures exist. To estimate models consistency in this paper, coupling measures [10] are the most appropriate because the interactions of models are the main characteristic of the proposed solution.

2. Data architecture constrained by functional EA

When EA is designed for an IS, telecom service architects have to deal with it during the transformation of the requirements into an architecture model.

2.1 PIM

The analysis of the functional requirements of a telecom service is an activity of the UP Analysis workflow. The static PIM (**PIM_S**) is the model of the **entities** which are involved in the use cases describing the telecom service. The dynamic PIM (**PIM_D**) represents the scenarios which instantiate a use case. For each scenario, an UML sequence diagram depicts the messages exchanged by the instances of entities. Each entity and each link between entities are extracted from use cases. The entity model is captured by an UML class diagram.

2.2 PSM

The **data** model of the telecom service is a delivery of the UP Design workflow. The significant feature of a data is that only one component produces it. This component is created during the design workflow. This model plays the role of static PSM (**PSM_S**). An UML class diagram represents the data and their dependencies. Each scenario is represented during the dynamic design with a UML sequence diagram where the messages exchanged by the instances of data during the course of scenario making up the dynamic PSM (**PSM_D**).

2.3 PDM

In our approach, functional EA view plays the role of PDM, because it constrains the transformation from the PIM to a PSM. It is designed by enterprise architects that are experts of the IS areas of telecom services. They design a functional view which corresponds to the objectives of the company. **Functional component** and dependency between these components are the main concepts of the functional view. In the case of France Télécom, the telecom services IS functional view is composed of about several tens of functional components. These quantities are comparable to the size of the business view (process and sub-process) designed for telecommunication companies such as described by the TMF (TeleManagement Forum) [11].

The dependencies between functional components must also translate the objectives of the company. For example, the enterprise architects may consider that, strategically, a mailbox has to refer to the identity of a user of a messaging service. In that case, in the functional view, they design a dependency from the *Identity Management* component to the *Messaging Box Management* component. An UML component diagram captures these functional components and their dependencies.

2.4 Model transformation

The rules formalizing the model transformation have the following inputs: the concepts of the static PIM or the dynamic PIM, and the concepts of the PDM. The result of a PIM_S transformation is a static PSM (PSM_S) and the result of a PIM_D transformation is a dynamic PSM (PSM_D).

- **Rule 1 (R1) – Data design** (see Figure 1)**.**
  Every PSM_S datum comes from a PIM_S entity (*is extracted from* relationship) and *is produced by* only one PDM functional component. We define the functional component which produces the datum as referencing (*references*) the entity from which the datum is extracted during the transformation.

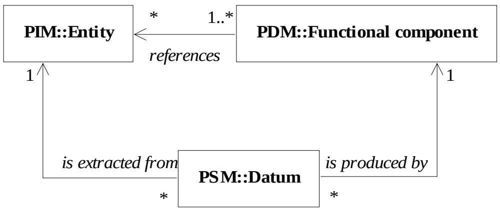

Figure 1. Data design

- **Rule 2 (R2) – Data dependency design** (see Figure 2)**.**
  A dependency between PSM_S data comes from a link between PIM_S entities (*is extracted from*), and *is consistent with* a dependency between PDM functional components which produce data. It means that a data dependency exists when a dependency between components producing data exists. We define the functional component dependency as referencing the link between entities (*references*).

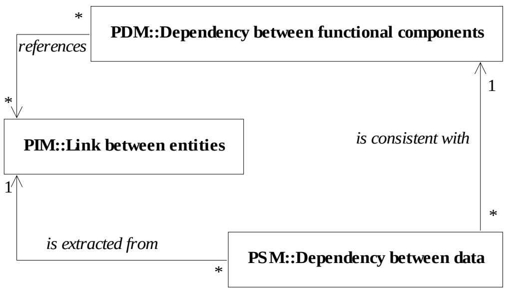

Figure 2. Data dependency design

- **Rule 3 (R3) – Message design** (see Figure 3)**.**
  A PSM_D message between instances of data is the result of the transformation of a message exchanged between instances of PIM_D entities (*is extract from*). It *is consistent with* a dependency between PDM functional components which produce data. It means that a message exchanges between instances of data exists when a dependency between components producing data exists. We define the dependency between functional components as referencing the message between instances of entities (*references*).

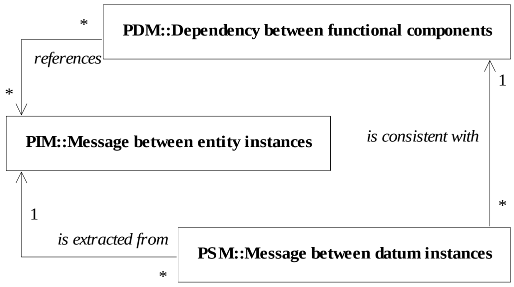

Figure 3. Message design

During the UP Analysis workflow, the telecom service architect references every PIM_S entity by a set of functional components which produce the PSM_S data extracted from the entity. The architect has to study every attribute of the entity to design the data.

A dependency between data is in a warning state when it does not satisfy the rule **R2**. It means that there is no dependency between functional components which produce these data, even though there is a link between the entities from which data are extracted.

A message exchanged between instances of data is in a warning state when it does not satisfy the rule **R3**. It means that there is no dependency between functional components which produce these data, even though there is a message exchanged between instances of entities from which data are extracted.

These warnings mean loss of information between the analysis of functional requirements and the data architecture according to the EA. This loss has to be assessed by an alignment measure to help the architect improve the final PSM.

3. Alignment measure and axiomatization

Static and dynamic alignment measures between PIM and PSM are proposed to evaluate the loss of information between the analysis of functional requirements and the data architecture.

These measures are used in a context of an iterative analysis-design process to help architects in the choice of the best solution between two iterations.

3.1 Measurement elaboration

The literature emphasizes the difficulty in constructing valid measurements [9] [12] [13] [14]. In this paper, since the measured factors first appear as quite abstract and unclear, we choose to make the axiomatization of the measures. Figure 4 illustrates the process of the measurement construction: the factor to be measured is first informally defined and significant and measurable attributes are identified (with intuitive and hopefully convincing arguments and assumptions). Then the intuitive properties of the factor behavior must be expressed using the chosen attributes: this is what we call axioms. An axiom is an expected and understandable property of the measurement that also has a meaning in the mathematical model. It makes the connection between the intuitive/real world and the formal one for the theoretical validation.

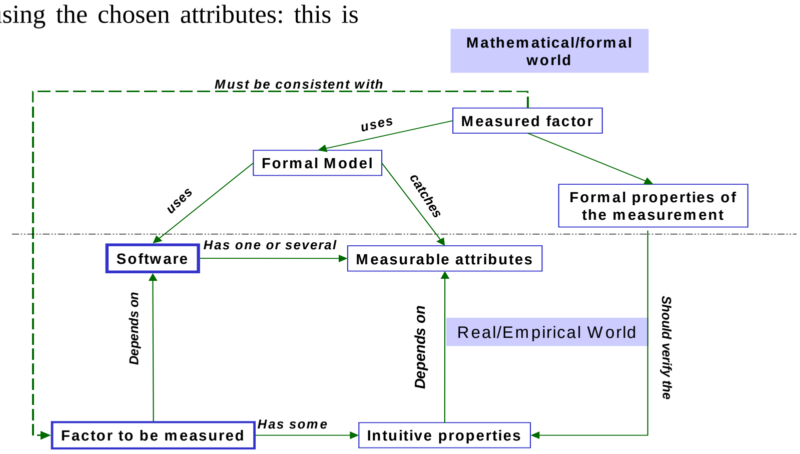

Figure 4. Measurement elaboration

3.2 Static alignment measure

By definition, a Static Alignment (**SA**) is an alignment of a PIM_S with its PSM_S. The entity referencing defined by **R1** rule and the link between entities referencing defined by **R2** rule allow a SA modelling. It concerns the links between entities (modeled during telecom service analysis), and the dependencies between data (which are the result of the model transformation described in Section 2.4). The entity model (i.e. PIM_S) and the data model (PSM_S) are thus the SA parameters.

$SA(PIM_S, PSM_S)$ is null if no link between entities has a reference among the dependencies between data. $SA(PIM_S, PSM_S)$ is perfect if all the PIM_S links have a reference among the PSM_S dependencies.

The following Static ALignment axioms (**SAL**) concern a part of the intuitive behaviour of a static alignment measure according to different statistic models PIM_S and PSM_S specifying one telecom service. This is the case of an iterative development (see Section 2.1). For each iteration, it is indeed possible to have a specific entity model which is transformed into a specific data model.

– **SAL1 – Dependency addition between data.** The static alignment resulting from the addition (denoted ◊) of a Data Dependency (DD) between two PSM_{S1} data is better than or identical to the static alignment of the model PIM_{S1} and the associated data model PSM_{S1}. It is identical when DD is extracted from no links in PIM_{S1}:

  If $PIM_S = PIM_{S1}$ and $PSM_S = PSM_{S1} \lozenge \{DD\}$
  $\Rightarrow SA(PIM_S, PSM_S) \geq SA(PIM_{S1}, PSM_{S1})$.

– **SAL2 – Dependency deletion between data.** The static alignment resulting from the deletion of a DD between two PSM_{S1} data is worse than or identical to the static alignment of the model PIM_{S1} and the associated data model PSM_{S1}. It is identical when DD is extracted from no links in PIM_{S1}:

  If $PIM_{S1} = PIM_S$ and $PSM_{S1} = PSM_S \lozenge \{DD\}$
  $\Rightarrow SA(PIM_S, PSM_S) \leq SA(PIM_{S1}, PSM_{S1})$.

A Static Alignment Measure (**SAM**) according to these axioms depends on two parameters of PSM_S:

– dependency in a warning state or not.

$$\text{Let, } SAM(PSM_S) = \frac{N_{dd}(PSM_S)}{N_d(PSM_S)} \quad \textbf{(1)}$$

where

– N_{dd} (PSM_S) is the number of dependencies between data which are not in a warning state
– Nd (PSM_S) is the number of data.

The SAM parameters are well known for a data model [15]. The transformation implies a same direction for the data dependency than the direction of the dependency existing in the functional components which produce these data. Therefore, the only dependencies which are not directed in PSM_S are those in a warning state.

3.3 Dynamic alignment measure

By definition, the Dynamic Alignment (**DA**) is the alignment of the PIM_D with the PSM_D. The entity referencing defined by **R1** rule and the message between instances of entities referencing defined by **R3** rule allow a DA modelling. It concerns the message exchanged:

- between instances of entities modeled during telecom service analysis and
- between instances of data which are the result of the model transformation.

The dynamic model of analysis (PIM_D) and the dynamic model of data (PSM_D) are thus the DA parameters. DA(PIM_D, PSM_D) is null if no message between instances of entities involved in PIM_D, has a reference among the messages exchanged between instances of PSM_D data. DA(PIM_D, PSM_D) is perfect if all the messages of PIM_D have a reference among the messages of PSM_D resulting of the transformation described in Section 2.4.

The following Dynamic ALignment axioms (**DAL**) concerns a part of the intuitive behaviour of a dynamic alignment measure according to dynamic models specifying one telecom service. This is the case of an iterative development.

- **DAL1 – Message addition between instances of data.**
  The dynamic alignment resulting from the addition (denoted ◊) of a message M_D exchanged between instances of data PIM_{D1}:
  $PIM_D = PIM_{D1}$ and $PSM_D = PSM_{D1} \Diamond \{M_D\}$
  $\Rightarrow DA(PIM_D, PSM_D) \geq DA(PIM_{D1}, PSM_{D1})$.
- **DAL2 – Message deletion between instances of data.**
  The dynamic alignment resulting from the deletion of a message M_D exchanged between instances of data involved in PSM_{D1}, is less good than or identical to the dynamic alignment of the model PIM_{D1} and the associated data model PSM_{D1}. It is identical when M_D references no message of the model of dynamic analysis PIM_{D1}:
  $PIM_{D1} = PIM_D$ and $PSM_{D1} = PSM_D \Diamond \{M_D\}$
  $\Rightarrow DA(PIM_D, PSM_D) \leq DA(PIM_{D1}, PSM_{D1})$.

A Dynamic Alignment Measure (**DAM**) according to these axioms depends on two parameters of PSM_D:

- message exchanged between instances of data in a warning state or not.

Let, $$DAM\left(PSM_D\right) = \frac{\sum_{i=0}^{N} i * NS_i(PSM_D)}{\sum_{i=0}^{N} NS_i(PSM_D)} \quad \textbf{(2)}$$

where:

- N is the maximal length of sequences, in terms of number of exchanged messages of PSM_D which are not in a warning state.
- $NS_i(PSM_D)$ is the number of sequences of i messages of PSM_D which are not in a warning state during the progress of the scenarios.

The sequences of messages taken into account in the DAM calculation are the sequences of maximal length. This length is estimated by the number of exchanged messages of PSM_D. A message is taken into account for this estimation if it is not in a warning state. The DAM parameters are usual for coupling measures which estimate class interactions in an object-oriented telecom service [16]. The DAM specificity proposed here is its parameterization with the sequences of datum instances carrying out a scenario of the use of the telecom service.

3.4 Complementarity of static and dynamic alignment measures

Both alignment measures are based on referencing by dependencies between functional components of the functional EA. The complementarity of the static and dynamic aspects is a fundamental characteristic of the analysis and the design of the telecom service architecture. This complementarity is thus an important property of the measures of static and dynamic alignments.

4. Illustration

A telecom service of messaging is used to illustrate the model transformation and the alignment measures, respectively defined in Section 2.4 and Section 3.

4.1 Static PIM and dynamic PIM

The use case analysed during the first iteration is the following one: a message is sent by a user, and the message is then stored in a box.

PIM_{S1} (see Figure 5) represents the participating entities of the use case: the *User's identity*, the *Message* and the *Messaging box*. The links between these entities relates the message to its user (a message *is sent by* a user) and the message to the messaging box (a message *is stored in* a messaging box).

Figure 5. PIM_{S1}

The nominal scenario derived from this use case is shown by Figure 6.

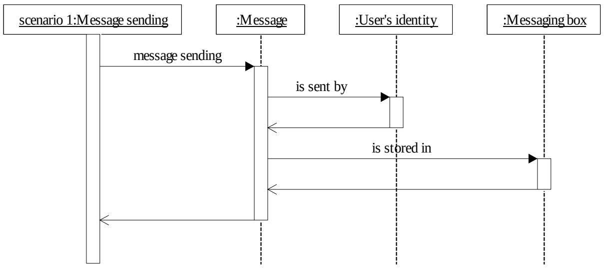

Figure 6. PIM_{D1}

The use case analysed during the second iteration is the following one: a message is sent by a user which owns a messaging box.

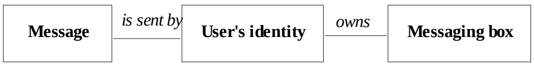

Figure 7. PIM_{S2}

PIM_{S2} (Figure 7) represents the participating entities and their links which are henceforth between the message and the user (a message *is sent by* a user) and between the user and the messaging box (a user *owns* a messaging box). The nominal scenario deriving from this use case is represented in Figure 8.

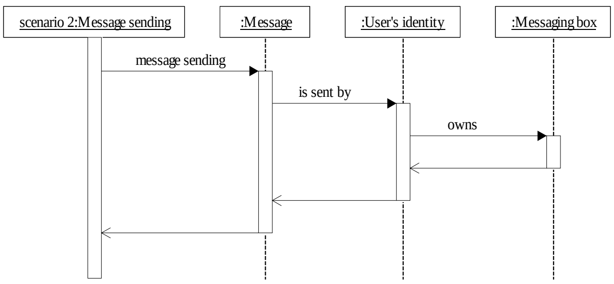

Figure 8. PIM_{D2}

4.2 PDM

The functional EA extract is bounded in Figure 9. It contains 3 functional components: *Message Sending*, *Identity Management* and *Message Storage*. The dependencies proposed by the enterprise architects are:

– from *Message Sending* to *Identity Management*, and
– from *Identity Management* to *Message Storage*.

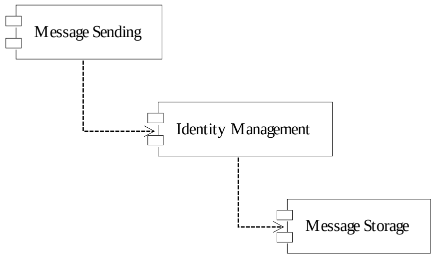

Figure 9. PDM

4.3 Static PSM and dynamic PSM

During the first iteration, PIM_{S1} is transformed into PSM_{S1} by **R1** and **R2** rules application (see Figure 10). According to **R1** rule, the telecom service architect defines data as follow:

– *User's identity – Identity Management* datum extracted from *User's identity* entity, and produced by *Identity Management* functional component.
– *Message – Message Sending* datum extracted from *Message* entity and produced by *Message Sending* functional component.
– *Messaging box – Message Storage* datum extracted from *Messaging box* entity and produced by *Message Storage* functional component.

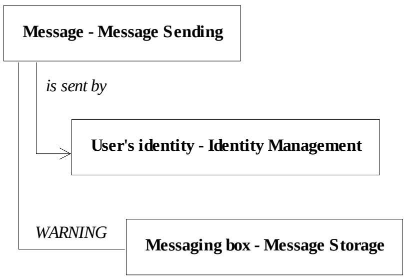

Figure 10. PSM_{S1}

According to **R2** rule, two data dependencies are then defined**:**

– *is sent by* dependency from *Message – Message Sending* to *User's identity – Identity Management*
  – It is extracted from *is sent by* link between entities, and it is consistent with the dependency from *Message Sending* to *Identity Management* functional components

– The dependency between *Message – Message Sending* and *Messaging box – Message Storage* data.
  – It is extracted from *is stored in* link between entities but it is not consistent with a PDM dependency because of the lack of dependency between *Message Sending* and *Message Storage* functional components (R2 rule). So its state is *WARNING*.

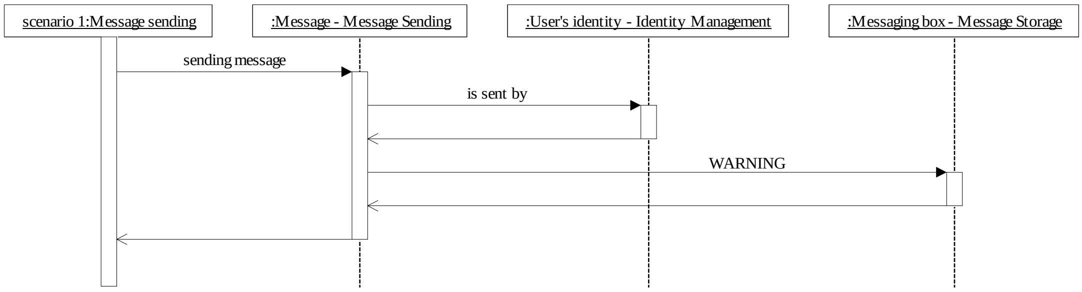

Figure 11. PSM_{D1}

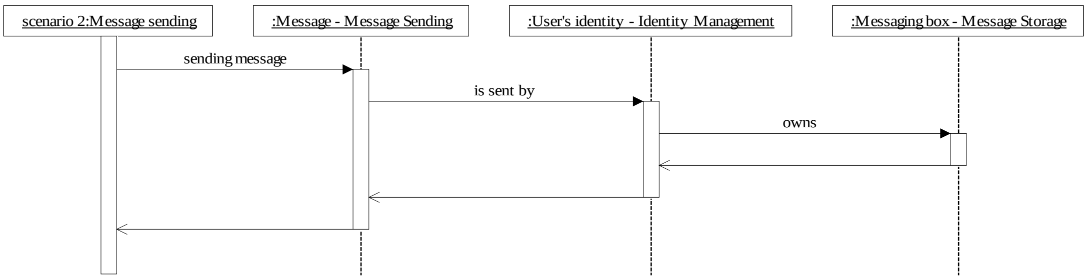

Figure 13. PSM_{D2}

The **R3** rule application for the transformation of PIM_{D1} returns the PSM_{D1} model (see Figure 11) as follow:

– *is sent by* message sent by an instance of *Message – Message Sending* datum to an instance of *User's identity – Identity Management* datum
  – It is extracted from *is sent by* message between the instances of entities and it is consistent with the dependency between the functional components from *Message Sending* to *Identity Management*
– The message sent by a *Message – Message Sending* instance to a *Messaging box – Message Storage* instance
  – It is extracted from *is stored in* message between the instances of entities but it is not consistent with a PDM dependency because of the lack of dependency between the *Message Sending* and *Message Storage* functional components (rule R3).So its state is *WARNING*.

For the second iteration, the **R1** and **R2** rules application on PIM_{S2} returns PSM_{S2} (see Figure 12).

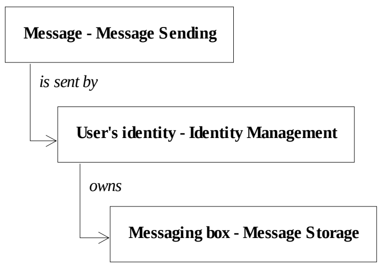

Figure 12. PSM_{S2}

The **R3** rule application on PIM_{D2} returns PSM_{D2} (see Figure 13).

4.4 Static alignment measure

According to the formula **(1)**, the SAM value for PSM_{S1} is: $SAM(PSM_{S1}) = \frac{1}{3}$, because PSM_{S1}

contains 3 data and 1 dependency between these data which is not in a WARNING state.

The SAM value for PSM_{S2} is: $SAM\left(PSM_{S2}\right)=\frac{2}{3}$.

4.5 Dynamic alignment measure

According to the formula **(2)**, the DAM value for PSM_{D1} is:

$$DAM\left(PSM_{D1}\right)=\frac{(0*2)+(1*0)+(2*1)}{2+0+1}=\frac{2}{3}$$

because PSM_{D1} contains:

- 1 sequence of 2 messages exchanged between the instances of the data
- 2 sequences of messages in a WARNING state.

The DAM value for PSM_{D2} is: $DAM\left(PSM_{D2}\right)=4$.

4.6 Results

The static alignment and the dynamic alignment are better for the second iteration than for the first one. The proposed example illustrates applications of the:

- axiom **SAL1** on the addition of a dependency between data,
- axiom **DAL1** concerning the addition of a message exchanged between instances of data,
- complementarity of the static and dynamic alignment measures.

5. Conclusion

The MDA-based process described in this paper enables to take into account the functional EA at the beginning of the telecom service design process. The automated transformations complete the design process with a measure of the loss of information between the functional requirement analysis, and the design of the data architecture according to the functional EA. The first benefit of this approach is to make the telecom service architect task much easier.

As several applications at France Télécom confirmed it, performing this measure as soon as possible during the telecom service development provides a better return on investment for the whole EA activity. Indeed, it enables an easier reuse of the components developed in an IS.

6. References

[1] J.A. Zachman, "A Framework for Information Telecom services Architecture", IBM Telecom services Journal 26, No. 3, 1987, pp. 276-292.

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

[3] OMG: Model Driven Architecture (MDA), http://www.omg.org/mda/.

[4] D.S. Frankel, "Model Driven Architecture – Applying MDA to Enterprise Computing", Wiley Publishing Inc, 2003.

[5] M. Woodside, G. Franks, D.C. Petriu, "The Future of Software Performance Engineering", Future of Software Engineering 2007, 2007, pp. 171-187.

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

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

[8] The Business Rules Group, "The Business Motivation Model", http://www.businessrulesgroup.org/bmm.shtml, 2005.

[9] C. M. Pereira, P. Sousa, "Enterprise Architecture: Business and IT Alignment", 2005 ACM Symposium on Applied Computing, 2005, pp. 1344-1354.

[10] S.R. Chidamber, 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] TeleManagement Forum: http://www.tmforum.org/.

[12] N.E. Fenton and R.W. Whitty, "Axiomatic approach to Software Metrication through Program Decomposition", The Computer Journal, vol.29, 1986, pp. 330-339.

[13] L.C. Briand, S. Morasca, V.R. Basili, "Property-Based Software Engineering Measurement", IEEE Transactions on Software Engineering, vol.22, 1996, pp. 68-86.

[14] B. Kitchenham, S.L. Pfleeger, N. Fenton, "Towards a Framework for Software Measurement Validation", IEEE Transactions on Software Engineering, vol.21, No. 12, 1995, pp. 929-943.

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

[16] L.C. Briand, J.W. Daly, J.K. Wüst, "A Unified Framework for Coupling Measurement in Object-Oriented Telecom services", IEEE Transactions on Software Engineering, vol. 25, No. 1, 1999, pp. 91-121.