Governance issues on heavy models in an industrial context

Sabri Skhiri[1], Marc Delbaere[2], Yves Bontemps[2], Gregoire de Hemptinne[1], and Nam-Luc Tran[1]

[1] Euranova S.A.
[2] SWIFT : Society for Worldwide Interbank Financial Telecommunication

Abstract. SWIFT is a member-owned cooperative providing secure messaging capabilities to the financial services industry. One critical mission of SWIFT is the standardization of the message flows between the industry players. The model-driven approach naturally came as a solution to the management of these message definitions. However, one of the most important challenges that SWIFT has been facing is the global governance of the message repository and the management of each element. Nowadays modeling tools exist but none of them enables the management of the complete life-cycle of the message models. In this paper we present the challenges that SWIFT had to face in the development of a dedicated platform.

Keywords: governance, meta-modeling, operational issues

1 Introduction

SWIFT is the leader in the banking communication and message transmission. One of its main missions is the management of the communication standards ISO-15022 and ISO-20022 that are used between banks in order to exchange messages. Those standards provide the definition of message payloads (i.e. which data fields can or must be included in which communication flow).

One of the difficulties of managing a worldwide business standard is the continuous need to evolve the standard to cater for new business requirements. From a model management point of view, this creates a lot of new definitions that then have to be organized properly.

In 2009, SWIFT Standards undertook a major strategic study aimed at defining a 5 year roadmap for standard capabilities evolutions. They identified a set of priorities: (i) the management of the content and the reuse, (ii) the ability to support specialized standards and market practices, and (iii) the management of changes.

Very recently, SWIFT and their customers have reiterated the role of ISO-20022 as a mechanism to facilitate the industry integration at the business level. In order to realize this vision, the need of a common and machine-readable definition has been established. This definition comprised the business processes, the data dictionary, the message definitions, the market practices and the mapping

rules. All of these definitions would be managed in a controlled manner over time (versioning). It should be possible for industry players to customize the definitions in order to fit their needs (e.g. local market practices, bilateral agreements) as well as the representations of their own formats in this environment (using the expressive power of the ISO 20022 global dictionary).

A thorough model-based approach has been set up in order to enable the proper governance of the whole model and the interoperability across the actors working on the model.

2 Technical and operational issues

All the issues related to the management of an industry-wide repository of structured business content are related to the concept of governance. This concept involves the control of the life-cycle of each model element, the management of the dependencies between them, the versioning, the design of new elements based on existing components and many other aspects related to governance. One additional complexity in the case of the ISO 20022 models is the existence of two layers of models: a business layer that describes the business at conceptual level (process definitions, business concept definitions at the semantic level) and a message definition layer that describes all of the data objects and the actual message definitions. In this architecture, each data element present in a message definition must be linked back to a semantic element. This additional layer makes it even harder to govern the repository content. In this section we briefly describe the different challenges met by such governance framework.

- **Versioning**: we should be able to maintain a version for each model element with a granularity at the atomic attribute. Most of the modeling frameworks (such as EMF) usually use XMI (XML Metadata Interchange) for the serialization of the models. We would then have to rely on the versioning at the file level, which does not meet the requirement of our versioning granularity.
- **Link between message and semantic objects**: when the first messages are created, the links between the business components and the content of the messages are clear. This relationship must remain clear over the evolution of the repository and the messages. How can we guarantee the relational integrity between business objects and message definitions over different versions?
- **Knowledge of the market practices**: each financial environment has its own needs. In order to let the model evolve on the right way we need to know the main market practices and how they are used. From a model viewpoint, the market practices are a set of restrictions applied on a view. A view is the projection of a subset of messages from the repository. We should be able to maintain a coherent market practice along the different versions of the messages that it constraints. Going further, we should be able to evaluate the impact of a message change on all the market practices using this message and also on the complete repository. This is a typical governance use case.

- **Content management**: the documentation and the structure of the models needs to be centralized in one place in order to promote the reusability of message components and to control the repository content. This includes, among others, the documentation, the search tools, the central repository, the message model templates and the comparison tools (e.g.: find isomorphic components).
- **Auditing tools**: we should be able to audit each modification in the repository and to track back detailed information on the change such as the identity of the requester, the version, the implementor, the impacted release, etc...

3 Solutions

The foundational element of the solution is the implementation of a meta-model above the current standard model. This meta-model allows to manage and to simplify the use of the standard model: instances of the meta-model are the models of the standard, and instances of the models are the message definitions. Both are defined in an XMI file.

- **Versioning and organization**: based on the standards defined before, we have developed a graphical interface built on a meta-model which allows us to create and manage models based on the meta-model and with respect to the standards.
- **Business objects**: these objects are based on the semantic model. The link between the semantic model and the message model is always maintained thanks to the object-oriented programming methodology. Since the message model and the semantic model are defined in the same meta-model, they can be linked and work together. Once the link between message and semantic is done, it is conserved.
- **Auditing tools**: thanks to the meta-model, the objects are always linked. All the variations of the model are based on differences between messages. Every definition has links to other instances of definitions. This makes possible the comparison of objects and the creation of auditing tool. It allows to add search functionalities, impact analysis and message definitions comparison.
- **Market practices**: meta-modeling is also a good choice for the creation of market practices. These are the usages in which the message definitions are used in practice by the users. Thus, these consist in restrictions added on a message definition. This can be seen as elements that are added to the base object. This is easy to accomplish thanks to the meta-modeling. The restriction should be easy to change and to remove. We therefore keep the initial object every time, and it becomes a restricted object.
- **Content management**: in order to structure the documentation, the model allows adding documentation on each object. Furthermore, using models to organize objects and data makes things clearer for non-expert people.

The data can then be stored on a centralized way in order to promote reusability. Modeled architecture data storage as provided by CDO is ideal in this situation.

4 Implementation

We have developed a framework answering the governance issues, based on the Eclipse Rich Client Platform (RCP) tools with the Eclipse Modeling Framework (EMF). We have created a visual editor in order to manage the models very efficiently. Mainly based on tree and editor views, this interface is intended to draw a graphical representation of the models. The documentation for each model element can be added almost everywhere and can be reviewed. We developed an authorization and authentication layer integrated with the editor. In such way, we are able to track back any operation on models. This makes it possible to formally separate the global standardization process from the definition of local market practices.

5 Illustration

In this section we chose to describe two important mechanisms of the Model editor. This description aims at giving a better understanding of the meta model and its usage.

5.1 View model

The View model is used for grouping elements within an high level view. Originally it was designed for resolving two important challenges of the ISO 20022 Repository: (1) finding a relevant content, as its relevance may rely on several criteria of different kinds (project, business domain, publication, etc.), (2) the need to define finer-grained scopes, since the dictionary size will dramatically increase in the coming years. Therefore, the view model is a mechanism for providing a view on specific set of message components. In additionn it offers informations to describe when this view was extracted. The root element of the model exposed in the Figure 1 is the ViewSet, which is nothing else than a set of views. Basically a View has a name, a description and a list of message components. A view does not contain components, it references them. A view can be checked out from the repository. In such a case, the editor will create a local copy of each element. A change request can then be associated with this view. That is why the change model is also linked with the view model. In case of modifications when the view is checked in, the user can use EMF compare and its 3-way comparison to evaluate the changes and merge the different versions. The mechanism of Publication View is based on the same concept. A publication is the mechanism by which the base standard is published, it uses the same concepts and adds additional elements: the previous and next Publication View. These attributes enables to link the publications between them.

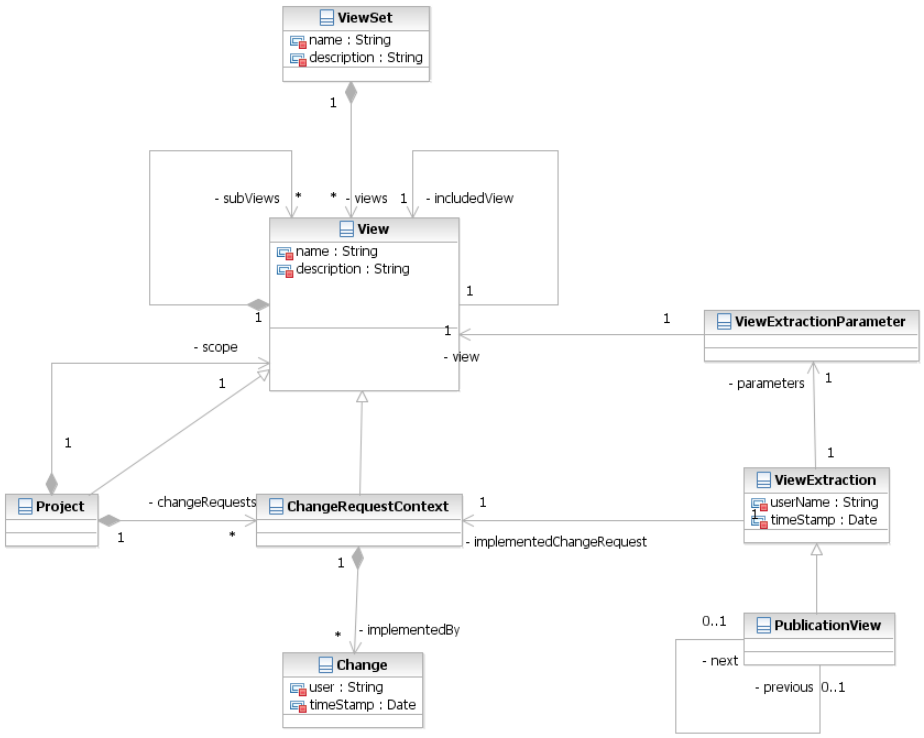

Fig. 1. The view model is used for checking out content, but also for publishing new base standard.

5.2 Semantic Traces

Semantic traces aim at simplifying the understanding of the message model. The ISO20022 repository is composed of a conceptual and a logical layer. The first is used for defining the normalized business model while the second layer is used for defining message components. The message components are a view of the business components, they make up the real message definitions used on the network. The mapping to the business layer aims at giving semantic meaning. In the example of the Figure 2, the PersonIdentification is the message component while its attributes are the message elements. We have to trace the message component to a business component, in this case the Person, but also each message element. A message element can be traced to a business element of the business component traced from its container (person in this case) but also to any other business elements of any business component connected to Person through a navigation path. For instance, a navigation path can specify that the birthPlace message element represents the IsoCode of the birthPlace of a Person.

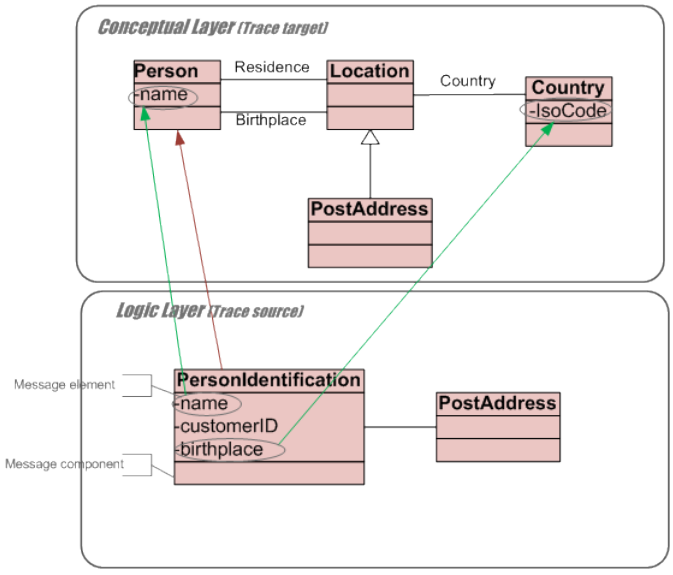

Fig. 2. The semantic trace between the business and logical layer. This trace is used for defining the impact of an element.

By linking two parts of the model that evolve at their own pace, semantic traces can pose governance issues, namely consistency problems.

Thanks to the meta-modeling approach, we could define the concept of impact, eg, a trace target X (a business component has an impact on a trace source y (a message component) only if the target of y is X, and implement an impact analyzer. Now, before any change is performed on the business model, analysts can assess the semantic impact on message definitions.

6 Conclusion

The governance of models of significant size is a real challenge and no framework today can cover all its different aspects. In this case, we have developed a simplified framework covering our prior requirements. However, there is a need for research and industrialization in this area. It has clearly been shown that the modeling approach brings more cohesion and clarity in the managed data. With this problem impacting significant models, we see how we can go back to a more elegant solution that allows easier governance by adding abstraction and meta-modeling.