Supporting Enterprise Cartography through Metamodel Integration

Extended abstract

Margarida Guerreiro Morais
Instituto Superior Técnico
margaridamorais@tecnico.ulisboa.pt

Abstract

Enterprise Architecture (EA) has taken on a major importance for the biggest problem Enterprises face today - "change". Enterprise Architecture enables the bridge between Business-Driven strategy and IT-Driven implementations and the establishment of an Enterprise environment suited to change.

Our motivation with this work is to understand the main differences between two EA specifications (TOGAF and ArchiMate) and following previous work done on TOGAF Framework and ArchiMate Modeling Language Harmonization achieve through the Enterprise Cartography tool, Atlas, a way to transform ArchiMate models into TOGAF models and vice-versa in order to have all the models consolidated in one EA repository.

The result obtained is a semi-automatic solution to transform TOGAF models to ArchiMate models and vice versa, being able to consolidate these models in the same EA repository. The solution proposed is applied to a simple model example that demonstrates the purpose of this research.

Key Words

Enterprise Architecture, Enterprise Cartography, TOGAF, ArchiMate, Atlas

1 Introduction

As much benefits as Enterprise Architecture can bring to the table, a study made in 2010 shows that 66 percent of the EA projects expected results are not achieved and that it is not only the implementation of an EA project itself that is a challenge but also the several difficulties in the maintenance and managing of that same EA. [1] The problem of keeping a set of accurate representations (i.e. models) of an enterprise is not an easy one for large enterprises.

The origin of this difficulty is that the planning process originates in many of the enterprise's communities without a consistent, coherent and complete systemic view of the enterprise to support them, individually and as a whole as well as the struggle to align the different elements designed by different architects resulting into an unconscious design [2].

Referring to Enterprise Cartography (EC) as the process of representing an Enterprise observed directly from reality. We can differentiate it from Enterprise Architecture (EA) because it focuses on producing representations based on observations and not including the purposeful design, as one expects in EA. [2] Adopting a federated approach Enterprise Cartography demises the constraints and inflexibility that one has on the EA modeling and provides a way to have concise and coherent views of the entire organization.

1.1 Problem Description

As previously introduced, it is usual nowadays enterprise's using different specifications in the development of Enterprise Architecture. Having spread across different domains of the enterprise, EA models created with different metamodels, often leads to the lack of an enterprise-wide view.

Given that the ArchiMate 3.1 and TOGAF 9.2 are two specifications widely used in the development of Enterprise Architecture due to the fact that they are both developed by The Open Group, they were designed to be able to be used harmoniously.

Our motivation with this work is to understand the main differences between two EA specifications (TOGAF and ArchiMate) and following previous work done on TOGAF Framework and ArchiMate Modeling Language Harmonization achieve through the Enterprise Cartography tool, Atlas, a way to transform ArchiMate models into TOGAF models and vice-versa in order to have all the models consolidated in one EA repository.

1.2 Document Structure

In the Introduction Section, we begin by referring the difficult task of maintaining an EA up to date and present

the Enterprice Cartography field, describing the problem we want to research in this work and the objectives of the report.

The Enterprise Architecture section gives some theoretical context about Enterprise Architecture, presents the TOGAF and ArchiMate specifications, and its harmonization.

In the Enterprise Cartography section, some theoretical context about Enterprise Cartography discipline is described and also the EC tool Atlas used in the scope of the research is presented.

Following in the Tool Integration Methodology section research on Model-based Integration Methodology, focusing on the XSLT approach is presented.

Problem Analysis is dedicated to problem analysis, contextualizing the current state of work already done in this area and what we want do develop with this work.

In the Objectives Analysis, we present the objectives that were intended to achieve during the dissertation, as well as the solution that resulted from these same achieved objectives.

With Demo section of the solution, the simple model example provided is presented in order to validate the model transformation solution.

The last section concludes with the conclusions about the work presented and presents ideas for future work that can be developed within the scope of the work already done.

2 Enterprise Architecture

Adopting the definition from [13] EA can be viewed as a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure.

The purpose of EA is to optimize across the enterprise the often fragmented legacy of processes (both manual and automated) into an integrated environment that is responsive to change and supportive of the delivery of the business strategy. [12]

2.1 TOGAF Framework

The Open Group Architecture Framework (TOGAF) is a conceptual model of enterprise architecture conceived in 1995 by The Open Group Architecture Forum, whose goal is to provide a global approach to the design, planning, implementation and governance of architectures, thus establishing a common language of communication among architects.

TOGAF is based on an iterative, reusable, cyclical process and supported by the best modeling practices involved in the activities of an organization, comprising four types of architecture that are commonly accepted as subsets of an enterprise architecture, namely: business, data, applications, and technology.

The content of TOGAF is structured in five parts: [12]

1. **Architecture Development Methodology (ADM)** – Defines the process of creating an architecture. Is a full life-cycle process for planning, designing, realizing and governing EA. This cycle has a preliminary part and eight core parts arranged in a sequential cycle order:
   A - Architecture Vision;
   B – Business Architecture;
   C – Information System Architectures;
   D – Technology Architecture;
   E – Opportunities and Solutions;
   F – Migration Planning;
   G – Implementation Governance;
   H – Architecture Change Management.

2. **Architecture Content Framework** – The Architecture Content Framework within the TOGAF framework identifies the main types of architecture building blocks that are relevant in the context of the ADM. Provides a structured model of types, relationships and attributes of building blocks that used informally or as the basis for configuration of an Enterprise Architecture modelling tool represent the basic elements of architecture in TOGAF. The Content Framework features a core and extension concept. The Core concept provides an introduction to the way in which the TOGAF framework employs a basic core metamodel and the extension modules address specific architectural issues in more detail, tailoring the model and making it more robust to answer specific parts of the Enterprise Architecture.

3. **Architecture Capability Framework** – Defines the processes, the skills, the roles and the responsibilities that an organization need to have to construct and develop an EA.

4. **Enterprise Continuum and Tools** – Defines taxonomies and tools to categorize and store the outputs of an EA.

5. **Reference Models** – Technical Reference Model (TRM) and the Integrated Information Infrastructure Reference Model (III-RM).

2.2 ArchiMate Language

ArchiMate is an open and independent modelling language for enterprise architectures, owned and developed by The Open Group [12] . ArchiMate provides instruments to describe, analyze and visualize the relationships among different Enterprise architecture domains in a simple and unambiguous way.

Clear concepts and relationships between architecture domains are presented in the ArchiMate language that offers a simple and uniform structure for describing the contents of these domains.

The ArchiMate language provides a core framework that allows for the modeling of the enterprise from different viewpoints. The aspects (passive structure, behavior and active structure) and the layers (business, application and technology) create the structure of the framework where the position within the cells highlights the concerns of the stakeholder. There is also some layers and an aspect that can be added to the Core Framework. The physical elements are added to the Technology layer for modeling physical facilities and equipment, distribution networks, and materials. Besides, an additional motivation aspect
and the implementation and migration elements are added.

3 Enterprise Cartography

Keeping design and representation together is fine for a centralized coordinated design process, but organizations are actually designed in an asynchronous, distributed process, involving many actors and many stakeholders, without formal mechanisms of communication between the different designers. The current EA patterns are not capable of handling the organization´s changing pace, neither for design purposes nor to keep the architectural representations up to date.

The emerging of the Enterprise Cartography keeps apart design from representation. [4]

3.1 Enterprise Cartography discipline

Enterprise cartography denotes the discipline dealing with the conception, production, dissemination and study of enterprise maps to support the collective understanding of a dynamically changing organization. [3]

Enterprise cartography doesn't focus on the enterprise design, but rather targets the abstraction and representation of the "enterprise reality".

- *enterprise reality* - refers to the present state of the enterprise, sustained on relevant facts captured in logs and represented through artifacts based on previously defined and agreed upon models. [2]

With Enterprise Cartography we can consolidate each partial architecture into a single global set of architectural maps and represent the enterprise architecture in different points in time, namely: [2]

- ***AS-IS*** - Set of all alive artifacts, their relations and states in the present time.
- ***TO-BE*** - Set of alive artifacts, their relations and state at a given time in the future.
- ***emerging AS-IS*** - set of all alive artifacts, their relations and states as observed after the successful completion of ongoing transformation initiatives.

But how does the adoption of an Enterprise Cartography project help in any way the EA process?

From the practical cases discussed in [2] we can see that an EC approach can be the solution to a wide range of challenges encountered in the development, management and maintenance of an EA, such as a way to consolidate models produced in different projects into a single and enterprise-wide view, as a way to create a central knowledge base to allow a descriptive and perspective approach for EA, as a way to integrate important information spread on different tools, having all the challenges essentially a common goal: deploy processes and tools to produce up-to-date architectural maps of the enterprise architecture with near zero effort.

4 Tool Integration Methodology

A rich variety of modeling languages and tools are available, not only to support the development of EA, but to support other domains, but there is no single language that can model all the specific concepts of each organization, so it is quite common that an organization resorts to several EA modeling tools each with its own language. The exchange of models among different tools and thus the integration of the respective languages becomes an important prerequisite to achieve an holistic view of the enterprise.

Due to the lack of interoperability, however, it is often difficult to use tools in combination. [6] To achieve the necessary interoperability between modeling tools various methods were developed on model-based integration.

4.1 Model Based Tool Integration

Model-based tool integration means that tools are integrated on basis of metamodels defining syntax and semantics of the modeling languages supported by the tools. [7] A metamodel is a special kind of model that represents the abstract syntax of a modeling language and conforms to a meta-metamodel.

According to [7] we can analyze the advances in model-based tool integration by two integration challenges, the integration of model syntax where model transformations occur (metamodel integration) and the field of semantic integration (concepts mapping).

4.2 XSLT Approach

Given that the Enterprise Architecture Modelling tools talked in the scope of this research can export (or import) models as XMI files, the research on XSLT declarative language model transformation approach it is the focus.

Because models developed with Enterprise Architecture Modelling tools can be serialized as Extensible Markup Language (XML) using the XML Metadata Interchange (XMI), implementing model transformations using XSLT, which is a standard technology for transforming XML, seems very attractive.

Extensible Stylesheet Language for Transformations (XSLT) is one of the W3C standards, it's a declarative rule-based programming language for transforming XML documents. An XSLT stylesheet consists of a set of rule templates, each rule template matches elements in source model, and produces output to the target model, transforming a XML file into another XML file. XSLT is the most common and powerful language for XML transformation and has strong support to complex pattern matching (Xpath).

XSLT stylesheets can also be easily executed and integrated into different system environments and platforms, without additional packages and libraries. [8]

5 Problem Analysis

This section explains more in depth the current state of ArchiMate and TOGAF metamodels harmonization.

5.1 Problem Context

The two specifications are widely used in the development of Enterprise Architecture and due to the fact that The Open Group develops them both, they were designed to be able to be used harmoniously.

However, as previously discussed, using different specifications in the development of EA, having spread across different domains of the enterprise, EA models created with different metamodels, often leads to the lack of an enterprise-wide view. In order to be able to have all models in the same repository and achieve an enterprise-wide view.

Having studied the framework and language, widely used in the development of EA, we found that there is space to create a solution for the integration of models based on the two specifications. This solution goes through the analysis of the work already developed in the harmonization of the two metamodels in question and in the creation, using the EC Atlas tool, of a semi-automatic mechanism for the transformation between ArchiMate and TOGAF models and vice-versa.

5.2 TOGAF and ArchiMate Harmonization

From Estrem and Gonzalez work developed in the TOGAF R Framework and ArchiMate R Modeling Language Harmonization white paper [9], the Content Metamodel and relationships harmonization between TOGAF and ArchiMate language is exposed.

The paper has the purpose of comparing and contrast the Content Metamodels of the TOGAF 9.1 and ArchiMate 2.1 standards. It is also intended to provide guidance to EA practitioners to better utilize both standards in the development of models for EA projects.

The two standards can be joined together but should remain independent of each other, they work well and are compatible and complementary for EA development.

They have been developed independently of each other and address different though related purposes. The TOGAF metamodel is an abstract, implementation independent model used to define the elements that are used within the artifacts used in the framework.

The ArchiMate language metamodel is more specific and defines entities and relationships used for architecture modeling.

The document is structured in two parts. The first part provides the examination of the entities defined in the Content Metamodel of the TOGAF 9.1 and Archimate 2.1 standards . The second part examines the relationships that are defined between the entities in the respective standards.

Estrem and Gonzalez give a detailed explanation of how the entities and relationships of each of the standards are related, in cases where there is mapping
and how strong this mapping is. In cases where entities that are present in the ArchiMate metamodels do not exist in the TOGAF metamodel and vice versa, the mapping of entities between the standards sometimes can be performed by either defining attributes (profiling) or through specialization and some specific examples of these mappings are provided.

The work exposed in the article is focused on harmonizing the specifications of both metamodels, but does not present a concrete solution of how we can materialize the transformation between a TOGAF model and an ArchiMate model or vice versa, which will be addressed in this research. The work also focuses on earlier versions of the TOGAF and ArchiMate specifications. This research will focus on the recent versions of the two specifications and the main changes of both standards are analysed to understand if the alignment between ArchiMate and TOGAF has increased in some aspects.

5.3 From ArchiMate 2.1 to 3.1 and TOGAF 9.1 to 9.2

The previous work on ArchiMate and TOGAF Harmonization was made following Version 2.1 of the Archiamte Specification and version 9.1 of the TOGAF Specification and their current versions stand on ArchiMate 3.1 and TOGAF 9.2.

An overview of the main changes of both standards were made to understand if the alignment between ArchiMate and TOGAF increased in some aspects.

In this subsection, the differences found between Version 2.1 and Version 3.1 of the ArchiMate specification will be presented. Then the differences found between Version 9.1 and Version 9.2 of the TOGAF specification will be addressed. Complementing this analysis with the presentation of new focuses of harmonization between the two specifications.

The following elements were introduced to the ArchiMate metamodel:

Outcome	An end result that has been achieved	Outcome
Course of action	An approach or plan for configuring some capabilities and resources of the enterprise, undertaken to achieve a goal.	Course of Action
Capability	An ability that an active structure element, such as an organization, person, or system, possesses.	Capability
Resource	An asset owned or controlled by an individual or organization.	Resource
Value Stream	A value stream represents a sequence of activities that create an overall result for a customer, stakeholder, or end user.	Value stream
Application process	A sequence of application behaviors that achieves a specific outcome.	Process Process
Technology process	A sequence of technology behaviors that achieves a specific outcome.	Process Process
Application event	An application behavior element that denotes a state change.	Event Event
Technology event	A technology behavior element that denotes a state change.	Event Event
Technology interaction	A unit of collective technology behavior performed by (a collaboration of) two or more nodes.	Collaboration Collaboration
Implementation event	A behavior element that denotes a state change related to implementation or migration.	Event Event

Figure 1. ArchiMate 3.1 - New Elements [10]

The following elements were introduced to the TOGAF Content Metamodel:

Business Capability	A particular ability that a business may possess or exchange to achieve a particular purpose.
Value Stream	A representation of an end-to-end collection of value-adding activities that create an overall result for a customer, stakeholder, or end-user.
Course of Action	Direction and focus provided by strategic goals and objectives, often to deliver the value proposition characterized in the business model.

Figure 2. TOGAF 9.2 - New Elements [11]

From the analysis of the changes in both specifications (ArchiMate 3.1 and TOGAF 9.2) we encountered new entity mappings between ArchiMate Capability element and TOGAF Business Capability element, ArchiMate Value Stream element and TOGAF Value Stream element and ArchiMate Course of Action element and TOGAF Course of Action element.

6 Objectives Analysis

In order to be able to have all models in the same repository, we present a solution that focuses on integrating the two specifications to be able to consolidate models developed with the two different metamodels, TOGAF and ArchiMate, in the same EA repository, through the Enterprise Cartography tool Atlas.

In order to achieve this objective the solution presented follows the following phases:

1. **Concepts Mapping**

To understand which mappings of entities can be used in the transformation between metamodels, in this case between TOGAF entities and ArchiMate entities, an analysis of the meaning of each one is necessary.

Following the work done in [9] and research on the new entities added in the TOGAF 9.2 specification and ArchiMate 3.1 specification already discussed in the relates work section, the result we want to achieve in this phase is the materialization of the mapping between ArchiMate entities and TOGAF entities and vice versa.

2. **Relationships Mapping**

To understand which mappings of relationships can be used in the transformation between metamodels, in this case between TOGAF entities and ArchiMate relationships, an analysis of the meaning of each one is necessary. The main challenge on this phase is the mapping between relationships that don't have a direct mapping and which need to be defined.

Following the work done in [9], the result we want to achieve in this phase is the materialization of the mapping between ArchiMate and TOGAF relationships.

3. **XSLT Transformation Approach**

Using the XSLT approach to create a metamodel transformation mechanism, it is necessary to define three files to be run in a batch and inserted in the Atlas repository. Two of the files contain the definition of the metamodels, a file with the definition of the entities that are in the specification TOGAF 9.2 and a file with the definition of the entities that are in the specification ArchiMate 3.1 need to be defined. The third file contains the mapping rules between entities and relationships from the origin model to the target model. The mapping rules defined in this XSLT file are not general to all TOGAF to ArchiMate transformations or vice versa.

Each model to be transformed must have a specific XSLT file with its mapping rules defined as the applied rules will be different from model to model. Since the

ArchiMate 3.1 specification is already available in the Atlas repository, the objectives for this phase are the definition of the XSLT file with the TOGAF 9.2 specification to be inserted in the Atlas repository and the understanding of the mapping rules that the Atlas tool offers and which ones will be used in order to achieve the model transformation without losing meaning nor coherence.

6.1 Concepts Mapping

Some of the concepts found in the two metamodels share very similar names and analyzing their description we find that they also share the same meaning, that is, they have a very strong alignment.

There are other concepts, which do not have a similar name but still maintain a good alignment, meaning that the concept describes a similar meaning. However, there are concepts that do not have this strong alignment, but that we can still find some kind of similarities.

And then there are concepts that do not have any type of mapping, that is, the concept does not exist in one of the metamodels.

Only concepts with mapping, even if weak, will be mentioned. Next we will analyse for each ArchiMate concept the TOGAF concepts that can be aligned with it.

Through the detailed semantic and comparative analysis of the entities of the respective metamodels, and after analyzing the mapping covered and presented in [9], the mapping between the entities of the TOGAF 9.2 standard and the ArchiMate 3.1 standard that will be used for the transformation between metamodels is exposed in the following images:

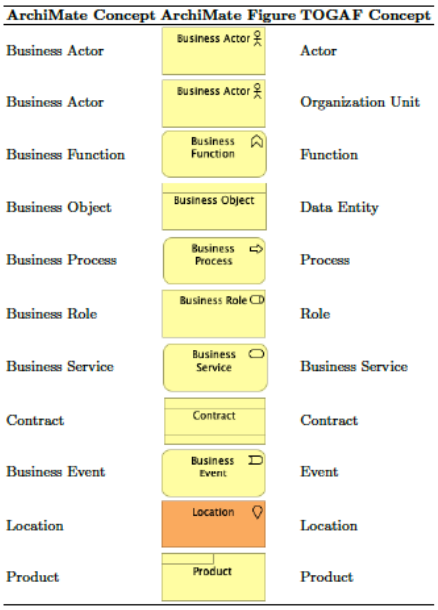


Figure 3. Concepts Mapping - Business Domain

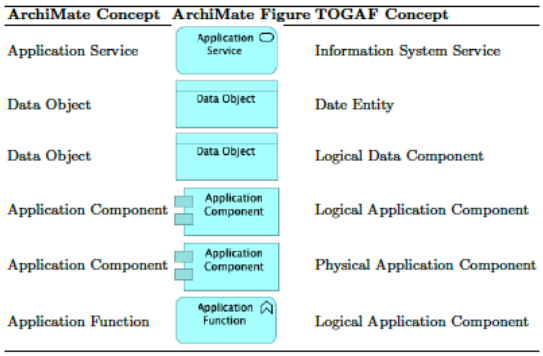


Figure 4. Concepts Mapping - Application Domain

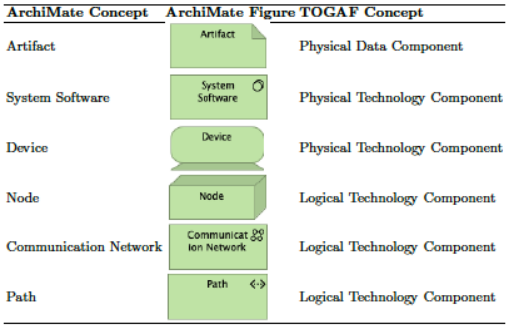


Figure 5. Concepts Mapping - Technology Domain

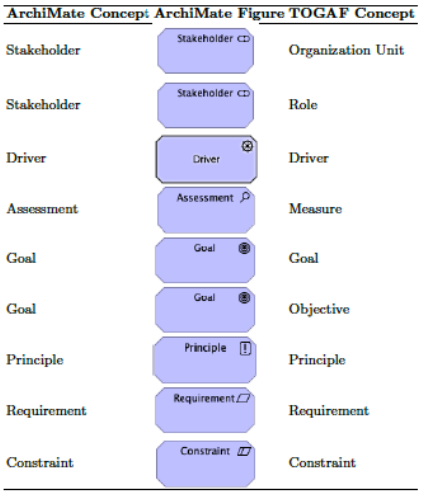


Figure 6. Concepts Mapping - Motivation Extension

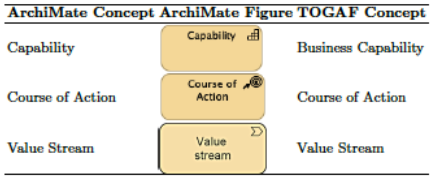


Figure 7. Concepts Mapping - Strategic Extension

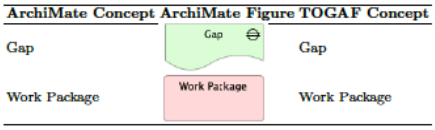


Figure 8. Concepts Mapping - Implementation and Migration Extension

6.2 Relationships Mapping

The TOGAF standard defines its metamodel with associations that are named to describe their meanings.

The relationship names are different going from one entity type to another. To understand which mapping should be done to one of ArchiMate's relationships its meaning must be decoded.

In TOGAF every element can have decomposition, this type of relationship can be seen in ArchiMate as the Composition relationship. It some ways the Aggregation relationship is also appropriate, even though the concept is not explicitly considered in the TOGAF standard, so the mapping would depend on the context.

For each TOGAF entity we will present all the other relationships mapping to ArchiMate.

In the solution proposed we categorized the relationship mapping in two groups:

– **Direct Relationships:** TOGAF relationships that have a direct mapping to one of ArchiMate's relationships.

– **Derived Relationships:** TOGAF relationships that don't have a direct mapping to one of ArchiMate's relationships and need a more complex transformation to achieve the same meaning.

A table with the examination of the relationships mapping retrieved from [9], will be presented. To note that the context of the model many times influences the type of relationship mapping.

ArchiMate Relationship	TOGAF Mapping
Composition	Decomposed
Aggregation	Could be decomposition even though it does not have the exact meaning, also the contain relationship can be used. For the product aggregation the set of relationships in the TOGAF standard can be applied (produced by, decompose, orchestrate, supports, and realized by. It is important to recall that the product concept is different in the TOGAF standard and the business and application collaboration concepts do not exist.
Assignment	Perform task in, contains, operates in and contains, resides, encapsulates, performs or is performed by, participates, involves, accesses or can be accessed. This could apply for relationships between actors, roles, and processes; however, do not apply for a business interface assigned to a business service. The same applies for the application and infrastructure interface concepts assigned to services and the application collaboration assigned to an application interaction. For business collaboration and business interaction, even though the concepts do not exist in the TOGAF standard, since a business interaction can be mapped with a process (behavior made by an aggregation of roles assigned to a behavior concept) in this case the previous mapping can also apply.
Realization	Orchestrates or decomposes/is realized by, is bounded by/provides governed interface to access, decompose or relate to. implements/is realized by, supplies or realizes. Some of the relationships could have a different meaning than the one stated as the realization so again the modeling would depend on the context.
Used By	Consumes/is provided to, owns and governs/is owned and governed by, indirect a role accesses a function which is bounded by a business service, orchestrates or decomposes/supports or is realized by, indirect, the actor performs task in a role and the role operates in a location. Some of these relationships in the TOGAF standard could have another meaning beyond the "used by" like, for example, in relationships like "bounded by" or "owns and governs".

	For the Application and Infrastructure layers the relationships between the concepts in the ArchiMate language are connected by the used by and can have relationships with different meanings such as implements/realized by, communicates with/decomposes, operates in, or supplies.
Access	Provides or consumes, is accessed and updated through, orchestrates or decomposes/supports or is realized by, provides or consumes/is accessed and updated through, is bounded by/provides governed interface to access, operates on, resides within. Similar to previous mappings, some of these relationships in the TOGAF standard could have another meaning beyond the one that is implicit in the "access" relationship like, for example, "provides", "governed interface", or "is bounded by"; for these relationships the mapping will depend on the context and might require additional explanations or comments in case the "access" relationship is used.
Association	Association in the ArchiMate language can also be used in a more specific way to relate a device with a network, which in this case would correspond with "decomposes/is dependent on" for the Technology layer mapping (physical and logic technology components). For the motivational concepts like driver and goal the mapping is done with the "creates" relationship in the TOGAF standard, for the plateau and gap, the mapping that could apply is "is delivered by" or "delivers".
Triggering	The dynamic relationship concept does not exist in the TOGAF standard; however, some other relationships can be mapped; such as: is generated by/generates, orchestrates .
Flow	Orchestrates, decomposes/supports is realized by, precedes or follows/decompose, communicate with. There is no flow relationship between these concepts in the TOGAF standard; the closest ones might be orchestrates, precedes, or follows.
Grouping	N/A
Junction	N/A
Specialization	The specialization concept does not exist explicitly in the TOGAF standard; however, a few relationships can be mapped; e.g., realizes/is realized by (a physical and technology component). If categories or taxonomies for TOGAF concepts are defined as part of the modeling effort, this can be represented using, for example, the decompose relationship.

Table 1. Relationships Mapping

6.3 XSLT Transformation Approach

As mentioned above, with the Atlas tool it is possible to have different models coming from different EA tools being consistent with each other in a single place, using the transformation model based on XSLT technology. [5]

The model transformation mechanism is based on two steps.

The first step deals with the configuration of the TOGAF and ArchiMte metamodels in the Atlas repository. For this, a specific file must be defined with the entities and attributes of each standard.

The ArchiMate 3.1 metamodel was already published in the Atlas repository, so for the solution we only had to defined the file for the TOGAF 9.2 metamodel entities.

The following images shows the way the XSLT file for the TOGAF 9.2 metamodel entities has been defined and the entities created in the Atlas repository after the XSLT file was processed.

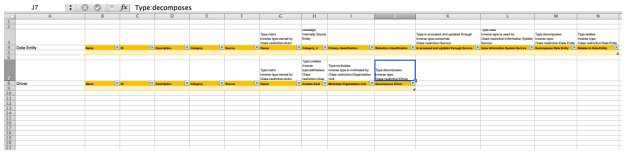


Figure 9.XSLT TOGAF Entities Definition File (1)

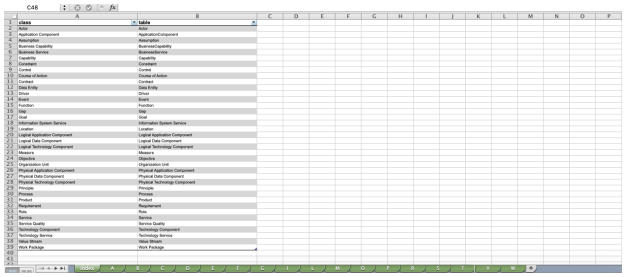


Figure 10.XSLT TOGAF Entities Definition File (2)

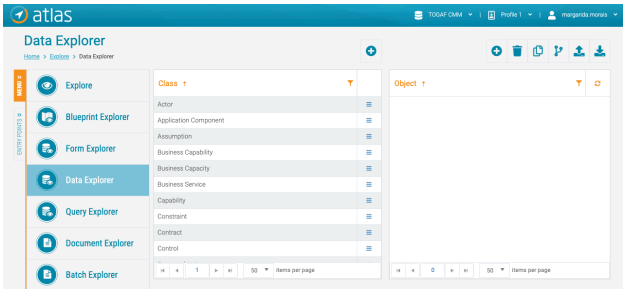


Figure 11.Atlas Repository - TOGAF Entities Imported

Atlas uses an high level type-based rules that operates on types, instances and relationships. To configure such rules, a XSLT file is defined with the transformation rules, that transforms the source model and objects into target model and objects. The structure of the relationships transformation file is based on a matrix where the Origin Metamodel and Target Metamodel entities are defined as well as the rules that must be applied to them.

7 Demonstration

Next, an ArchiMate model example will be presented that will be used to demonstrate the transformation between the metamodels of the TOGAF and ArchiMate standards.

The example used represents a very simple view of the enterprise architecture of an insurance company that offers claims services. This insurance company offers its customers a registration service, a service for attaching claim information and a payment service.

These services are supported by a process of handling claims and its distinct functions as well as by an application and technological layer.

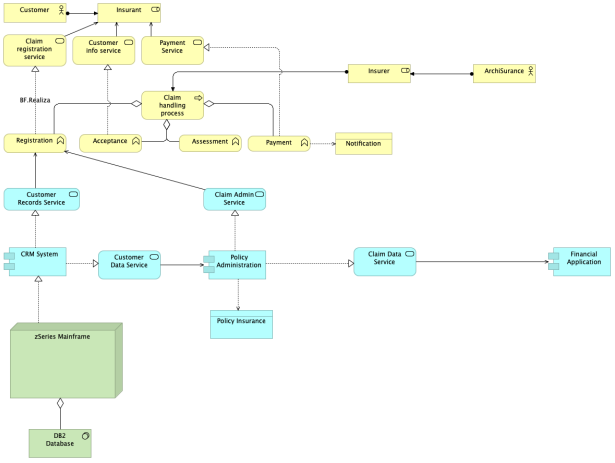


Figure 12.ArchiMate - Insurance Company Example

When transforming a model between TOGAF and ArchiMate standards, and vice-versa, many considerations must be taken into account, the context influences immensely what type of relationships will be used when transforming the model.

Following the concepts and relationships mapping above it is necessary understand how can we, using the Atlas rules, transform the example model into a TOGAF model.

Below the TOGAF model we want to achieve after the transformation of the ArchiMate model presented.

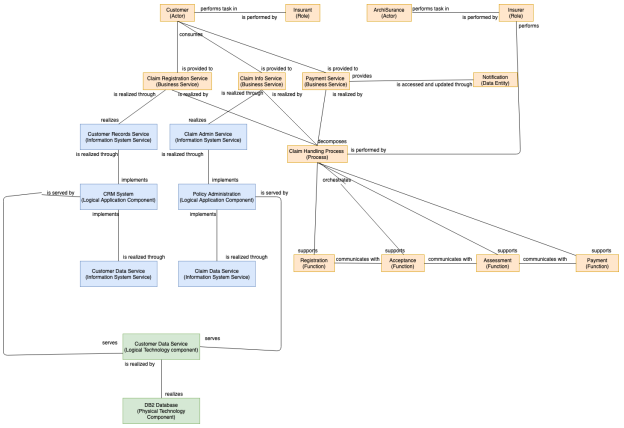


Figure 13. TOGAF - Insurance Company Example

Following the Atlas transformation rules the model transformation file must be defined. In the XSLT file we can define mappings based on 3 types of transformation rules:

Firstly the **Delete Data Type Rule** is applied to all the ArchiMate entities that do not incorporate the model.

Also the **Rename Data Type Rule** is applied to the ArchiMate entities that appear in the model so that the transformed model will have the description of the entities of the TOGAF metamodel.

These types of transformation rules only use the source class as an argument to convert the reference.

Finalizing step one of processing class rules, the rules applied next are rules based on source class, relationship type and target class.

In the example model we apply this type of rules in the following relationships:

- Business Actor and Business Role
- Business Actor and Business Role
- Business Role and Business Process
- Business Process and Business Function
- Business Function and Business Function
- Business Function and Business Service
- Application Component and Application Service
- Business Service and Business Role
- Node and Application Component
- Node and System Software
- Business Function and Business Object
- Application Component and Data Object
- Application Service and Business Function

The **Filter Property Rule** is applied to teh relationships mentioned above.

Due to the fact that there are some class-to-class relationships in ArchiMate that do not have direct mapping in the TOGAF metamodel, it is necessary to make additional rules to be able to recreate the same meaning.

In the ArchiMate model example, it is the Business Role that has a relationship with the Business Service, since this relationship it is not known in the TOGAF metamodel a rule must be define to create a derived relationship that will propagate the same meaning.

For this a **Copy To Rule** is defined so that in the TOGAF model transformation is the Actor that will have this property.

There is also the possibility to define in the file another type of rules that,converts the reference, based on the source class and relationship type.

This type of rules are applied to classes of origin that regardless of the class of destination,for a specific type of relationship, always have the same mapping.

In this specific example we did not defined any rules for this type of transformation.

Below images of the XSLT transformation file defined for the exemple are presented.

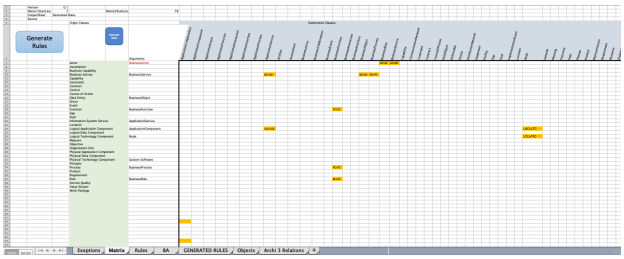
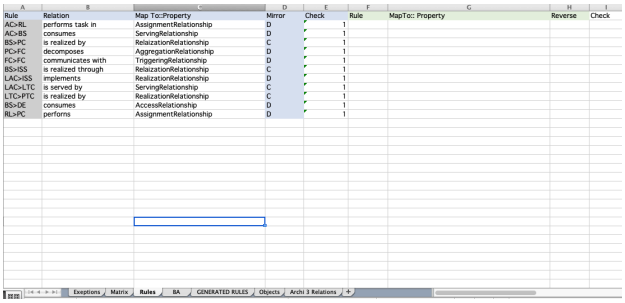


Figure 14. XSLT Transformation File Example (1)



Rule	Relation	Map To::Property	Mirror	Check	Rule	MapTo:: Property	Reverse	Check
AC>RL	performs task in	AssignmentRelationship	D	✓	1			
AC>BS	consumes	ServingRelationship	D	✓	1			
BS>PC	is realized by	RelazationRelationship	C	✓	1			
PC>FC	decomposes	AggregationRelationship	D	✓	1			
FC>FC	communicates with	TriggeringRelationship	D	✓	1			
BS>ISS	is realized through	RelazationRelationship	C	✓	1			
LAC>ISS	implements	RealizationRelationship	D	✓	1			
LAC>LTC	is served by	ServingRelationship	C	✓	1			
LTC>PTC	is realized by	RealizationRelationship	C	✓	1			
BS>DE	consumes	AccessRelationship	D	✓	1			
RL>PC	performs	AssignmentRelationship	D	✓	1			

Figure 15.XSLT Transformation File Example (2)

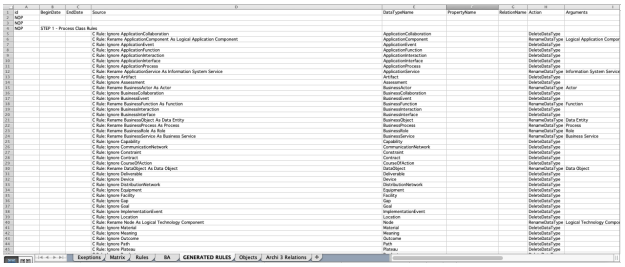


Figure 16.XSLT Transformation File Example (3)

Having all the rules defined in the XSLT file, all that remains is to run the batchin the Atlas tool, which is responsible for integrating the new model transformedinto the Atlas repository.

8 Conclusion and Future Work

In this section, the conclusions of this research will be presented, as well as a brief summary of the possible future work that can be further developed in the scope of this work.

8.1 Conclusion

Through this report we examine the importance of an EA solution in organizations, as a form of adaptation to the constant and rapid change within the organization and the need to establish a defined process to manage and maintain the EA repository.

To be able to benefit from the EA, we verified the need to have a consolidated EA repository, which is certainly a challenge within enterprises where thee planning process originates in many of the enterprise's communities without a consistent, coherent and complete systemic view of the enterprise to support them, individually and as a whole as well as the struggle to align the different elements designed by different architects resulting into an unconscious design.

We look to EC discipline, what characterizes it and how it can be used as a tool to assist the process of managing, maintaining and integrating tools with the EA repository, as suggested in practical cases [2].

Having studied the framework and language, widely used in the development of EA, the TOGAF framework and the ArchiMate language, we found that there is space to create a solution for the integration of models based on the two specifications.

This research goes through the analysis of the work already developed in the harmonization of the two metamodels in question and in the creation, using the EC Atlas tool, of a semi-automatic mechanism for the

transformation between ArchiMate and TOGAF models and vice-versa.

From the research done in this work we can conclude that, through the Atlas tool, it is possible to achieve the model transformation using a well defined XSLT file where are presented the mappings based on 3 types of transformation functions whose arguments are:

- Convert Reference based on source class only
- Convert Reference based on source class and type of reference
- Convert Reference based on source class, reference type and target class.

As seen in the Demonstration Section, for the example presented, there are some cases where derived relationships are needed for the transformation and additional rules must be defined.

As for the reversibility of the conversions, the XSLT file must be modified so that the source and destination model are exchanged, that is, the transformation rules applied will have to be implemented according to the direction of the transformation, but the mapping between the two specifications, both entities and relationships, presented in the Relationships Mapping remains the same.

As seen in the Concepts Mapping Section there are entities between both specifications that do not have any type of mapping. In these cases where the model to transform has entities that do not have any mapping in the target model, the entities would not appear in the model transformed.

As seen in the Demonstration Section in the rules created to transform the model example, the entities that do not incorporate the model to be transform will suffer a Delete Data Type Transformation Rule allowing to delete a Data Type from the initial model. Deleting a Data Type also deletes corresponding Data Instances, and therefore also deletes references to deleted instances.

Being the result presented, a simple example of a way to consolidate models based on the two metamodels in a single repository.

8.2 Future Work

Given the capacity found in the Atlas tool for integrating models from different tools, there is a lot of work that can be done in this field and in the integration between EA tools metamodels.

Within the work carried out for the elaboration of this research it would be a good practice to have several models and views to convert that encompass all entities with mapping between the two metamodels (TOGAF and ArchiMate), to analyze and test the transformation and the use of the solution in a specific use case to verify its usability and versatility.

It would also be interesting to study the creation of an XSLT transformation file, for each of the transformation directions, TOGAF to ArchiMate and ArchiMate to TOGAF, with defined set of generalized rules that could be used as a basis for the transformation of several models, instead of having to create a transform file from scratch every time there is the need to transform a specific model / view.

9 Referências

[1] Roeleven, S. (2010). Why Two Thirds of Enterprise Architecture Projects Fail. Business White Paper

[2] Sousa P.,Tribolet J., & Guerreiro S., Enterprise Cartography: From Theory to Practice

[3] Tribolet, J., Sousa, P., & Caetano, A. (2014). The Role of Enterprise Governance and Cartography in Enterprise Engineering. Enterprise Modelling and Information Systems Architectures, 9(1), 38-49.

[4] Sousa, P., & Carvalho, M. (2018). Dynamic Organizations Representation. Linking Project Management with Enterprise Architecture. 2018 IEEE 20th Conference on Business Informatics (CBI).

[5] W3C.(1999). XSL Transformations (XSLT).

[6] Kappel, G., Kapsammer, E., Kargl, H., Kramler, G., Reiter, T., Retschitzegger, W., . . . Wimmer, M. (2006). Lifting Metamodels to Ontologies: A Step to the Semantic Integration of Modeling Languages. Model Driven Engineering Languages and Systems Lecture Notes in Computer Science.

[7] Kapsammer, E., Reiter, T., & Schwinger W. (2006). Model-Based Tool Integration-State of the Art and Future Perspectives.

[8] Li, D., Li, X., & Stolz, V. (2011). QVT-based model transformation using XSLT. ACM SIGSOFT Software Engineering Notes, 36(1).

[9] Estrem, W., Gonzalez, S. (2014, December). TOGAF® Framework and ArchiMate® Modeling Language Harmonization Content Metamodel Harmonization: Entitles and Relationships

[10] (n.d.). Retrieved from https://pubs.opengroup.org/architecture/archimate3-doc/apdxe.html

[11] (n.d.). Retrieved from https://pubs.opengroup.org/architecture/togaf9-doc/arch/

[12] The Open Group. (2011). TOGAF Version 9.1

[13] Lankhorst, M. (2013). Enterprise architecture at work: Modelling, communication and analysis.

Berlin: Springer.