Uncovering the organisational modelling and business process modelling languages in the ARIS method

Article *in* International Journal of Business Process Integration and Management · May 2011

DOI: 10.1504/IJBPIM.2011.040205

3 authors, including:

Paulo Sérgio Santos Júnior
Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo…

João Paulo A. Almeida
Universidade Federal do Espírito Santo

Uncovering the organisational modelling and business process modelling languages in the ARIS method

Paulo S. Santos Jr.*, João Paulo A. Almeida and Thiago Lavarezi Pianissolla

Ontology and Conceptual Modelling Research Group (NEMO),
Computer Science Department,
Federal University of Espírito Santo (UFES),
Av. Fernando Ferrari, s/n, 29060-970 Vitória,
Espírito Santo, Brazil
E-mail: paulossjunior@inf.ufes.br
E-mail: jpalmeida@ieee.org
E-mail: tlpianissolla@inf.ufes.br
*Corresponding author

Abstract: In this paper, we propose an approach to excavate and define the metamodels of the organisational modelling and business process modelling languages of ARIS method. This approach uses information obtained with user interactions over the modelling environment called ARIS toolset and extra information from tool documentation. The application of this approach results on well-defined language metamodels, clarifying the language's main modelling elements and their relationships. The metamodels serve as a starting point for the definition of the semantics of the language and allow the construction of tools to manage modelling, simulation, analysis and transformation of organisational models and business processes. To validate the metamodels we define a set of transformations which enables one to create instances of the metamodels using as a starting point models in the ARIS toolset serialisation format (the ARIS markup language-AML).

Keywords: business process modelling; ARIS method; enterprise architecture modelling; metamodelling.

Reference to this paper should be made as follows: Santos Jr., P.S., Almeida, J.P.A. and Pianissola, T.L. (2011) 'Uncovering the organisational modelling and business process modelling languages in the ARIS method', *Int. J. Business Process Integration and Management*, Vol. 5, No. 2, pp.130–143.

Biographical notes: Paulo S. Santos Jr. holds an MSc in Computer Science from the Federal University of Espírito Santo, Brazil. This paper is one of the results of his MSc thesis, in which he has studied model-driven approaches for enterprise modelling. Currently, he collaborates with the Ontology and Conceptual Modeling Research Group (NEMO), which is part of the Computer Science Department of the Federal University of Espírito Santo, Brazil. His main research interests include: conceptual modelling, model-driven design, enterprise architecture and business process modelling.

João Paulo A. Almeida is an Associate Professor at the Federal University of Espírito Santo and a Senior Member of the Ontology and Conceptual Modeling Research Group (NEMO). He holds a PhD in Computer Science from the University of Twente, The Netherlands. In 2006, he worked as a Scientific Researcher for the Telematics Institute (now Novay) on the application of model-driven approaches to the design of services and service-oriented architectures. His current research areas include: model-driven design, service-oriented design, enterprise architecture, conceptual modelling and business process modelling. He currently serves in the steering committee of the IEEE EDOC Conference.

Thiago L. Pianissolla graduated with a degree in Computer Science from the Federal University of Espírito Santo in 2008 with a final assignment on use of semantic web service to support the identification of human resources based on their competence. He is a former member of the Ontology and Conceptual Modeling Research Group (NEMO), which is part of the Computer Science Department of the Federal University of Espírito Santo, Brazil. His main research interests include conceptual modelling and enterprise architecture.

This paper is a revised and expanded version of a paper entitled 'Escavando as linguagens de modelagem organizacional e modelagem de processos do ARIS method (in Portuguese)' presented at 2nd Workshop on Business Process Management, Vila Velha, ES, Brazil, October 2008.

1 Introduction

Organisational management is a task which involves a significant level of complexity, since it aggregates several perspectives and domains of knowledge [such as business processes (Van der Aalst et al., 2004), information technologies and infrastructure]. This variety of domains of knowledge and perspectives involves several factors which can potentially provide conflicting quality criteria which affect the performance of an organisation. According to Jonkers et al. (2004), this is an important problem because a change in the strategy and in the goals of the organisation can bring about significant modifications in several areas of the organisation.

An analysis of how these factors are interconnected and how they should be prioritised becomes possible with the application of the concept of enterprise architecture. An enterprise architecture describes an organisation's structure, its features and functions, as well as the dynamic interaction between the several components that constitute it. According to Lankhorst (2005), an enterprise architecture "is a complete and coherent set of principles, methods and models that are used in the design and development of an organization's structure, business process, information technology and infrastructure."

A complete and consistent enterprise architecture is the key to risk reduction and management of the several domains of knowledge of the organisation, because it drives the application and evaluation of new investments (new information technologies, changes in the infrastructure and new business processes, among others) and their impact on the organisation's goals and components (DoDAF – US Department of Defense, 2007).

The enterprise architecture concept has attracted significant academic and governmental attention, with many efforts resulting in enterprise architecture frameworks, including the Zachman framework (Zachman, 1987), The Open Group Architecture Framework (TOGAF, 2007); Department of Defense Architecture Framework (DoDAF) (DoDAF – US Department of Defense (2007), ISO's reference model for open distributed processing (RM-ODP) (RM-ODP-ISO-ISO/ITU-T (1995), architecture of integrated information systems (ARIS) (Scheer, 1999) and Archimate (Archimate Consortium).

The use of frameworks not only allows systematic structuring of an organisation's knowledge into enterprise architecture descriptions, but also provides tools for enterprise architecture analysis and facilitate the identification of the stakeholders' needs for information system development.

This work is especially interested in the ARIS framework (Scheer, 1999) due to its importance in several industrial contexts. The ARIS framework enables a modeller to construct a general view of the whole organisation and also supports information system development projects (Lankhorst, 2005).

The importance of EPCs in the practice of enterprise architecture modelling can be attested by the existence of several successful commercial tools which offer support for these diagrams, e.g., IDS Scheer's ARIS toolset, Microsoft's Visio and BOC's ADONIS. Further, the business process modelling notation of the ARIS method have been used in the documentation of the widely-employed SAP R/3 enterprise resource planning system, which has led to the SAP reference model with over 600 business process models (Mendling, 2009).

The ARIS framework has different views (organisational, data, control, function and output) and abstraction layers (Eclipse Foundation, 2008b). In this article, we are interested in the organisational and control views. The organisational view describes the hierarchy of organisation, i.e., the communication and relationships between organisational units as well as the roles played by individuals. The control view describes the processes which transform information through a function or a set of functions. Since these functions represent potentially complex organisational tasks, the control view is used for modelling business processes into ARIS.

Each view of ARIS framework has its own language, which can be defined through its syntax and semantics. Syntax focuses purely on the language's notation aspect, ignoring completely the meaning of the syntactic elements, which are revealed only through the semantics of the language. The semantic definition of the language L, or simply its semantics, is composed of two parts: a semantic domain and a semantic mapping between the syntax elements and a semantic domain (Hsi, 2005).

Scheer has introduced the languages that represent each view of the ARIS method in Scheer (1999). However, there are divergences between the original definition of the language and its current use in the ARIS toolset. As a consequence, there are elements in Scheer (1999) which are not present in the modelling tools and vice-versa. Further, the semantics of the ARIS languages is not rigorously defined in Scheer (1999) nor in subsequent efforts, with the exception of the subsets of the ARIS EPC process modelling notation in Mendling et al. (2005a) and Mendling et al. (2005b).

With this deficiency in the language definition, any study, analysis and construction of tools based on the metamodels in Scheer (1999) becomes problematic, since this source no longer represents the actual set of syntactic and semantic elements of the language as used in the industry nowadays. In fact, the definitive source for the study of the syntactic elements of the ARIS language is the language's support by the ARIS toolset given its widespread usage and acceptance.

This paper has the objective of uncovering and defining the organisational and business process modelling languages of the ARIS method. The result of our effort is a metamodel that allows us to manipulate ARIS domain models in a model-driven approach. Further, the availability of a metamodel for the ARIS method allows us to identify the semantic elements of the organisational and business process domains of the ARIS Method.

In Section 2, we present a simple approach for excavation and definition of the domain languages of the ARIS toolset, which allows us obtain the semantic elements of ARIS method. We focus on the organisational domain (organisational view) and the business process domain (functional view). In Section 3, we present the validation of the domain languages introduced in Section 2, through MDD transformation. Section 4 elaborates on some final consideration of this article.

2 Excavating the organisational and business process modelling languages of ARIS method

The approach proposed in this section has the objective of excavating and defining a domain-specific language, identifying the notational elements (or concrete syntax), the elements that constitute the abstract syntax (as captured in a metamodel), the semantics of each element which constitute the metamodel and the relationships among the elements of the metamodel.

In order to perform excavation and definition of the domain-specific language of the ARIS method as implemented in ARIS toolset, the following the steps are proposed:

1 To identify the model type (diagram) used by the ARIS toolset to represent the selected domain.

2 To identify the main elements of a particular model type. In this step we distinguish between the elements which are purely notational (i.e., which belong to the concrete syntax) and the elements which are inherent to the abstract syntax of the domain language (i.e., which are the subject of a semantic mapping).

3 To survey for information into tools and other sources which enable us to identify the semantics of the elements selected at Step (2).

4 To interact with the modelling tool to exercise the modelling elements selected at Step (2) in order to uncover the potential relationships between these elements.

5 To build the domain metamodel with the information obtained in Steps (1), (3) and (4).

6 To complete the metamodel with the specialisation of selected elements of Step (2).

The next section will introduce the definition of a fragment of the organisational and business process domain languages as implemented into ARIS toolset iterating through Steps 1–6 of approach.

2.1 Excavation and building of the organisational metamodel of ARIS method

2.1.1 Identifying the model of the organisational domain

The ARIS toolset uses the *organisational chart* model to represent the organisational units and its inter-relationships with other organisational components. The modelling environment makes available the following elements for this model: organisational unit type, organisational unit, cost centre, position type, position, system organisational unit type, system organisational unit, person type, internal person, external person, group, location, workstation e-position description.

Table 1 Object type and symbol type of organisational chart

Object type	*Symbol type*
Organisational unit type	Organisational unit type
	Position type
Organisational unit	Organisational unit
	Cost centre
Position	Position
System organisational unit type	System organisational unit type
System organisational unit	System organisational unit
Location	Location
	Workstation
Person	Internal person
	External person
Person type	Person type
	Position description
Group	Group

2.1.2 Identifying the main elements of the domain model

The ARIS toolset differentiates the elements above into two categories:

1 object types

2 symbol types.

The first category represents an abstract type of element. The second category represents the possible notations (shapes) of an object type into tools. Consequently, an element of a specific object type can be graphically represented through several symbols and thus may assume different meanings according to this symbol. For instance, the object type person can represent an internal person in an organisation (internal person) or an external person (external person) simply by changing its symbol type. We can conclude that the object type is specialised through an association of an object type with a symbol type. Table 1 shows the symbol types which are associated with the object type into an organisational chart diagram.

Considering the object type as candidates for metaclasses, we have identified the following metaclasses of the organisational model of ARIS method: organisational unit type, organisational unit, position, system organisational unit type, system organisational unit, location, person, person type e-group. The specialisation of theses metaclasses (symbol type) will be incorporated in the metamodel only after the identification of the semantics of the main elements and their relationships (since their relationships are inherited directly from the main elements).

2.1.3 Identifying the semantics of selected elements

Once the main modelling elements are identified, the semantics of each element must be described. The online documentation of the ARIS tool (help) has been used as reference in this step.

An organisational chart is intended to describe the organisational units (organisational unit) and their relationships with the other organisational components. An organisational unit is an entity which is responsible for performing the activities which achieve the organisational goals.

The relationships among the organisational units, as well as the relations of superiority and subordination, are represented in the model in the form of relationships (lines that join two or more modelling elements in the ARIS toolset) which can assume the following meanings:

1 an organisational unit is technically superior to other organisational unit (*is technically superior to*)

2 an organisational unit is disciplinary superior to other organisational unit (*is disciplinary superior to*)

3 an organisational unit is a component of the other organisational unit (is a component of).

The documentation of the ARIS toolset defines that the smallest organisational unit is represented by a position (position) which an individual (person) can assume in the organisation. According to Scheer (1999), positions can be allocated according to the size of the units, the business rules and the organisational structure. Several positions can be associated with an organisational unit. The responsibilities and obligations of some position (position) are defined in the position description.

The association of a person (person) and an organisational unit represents that this person is an employee of the organisational unit and the association of a person with a position (position) defines the status of this person in the organisation (the functions and responsibilities).

The organisational units (organisational unit) are grouped based on similar features, e.g., based on responsibilities and obligations. The same criteria can be used for creating groups of persons (person), for instance, a set of employees which have the same abilities. Thus, the ARIS method allows one to create organisational unit types (organisational unit type) such as departments and person types (person type) such as department manager, team leader or project manager.

Using classes which represent types (for example, an organisational unit type or a person type) opens the possibility of representing generic business rules which are applied to these classes. Then, this enables one to specify business rules which are applied to all the instances of these classes. For example, we can state that only a specific person type (person type) is allowed to perform some function or to have access to some specific information.

The group element represents a group of employees (person) – with or without a position in the organisation – or organisational units which perform activities together for a given period of time in order to achieve a common goal.

Location is a modelling element of ARIS that is aimed at representing the geographic positions of an organisational unit (organisational unit), a person (person), a position (position) or an organisational resource. A location can represent a place, a city, a building, a room, workstation or a plant or factory floor.

2.1.4 Identifying the selected relationships and building the metamodel

The fourth step of our approach seeks to identify the relationships among the identified elements in the second step. By identifying the relationships among the elements of the model, it is possible to identify concepts which have not been properly explained or even not discussed in the documentation of the tool.

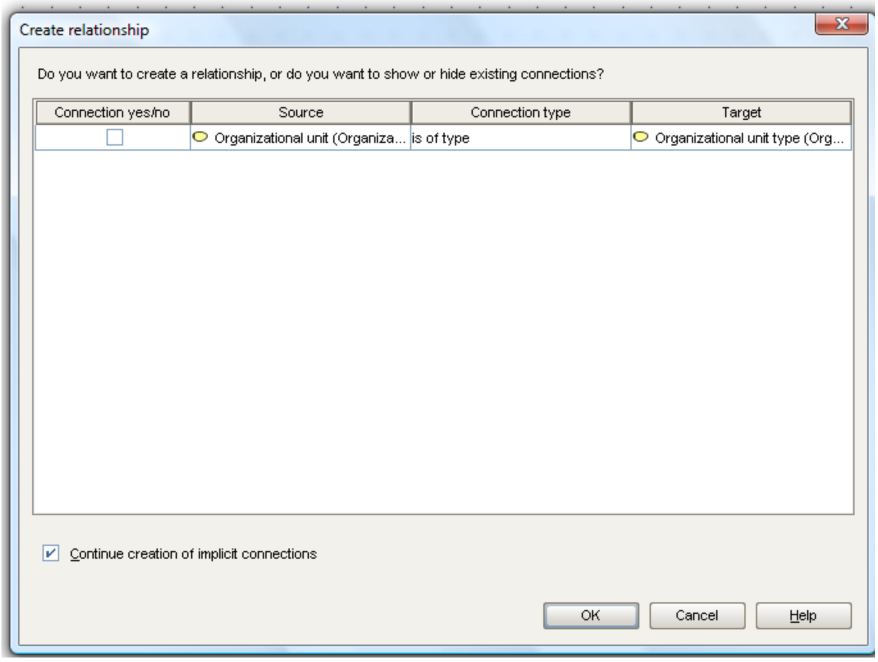
The excavation of the relationships is done through an analysis of the information displayed as a result of interaction between the user and the tools, during the modelling activity. Figure 1 presents how the tool deals with the relationships between two elements.

When we try to create a relationship between two elements of different types or between two elements of same type into the modelling environment (depicted in Figure 1), the modelling tool displays the relationships between these elements.

The modelling environment of ARIS toolset (on the right side of Figure 1) specifies the following pattern to identify the source element, the relationship name and the target element of the relationship: <SOURCE ELEMENT> – <RELATIOSHIP NAME> – <TARGET ELEMENT>. From this pattern, we are able to extract the information of relationship shown in Figure 1:

- source element: organisational unit;
- relationship name: *is of type*;
- target element: organisational unit type.

Figure 1 Uncovering possible relationships between two elements through interaction with the ARIS tools (see online version for colours)



In this work, we have developed a systematic approach for excavating the relationships between the elements in the abstract syntax of the domain-specific language:

1 the auto-relationships of the elements are identified, such as the relationships among the organisational unit elements

2 the relationships among the elements of different types are also identified, such as the relations between organisational unit elements and organisational unit type elements.

To exemplify our approach, we apply the approach at the following elements: organisational unit and position.

We were able to identify that the ARIS toolset has the following concepts implied by the relationships excavated:

1 an organisational unit may be composed (*is composed of*) by the other organisational units

2 an organisational unit may have a hierarchy relationship of superiority (*is superior*) with the other organisational units

3 an organisational unit may be technically superior to other organisational units (*is technical superior to*) or disciplinary superior (*is disciplinary superior to*)

4 an organisational unit may be responsible (*is responsible for*) for one or more organisational units

5 an organisational unit (organisational unit) is subordinated (*is subordinate*) to the other organisational units.

The following auto-relationships of the position have been excavated:

1 a position can be a substitute for (*substitutes for*) one position or more than one position

2 a position is technically superior to (*is technical superior to*) or is disciplinary superior to (*is disciplinary superior to*) zero, one or many positions

3 a position is responsible for an organisational manager (*is organisation manager for*) for zero, one or many positions.

According to the previously introduced semantics, a position (position) must have all the auto-relationships of an organisational unit (since, by definition, a position is the minor organisational unit). However, the excavation of the auto-relationships of the position elements has revealed that these relations are not the same auto-relationships of the organisational unit elements. Thus there is no generalisation relationship between the organisational unit metaclass and the position metaclass, somewhat contradicting both the tool documentation and Scheer's book (Scheer, 1999).

The second step in the identification of the relationships is performed between elements of different types, such as the excavation of the relationships between a position element and an organisational unit element.

The excavation of relationships between a position and an organisational unit has revealed the following relationships:

1 An organisational unit can be technically superior to or disciplinary superior to one or more positions. The inverse, i.e., a position (position) can be technically superior (*is technical superior to*) or disciplinary superior (*is disciplinary superior to*) to one or more organisational units also applies.

2 An organisational unit is composed of zero or more positions (position).

3 A person with a specific position in the organisational can manage one or more organisational units (*is organisation manager for*).

The next step is concerned about identifying the cardinality of each relationship. For each element (the source element), we have found necessary to test its relationships with all the other remaining elements (target elements) which pertain to the organisational model. This testing step is necessary to identify the cardinality of the relationships between the source elements and the target elements which are permitted by the tool. As an example, let us consider the cardinality of the *is support* relationship and the *is composed of* relationship. The *is superior* relationship [shown in Figure 2(a)], represents the relationship that an organisational unit (organisational unit) can have with other organisational units (in the metamodel). As shown, the ARIS toolset allows this element to have zero or more elements of this type which implies a cardinality of 0..* for this relationship in the metamodel. Figure 2(b), shows the *is composed of* relationship between an organisational unit (organisational unit) and a position (position). This relationship has 0..* cardinality.

Figure 2 Example of relationships between elements (see online version for colours)

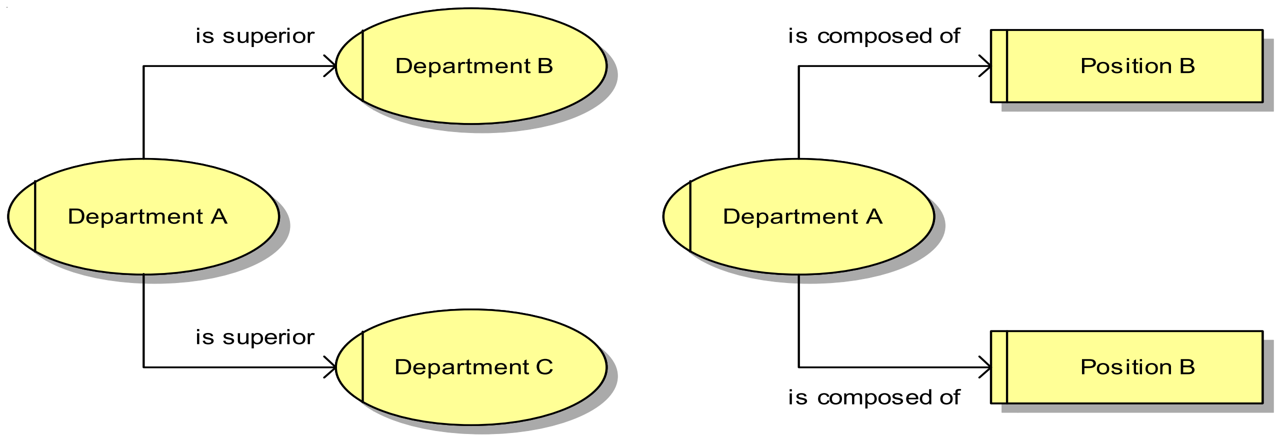


Figure 3 Fragment of organisational domain language (see online version for colours)

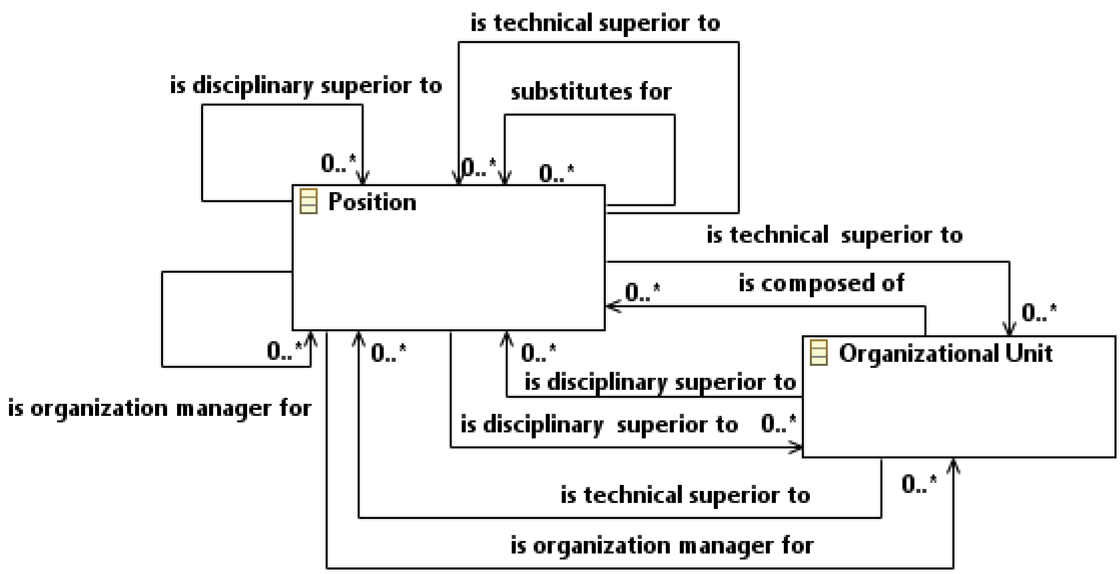
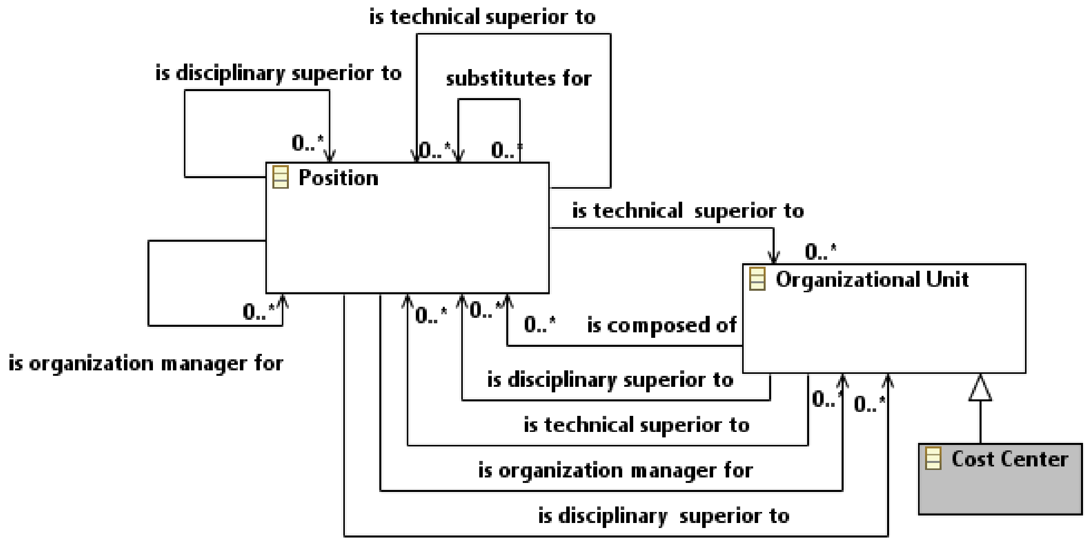


Figure 4 Fragment of organisational domain language with notation element (see online version for colours)



Considering the relationships excavated and the concepts introduced, we now are able to build a metamodel which represents the abstract syntax of the organisational domain language of ARIS tools. Figure 3 shows a fragment of this metamodel which has been excavated using our approach. The metamodel has been defined in the eclipse modelling framework (EMF).

2.1.5 Adding notational elements in the metamodel

The sixth (and last step) for building our organisational metamodel of ARIS consists of adding the elements which were considered notational elements at the second step. Figure 4 shows a metamodel fragment obtained after the application of the sixth step.

The position element does not possess any associated notational element and the organisational unit element has only the notational element cost centre (in gray). Thus, the cost centre element has been added in the metamodel as a specialisation of the organisational unit class.

The result of this last step is a fragment of the metamodel for the language which represents the organisational domain of ARIS method.

2.2 Excavation and building of the business process metamodel of ARIS method

We have applied the same steps used to build the ARIS method business process modelling metamodel. To avoid repetition, the following sections illustrate other aspects of the definition of metamodels in our approach, such as the need for refactoring the metamodel and adding abstract classes to improve the representation of the elements.

2.2.1 Identifying the business process domain model

The ARIS toolset utilises an extended event-driven process chain (eEPC) model to represent the organisational processes of an organisation. The environment for modelling provides several modelling elements for this model. To demonstrate the use of the proposed approach, we have selected a fragment of this set. The fragment has been divided in the following sets:

- *modelling elements to represent the participation of organisational elements into a business process:* organisational unit type, organisational unit, cost centre, position type, position, system organisational unit type, system organisational unit, person type, internal person, external person, group, location, workstation, position description and employee variable
- *modelling elements to describe the transformation of information and the execution of tasks as part of a business process*: function and process interface
- *modelling elements to enable the composition of tasks into complex processes*: AND rule, XOR rule, OR rule, rule and gateway
- *modelling elements to represent events:* event, start event, intermediate event e-end event.

2.2.2 Identifying the main elements of domain model

To identify the main elements of the model, we have performed the same steps as followed for the identification of the organisational metamodel. Table 2 shows the object types and symbol types of an eEPC diagram.

The employee variable is the unique element of organisational set has occurrence only in eEPC model. The others organisational elements (organisational unit type, organisational unit, cost centre, position type, etc.) occur in both organisational and business process domain (and are thus omitted from Table 2). Only some symbol types are shown for the sake of brevity.

Table 2 Object and symbol types of an eEPC

Set	*Object type*	*Symbol type*
Organisational	Employee variable	Employee variable
Transformation of information	Function	Function
		Process interface
Rules	Rule	AND rule
		XOR rule
		OR rule
Events	Event	Event
		Start event
		Intermediate event
		End event

2.2.3 Identifying the semantics of selected elements

An event represents a state which is relevant to the process management and that somehow affects the flow of execution. In other words, events are said to establish the preconditions and postconditions for each stage of the process. Preconditions represent a state of reality which triggers one or more tasks, while postconditions, in their turn, represent a state of reality that exists only after the task has been performed. In ARIS, events may be the result of tasks in a business process or may be created by actors external to the process.

The function metaclass is a basic element for EPC process modelling. According to the ARIS toolset online documentation, the function element represents either a technical task or a task performed on some object with the purpose of achieving one or more business goals. A task can be performed by either a person or an application system, and has inputs – such as information or raw material – and outputs, such as new information or products. Furthermore, tasks can consume and create organisational resources during their execution.

The rule element controls the flow of the process model on the basis of the results and effects of its preceding tasks.

An employee variable represents a placeholder for a person whose involvement in the process can be identified (although the specific person is not yet identified). This placeholder is to be filled in at a later stage by a specific person.

2.2.4 Identifying the selected relationships and building the metamodel

When we performed the excavation of the relationships between the Function element and the elements of the organisational set, we have perceived that, except for the system organisational unit element, the system organisational unit type and the location elements, all elements of the organisational set have the same relationships with the function element. Therefore, we have created participant abstract metaclass to simplify the metamodel.

According to the ARIS toolset, a system organisational unit element can be assigned to (*is assigned to*) to one or more tasks of one or more processes. In the same way, the system organisational unit type element can also be assigned (*can be assigned to*) to one or more tasks of one or more processes.

The ARIS toolset also determines that a task (function) is performed at location, as a result of this, there is a *is carried out* relationship between a location and a function.

The excavation of the relationships between the participant element and the function element has identified the following relationships:

1 a participant is technically responsible for (*is technically responsible for*) a task

2 a participant carries out a task (carries out)

3 a participant decides on task (*decides on*)

4 a participant contributes to the realisation of a task (*contributes to*)

5 a participant must be informed about the execution of a task (*must be informed about*)

6 a participant agrees or accepts (*accepts*) to realise a task

7 a participant has a consulting role in a task (*has consulting role in*)

8 a participant must be informed about the cancellation of a task (*must be informed on cancellation*)

9 a participant must be informed about the result of a task (*must inform about result of).*

The *is predecessor of* auto-relationship of the function element is used to indicate that a particular task precedes another task in the business process.

The *activates* and *creates* meta-associations between the elements function and event are used to indicate, respectively, that a task is triggered by one or more events and that one or more tasks can trigger one or more events.

The *activates* meta-association between the elements rule and function specifies which functions are activated by the rule. The *leads* to meta-association represents the connection between the task which precedes the rule (rule) and the events (event) which are created by the task. The meta-association *links* of the rule element is used to specify behavioural rules of the business process of higher complexity, i.e., rules which can be constructed by the combination of more elementary rules.

2.2.5 Building of a process metamodel fragment

Figure 5 shows the fragment of the resulting metamodel for the business process language, revealing organisational elements and their relationships with the function element. Figure 6 shows the fragment of the business process metamodel with the following elements: functions, event and rule and their relationships.

2.2.6 Adding notational elements in the business process metamodel

Figure 7 shows an example of the result obtained in the execution of the last step of the approach. We have added only the notational elements (in gray) which correspond to the set of the organisational elements which pertain to the business process domain of the ARIS method.

2.3 Discussion of the excavated models

Several techniques for conceptual models building based on extraction of information have been developed. Most of them are aimed at extracting software components, e.g., database (Volz et al., 2002), XML schemas (Ferdinand et al., 2004) and graphic interfaces (Hsi, 2005). Differently from these techniques, our approach of excavation is aimed at excavating and defining a domain-specific language which is implemented in a tool.

Figure 5 Fragment of the business process modelling metamodel with the organisation elements (see online version for colours)

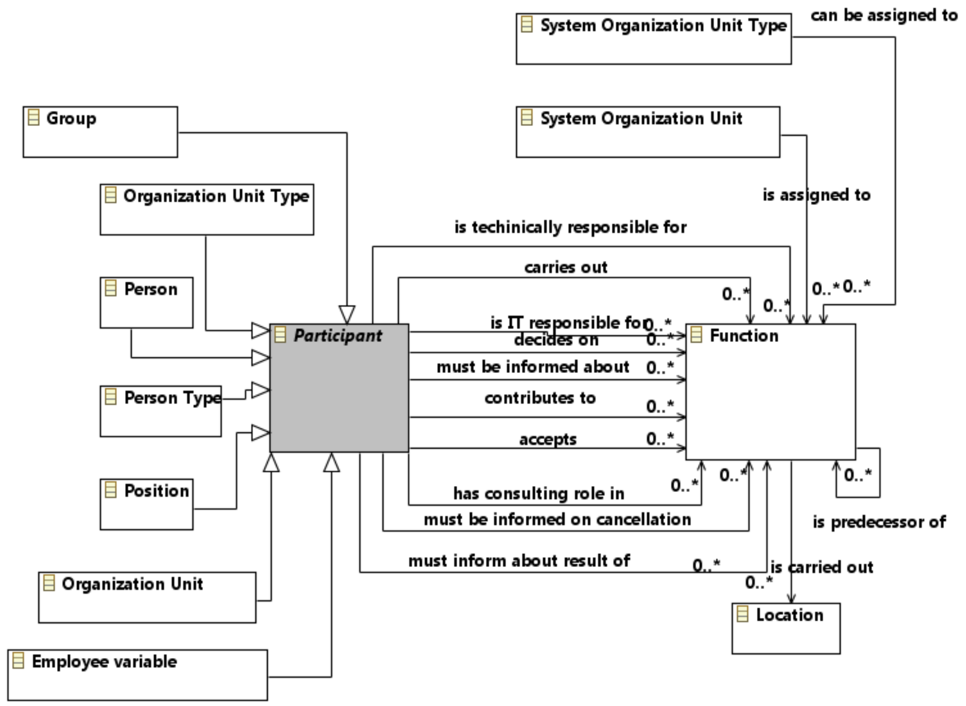


Figure 6 Fragment of the business process metamodel with the elements: functions, event and rule and their relationships (see online version for colours)

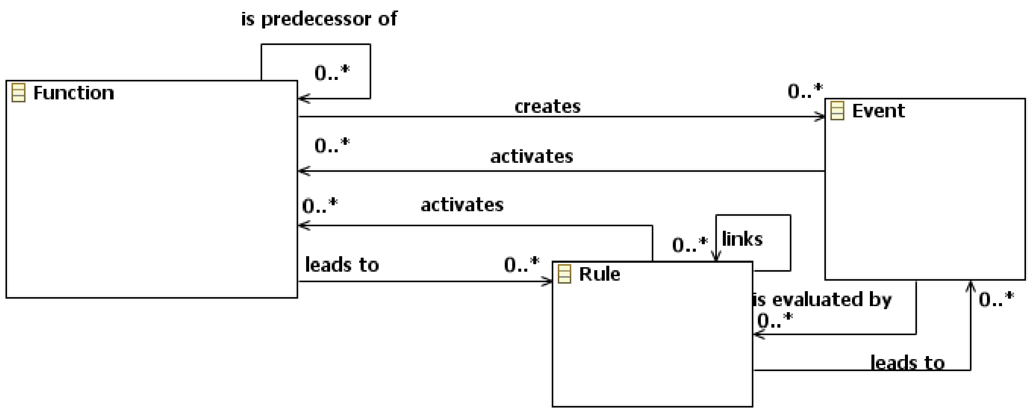
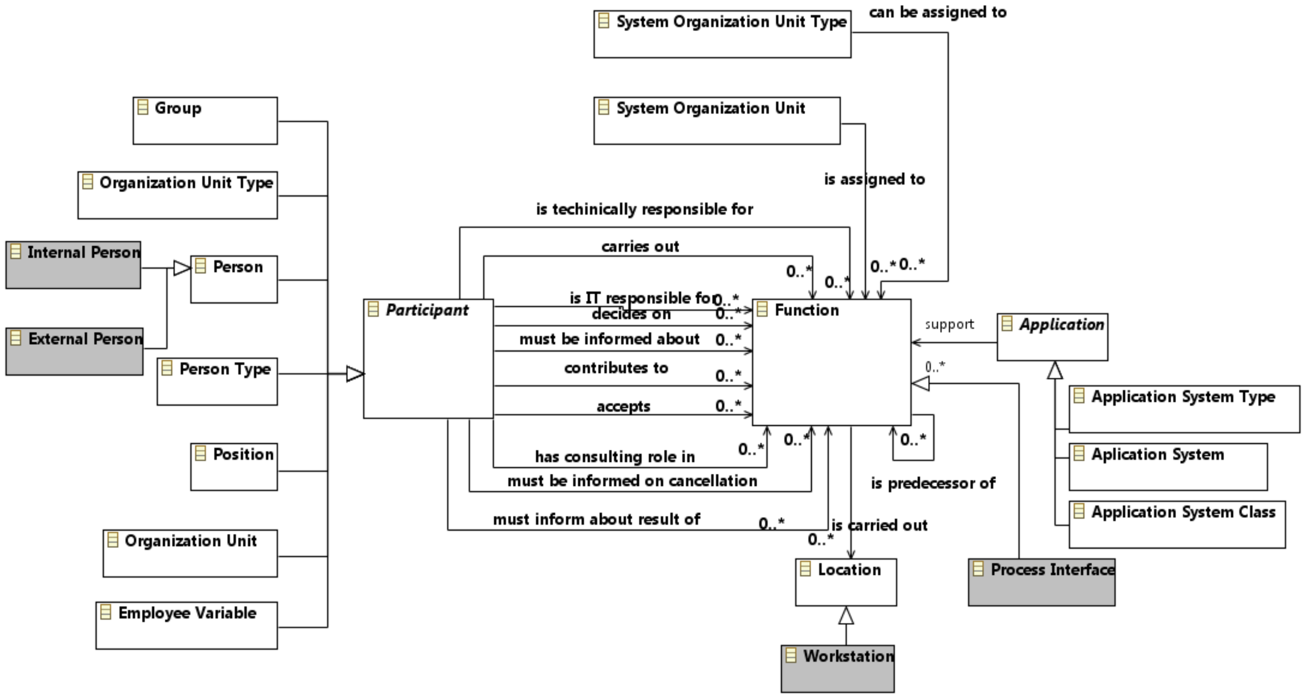


Figure 7 Fragment of business process domain language with notation element (see online version for colours)



An approach which seems to have similar objectives as ours was presented by Kern and his colleagues in (Kern and Kühne, 2007). That work has proposed a metamodel for the ARIS toolset repository which does not differentiate the kinds of objects of the ARIS method, for instance, it does explicitly identify metaclasses for the organisational units, functions and events (Scheer, 1999). These types are encoded through the assignment of numeric identifiers (attributes of a generic object definition metaclass). Consequently, one cannot differentiate the object types that represent quite distinct conceptual categories, such as organisational units and functions through quick inspection. On the contrary, the metamodels we have proposed here are defined at a higher level of abstraction when compared to those ones proposed by Kern and facilitate a mapping to a semantic domain.

3 Evaluation of concepts

To evaluate the metamodels of the organisational and business process languages of the ARIS method, this section introduces a model-driven engineering (MDE) approach. This approach creates instances of theses metamodels and uses information extracted from an ARIS markup language (AML) document to build the organisational and business process models.

AML is an XML format used to serialise ARIS models in the ARIS toolset. Each element and relationship depicted in the organisational and business process metamodel of ARIS method is codified in 'tags' in an AML document. For example,

an organisational model is identified by the 'MT_ORG_CHRT' tag of the 'Model.Type' attribute

the '*is predecessor of*' relationship of the business process metamodel is identified by the 'CT_IS_PREDEC_OF_1' tag of the 'CxnDef.Type' attribute. Therefore, a parser can be used in an AML document to build an organisational model or a business process model.

In order to extract model instances from an AML document, an AML metamodel is required. To create this AML metamodel, we employ two main steps at the metalevel.

The first step maps all the AML DTD elements to EMF (eclipse modelling framework) metaclasses and relationships, i.e., an isomorphic translation between the XML and EMF *technological spaces* (Kurtev et al., 2002). The objective of this translation is to enable access to model elements with MDE techniques in eclipse. We have performed excavation using an automated DTD to XSD (XML schema) transformation (*XML tools: DTD, XML schema and XML document conversion software tool: XML utilities)* and then an automated XSD to EMF transformation (Eclipse Foundation, 2008b). A fragment of the resulting EMF metamodel is presented in Figure 8 and discussed further in Section 3.1.

The second step is discussed in Section 3.2. It consists of analysis and restructuring of the AML metamodel. The objective is to improve the representation of modelling elements while preserving all model information.

3.1 AML metamodel

This section discusses the metamodel obtained in the first step (by direct translation of the DTD) using semantics inferred from (*ARIS platform: XML export and import)*, knowledge acquired by using the ARIS toolset and analysis of AML documents.

All metaclasses in the fragment presented in Figure 8 are contained in a top-level AML metaclass, which is the root of an AML document. (infrastructural metaclasses are omitted here. These metaclasses relate to information about tool versioning, database configuration, user management, date and version management, internationalisation support, etc. and are not directly related to the content of models.).

The group metaclass is a container for all diagrams (instances of model), objects and links among objects. The objects are instances of ObjDef which represents the basic modelling elements of the ARIS method such as organisational units, tasks (function) and events.

This fragment shows two important distinct concepts in the ARIS toolset: the separation between the content of a model (the definition of objects such as organisational units, tasks, events, rules, etc. and their links) and the diagrams (instance of models) which include the occurrence of objects and links, besides diagramming information. Using this separation between the content of a model and the content of a diagram it is possible for a definition element (i.e., an instance of ObjDef, CxnDef, AttrDef, etc) to be instantiated in different models at the same time. Further, by using occurrence objects (i.e., an instance of ObjOcc, CxnOcc, AttrOcc, etc.) in different diagrams with different properties one may instantiate the same object in with different notations. This concept is similar to that presented in Jonkers et al. (2004) under the name model/view/visualisation. The metaclasses which correspond to occurrence objects are shaded in Figure 8. With the exception of model and group, other metaclasses of the model are definition objects (in white in Figure 8.).

Definition objects carry the following information: name, type (typeNum), default symbol (symbolNum), links with other objects and links with other diagrams.

The element which links two objects in a model is denominated as a connection. It represents the link between a source element and a target element. A connection has the following attributes: type and name. The concepts of definition objects and occurrence objects are also applied to connections. These concepts are represented by CxnDef and CxnOCC, respectively.

All elements (connections, models and modelling objects) have a list of attributes. Each attribute is represented by an element named as AttrDef. The AttrDef has the following properties: type, name and content (AttrValue element).

Figure 8 Fragment of the AML metamodel (see online version for colours)

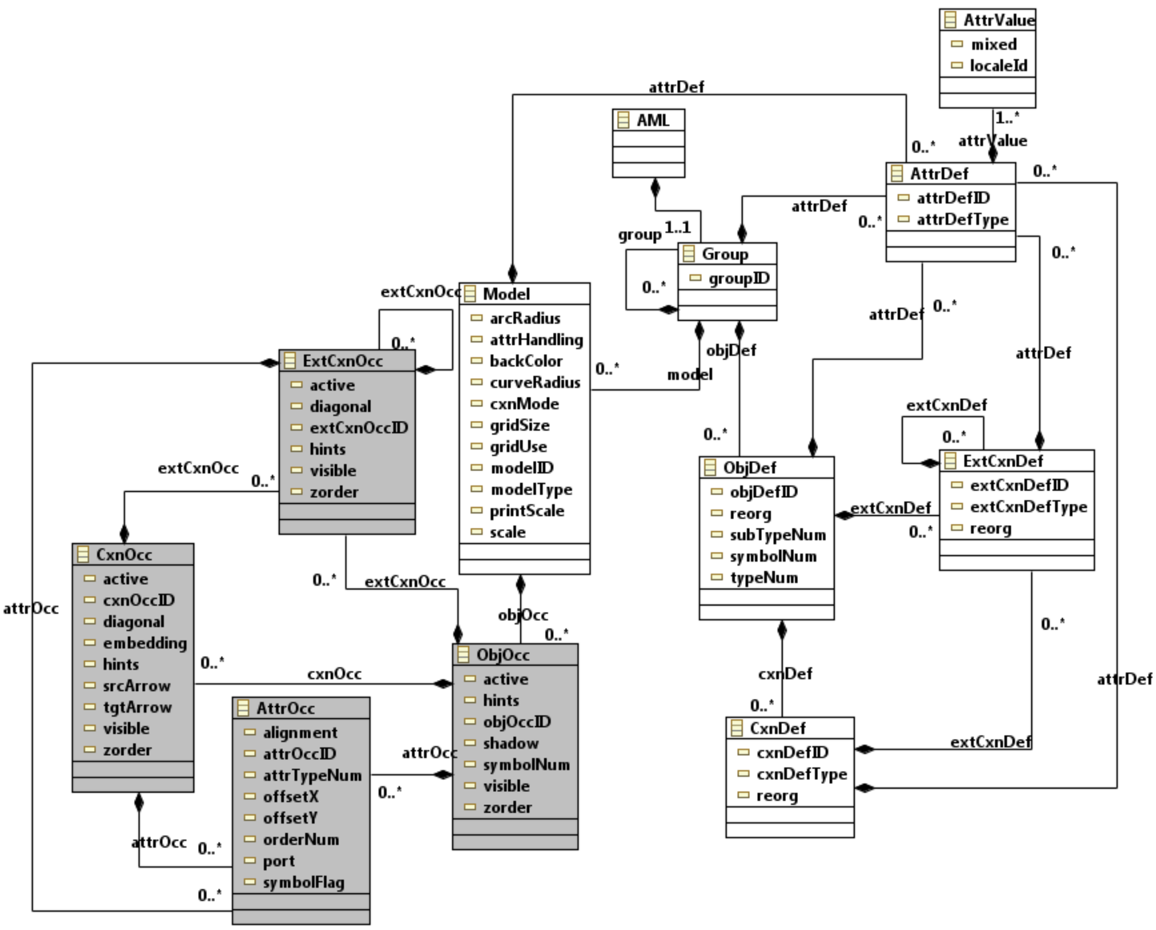
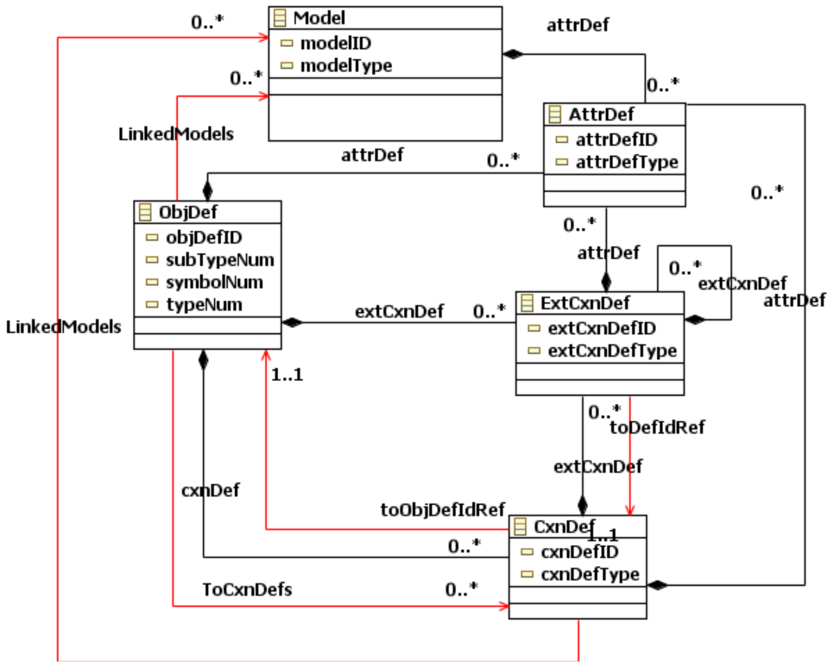


Figure 9 Fragment of refactored AML metamodel (see online version for colours)



3.2 *AML metamodel refactoring*

An analysis of the metamodel presented in Figure 8 will reveal that several relationships between metaclasses are encoded implicitly into attributes of type IDREF or IDREFS. For example, the relationship between an object occurrence (ObjOcc) and its corresponding object definition (ObjDef) is represented by the metaattribute objDefIdRef whose value identifies an instance of ObjDef.

By refactoring this metamodel, we replace these attributes with their corresponding meta-associations that do not appear directly in metamodel. The cardinalities of these associations must be obtained by analysing the AML DTD.

This is done by analysing two properties of the attribute:

1 whether it is mandatory (with #REQUIRED) or not (with #IMPLIED); (defining the lower bound to be 1 or 0)

2 whether the attribute is an IDREF type or an IDREFS type (defining the upper bound to be 1 or *).

Figure 9 shows a fragment of refactored AML metamodel with new relationships of association among the metaclasses (all non-containment associations).

3.3 *A discussion on the AML metamodel*

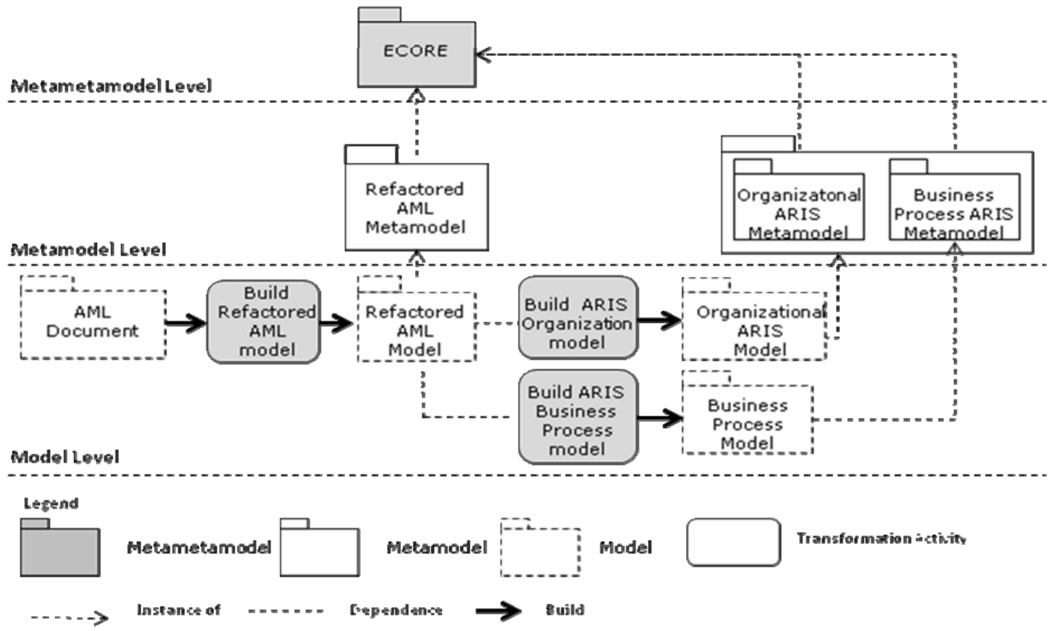
As shown in the AML metamodel fragments in Figure 8, the different types of objects of the ARIS method, such as, organisational unit, functions and event are not explicitly captured as metaclasses. In fact, these types are encoded in identifiers (IDs) in the respective elements, such as, ModelType, TypeNum and cxnDefType, which are attributes of model, ObjDef and CxnDef. Consequently, a quick inspection in these models does not allow one to differentiate the types which represent each conceptual category, such as, organisational units and functions. This kind of representation makes it cumbersome to determine the type of an object and the types of connections in which it may engage in a diagram.

Taking this into account, we have concluded that the AML metamodel is practically neutral with respect to the conceptual distinctions of ARIS method, and so, it is similar to a metametamodel. A salient feature of this model is that it captures the relations between object definitions and object occurrences. This a particular feature of the ARIS method which is not presented in metamodels such as EMF and MOF.

3.4 *Building the organisational and business process model from AML model*

Figure 10 shows the required steps to build an organisational and business process model from AML model. We have implemented the build refactored AML model, the build ARIS organisation model, and the build ARIS business process model transformations. The *build refactored AML model* is responsible for building a refactored AML model (instance of the refactored AML metamodel) with the content of an AML document. Both the build ARIS organisation model and the build ARIS business process model transformations use the attribute *type* of class *model* to identify the type of the model to be instantiate and use the attribute *type* of a class CxnDef to identify the types of connections between two elements of a model. We have tested the transformations on a number of AML documents exported from the ARIS toolset to evaluate coverage of modelling elements in our metamodels. As a result, the organisational and business process models of the ARIS toolset can be manipulated using model-driven techniques in the eclipse platform.

Figure 10 Building organisational and business process model from AML model



4 Conclusions and future work

The construction of metamodels as presented in this work allows one to develop a large number of tools for process analysis, such as modelling tools and simulation tools through the adoption of model-driven design (MDD) techniques, effectively opening up the contents of enterprise architecture models defined in the ARIS method. Furthermore, the metamodels presented here enable one to build computational tools which implement model transformations at a high level of abstraction without performing transformation at the document level, as the case of the XSLT transformation shown in Mendling and Nüttgen (2004) to transform ARIS EPCs into EPML (EPC markup language) documents.

To conclude, as future work, we intend to use foundational ontologies, such as UFO (Guizzardi, 2005; Guizzardi et al., 2008) to allow a precise definition of the semantics of the organisational and business process metamodels of ARIS method. By using semantic definition techniques, we are able to identify inadequate modelling elements of each domain through a set of criteria for systematic evaluation, such as clarity, expressiveness, completeness, parsimony, soundness, correctness and consistency (Guizzardi et al., 2005). Evaluation based on these criteria will enable us to establish well-founded recommendations to improve the quality of these modelling languages. The metamodels presented here are required as a starting point for this evaluation, since there can be no consistent semantic analysis and definition without precise abstract syntax definition.

Acknowledgements

This work was supported by FAPES in the scope of grants number 45444080/09 and 36316202/07 and CNPq in the scope of grants number 481906/2009-6 and 309059/2008-9.

References

ARIS Platform: XML Export and Import, White paper.

XML Tools: DTD, XML Schema and XML Document Conversion Software Tool: XML Utilities, available at http://www.hitsw.com/xml_utilites/.

Scheer, A.W. (1999) *ARIS – Business Process Modeling*, 3rd ed., Springer.

Archimate Consortium, *Archimate Resource Tree*, available at http://www.telin.nl/NetworkedBusiness/Archimate/ART.

DoDAF – U.S. Department of Defense (2007) *DoD Architecture Framework Version 1.5*, Vol. 2, Product descriptions.

Eclipse Foundation (2008b) *Generating an EMF Model using XML Schema (XSD)*, available at http://eclipse.org/modeling/emf/docs/2.x/tutorials/xlibmod/xlibmod_emf2.0.html.

Ferdinand, M., Zirpins, C. and Trastour, D. (2004) 'Lifting XML schema to OWL', *Web Engineering – 4th International Conference*, ICWE 2004, LNCS 3140, Springer, pp.26–30.

Guizzardi, G. (2005) 'Ontological foundations for structural conceptual models', PhD with Cum Laude, Telematica Instituut Fundamental Research Series, Vol. 015, Enschede, The Netherlands, Telematica Instituut.

Guizzardi, G., Ferreira Pires, L. and van Sinderen, M. (2005) 'An ontology-based approach for evaluating the domain appropriateness and comprehensibility appropriateness of modeling languages', *ACM/IEEE 8th International MODELS Conference,* LNCS Vol. 3713, Springer-Verlag.

Guizzardi, G., Vascocelos, J., Segrini, B., Falbo, R. and Guizzardi, R. (2008) 'Grounding software domain ontologies in the unified foundational ontology (UFO): the case of the ODE software process ontology', *IDEAS, XI Workshop Iberoamaricano*, Brasil, Pernambuco.

Hsi, I. (2005) 'Analyzing the conceptual integrity of computing applications through ontological excavation and analysis', PhD thesis, Georgia Institute of Technology.

Jonkers, H., Lankhorst, M., Van Buuren, R., Hoppenbrouwers, S. Bonsangue, M.M. and Van der Torre, L. (2004) 'Concepts for modeling enterprise architectures', Special issue on *Architecture in IT of the International Journal of Cooperative Information Systems*, September, Vol. 13, No. 3, pp.257–287, World Scientific.

Kern, H. and Kühne, S. (2007) 'Model interchange between ARIS and eclipse EMF', *The 7th OOPSLA Workshop Domain-Specific Modeling*.

Kurtev, I., Bézivin, J. and Aksit, M. (2002) 'Technological spaces: an initial appraisal', *CoopIS, DOA'2002 Federated Conferences*, Industrial track, Irvine, USA.

Lankhorst, M. (2005) *Enterprise Architecture at Work – Modelling, Communication and Analysis*, Springer.

Mendling, J. (2009) *Metrics for Process Models, Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness*, pp.17–57.

Mendling, J., Neumann, G. and Nüttgens, M. (2005a) 'Yet another event-driven process chain (extended version)', *Technical Report JM-2005-05-27*, Vienna University of Economics and Business Administration.

Mendling, J., Neumann, G. and Nüttgens, M. (2005b) 'Towards workflow pattern support of event-driven process chains (EPC)', M. Nüttgens and J. Mendling (Eds.): *Proc. of the 2nd GI Workshop XML4BPM – XML for Business Process Management at BTW 2005*, March, Karlsruhe, Germany, CEUR Workshop Proceedings, ISSN-1613-0073, Vol. 145, pp.23-38.

Mendling, J. and Nüttgen, M. (2004) 'Transformation of ARIS markup language to EPML', *Proc. of the 3rd GI Workshop on Event-Driven Process Chains*, pp.27–38.

RM-ODP-ISO-ISO/ITU-T (1995) *Open Distributed Processing – Reference Model*, International Standard ISO/IEC 10746.

TOGAF – The Open Group (2007) *The Open Group Architectural Framework* (TOGAF 8.1.1 'The Book'), Van Haren Publishing.

Van der Aalst, W. and Van Hee, K. (2004) *Workflow Management Models, Methods, and Systems*, MIT Press.

Volz, R., Oberle, D., Steffen, S. and Rudi, S. (2002) *OntoLiFT Prototype - WonderWeb: Ontology Infrastructure for the Semantic Web*.

Zachman, J.A. (1987) 'A framework for information systems architecture', *IBM Systems Journal*, Vol. 26, No. 3, pp.276–292.