

On the Support for the Goal Domain in Enterprise Modelling Approaches

Evellin Cristine Souza Cardoso, João Paulo A. Almeida and Renata S. S. Guizzardi

Ontology and Conceptual Modelling Research Group (NEMO)
Computer Science Department, Federal University of Espírito Santo (UFES)
Av. Fernando Ferrari, s/n, Vitória, ES, Brazil
ecardoso@inf.ufes.br, jp Almeida@ieee.org, rguizzardi@inf.ufes.br

Abstract— This paper reviews the support for the goal domain in several enterprise modelling approaches and enterprise architecture frameworks, namely, ARIS, ISO RM-ODP Enterprise Language, DoDAF, MODAF, BPMN/BMM, ArchiMate and its ARMOR extension. We describe how these efforts propose to align goals with the remaining elements in enterprise architectures (with special attention to the alignment of goals and business processes). Our aims are twofold: (i) to provide an integrated overview of the support for goal-related concepts in existing enterprise modelling approaches and, (ii) provide guidance for the selection of the frameworks with respect to the support for goal-related concepts.

Keywords: goals, enterprise modelling, business processes, survey

I. INTRODUCTION

Enterprise modelling approaches structure an enterprise architecture in terms of various related architectural domains or viewpoints [1][2][3] which focus on specific aspects of the enterprise. This division into related domains fosters separation of concerns and is a manner of mastering the intrinsic complexity of organizations, which encompass: (i) organizational activities structured in business processes and services; (ii) information systems supporting organizational activities; (iii) underlying information technology (IT) infrastructures, and (iv) organizational structures.

Among the various architectural domains, the domain of “motivation” has been recognized as an important element of enterprise architectures [4] as highlighted in Zachman framework’s motivation column [5]. Goal modelling is the artefact employed for capturing the motivational aspect and strategies behind the organizational practices [6], helping in clarifying interests and intentions from different stakeholders [7]. By adopting goal modelling, an enterprise architect can motivate particular choices for an enterprise’s elements and explore new possible configurations for an organizational setting. This is essential for business improvement once changes in a company’s strategy and business goals have significant consequences within all domains of the enterprise [8][9].

While the goal domain of enterprise architectures focuses on “why” [6][7], a behavioural business process domain also has significant importance in enterprise architectures since it addresses the way the enterprise organizes work and resources to fulfil its strategies [10], focusing on “how”

business activities are performed and supported by information systems.

This paper reviews the support for the goal domain in several enterprise modelling approaches and enterprise architecture frameworks, namely, ARIS, ISO RM-ODP Enterprise Language, DoDAF, MODAF, BPMN/BMM, ArchiMate and its ARMOR extension. We describe how these efforts propose to align goals with the remaining elements in enterprise architectures (with special attention to the alignment of goals and business processes). Our decision for the selection of these particular enterprise architecture approaches is based on their wide usage and their association with corresponding enterprise modelling languages.

The main contributions of this paper are twofold: (i) to provide an integrated overview of the support for goal-related concepts in existing enterprise modelling approaches (as a means to identify gaps in these frameworks for future research efforts); and (ii) to provide guidance for the selection of the frameworks with respect to the support for goal-related concepts.

This paper is further structured as follows: Section II discusses the importance of the goal domain for enterprise descriptions. Section III presents how the various modelling approaches address the goal domain and how they support the integration between the goal domain and the remaining viewpoints of the enterprise. Section IV presents an integrated analysis of the languages with respect to the support provided for goal-related concepts. Finally, Section V concludes the paper with some remarks about the current support for the goal domain in enterprise architecture frameworks and outlines some future work.

II. THE GOAL DOMAIN

Goal modelling explicitly captures the goals of an enterprise, documenting an enterprise’s strategy [11]. The definition of goals is related with objectives in a broad scope within the enterprise, i.e., goal statements range from high-level concerns in an enterprise (expressing the vision and mission of the organization) to declarations of the values that must be achieved by business process execution on behalf of stakeholders.

Although the benefits from adopting goal modelling for increasing the value of process models have been explored in literature [7][12][13][14][15], little attention has been devoted to the association between goal models and other elements of the enterprise architecture in the context of enterprise modelling. By aligning goal models with other elements in enterprise architectures such as physical or

informational resources, one can visualize how the elements of the enterprise architecture are manipulated in business processes in order to attain enterprise's goals. Thus, the motivation behind this association can be explained by the necessity of providing a strategic dimension for these elements, acknowledging the fundamental role of the whole set of elements of an enterprise architecture in achieving the enterprise's goals.

Since enterprise architectures are a powerful mechanism for managing the complexity of organizations, the first step to take the advantages of using some architectural approach is the documentation of key organizational aspects. In this context, *modelling languages* are employed with the aim of capturing enterprise descriptions for supporting decision-making and communication among stakeholders [3]. These modelling languages are based on adequate architectural concepts which in turn are represented by notational elements [2]. Thus, we include here only enterprise architecture frameworks that are supported by enterprise modelling languages. (Although BMM is an exception here since it is not a language, but a conceptual model instead, its importance in the context of current enterprise description, acknowledged by the effort of the OMG in standardizing it justifies its insertion in our analysis.)

Further, we restrict our attention to enterprise architecture frameworks that explicitly address the goal domain. The first aspect of evaluation thus concerns the availability of goal-related concepts in the enterprise modelling approach.

Although we focus on the goal domain and goal-related concepts, the relations of this domain with the rest of the enterprise architecture are essential to employing the goal domain meaningfully. The importance of the relations between architectural domains is emphasized in the more general observation that "(organizational) effectiveness is driven by the relationships between components rather than by the detailed specification of each individual component" [16]. This is certainly true for the relations between the goal domain and other architectural domains: goals affect the other architectural domains by providing a strategic dimension for business processes, resources, organizational actors and roles, etc. An ultimate objective of an enterprise architecture that addresses goal-concerns should be to reveal how an enterprise's goals can be translated into operational elements.

III. SUPPORT IN ENTERPRISE MODELLING APPROACHES

We start by reviewing Zachman's Framework and proceed to consider the support for the goal domain in ARIS, ISO RM-ODP Enterprise Language, DoDAF, MODAF, BPMN/BMM, ArchiMate and its ARMOR extension.

A. Zachman's Framework

The Zachman framework for Information Systems Architecture (also known as Zachman's framework) has been introduced in 1987 by John Zachman [5] and further extended in 1992 [17].

The framework considers the intersection between aspects of the enterprise architecture. These aspects can be summarized as the roles who participate in the process

(represented in the vertical axis) and the product abstractions (represented in the horizontal axis). The roles corresponds to the planner, owner, designer, builder and subcontractor and the product abstractions corresponds to "what" (material or information), "how" (process for product production or processing), "where" (corresponds to the geometry that components are deployed in relation to each other). More recently, the need for including additional product abstractions, namely, "who" (who executes the work?), "when" (when are the products processed?) and "why" (what motivates the work?) has been identified.

Although Zachman's framework has recognized the importance of the goal domain in its "motivation" column, Zachman did not define basic concepts for this column, justifying that "there is a scarcity of good examples in the people, time, and motivation columns" and stating that "the "why" column would be comprised of the descriptive representations that depict the motivation of the enterprise, and the basic columnar model would likely be ends-means-ends, where ends are objectives (or goals) and means are strategies (or methods)" [17].

B. ARIS

ARIS (ARchitecture for integrated Information Systems) [18] has been developed in Saarbrücken (Germany), in 1992, with the main aim at providing an architectural framework for enterprise description. The framework has been recognized as leader in the business community, having a large market share due to its integration with the SAP suite in the corporate and government sectors [15][19]. According to Davis [19], this leadership can be accounted by the fact that it provides support for modelling several aspects of complex business models (including processes, data, organizations, systems and so on), provides support for modelling software systems (using UML) among others.

The framework is composed of three abstraction levels (Requirements Definition, Design Specification and Implementation Description) which intend to describe the organization with different levels of detailing and four viewpoints (Organizational, Control or Process, Function and Data) which support the description of different enterprise domains and their relationships. While the division in abstraction levels allows one to capture the information about the enterprise in different granularities, the viewpoints aim at specifying details referring to each organizational aspect. Among these viewpoints, we concentrate in the Functional view since it captures the concept of goal as its associations with the other enterprise elements.

The Functional View defines the organizational functions which can be described in various aggregation levels in a hierarchical way. Functions designate activities or tasks which must be executed for the production of goods or services. The term function is also used synonymously with the terms complex function, process, activity or task [18].

The Functional View captures the static functional structure, whereas the Process View (or Control View) reflects the dynamic models that show the behaviour of processes and how they are related to resources, data and

functions of the business environment [19]. Static function structure means that functions are hierarchically organized independently of belonging to specific business process (as opposed to describing the dynamic behaviour of functions).

The language has opted for modelling the relationship between goals (represented as green pentagons) and functions in this view since the execution of functions can be seen as operations applied to objects for the purpose of supporting one or more goals [18]. According to ARIS toolset on-line documentation, an objective (or goal) is the definition of future business objectives which are supposed to be reached by supporting the critical factors and realizing new business processes. The relationship between objectives and functions is denominated as “supports of” relationship, although its semantics is not clear in the ARIS literature. Figure 1 shows the function *Prescribe patient’s treatment* which supports the goal *Minimize patient’s physical suffering and symptoms* (it is said that *Prescribe patient’s treatment* has a “supports of” relationship with the *Minimize patient’s physical suffering and symptoms* goal).

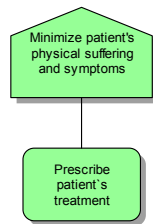


Figure 1 Relationship between goals and functions

Goals and their relationships (shown in Figure 2) are also modelled in this view. Goals can be linked with one another with a subordinate goal supporting several overriding goals. The allocation between functions and goals can be inherited by higher levels, i.e., an allocation at a lower level can be inherited by the upper levels [18]. Thus, the function *Provide medical care to patient* also supports the overriding goal *Heal the patient*.

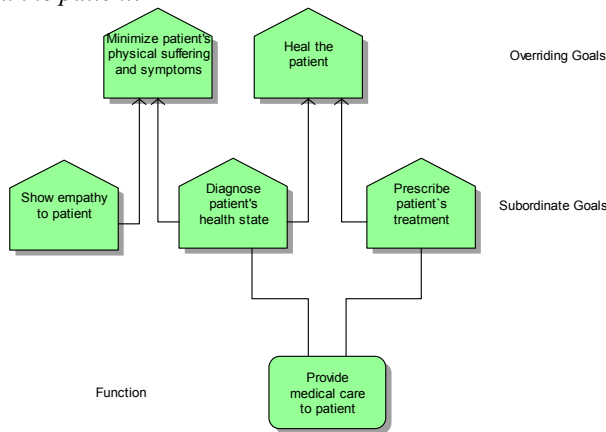


Figure 2 Goals and their relationships

Observe that the semantics of the relations between goals are also not clear (denominated as “belongs to” relationship).

Summing up, it is an n-to-n relationship between goals, i.e., a (overriding) goal can be overridden by N (subordinate) goals and a (subordinate) goal can override N (overriding) goals. Following with the example, *Heal the patient* (overriding goal) is overridden by *Diagnose patient’s health state* and *Prescribe patient’s treatment* (subordinate goals). We believe that this relation may indicate that the overriding goals consist of a disjunction or a conjunction of subordinate goals. However, the language is not expressive enough to distinguish between these cases.

C. ISO RM-ODP Enterprise Language

Differently for the other languages (or frameworks), the creation of the Reference Model of Open Distributed Processing (RM-ODP) by the ISO [20] has not been motivated by the need of providing a consistent framework for enterprise description, but by the need of manage the development and interoperation of distributed processing systems. For this reason, the RM-ODP consists of concepts and structuring rules for the specification of distributed systems. Although it has developed for managing system concerns, the language can also be employed for capturing the architecture of general enterprises, since the reference model has been defined to consider a system’s role in an overall (distributed and federated) enterprise setting.

The reference model defines a framework for the specification of ODP systems which has five viewpoints, called enterprise, information, computational, engineering and technology, which provide a basis for the specification of ODP systems. In each of the aforementioned viewpoints, a viewpoint language defines the concepts and rules for specifying ODP systems from that viewpoint. The reference model is accompanied by a UML profile developed recently with the purpose of specifying systems using the concepts in the reference model [20].

The enterprise viewpoint includes a concept of Objective, which is defined as “practical advantage or intended effect, expressed as preferences about future states.” Objectives in an enterprise specification are closely associated with the notion of Enterprise Community: “a community is a configuration of enterprise objects that describes a collection of entities (e.g. human beings, information processing systems, resources of various kinds and collections of these) that is formed to meet an objective.”

An enterprise specification states the objective of a community, how it is structured, what it does, and what objects comprise it. Every community has exactly one objective. The objective is expressed in a contract which specifies how the objective can be met. More specifically, policies in the enterprise contract constrain the behaviour and membership of communities in order to make them achieve their objectives. (See Figure 3 for the relation between Community, Contract, Policy and Objective).

An enterprise specification may decompose the objective of a community into sub-objectives (See Figure 3 for the refinement relation between Objectives). A sub-objective may be assigned to a collection of roles; in that case, the behaviour of the collection of roles is specified to meet the sub-objective and the sub-objective is met by the collection

Therefore, the relation among these concepts and the satisfaction of the goals within the framework can be described in a straightforward manner: enterprise goals (EnterpriseGoal) may be achieved through a set of enduring tasks (EnduringTask). Differently from other frameworks, it explicitly associates strategic elements with an “as is” or “to be” enterprise phase.

F. OMG Standards

The Business Process Modelling Notation (BPMN) is a standard graphical notation for specifying business processes [26]. The BPMN specification is constrained to support only the modelling concepts that are applicable to business processes. This means that the other domains of the organization are out of the scope of the current version of the language. For instance, the definition of organizational models and resources, data and information models, business rules and strategic concerns cannot be captured within the language.

Although BPMN does not provide concepts for modelling the motivational aspects, some works in research [27] use the annotation element (a type of Artefact) to capture intentional aspects without altering the basic structure of processes. This solution, however, has a clear disadvantage: the strategic aspect is captured only in natural language with no structure or rigor in the representation of goal-related concepts.

Another solution for capturing strategic concerns within the language is to adopt the concepts in the Business Motivation Model (BMM) [33] also standardized by the OMG for schematizing or structuring the development, communication and management of business plans in enterprises. BMM enables one to express the business environment from a strategic perspective, being useful as an instrument in corporate governance, strategic planning and business transformation activities. It serves as an instrument in the communication among diverse groups and interests and provides a common ground for discussion and decision-making (although it prescribes no language for the representation of strategic concerns.)

This model comprehends a set of built-in concepts which can capture the factors that motivate the establishing of business plans, the elements which comprise the business plans and the interrelationships between these elements. These built-in concepts define the elements of business plans and are characterized by basic attributes (identifier and text description) and associations (optional and many-to-many). Besides these concepts, there are three concepts (namely, business process, business rule and organization unit) which participate in associations within the Business Motivation Model. These concepts are regarded as references to elements that will be defined and maintained outside the scope of an enterprise’s Business Motivation Model.

BMM does not comprehend a methodology or a language, but rather it is an abstract specification which supports a wide range of approaches for creating and maintaining models for an enterprise. Further, although providing a conceptual model, BMM does not provide software tools neither for the construction of the deliverables

(organized business plans), nor for the management of the process of development. This characteristic makes the practical application of the BMM reliant on languages for modelling the concepts which are covered by the specification. Commonly, it is necessary to apply several modelling languages which focus on different architectural domains in conjunction to express the business context in its totality.

G. ArchiMate

ArchiMate is a modelling language for describing enterprise architectures which presents a well-defined set of concepts and relationships between architectures domains, [28].

The design of the language has been guided by the issue that managing the complexity of the organizations requires the description of the language in terms of several domains of knowledge as well as the relations within these domains. Therefore, the main goal of the language is to promote the integration of the several viewpoints of the organization, albeit it is not intended to introduce a language that can replace all existing domain-specific languages.

The central characteristic of the architecture is the service orientation. According to [16], “services are defined as the unit of functionality that some entity (e.g. a system, organization or department) makes available to its environment and which has some value for certain entities in the environment (typically the ‘service users’)”. The service orientation proposed in the ArchiMate enables a layered view of the architectural models, where the concept of the service is one of the main linking between the different layers. Services are made available to higher layers through service layers which in turn are interleaved with implementation layers that actually realize these services. Within a layer, there may also be internal services.

The language distinguishes three main layers (abstraction levels): the **Business Layer** which offers products and services to external customers realized by business processes executed by actors or roles; the **Application Layer** which supports the business layer with software applications services; and the **Technology Layer** which offers infrastructural services for software applications (composed by software systems, computer and communication devices). Each of these layers can be decomposed in sub-layers. However, the level of abstraction must be chosen according to stakeholders concerns. The layer distinction is depicted in Figure 6.

The arrows in the left represent use relations which show how the higher layers make use of the services of the lower layers [29]. For example, the business layer, where the business processes realize the products of the company, can make use of the generic services offered by supporting applications (in the application layer). These applications in turn can take the advantages of the infrastructural services realized by the lower layer (the technology layer) [16].

Besides the layered view proposed in ArchiMate, the architecture is structured in terms of other orthogonal dimension: the aspects. The aspect dimension distinguishes

between behavioural, structural and informational aspects of the enterprise [30].

In the **structural** and **behavioural** aspects, the behavioural concepts are assigned to structural concepts to show who or what displays the behaviour. For instance, in the business layer, roles, interfaces and collaborations (structural concepts) are assigned to business processes, business services and business interactions (behavioural concepts) respectively. The structural concepts are further categorized as active structural elements (concepts which display the behaviour, e.g., roles, interfaces and collaborations) and passive structural elements (objects to which behaviour is applied, e.g., information business objects). With respect to the **informational** aspect, it represents the problem domain knowledge that is used by and communicated between the stakeholders in performing their behaviours.

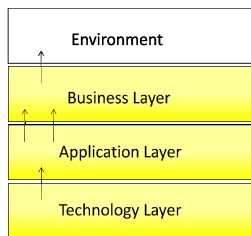


Figure 6 Layered View (adapted from [29])

The language puts emphasis on concepts which capture the design and realization of the organization, without however providing concepts for capturing the intentional aspects beyond the concepts of Value.

The Value concept represents the value of a product or service which makes some stakeholder to appreciate it (either for providing or for acquiring the product/resource). It can be associated with organizational services and, indirectly, with products [3]. For instance, the value *Be Insured* is the expression of what the service *Provide Insurance* aggregates to some client [3]. In other words, the Value concept represents what is the purpose of some service to a stakeholder and thus, can be used to capture the goals to be achieved with some service. Since the service concept links the layers within ArchiMate, (and the value is associated with the services) the value concept thus is strategic to capture what each layer aggregates to the overall architecture (addressing the reasons for the existence of the layers). The ArchiMate specification permits the use of composition, aggregation, association and specialization between Value elements but does not specify the semantics of these relations.

H. ARMOR/ArchiMate

The marginal support devoted to “motivation” in the core ArchiMate modelling framework led researches to extend it by proposing the ARMOR language in [30]. Basically, the ARMOR language concerns the motivational aspect of the enterprise and is defined on the basis of existing works in the community of goal-oriented requirements modelling.

To support the modelling of intentional properties, the core ArchiMate framework is extended as depicted in Figure

7. This extension comprises in the introduction of the *motivation* and *meaning aspects* as well as the *value layer* (observe that the Value and Meaning concepts of the previous section can be extended to represent these aspects in this new version of the language).

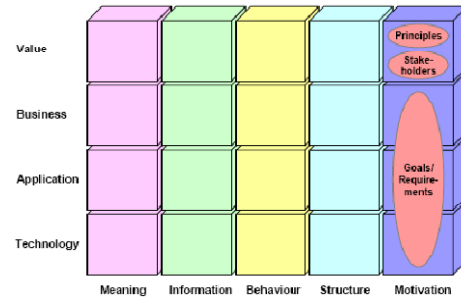


Figure 7 The extension of ArchiMate modelling framework [30]

The *value layer* represents the value of the services and products that are offered to customers, such as cost, and networks of value exchange. Concerning the aspects, whilst the *meaning aspect* represents concerns that are related to the semantics of enterprise (architecture) artefacts (i.e., the different ontologies that could be used by the enterprise and how it handles these ontological differences), the *motivation aspect* is concerned with the goals and intentions of the enterprise.

The remainder of this section is devoted to describe only the domains within the *motivation aspect* (the *value layer* and *meaning aspect* are not explored in the proposal and are left as topics of a forthcoming work). There are three essential domains within this layer: the *stakeholder domain*, the *principles domain* and the *requirements domain*.

The *stakeholder domain* models the stakeholders of the enterprise, including their concerns (some area of attention or interest) and the assessment of these concerns. The *principles domain* represents, amongst others, the vision, mission, strategies, policies, principles and guidelines of the enterprise, constituting the high-level constraints for the design of the enterprise architecture. Finally, the *requirements domain* is concerned about the goals, requirements and expectations that further constrain the design of the enterprise architecture.

The design of the ARMOR language is driven by a rigorous analysis of existing works in the area of elicitation/modelling. For this reason, many concepts of other languages for goal modelling are incorporated within the ARMOR language. Figure 8 depicts its meta-model:

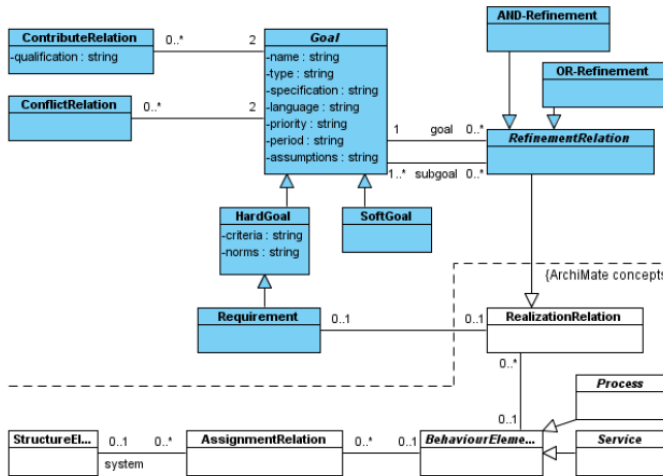


Figure 8 The ARMOR Meta-model [30]

According to this figure, the language has the following concepts:

Goal concept: The key concept of the language. A goal is defined as some desired effect in the problem domain, or some desired properties of a solution;

Requirement concept: the language has the principle of distinguishing between goals that can and can not (yet) be assigned to actors. For this reason, it provides the concept of requirement which can be defined as a goal that can be assigned to a single system;

Hardgoal/Softgoal concept: the distinction between hard and soft goals (of i* and KAOS, for instance) is considered significant and is therefore also supported in ARMOR. In particular, soft goals are useful in the evaluation of alternative designs;

Goal refinement¹: the current languages support the two types of refinement of goals into sub-goals: the means-end relationship and the decomposition relation. However, it is not always clear when should a refinement be considered as a decomposition and when as a means-end relationship. For this reason, ARMOR supports only the AND/OR refinement relation among goals, with the possibility to specialize this relationship later on if this is felt necessary;

Other relations between Goals: the ARMOR language supports the following relations among goals: **Positive and negative contributions** on the satisfaction of hard and soft goals, in order to facilitate the evaluation of alternative goal refinements. The need to be able to qualify the strength of the contribution, and in what detail may depend on the situation at hand. These contributions have different qualification ranges, such as the range 0..10 or the range ++, +, +/-, -, --. **Conflict relation** between two goals G1 and G2, such that the satisfaction of G1 inhibits the satisfaction of G2, and vice versa. A conflict is only possible between hard

¹ The refinement of some goal may be based on certain assumptions about (elements in) the problem domain. i* and KAOS introduce the notions of assumption, belief and domain property for this purpose. Since it is considered useful to make such assumptions explicit, ARMOR also supports the general notion of ‘assumption’.

goals (and requirements), since the criteria for the satisfaction of soft goals is unclear; i.e., it is unclear when the satisfaction of a soft goal inhibits the satisfaction of another goal.

Since ARMOR is an extension of the current version of the ArchiMate language, some concepts are ‘borrowed’ from ArchiMate, such as the realization relation concepts which is used to represent a link between a requirement and a design artefact, as for example, a realization relation between a requirement and the services and processes that implement this requirement). The ARMOR concrete syntax is shown in Figure 9 as well as some concepts of ArchiMate which can be used in combination with ARMOR concepts.

Abstract	Concrete	Abstract	Concrete
Hard goal	Hard goal (with goal icon)	Soft goal	Soft goal (with goal icon)
Requirement	Requirement (with requirement icon)	Refinement/Realization	Refinement/Realization (with realization icon)
And-refinem.	And-refinem. (with and-refinement icon)	Or-refinem.	Or-refinem. (with or-refinement icon)
Contribute	+/-	Conflict	Conflict (with conflict icon)
Business service	Business service (with service icon)	Business process	Business process (with process icon)
Used by	Used by (with used-by icon)	Association	Association (with association icon)

Figure 9 The ARMOR concrete syntax [30]

The linkage of the goal domain with the other viewpoints is established through the concept of Stakeholder. Observe the meta-model of Figure 10 which (indirectly) links Goals to Stakeholders (via Assessments and Concerns). A RealizationRelation is also another way of linking Goals and Services/Processes (Figure 8). As argued in [30], the relation between Goals, Services/Processes and Stakeholders is: stakeholder concerns to the services/processes (and possibly the supporting applications and technology) that ‘solve’ these concerns.

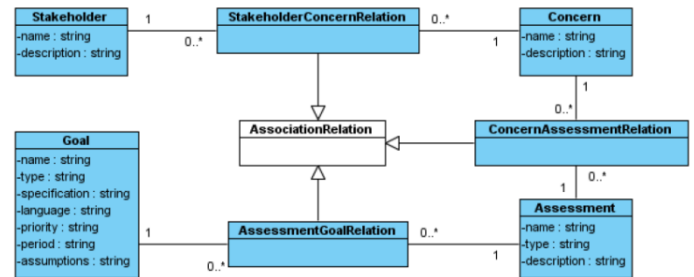


Figure 10 The meta-model of the stakeholders domain [30]

IV. SUMMARY OF SURVEYED APPROACHES

The previous sections have presented an overview of the support provided by the current frameworks within the enterprise architecture area with respect to capture goal-related concepts. They demonstrated that the support for goal-related concepts varies in the current frameworks, ranging from a rudimentary notion of intentional aspects (such as provided by ARIS) to more sophisticated support as provided by the ARMOR language. Table 1 summarizes the goal-related concepts and their relations to other elements in

the enterprise architecture for each of the architectural frameworks we have investigated.

V. CONCLUSIONS

Methods for enterprise architecture modelling that address strategic (or motivational) concerns aim at describing organizations to facilitate the decision-making process which ultimately determines how an enterprise's objectives are operationalized. In this paper, we have surveyed how current enterprise modelling approaches address these strategic concerns. More specifically, we have investigated the concepts for goal modelling and discussed how the aspect of "motivation" is integrated with other enterprise aspects. With respect to the particular enterprise frameworks surveyed in this paper, we can draw the following conclusions:

With respect to ARIS, our findings corroborate the observations of [15], which have concluded that ARIS lacks a comprehensive goal-modelling framework required for goal-oriented business process modelling. In this respect, the deficiencies of the ARIS framework includes its limited

expressiveness for capturing complex relationships among goals (such as causal, logical and influencing relationships and evaluation mechanisms [15]) and its limited capability in describing the integration between goals and business processes (and with the other elements of the enterprise architecture via the concept of business process). In particular, we have observed that the lack of clarity for the semantics of the "supports of" relationship in ARIS poses a significant problem for its application. The literature does not specify whether the execution of a function entails goal achievement or just positively contributes to the goal realization. Further, the relationship between goals and the other elements of the enterprise architecture (such as organization units, resources, and so forth) can only be indirectly established through functions. This kind of problem motivates the search for more expressive languages for goal modelling to be combined with the ARIS framework.

With respect to the RM-ODP Enterprise Viewpoint, we can conclude that the reference model defines rich relations between Objectives and its related concepts (Community,

Table 1 Summary of Goal-Related Concepts and their Relations

	Goal Domain		Other Domains	
	Main concepts	Relations between Concepts	Related Concepts	Relations
Zachman's Framework	Objective and Strategy	Ends-Means-Ends (between objectives, strategies and objectives)	(not discussed explicitly)	
		Conflict (between objectives, shown only as part of an example)		
ARIS	Objective	Objective "belongs to" Objective (no distinction between disjunctive and conjunctive)	Function	Function "supports of" Objective (no clear semantics)
RM-ODP Enterprise Viewpoint	Objective	Refinement between objectives	Community, process and role	A Community has one objective
				Sub-goals may be assigned to processes
				Sub-goals may be assigned to roles
OMG (BPMN/ BMM)	Means, Ends, Assessment and Influencer	Means-Ends relations	BPMN: Organization Unit (OU), Business Process (BP) and Business Rule (BR)	OU (defines Ends, establishes Means, makes Assessments, recognizes Influencers, may be defined by a Strategy, may be responsible for Business Processes), BP (realizes Courses of Action, is governed by Business Policies, may be guided by Business Rules, is the responsibility of one or more Organization Units), BR (is derived from Business Policy, may guide Business Processes, may have an enforcement level effected by a Tactic)
DoDAF	Vision and DesiredEffect	visionIsRealizedByDesiredEffect	Activity	desiredEffectDirectsActivity
MODAF	EnterpriseVision and EnterpriseGoal	EnterpriseVision has goals EnterpriseGoals	EnterprisePhase and EnduringTask	EnterprisePhase has vision EnterpriseVision
				EnterpriseVision has tasks EnduringTasks
ArchiMate language	Value	No specific relations (although generic composition, aggregation, association and specialization can be used with no specific semantics)	Product and Service	Association (a generic relation in ArchiMate) between Value and Product and Value and Service
ARMOR language	Goal (HardGoal, SoftGoal), Requirement, Concern, Assessment	AND/OR Refinement (between goals)	Services/Processes and Assessment/Concerns (Stakeholder)	RealizationRelation (services and processes) and AssociationRelation (Stakeholders)
		Conflict (between goals)		
		Positive/Negative contribution (between goals)		

Contract, Roles, Behaviours, Processes or Sub-Objectives). However, the UML profile for ODP does not allow a modeller to put this conceptualization at work in an enterprise model, restricting itself to a simple relation between objectives and its sub-objectives (modeled as composition) and the assignment of an objective to a community. Further extension of the profile seems necessary to capture complexity of enterprise settings.

With respect to DoDAF, we have observed that the support for the integration of intentional concepts is rudimentary, not allowing one to depict complex relationships between goals and the other enterprise elements (the sole relation which is possible to capture is the DesiredEffect-Activity relation). In relation to DoDAF, MODAF also has a limited expressivity for expressing intentional concepts (it has just the EnterpriseGoal and EnterpriseVision for capturing intentional concepts).

BPMN is even more problematic for expressing goal-related concepts: the language does not provide constructs for capturing strategic issues, restricting to model concepts which are solely related with the behaviour of business processes.

With respect to ArchiMate, we can conclude that the language does not provide the “motivational” dimension as one of its aspects, although there is a vague idea of intentionality in the current version of the language in linking the concept of value and service. Some recent effort [30] presents the motivational aspect of the ArchiMate language and introduces the ARMOR language used to capture goals and intentions of the enterprise. This language is strongly based on i*/Tropos (which is further explored in our own work [31]), with some additional concepts introduced as a means of capturing conflicting relations between goals and goals that a system TO-BE is expected to achieve (denominated as Requirements). The ARMOR metamodel inherits most of concepts of i*/Tropos, which suggests that most of the benefits of the usage of this latter methodology can be reproduced with the usage of ARMOR. We intend to investigate further the suitability of the ARMOR language as well as its expressiveness for modelling real-world organizations.

As a general conclusion we can observe that current approaches for enterprise modelling offer rudimentary support for intentional modelling as well as for the incorporation of intentional aspects in the other viewpoints; this conclusion is corroborated with observations by [30] and [32]. More specifically, we can conclude that the problem of capturing strategic concerns is an issue commonly neglected in the current languages, motivating us to set up a research agenda to explore methodologies for goal modelling and their integration into enterprise architecture and enterprise modelling approaches. In that respect, some research works in literature have intended to bridge this gap, such as [31] and [21].

REFERENCES

[1] IEEE. IEEE Recommended practice for architectural description of software-intensive systems. IEEE Std 1471-2000. 2000.

[2] R. Dijkman, "Consistency in Multi-Viewpoint Architectural Design". Ph.D. thesis. The Netherlands: University of Twente, 2006.

[3] Telematica Instituut. Archimate Resource Tree. <http://www.telin.nl/NetworkedBusiness/Archimate/ART/>, last accessed at 14/07/2009.

[4] E. Yu, M. Strohmaier, and X. Deng. "Exploring Intentional Modelling and Analysis for Enterprise Architecture". In Proceedings of the EDOC 2006 Conf. Workshop on Trends in Enterprise Architecture Research, TEAR. 2006.

[5] J. Zachman. "A Framework for Information Systems Architecture". IBM Systems Journal, pp. 276-292, 1987.

[6] E. Yu. "Modelling Strategic Relationships for Process Reengineering". Ph.D. Thesis. Department of Computer Science, University of Toronto, 1995.

[7] P. Halleux, L. Mathieu, and B. Andersson. "A Method to Support the Alignment of Business Models and Goal Models". Proceedings of 3rd International Workshop on Business/IT-Alignment and Interoperability (BUSITAL'08) CEUR Workshop Proceedings. 2008.

[8] H. Jonkers, et al. "Concepts for Modelling Enterprise Architectures". International Journal of Cooperative Information Systems. 2004.

[9] H. Jonkers, et al. "Enterprise architecture: Management tool and blueprint for the organization". Information Systems Frontiers, Netherlands, Springer, 2006. Vol. 8.

[10] A. Sharp and P. McDermott. "Workflow Modelling Tools for Process Improvement and Application Development". Artech House, 2001.

[11] B. Andersson, et al. "Enterprise Sustainability through the Alignment of Goal Models and Business Models". Proceedings of 3rd International Workshop on Business/IT-Alignment and Interoperability (BUSITAL'08) CEUR Workshop Proceedings. 2008.

[12] P. Kueng, and P. Kawalek. "Goal-based business process models: creation and evaluation". In Business Process Management Journal, pp. 17-38, 1997.

[13] I. Markovic, and M. Kowalkiewicz. "Linking Business Goals to Process Models in Semantic Business Process Modelling". In Proceedings of 12th International IEEE Enterprise Distributed Object Computing Conference (EDOC '08). 2008.

[14] G. Koliadis, and A. Ghose. "Relating business process models to goal-oriented requirements models in KAOS". Advances in knowledge acquisition and management (Pacific Knowledge Acquisition Workshop). 2006.

[15] D. Neiger, and L. Churilov. "Goal-Oriented Business Process Engineering Revisited: a Unifying Perspective". in Proceeding of the First International Workshop on ComputerSupported Activity Coordination (CSAC 2004). 2004.

[16] M. Lankhorst. "Enterprise Architecture at Work - Modelling, Communication, and Analysis". Springer Verlag. 2005.

[17] J. F. Sowa and J. Zachman. "Extending and formalizing the framework for information systems architecture". IBM Systems Journal. vol. 31, 3, 1992.

[18] A. Scheer, "ARIS – Business Process Modelling", Springer, 2000.

[19] R. Davis. "Business Process Modelling with ARIS - A Practical Guide". Springer, 2001.

[20] ISO - International Organization for Standard. "ISO/IEC 19793:2008, Information technology — Open distributed processing — Use of UML for ODP. <http://www.iso.org/iso/home.htm>, last accessed at 04/05/2010.

[21] E. Cardoso, et al. "Semantic Integration of Goal and Business Process Modelling", unpublished.

[22] ISO - International Organization for Standard. "Information Technology - Open Distributed Processing - Reference Model - Enterprise Language ISO/IEC 15414 | ITU-T Recommendation X.911. 2004".

[23] USA Department of Defense. "DoD Architecture Framework version 1.5 Volume I: Definitions and Guidelines". The United States Department of Defence. <http://www.defenselink.mil/cio-nii>, last accessed at 04/05/2010.

- [24] British Ministry of Defence. "Ministry of Defence", <http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/InformationManagement/MODAF>, last accessed at 04/05/2010.
- [25] British Ministry of Defence. "Modaf Meta-Model", <http://www.modaf.org.uk/m3>, last accessed at 04/05/2010.
- [26] Object Management Group (OMG). "Business Process Modelling Notation Initiative", <http://www.bpmn.org/Documents/FAQ-WG.htm>, last accessed at 04/05/2010.
- [27] G. Koliadis, et al. "A Combined Approach for Supporting the Business Process Model Lifecycle". in Proceedings of the Asia-Pacific Conference on Information System. 2006.
- [28] The Open Group. What's ArchiMate. <http://www.archimate.org>, http://www.archimate.org/en/about_archimate/what_is_archimate.html, last accessed at 04/05/2010.
- [29] M. Lankhorst, "ArchiMate Language Primer". Enschede: Telematica Instituut, 2004.
- [30] D. Quartel, et al. "A Goal-Oriented Requirements Modelling Language for Enterprise". Proceedings of the 13th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2009. Auckland, New Zealand, 2009.
- [31] E. Cardoso. "On the Alignment between Goal Models and Enterprise Models with an Ontological Account". Master Thesis. Federal University of Espirito Santo, Brazil, 2009.
- [32] P. Soffer and Y. Wand. "On the Notion of Soft-Goals in Business Process Modelling", pp. 663– 679, 2005.
- [33] Object Management Group (OMG), "Business Motivation Model (BMM)", <http://www.omg.org/spec/BMM/1.0>, last accessed at 04/05/2010. <http://cio>
- [34] USA Department of Defense. DM2 - DoDAF Meta-Model. nii.defense.gov/sites/dodaf20/DM2_HTML/index.htm?goto=1, last accessed at 30/06/2010.