Tartarus – Una estrategia para construir y expresar Arquitecturas Empresariales

Tartarus – A MDE Strategy to build and express Enterprise Architectures

Mario Elkin Rodríguez Alarcón, Esp[1]., Francisco Alexander Murcia Bermúdez, Esp[2]. & Darío Ernesto Correal Torres, Ph.D.
1. Esp. Construcción de Software. Est. Maestría en Ingeniería de Sistemas y Computación.
2. Esp. Construcción de Software. Est. de Maestría en Ingeniería de Sistemas y Computación.
3. Ph.D en Ingeniería. Profesor Asistente – Universidad de los Andes.
Universidad de los Andes.
me.rodriguez373@uniandes.edu.co, marioera@gmail.com; fa.murcia68@uniandes.edu.co, cycomantis@gmail.com; dcorreal@uniandes.edu.co.

Recibido para revisión 22 de septiembre de 2010, aceptado 03 de enero de 2011, versión final 09 de febrero de 2011

***Resumen*— Las compañías emplean importantes esfuerzos en establecer diagnósticos empresariales y planes estratégicos de inversión, para lograr y/o mantener lugares competitivos en el mercado. El profundo análisis organizacional requerido para tal fin, debe ser idealmente apoyado en metodologías para el levantamiento, depuración y comunicación de la información. Los exhaustivos resultados que emergen de un estudio a nivel empresarial, son susceptibles a la ambigüedad, inconsistencia y mala interpretación, dificultando la importante tarea de transmitir hechos relevantes para el mejoramiento de la empresa. En este artículo, presentamos un conjunto de estrategias para la construcción de modelos empresariales, que permiten describir la realidad organizacional y obtener artefactos consistentes para comunicarla. Dichos modelos son definidos a través de un meta-modelo, que abarca los principales conceptos empresariales, abstraídos de algunos de los marcos arquitecturales que gozan de mayor reconocimiento en el mercado. Adicionalmente, ofrecemos un ejemplo, para ilustrar el modelado de un fragmento de la arquitectura empresarial de una organización financiera, con la motivación de derivar artefactos que propicien su divulgación.**

***Palabras Clave*— Meta-Modelo, Ontología, Marco Arquitectural, Lenguaje Específico de Dominio**

***Abstract*— Companies devote considerable efforts to establish business assessments and strategic investment plans in order to attain and/or maintain a competitive spot in the market. The thorough organizational analysis required for this endeavor should be ideally supported by information capture and communication methodologies. The comprehensive results that emerge from a company-wide study are susceptible to ambiguity, inconsistencies and misinterpretation, thus hindering the important task of communicating relevant data that could be used for company improvement. In this article, we present a set of strategies used to build business models that can describe the organizational reality as well as attain consistent ways of describing it. These models are defined through a Meta-Model that encompasses fundamental business ideas, taken from some of the most recognized architectural frameworks in the market. In addition to the aforementioned, we provide an example to illustrate the modeling of a fragment of a financial company's enterprise architecture in order to obtain artifacts that foster its dissemination.**

***Keywords*— Meta-Model, Ontology, Architectural Framework, Domain Specific Language**

I. INTRODUCTION

Managing the complexity of an enterprise, because of the diversity of processes, services, policies, technological resources and other challenges, has brought about the evolution of Enterprise Architecture (EA) over the last two decades. The experience acquired by organizations in this field, expressing reality through documents, models and other artifacts, has produced sets of patterns and guidelines that are clustered in well-known Architectural Frameworks (AF), like Zachman[30], TOGAF [27], FEA [4] [3], DoDAF [28], E2AF [13], among others. Each AF has been built on a set of different fundamental principles (functional decomposition, modularization, standardization [27] and efficient use of resources [4], etc.), which has derived different forms of organizing and structuring the artifacts that make up an EA.

Since each AF supports and tends to satisfy certain principles, the artifacts that constitute it are designed and organized so

as to favor particular qualities [19]. Therefore, an enterprise architect must review different AFs and take from them the guidelines and suggestions necessary for a complementary set of components needed when building an EA. For example, it is common practice to use Zachman's guide to categorize artifacts and to be guided by their creation process, proposed by TOGAF [25]. However, despite the complementarity of AFs, right now there is no common knowledge environment in EA. Each AF has and defines its own glossaries of terms, deliverables and structure, hindering not only the neutral and objective construction of an EA, but the simple and consistent integration of artifacts.

Enterprise architects invest great amounts of time and expenditure in work related to the coherent integration of an EA's various components, which are built on the basis of different guidelines, depending on the AF used. The degree of inconsistency and ambiguity that can penetrate an EA jeopardizes the accuracy with which the architects want to communicate the corporate vision, even more considering the inherent propensity of the language to be misinterpreted. The potential consequences for a company because of mistakes in its EA's consistency and expression could result in poorly focused and improperly justified investments, to the detriment of its improvement and competitive position in the market.

With the intent of overcoming the difficulty stated herein, this article presents a strategy based on the principles of Model-Driven Engineering (MDE) and Ontologies that we have called Tartarus, which will assist the task of an enterprise architect when he is ready to model and communicate a company's actual status, opening the possibility of obtaining architectural artifacts in terms of one or more specific AFs. As an example, the article also presents a fragment of a financial institution's EA, modeled and expressed through Tartarus, which can develop architectural artifacts in terms of particular AFs.

The rest of the article is organized as follows: Section 2 describes the context that frames the problem stated herein; section 3 describes fundamental aspects of the proposal and its construction process. Section 4 details sub-sections of the strategy through a scenario. Section 5 presents some related work and the conclusions reached.

II. CONTEXT

An Enterprise Architecture (EA) can be understood as the logical process of coherently organizing business processes and technological infrastructure to describe the relationships between applications and systems, with the business objectives and motivators of an organization. An EA can be visualized as strategic plan and faithful regulator of the enterprise's principles, which encompasses the organization, processes, data and technology [17][16].

Our Tartarus proposal seeks to provide different mechanisms to define and model EAs. This objective is achieved through the coordinated use of several technologies and strategies. Among the most relevant are the use of ontologies and domain specific languages. The following section describes the main concepts and ideas regarding these technologies. Section 4 provides a detailed explanation of how these technologies are used to support the process of EA modeling and analysis.

A. Domain Ontologies

A Domain Ontology is a formal description of a set of concepts and its relationships in a specific domain [9]. They are widely used to aid the capacity to communicate by establishing a common vocabulary. At present, there are standards for the expression of ontologies, as well as languages to perform knowledge derivation queries and operations pertaining to them, such as Web Ontology Language (OWL) [29].

Different methodologies have been proposed for building an ontology. In this work, we adopted Methontology [5] [8], which establishes a sequence of steps that go from building a glossary of terms and a conceptual taxonomy to defining the set of binary relationships between the concepts, as well as their attributes, axioms and norms. Clear, structured and supported in processes of knowledge acquisition, integration, evaluation, documentation, control and quality: the result is an ontology that is made correctly in the chosen knowledge domain.

B. Model-Driven Engineering (MDE)

MDE is based on the use of a set of standards formulated by the Object Management Group (OMG) that seek, from a holistic perspective, to improve the life cycle of specification, architecture, design, development, deployment, maintenance and integration of information technologies through models [10]. One of its basic premises is the capacity of expressing, in platform-independent terms, the fundamental concepts of a problem through the use of models (diagrams, rules, restrictions) which disconnect it from the technological complexity needed to solve said problem. The use of models results in detailed analyses and more cohesive designs, enabling the derivation and automatic generation of platform-independent artifacts (even executable code) [23].

The models have to be in line with an archetype (Meta-Model) on which it is possible to define transformations that contribute to the automatic derivation of new models that propitiate its evolution and expression. The relationship between a Meta-Model and a Model is similar to the one between UML and a particular class diagram. It is said that the diagram is in line with UML, just like a Model is in line with its Meta-Model.

In MDE, the tasks consist then, in the definition of Meta-Models, which express the language and rules to create models. These models can be transformed and analyzed to obtain a new model that can be expressed in a particular technology or programming language. Following the same principle, Domain-

Specific Modeling, proposes the use of models to represent the concepts of a particular domain and their relations. Differing from general MDE strategies, models in DSM are not intended to be for general purpose use. For example, a model could be constructed to represent the tasks associated to a project. In general, a project has tasks and responsible persons associated to them, and also deadlines and products associated to those tasks. With the concepts expressed, any project could be described in a general view at the model level. However, in a DSM, the concepts become more specific. Following our example, we could talk about software projects. In this case we have analysis, design, codification, implementation and testing tasks. These concepts are now used in the model to identify in a more exact way and using domain specific terms, fundamental concepts of the problem solution.

C. Domain-Specific Languages

Domain-Specific Languages (DSL) make the comprehension of programming code more easy for the persons with expertise on a particular domain but that not necessarily have informatics knowledge. The idea is not that the final user programs using a DSL, a task that will probably will continue to be carried out by a programming expert. The idea is that the final user can easily understand the produced code semantics by seeing as part of the application elements that belong to his domain [20].

Different types of DSL exist, particularly two of them are commonly used. The first ones are the graphical DSLs, in which the user expresses his requirements using typographic elements with an unambiguous semantic. The second type, are the ones oriented to textual use, in which user requirements are expressed by using a simple text editor. Never the less, it is possible to have combinations like DSLs that integrate both the graphical and textual interface.

III. TARTARUS-EAMM: A META-MODEL FOR ENTERPRISE ARCHITECTURES

This section describes the building of the Tartarus proposal, which began with the definition of an EA ontology that establishes a detailed, homogeneous conceptual framework. Using the conceptual debugging found in the ontology, a Meta-Model was built which gives the Enterprise Architect greater capacity for expression. This model can be used for modeling architectural views independently of any AF, which in turn can give a neutral perspective of the problem, the organization and its multiple components. EA models pursuant to the Meta-Model presented can be used in accordance with the principles offered by MDE, to be transformed, derived, processed and even validated. The model's inherent properties help to facilitate the traceability and integrity of each modeled concept, providing artifacts that are consistent and adaptable to change.

A. Step 1- Defining an Enterprise Architecture Ontology

Following the guidelines of the methodology selected to build Ontologies, Methontology, the first step is to prepare and debug an EA concepts glossary. This glossary is abstracted from multiple sources, such as vocabularies from some of the most attractive and best known AFs in the market (Zachman [30], TOGAF [27], FEA [4] [3], DoDAF [28], E2AF [13]), standards and guides (IEEE [12], ITIL [14], Carnegie Mellon [18], OASIS [21]) and informational resources from organizations and academies (ICH [11], ACM [1]), among others.

The glossary obtained is used as the main input component in preparing the EA Ontology, which constitutes a comprehensive and rigorous conceptual model populated with individuals that represent basic parts of the selected sources to validate its integrity. Each concept is then listed and defined in accordance with its meaning in the aforementioned sources, showing that different AFs use slightly different words to refer to the same concept. For this reason, it is necessary to group some definitions in a way that is semantically consistent.

To continue with the steps established in Methontology, we decided to make a mental map that allowed us to see the distribution of the terms selected in their entirety, to try to find hierarchies that would infuse synergy to the work done, and therefore constitute the base conceptual taxonomy on which to build the ontology.

B. Step 2- Obtaining an EA Meta-Model

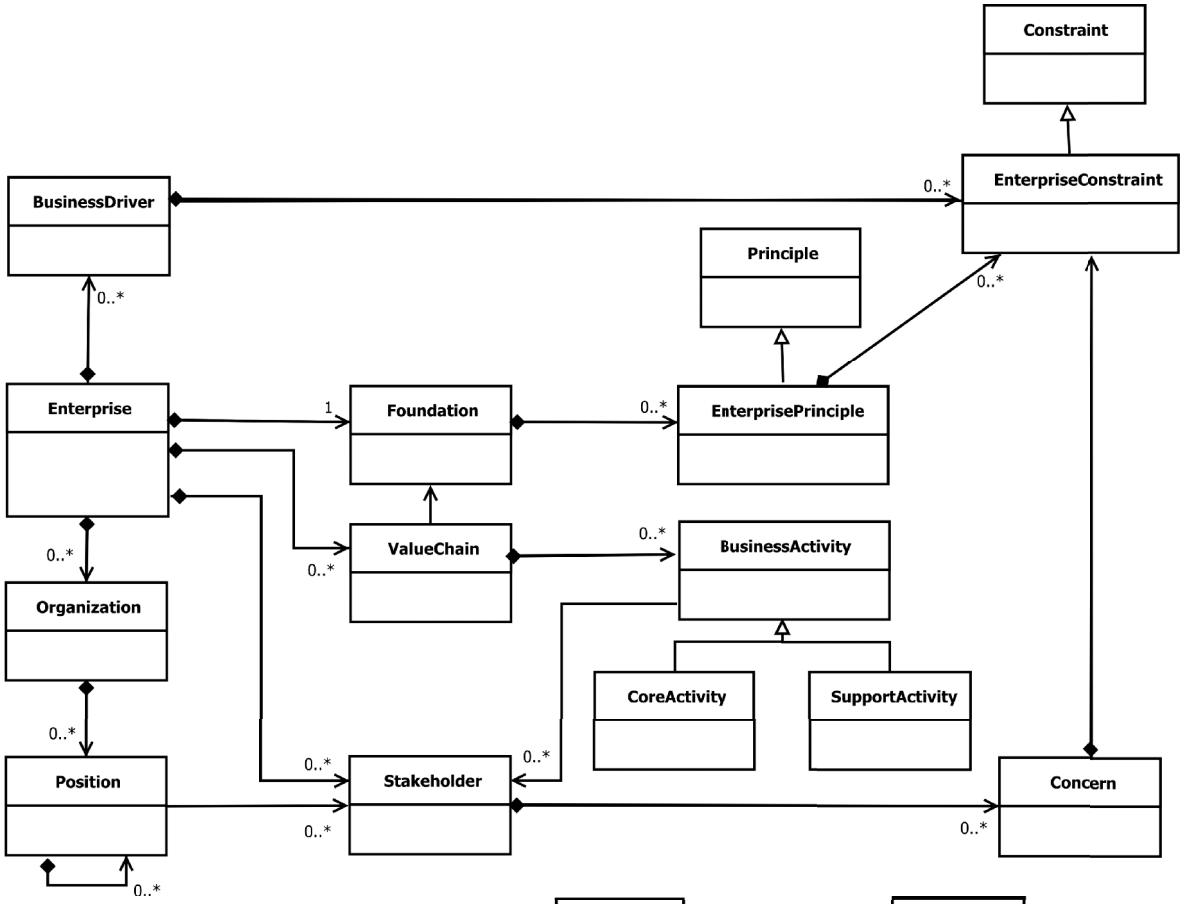
Based on the work accomplished in creating the ontology, five (5) fundamental concept groups were identified: Enterprise, Environment, Management, Continuum, and Architecture.

Enterprise, encompasses the concepts that describe the organization whose architecture will be modeled: mission, vision, value chain, capacity, business motivators and stakeholders that participate, affect and motivate the preparation of the EA. Figure 1, presents these components and their relationships.

Environment represents a set of conditions and statuses where the EA unfolds. Including, but not limited to: Human capital, Organizational Culture, Processes and Technology under which the architecture is raised, nurtured and developed. Management is the common thread of the EA, which marks the set of strategies that watch the appropriate use and evolution of the architecture's inherent artifacts. Continuum refers to the evolutionary, staggered and continuous process of maturation, development and implementation of the EA in the organization. Architecture is made up of the concepts that constitute the means to describe an organization in terms of business, information, technology and applications, and the way to validate quality attributes and scenarios, as presented in Figure 2. The architecture is expressed through views and descriptions that include deliverables (Documents, Models and other artifacts).

Revista Avances en Sistemas e Informática, Vol.8 No.1, marzo de 2011 - Medellín. ISSN 1657-7663

Figure 1. Enterprise Architecture Meta-Model



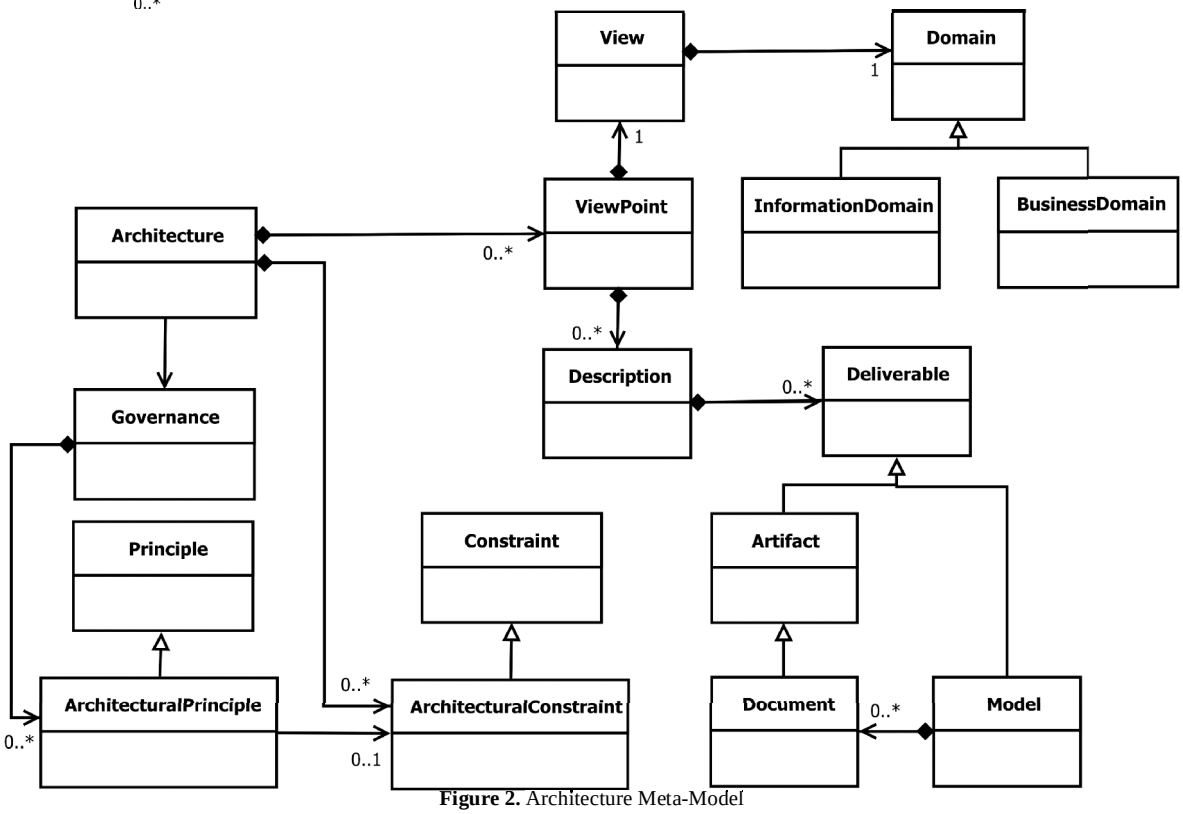


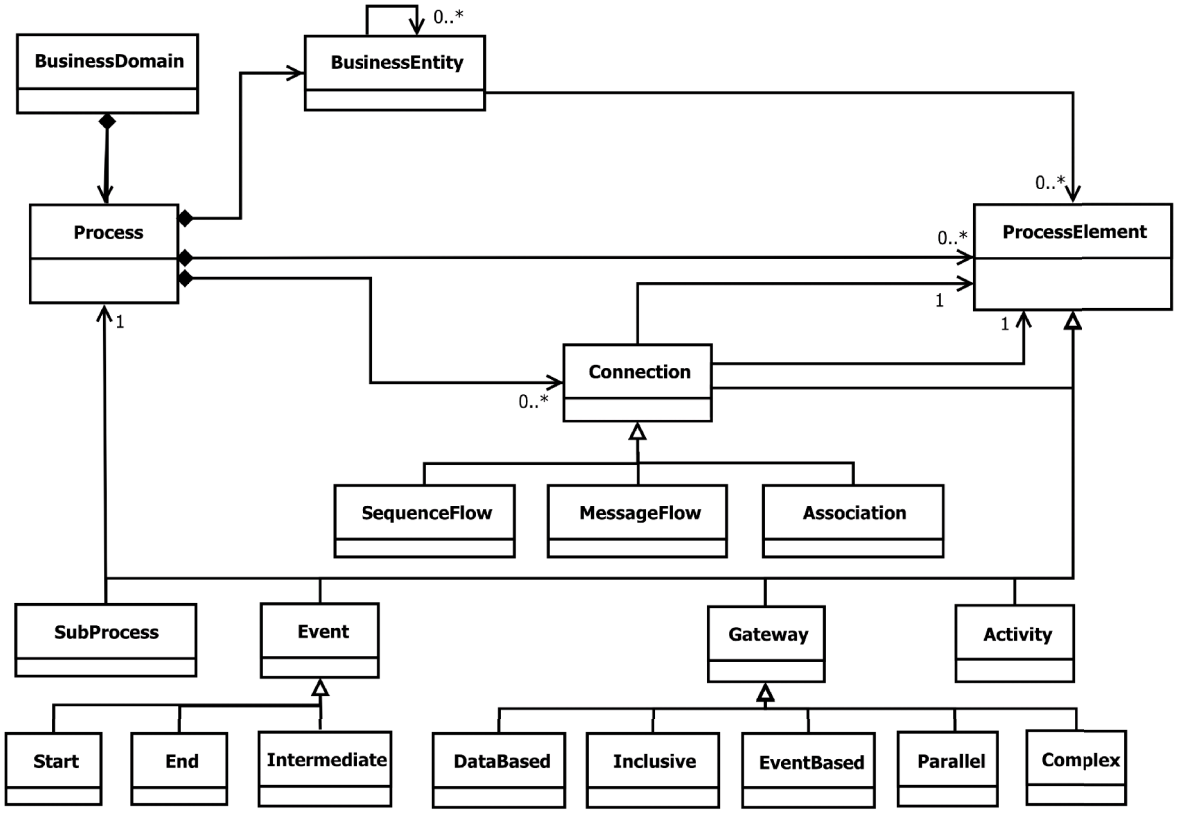
Figure 2. Architecture Meta-Model

One of the basic pillars of this last category is made up of Architectural Domains that can break down the company's complexity into specialized and interrelated views. Architectural domains are concepts commonly found in different AFs, which provide an overall vision of the company's physical, logical and structural make up. As a result of the conceptual debugging, several domains were identified, including:

--The Technology and Applications domain: Provides a vision of the infrastructure that supports the company's foundations. It is an overarching vision of the company's IT resources, which includes everything from the hardware to the services provided by software applications. It also shows how the aforementioned infrastructure is related to the company's business processes.

The Business domain defines the company's inherent business processes, which largely determine the business vertical the company is under and differentiate it from its peers. Figure 3, shows the basic concepts and relationships defined in the Meta-Model that describe business processes. It highlights the composition of a process by BusinessEntities, which represent the business players (Company, Areas, etc.), and which through Participants carry out activities (ProcessElements) linked by Connections. The Meta-Model encompasses different types of activities, events and flows derived from the BPMN nomenclature.

Figure 3. Business Domain Meta-Model



IV. TARTARUS-EADL: A DOMAIN-SPECIFIC LANGUAGE FOR ENTERPRISE ARCHITECTURES

In today's world, it is ever more common to see the use of specialized languages in the definition of Software Architecture (ADL) as a mechanism to describe relevant components of solution architecture and its relationships. However, there are few languages to define and analyze EAs in a business environment.

Tartarus gives the enterprise architect a set of tools to build, analyze and refine EAs. Additionally, Tartarus makes it possible to generate artifacts according to a specific AF. All of this is possible because of the Model-based focus used to build this tool.

To facilitate the definition and analysis of an EA model, Tartarus provides a specific language domain called Enterprise Architecture Description Language (EADL). Using EADL, enterprise architect is able to define the architecture's main

elements and their relationships, using terms and concepts inherent to the EA domain, without having to worry about particular details imposed by an AF.

The objective is therefore to provide a specific domain language (EADL) that facilitates the definition of EAs independently of any AF. These definitions are subsequently analyzed and processed to build models according to Tartarus-EAMM. The models obtained provide an abstraction of the organization in terms of its EA. And finally, the architect can derive artifacts for the preferred EA, based on the EA model. As we have already mentioned, AFs are regularly used together to complement each other, so that based on the EA model, the architect can generate the artifacts required by TOGAF and Zachman and thus enrich his business vision. Below is a detailed description of each step discussed herein, with examples of how they can be used.

A. Modeling an EA through EADL

The first step in defining a company's EA is to establish the entry point of the model. In order to do this, EADL language can create an EnterpriseArchitecture component which becomes the recipient of all the relevant concepts for the architecture. An EnterpriseArchitecture component is defined in terms of: Enterprise, Architecture, Environment, Management and Continuum, as explained in section 3.

Prog. 1, shows how enterprise (line 5) and architectures (line 6) concepts relate, within the main concept FinancialEnterpri seArchitecture.

```
1 import "enterprise/Enterprise.eadl";
2 import "architecture/CurrentArchitecture.
  eadl";
3
4 E n t e r p r i s e A r c h i t e c t u r e
  FinancialEnterpriseArchitecture {
5 enterprise { FinancialCorporation };
6 architectures { CurrentArchitecture, ... };
7 };
```

Prog. 1, EnterpriseArchitecture.eadl - Main model definition file.

Prog. 2, presents the definition of FinancialEnterprise. In the example we can see the definition of a Stakeholder (line 15), along with his pertinent drivers and position within the organization. The BusinessDriver IncreaseCreditCardSales (line 20), shows related processes (targetProcesses) and the qualification (weight) given by the architect according to the motivator's rank within organizational objectives.

```
1 import "Foundation.eadl";
2 import "ValueChain.eadl";
3 import "Organization.eadl";
4
5 Enterprise FinancialEnterprise {
6   drivers { IncreaseCreditCardSales, ... };
7   foundation { EnterpriseFoundation };
8   experience { FinancialValueChain };
9   chart { FinancialOrganization };
10  stakeholders { ChiefExecutiveOfficer,
  ExecutiveBoard,
11 RegionalManager, BranchManager, BranchEmployee
  };
12 continuum {};
13 };
14
15 Stakeholder ChiefExecutiveOfficer {
16  drivers { IncreaseCreditCardSales, ... };
17   occupies { GeneralManagement };
18 };
19
20 BusinessDriver IncreaseCreditCardSales {
21  description: "Increase in sales of credit
  card products.";
22   weight: 5.0;
23  targetProcesses { CreditCardProcess, ... };
24 };
```

Prog. 2, Enterprise.eadl - Company's foundation definition.

EADL operates under a composition structure. This means that each language component is made up of simple attributes (line 21) and complex components that refer to other components defined in the same file or in imported files (lines 7, 8 and 9).

Architecture is one of the other five (5) main concepts that can be defined within an EnterpriseArchitecture. This concept establishes the starting point for definition of architectural domains explained in section 3. A Domain relates to an architecture through a ViewPoint and a View. The viewpoint is also related to a Description that allows the enterprise architect to establish Deriverables (Model and/or Document) which would be used as its representation. EADL supports the definition and association of several architectures, allowing the architect to compare different solutions.

Prog. 1 - line 6 shows the relationship between the EA and Architecture which we have called CurrentArchitecture. The code used for this example is not listed because of space constraints, but it is listed in the next section as the model that resulted from the first transformation.

B. Transformation to Models

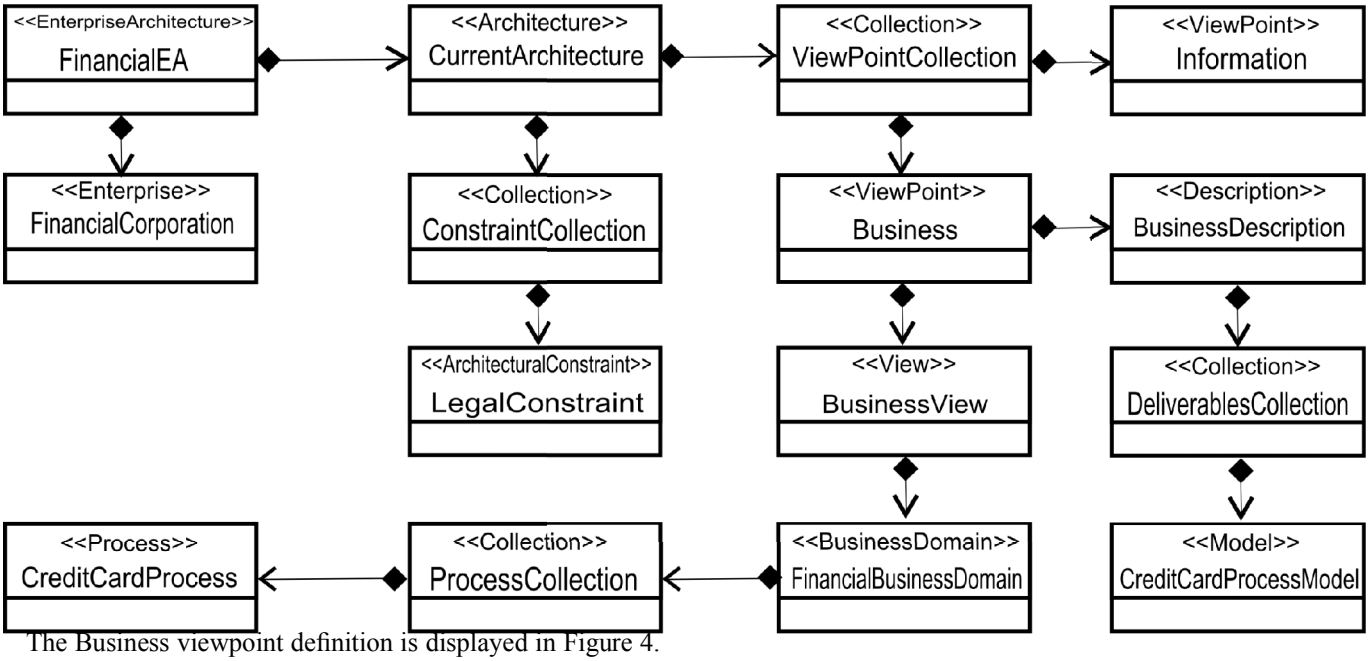
Once the architect has defined the EA through EADL, the next step is to pull these definitions together into a model according to the MM Tartarus-EAMM.

The EA model is obtained as a result of analyzing and interpreting the code fragments expressed in EADL, thanks to the tools provided by the chosen development environment:

OpenArchitectureWare (OAW). In our case, EADL was defined through OAW, which gives us two advantages: First, an EADL language editor, and second, an interpreter that receives EADL programs and generates a model according to the Meta-Model EAMM.

Figure 4, provides an example of a model obtained from the interpretation of an EA definition using EADL. The model obtained corresponds to the Meta-Model and will allow the architect to analyze, validate and refine the organization's architecture.

Figure 4. EADL graphical representation.



The Business viewpoint definition is displayed in Figure 4. It is related to two components called BusinessDescription and BusinessView. The first one defines the deliverables associated to the viewpoint. In this case, we can see a Model CreditCardProcessModel that provides guidelines for generating the artifact needed for the CreditCardProcessModel, found in the BusinessView. The view represents the viewpoint matching the corresponding BusinessDomain. As we saw in the EADL that defines the business motivators (line 23 – Prog. 2), the CreditCardProcess is related to the motivator IncreaseCreditCardSales. The MM lets the user establish relationships between concepts defined in different categories, giving an integrated perspective of the organization's business, technological and strategic aspects.

C. Artifact refinement and derivation

The models obtained from the transformation of EADL code artifacts are conformant with the MM EAMM. They enjoy inherent refinement qualities sought by MDA, which allows us to outline a new transformation with the goal of obtaining architectural artifacts as per the selected AFs.

Stakeholder	Description	Positions	Concerns	Business Drivers
Chief Executive Officer	Highest-ranking corporate officer	General Management	Effective translation of drivers, goals and objectives into IT architecture	Increased Credit Card Sales
Executive Board	Board of directors who jointly oversee the activities of the organization	Stockholder		
Regional Manager	Oversees a geographical location	Regional Management	Traceability of business processes.	
Branch Manager	Oversees a branch office	Bogota Branch Management	Real time customer information queries.	
Branch Employee	Person hired t o provide services to the company	Bogota Branch Cashier		

Figure 5. TOGAF Stakeholder map matrix.

Revista Avances en Sistemas e Informática, Vol.8 No.1, marzo de 2011 - Medellín. ISSN 1657-7663

For example, included in TOGAF guidelines are a large number of artifacts related to each phase. For Phase A, Architecture Vision, TOGAF has the Stakeholder Map Matrix artifact which describes the stakeholders, indicates their positions in the organization and also their concerns and business motivators they pursue. In Prog. 2 - line 15, the architect defined a stakeholder called ChiefExecutiveOfficer. Figure 5 shows an example of how this Stakeholder and his attributes are transformed to be part of the artifact suggested by TOGAF.

The MM helps to define deliverables generated from the defined components, thanks to the Deliverable definition. These can be specified as static and/or dynamic diagrams, catalogs, charts, matrices, etc. It is even possible to specify the export format; for example, the matrix in Figure 5, could be exported to a spreadsheet or as a PDF report.

Another diagram specified in TOGAF Phase A is the value chain. In Prog. 2 - line 8, the architect related the FinancialEnterprise with its ValueChain using the experience attribute. Figure 6, shows how the concept is transformed to the artifact, clearing differentiating the CoreActivity from the SupportActivity.

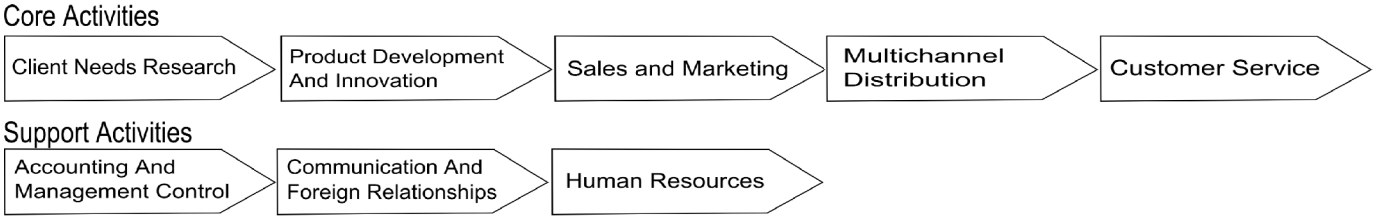


Figure 6. TOGAF Value chain diagram.

Artifacts in different AFs can be defined in the same way for complementation. The architect can create as many Deliverable as he deems necessary.

Together with the artifacts already defined for TOGAF, he can add definitions used by Zachman, such as Process Identification which is a catalog of the organization's processes, or Organization Configuration, which lists roles or positions along with job responsibilities within the organization in a matrix, etc.

V. RELATED WORK

Many efforts to consolidate ontological approaches in EA have been outlined in recent years (See [22], [15], [24], [26], [7], [2]).

The degree of AF complementary is illustrated in [25], with a case study, to determine how four AF (Zachman, TOGAF, FEA and Gartner) would address it, with a resulting comparison table by criteria which shows the major differences existing in the approximations. However, the conclusion recommends a mix of the different AFs, given the degree of complementarity among them, and to take advantage of this scenario for the needs of the company. Tartarus encompasses a set of mechanisms that enable the AFs' synergy, facilitating the work of enterprise architects by providing tools derived from a comprehensive ontological study to express and communicate EAs consistently.

Mechanisms to effectively build business models have been postulated in recent years. In particular, we point out interesting contributions made by MEMO (Multi Perspective Enterprise Modeling) [6], which has, among other elements, a set of visual languages aimed at building interrelated models that describe various aspects of a company (Business processes, organizational structure, etc.).

The tool defined in Tartarus (EADL), unlike MEMO, allows the construction of text models based on concepts that emerged from the EA ontology. Models built with EADL (in accordance with Tartarus EAMM) are transformed to be expressed in terms of particular AFs, and could even be derived to be expressed in terms of MEMO to be able to get visual perspectives of interrelationship between business models.

VI. CONCLUSION

The set of contributions that we are offer the community, which is the Tartarus proposal, is made up of the following:

--A comprehensive conceptual debugging of the terminology inherent to the EA knowledge domain, built on multiple sources of information that have good market recognition and standing.

--The EA Ontology is built following strict methodological and procedural norms. This Ontology is the mechanism which helped to outline a semantic agreement to reduce the conceptual ambiguity surrounding EA, thus setting solid bases of knowledge for the other elements of or Tartarus proposal.

--The definition of an EA Meta-Model, built on the Ontology. This Meta-Model is the archetype of infinite EAs which can express fundamental aspects of a Company, its Architecture,

Views, Domains, Processes, Information, and Resources through models, without introducing particular influences inherent to AFs. The models based on the EA Meta-Model do not tend to favor intrinsic aspects of any AF, giving the enterprise architect the chance to build a neutral EA, without any particular tendencies, and which reflects the organization's implicit reality.

--A set of transformations between models, which enable the derivation of architectural artifacts in terms of particular AFs, with a view to facilitating communication and transmission of an EA. The enterprise architect can then take advantage of the AF's complementarity, deriving models that are consistent with the defined EA and the AFs chosen to express it.

--The definition of a DSL to build EAs, which we have called EADL. This language can be used to quickly build architectural models with a broad capacity for expression, using the terms found in the Ontology and which make up the Meta Model proposed in Tartarus.

A validation of the proposal was made in a limited scenario which is common to financial institutions. The validation showed the coherence of the models created by Tartarus to express an EA, by changing to consistently express subsections of the organization's business domain in terms of particular AFs.

The components that comprise Tartarus are only the beginning of a set of proposals that we are currently conceptualizing and developing. We want to take advantage of the models' capacity for expression to capture a business vertical according to the EA Meta-Model and to build a model-guided product line (MD-SPL) in the EA's knowledge domain. This MD-SPL could manage business motivators, market restrictions, external forces, etc., as variables that allow the introduction of modifications to the modeled business vertical, resulting in EAs that satisfy organizational idiosyncrasies or particulars. The EAs generated could serve as a baseline for enterprise architects to perform automatic variations for enterprise studies and analysis on architectural artifacts expressed in models.

Other possible uses and specializations for Tartarus involve the introduction of elements to resolve situations that come up on a daily basis, whereby, despite efforts to ensure consistency of an EA's artifacts, documents inevitably lose validity at breathtaking speeds. By having the EA expressed in models, it is possible to perform verifications and analyses of the consistency between its components.

Countless scenarios can emerge from Tartarus or other MDE approximations to solve everyday problems in the areas of EA administration, derivation and analysis, which are becoming more demanding and require innovative solutions very quickly.

REFERENCES

[1] ACM., 2009. The ACM digital library. Available: http://portal.acm.org/portal.cfm

[2] EAS Enterprise architecture solutions., 2000. The essential project. Available:
http://www.enterprise-architecture.org

[3] FEA-PMO., 2006. Federal transition framework metamodel reference version 1.0. Available:
http://georgewbush-whitehouse.archives.gov/omb/egov/documents/FTF_Metamodel_Reference_Pilot_Final_December_2006.pdf

[4] FEA-PMO., 2007. FEA consolidated reference model document version 2.3. Available:
http://www.whitehouse.gov/omb/assets/fea_docs/FEA_CRM_v23_Final_Oct_2007_Revised.pdf

[5] Fernández-López M.; Gómez-Pérez A. y Juristo N., 1997. Methontology: from ontological art towards ontological engineering. En: Proc. Symposium on Ontological Engineering of AAAI.

[6] Frank U., 2002. Multi-perspective enterprise modeling (MEMO) - Conceptual framework and modeling languages. En: HICSS 2002: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS'02), Vol. 3.

[7] Franke U.; Hook D.; Konig J.; Lagerstrom R.; Narman P.; Ullberg J.; Gustafsson P. and Ekstedt M., 209. EAF2- a framework for categorizing enterprise architecture frameworks. En SNPD 2009: Proceedings of the 2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing, Washington, DC, USA, IEEE Computer Society, pp. 327-332.

[8] Gómez-Pérez A.; Fernández-López M. and Corcho-Garcia O., 2003. Ontological Engineering with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. New York: Springer-Verlag Inc., Secaucus, ISBN: 1852335513.

[9] Gruber T.R., 1993. A translation approach to portable ontology specifications. En: Knowledge Acquisition. Vol. 5(2), pp. 199-220.

[10] Guttman M. y Parodi J., 2006. Real-Life MDA: Solving Business Problems with Model Driven Architecture (The OMG Press). San Francisco: Morgan Kaufmann Publishers Inc.

[11] ICH., 2009. ICH architecture resource center. Available: http://www.ichnet.org

[12] IEEE., 2000. IEEE recommended practice for architectural description of software-intensive systems. IEEE Technical report.

[13] IFEAD., 2006. Extended enterprise architecture framework essentials guide, version 1.5. Available:
http://www.enterprise-architecture.info/Images/E2AF/ExtendedEnterpriseArchitectureFrameworkEssentialsGuidev1.5.pdf

[14] ITIL., 2006. ITIL open guide. Available: http://www.itlibrary.org

[15] Kang D.; Lee J.; Choi S. y Kim K., 2010. An ontology-based enterprise architecture. En: Expert Syst. Appl, Vol. 37(2), pp. 1456-1464.

[16] Land M.; Proper E.; Waage M.; Cloo J. and Steghuis C., 2008. Enterprise Architecture: Creating Value by Informed Governance. Springer Publishing Company, Incorporated.

[17] Lankhorst M., 2005. Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer.

[18] Lewis G.; Comella-Dorda S.; Place P.; Plakosh D. y Seacord R., 2001. An enterprise information system data architecture guide. Available: http://www.sei.cmu.edu/reports/01tr018.pdf

[19] Martin R. y Robertson E. L., 2003. A comparison of frameworks for enterprise architecture modeling. En: Conceptual Modelling, Vol. 2813 of Lecture Notes in Computer Science, Springer, pp. 562-564.

[20] Mernik M.; Heering J. y Sloane A.M., 2005. When and how to develop domain-specific languages. En: ACM Comput. Surv, Vol.37(4), pp. 316-344.

[21] OASIS., 2010. Oasis wiki. Available: http://wiki.oasis-open.org

[22] Ohren O. P., 2005. An Ontological Approach to Characterizing Enterprise Architecture Frameworks. En: Knowledge Sharing in the Integrated Enterprise. Springer Boston.

[23] OMG., 2003. MDA guide v1.0.1. Available: http://www.omg.org/cgi-bin/doc?omg/03-06-01

[24] OMG., 2008. OMG-SBVR: Documents associated with semantics of business vocabulary and business rules. Available: http://www.omg.org/spec/SBVR/1.0/

[25] Sessions R., 2007. A comparison of the top four enterprise architecture methodologies. Available:
http://www.objectwatch.com/whitepapers/4EAComparison.pdf

[26] Tang A.; Han J. and Chen P., 2004. A comparative analysis of architecture frameworks. En: APSEC 2004 Proceedings of the 11th Asia-Pacific Software Engineering Conference, Washington, DC, USA, IEEE Computer Society, pp. 640-647.

[27] TOGAF., 2009. The open group architecture framework - version 9. Available:
http://www.opengroup.org/architecture/togaf9-doc/arch/index.html

[28] US-DOD., 2007. Department of defense architecture framework version 1.5. Available:
http://cio-nii.defense.gov/docs/dodaf_volume_i.pdf.

[29] W3C., 2004. Owl Web Ontology Language guide. Available: http://www.w3.org/TR/owl-guide/.

[30] Zachman J. A., 1987. A framework for information systems architecture. En: IBM Systems Journal, Vol. 26(3), pp. 276-292.