

An Ontology-Matching based Proposal to Detect Potential Redundancies on Enterprise Architectures

Camilo Castellanos
University of Los Andes
Bogotá, Colombia
cc.castellanos87@uniandes.edu.co

Dario Correal
University of Los Andes
Bogotá, Colombia
dcorreal@uniandes.edu.co

Francisco Murcia
University of Los Andes
Bogotá, Colombia
fa.murcia68@uniandes.edu.co

Abstract—Our work presents an approach to automatic detection of potentially redundant elements within an Enterprise Architecture (EA). Unintended or unidentified redundancies can affect the data quality, duplicating efforts which may lead to inconsistencies. Evaluating and identifying manually redundant elements in large information systems (IS) is a tedious, errorprone and time-consuming task. Untimely, incomplete or inaccurate analysis, could affect the dynamics and organizational flexibility that promotes an EA. Our proposal is a MDA (Model- Driven Architecture) and Ontology-Alignment based. The main idea is to perform a transformation from an EA model towards ontology in OWL format (Ontology Web Language) and exploit ontology matching tools to infer correspondences between concepts. Our objective is to support architects analysis and making decisions with an integral view which describes EAs data and processes, and its inner relationships.

Keywords-Enterprise Architecture Alignment; Information Architecture Alignment; Ontology Matching; Redundancy Detection;

I. INTRODUCTION

The Enterprise Architecture (EA) definition improves the business and Information Systems (IS) alignment within organizations. The alignment concept is based on ideas commonly used in EA frameworks, where business and Information Technology (IT) are described, and is a key issue in business because of the impact on the entire organization. The need for duplicated data insertion in different systems, the effort required to keep multiple coherent replicas of the same information and the lack of business information are common examples of such misalignment [1].

In [2] some concerns are mentioned that arise when a company faces the challenge of aligning IT and business: i) Analyze the current situation and determine the future business strategy. ii) Document the current architecture state and design the future architecture state. These evaluations require an accurate and complete diagnostic of the actual state of the company in all its domains (organizational structure, business processes, services, applications, infrastructure and information). The EA frameworks are mostly informal, so there is a lack of EA tools that can help enterprise architects to check this alignment [3].

In particular, inside the information domain of an EA, analysis of actual information assets should be conducted to detect and / or eliminate unplanned redundancies [4]. The causes of designs with unintentional redundancies range from strategies to improve performance or IS availability, to the lack of communication and cooperation within an organization. Therefore a controlled and intentional redundancy can be differentiated from an unplanned one that leads to performance and security problems, overcosts and inconsistencies [5].

A. Problem Description

The redundant information may lead to inconsistencies, overlapping entities, and duplication of efforts in design and development [5]. Lets consider an organization with two ERPs (Enterprise Resource Planning) which need to be analyzed as part of, for example, a SIs migration process, merger or corporate acquisition. In both IS there will be concepts such as: *Supplier*, *Purchase Order*, *Bill*, etc. These concepts may not have exactly the same structure (name, attributes, data types, metadata, etc.). Similarly, processes such as *Approving Purchase Orders* and *Manage Bills*, may be common to these two systems, despite the syntactic or functional differences that may exist.

The descriptions in the EA should facilitate the identification of these matches, so it can support the migration, integration or unification strategies definition. With the current tools, architects must manually review and compare the set of artifacts that describe both systems and write down the results of their comparisons in other non-formal artifact (spreadsheets, text documents, images, etc.).

A manual alignment procedure implies a high probability of error and a large investment of time and resources [6], due to errors, omissions, or delays in the identification of such situations. The problem of finding similarities within heterogeneous schemas has been addressed from the perspective of ontology alignment, given the similarities between ontologies and database schemas [7]. Different systems of automatic ontology alignment have been proposed, however, they address the schema matching in an isolated way and the results of such comparisons are not formally incorporated

in the EA. Within its scope, information architecture relationships with processes, applications and services are not included, thus offering a partial view of the organization.

Automating the alignment task is not a trivial problem, since the exact semantics of the model is only fully understood by its designers and can not be fully expressed by the scheme itself [8]. The alignment of schemes is a task that has two main challenges: i) The size of the schemes and the number of matches to be performed are increasing. ii) The complexity of comparing data sources that have syntactic, semantic and terminological heterogeneity [9].

B. Objectives y Contributions

Our proposal main objectives are: i) Semi-automate the process of identifying redundant entities in the context of an EA, exploiting ontology alignment techniques. ii) To enrich the formal descriptions of the information architecture current state, adding the implicit relationships among its elements. iii) Supporting impact or gap analysis task, through a view that describes these relationships.

The main contributions of this work are: A EA metamodel is extended to express elements of the information domain and explicit and implicit relationships that exist among them. We developed a tool that automatically transforms information architecture models to ontologies and compares them with the aim of finding similarities. We implemented a user interface that allows to verify (confirm / edit / reject) the candidate mappings generated by the matching engine. We developed a graph-based report that displays the correlations detected inside the information domain.

C. Document Structure

The rest of this paper is structured as follows: Section II provides a study case to motivate this work. Section III describes the context in which our proposal is framed. Section IV develops the proposed solution. Section V outlines the experimental process. Section VI talks about related work and finally Section VII offers the conclusions.

II. MOTIVATION

In order to better describe the problem, let's take as an example two ERP systems from an organization used in the Enterprise Architecture Laboratory at the University of Los Andes. Figure 1 shows a part of the two SI involved: On the left the ERP1 system is described and on the right the ERP2 system is depicted. These views correspond to artifacts that formally describe the architecture. Business processes are expressed in BPMN (Business Process Modeling Notation). Information assets are defined in an entity relationship model.

In the scenario of companies acquisitions, companies mergers or SIs migration processes is key to identify redundancy or correlation of these elements in the different

ERP1 Element	EPR2 Element	Sim
Create Person	Create Account	0.9
Activate Person	Approve Account	0.8
Person	Customer	0.9
Person	Account	0.75

Table I
CANDIDATE MAPPINGS BETWEEN ERP1 AND ERP2

domains involved. The figure shows some coincidences (dotted lines) that should happen after an alignment process. At a process level, some tasks can be duplicated, despite some syntactic or structural differences. For example, described business processes share similar tasks: *Create Person* \simeq *Create Account*. In the information domain, the entities that support business processes can be overlapping, like *Person* \simeq *Account* and *Person* \simeq *Customer*. Identifying these coincidences is a key part of the evaluation that the architect makes so he can take decisions like: Combine, synchronize or unify processes and / or entities.

The artifacts of the current EA frameworks that describe these components are designed to be read and understood by people, therefore do not provide tools to support this analysis. As a result, these comparison tasks are performed manually. Recent publications discuss the automation of the alignment schemes [10] [11] [12] [13] [14] [15] and processes alignment [16] [17] [18] [19], but in an isolated way, outside the EA framework. These proposals traditionally generate a set of relationships between elements from each domain. Table I shows a possible output generated by these tools. These results are not formally incorporated into the EA, but become a part of the heterogeneous set of artifacts (text documents, images, charts, spreadsheets).

III. BACKGROUND

A. Enterprise Architecture

An Enterprise Architecture (EA) offers an integral and structured way to describe an organization, its IS, and how these are integrated to achieve the business goals supported by Information Technologies (IT). This description is made out of documents, diagrams and other artifacts that formalize different points of view of the organization, so that they are a reference and support for decision making. Nowadays, various AEs frameworks are widely used, like Zachman [20], Department Of Defense Architecture Framework (DoDAF) [21] and The Open Group Architecture Framework (TOGAF) [22]. These frameworks have in common the disaggregation in EA dimensions: i) Business Architecture defines the strategy, governance, organization and key business processes. ii) Data Architecture describes the logical and physical organization data assets structure and the data management resources. iii) Application Architecture provides a model for applications to be deployed, their

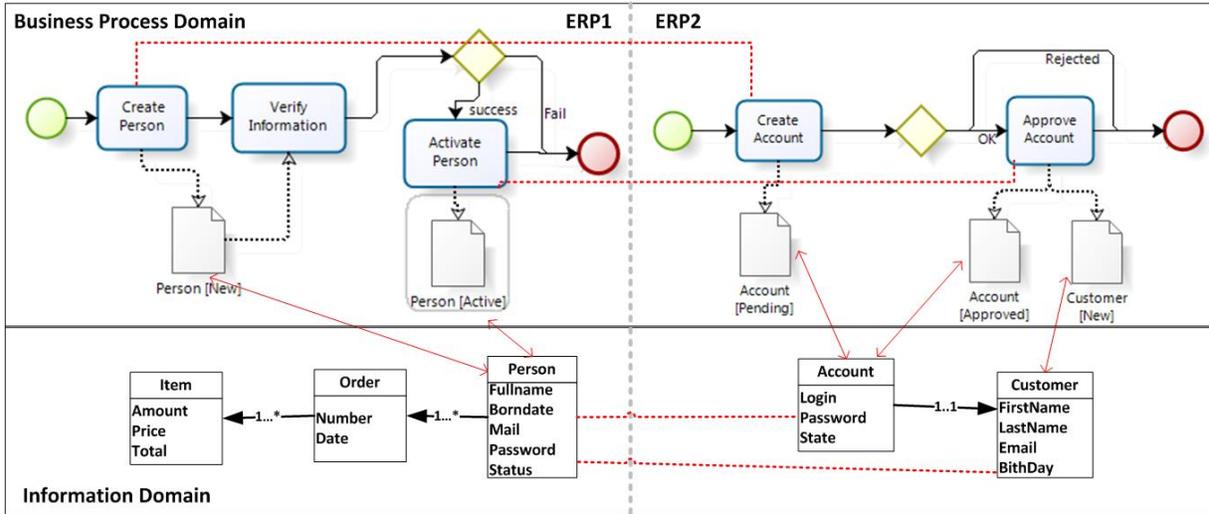


Figure 1. Example of business and information domain alignment

interactions and relationships with key business processes of the organization. iv) Technology Architecture describes the hardware and software capabilities that are required for the deployment of business services, data and applications.

Data architecture domain is the focus of our contribution and has as an objective to define types and data sources needed to support the business, so they can be: Complete, consistent, stable and understood by users [22].

B. Tartarus: An EA Metamodel

Model-Driven Architecture (MDA) is a proposal by the OMG to address software development by providing a set of guidelines to structure specifications expressed in models. It is neutral in terms of technology and provider, and seeks to significantly reduce the development effort, separating the system architecture from the platform architectures. One of the MDA key elements is the Platform Independent Model (PIM) that describes the structure and behavior of a system, but not their implementation. The particular platform implementation (JEE, .NET, etc.) is defined in a Platform Specific Model (PSM), which originates from the PIM. To realize this conversion, changes are made based on detailed templates for each platform, which map elements of the PIM to PSM elements.

Tartarus is a MDA approach, for the EAs analysis, of the *Moosas Project* at the University of Los Andes [23]. Its core is a metamodel that allows the definition of EA models. Tartarus arises as a solution choice to the current variety of frameworks, standards, tools and formats that are part of the definition of an AE [24]. A first phase of this project focused on finding differences between business processes.

Figure 2 provides an overview. The project is comprised of five packages: *Enterprise* contains the structure, value chain, principles, organizational incentives and other strate-

gic elements. *Continuum* meets the definitions to describe how the EA evolves. *Management* has the necessary factors to evaluate the artifacts that form an architecture. *Environment* is the elements set that describes the environment in which business operates. *Architecture* brings together the key concepts to visualize and structure the EA. *Architecture* package is divided into three domains: *Business Domain*: Describes the business processes. *Technology Domain*: Includes software and hardware capabilities that support services and information business. *Information Domain*: Structures data components that are a part of the company information.

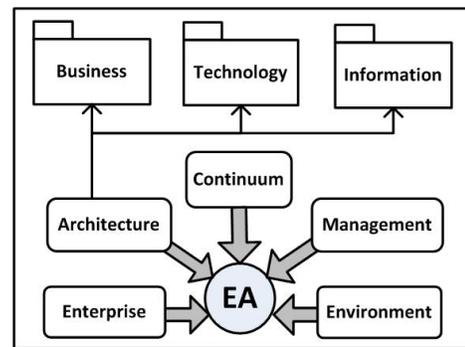


Figure 2. Tartarus overview and architecture domains

C. Ontology

Ontology is, basically, an explicit description of a specific knowledge domain, defined in terms of their concepts, properties, attributes, constraints and individuals [25]. Formally we can define ontology as: $O = \{C, P, H^C, H^P, A^O, I, R^I\}$. Where C is the set of concepts,

P is the set of properties and H^C is the hierarchy of relationships between concepts such that $H^C \subset C \times C (c_i, c_j) \in H^C$ denotes that the concept of c_i is a subconcept of c_j . In the same way H^P defines the hierarchical relationships among properties. A^O is the set of axioms. I refers the set of individuals, meaning, concepts and properties instances who are associated by relational instances R^I . One of the main advantages of ontologies is to provide useful features for intelligent systems, knowledge representation and engineering [26].

D. Ontology Alignment

An ontology alignment function can be formally defined as: $f(O_1, O_2) = \{e_{i1}, e_{i2}, i_i, r_i\}$ [9] [27], where O_1 and O_2 are input schemes/ontologies, respectively called source and target. e_{i1} and e_{i2} are two compared entities; i_i corresponds to the index of similarity or confidence (measured between 0 and 1); and r_i is the relationship (*equality*, *specialization*, *generalization*) that may exist between e_{i1} and e_{i2} . Detecting similar elements among different information sources is also a core need in assessment, migration, and integration processes, evolution of SI, exchange of information in P2P systems and composition of web services [9].

In 2004 Ontology Alignment Evaluation Initiative (OAEI) arises, an initiative that annually evaluates ontology alignment systems. The OAEI objective is to compare different proposals, with the aim of offering conclusions about the best techniques and strategies, for which, it is provided some test cases on which different systems experiment. Among the evaluated issues we can find: The benchmark track, The directories and thesauri track, Instance matching. Performance on these tests is typically measured under three indicators: *Precision* refers to the fidelity and expresses the proportion of effective elements of total items retrieved. *Recall* is associated with completeness and measures the ratio between the total of retrieved items over the total number of elements that should have been identified. The *F-Measure* combines precision and recall to determine a balance between them and ranges from 0 to 1: $F-Measure = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$.

There are several proposed methods for automatic alignment of ontologies [10] [6] [7] and our proposal includes some of them. The alignment main techniques are scheme-, content-, and combined-based. The *scheme-based* only take into account the structural information of the scheme, not its content. Within this group linguistic, textual, structural and restrictions comparisons are applied. *Content-based* strategies involve statistics, patterns or even the same data to infer correspondences. The *combined techniques* apply the above approaches for better results. This combination can be manually or automatically configure, using machine learning. This paper does not seek to determine the best way to detect possible redundancies, but to adapt and apply techniques and advances in the field of ontology alignment to the EA context.

IV. PROPOSED SOLUTION

Our proposal is a model-driven work supported by ontology alignment that semi-automatically identifies potentially redundant elements. This approach allows to formally expressing the organization data schemas in a central and homogeneous repository that favors analysis and evaluations of the organization data assets. Our contribution is a part of a work in progress, which as a first step defined the metamodel with the basic concepts that describe an EA [24]. That work also generated: A importer of processes expressed in XPDL (XML Process Definition Language) and a mechanism to find automatically the differences between two versions of a business process. Our contribution in particular, addresses the information domain, but it additionally includes base elements to the model to align other domains of EA in the future work.

The system imports the structures from an XML schema or a database connection and puts it in a model conforming to Tartarus metamodel. Starting from the model a transformation to OWL is made, to infer similarities between elements using an ontology alignment system. The results of these inferences are loaded back to the repository. Our proposal is divided into six stages and an overview is shown in Figura 3.

A. Import Schemes

Initially business processes and entities formal description is required. These definitions can be expressed manually or imported from a JDBC, XML and XPDL source, to obtain a model containing processes, entities and their relationships, in terms of Tartarus concepts. In Figure 4 our information architecture metamodel (left) is detailed, which is an adaptation of the work proposed in [28], enriched with the definitions of the inferred relationships between entities, tables comments and columns comments. The *Schema* metaclass represents the schemes contained in the EA. In our case, the ERP1 scheme becomes the instance *Schema:SI*. The *Attribute* metaclass is specialized into two subclasses: *SimpleAttribute* defines columns in the database or XML Schema primitive types, they have a data type (*INTEGER*, *DOUBLE*, *STRING*, etc.). On the other hand, *Abstract* refers to entities in a relational model or complex data types in XML Schema.

For example, the entity *Person* becomes an object *Abstract:SI.Person* and each of their fields (*Fullname*, *Born-date*, *Mail*, etc.) are objects of type *SimpleAttribute* with their respective types of data. In *BinaryAbstractAggregation*, the relationships between each pair of *Abstract* elements are defined. The relationship between the entities *Person* and *Order* is represented by the association *BinaryAbstractAggregation:Person_Order*.

The *MatchResult* metaclass maps inferred correspondences between *Abstract* elements, after the alignment execution. At point zero there are no results of alignments, these

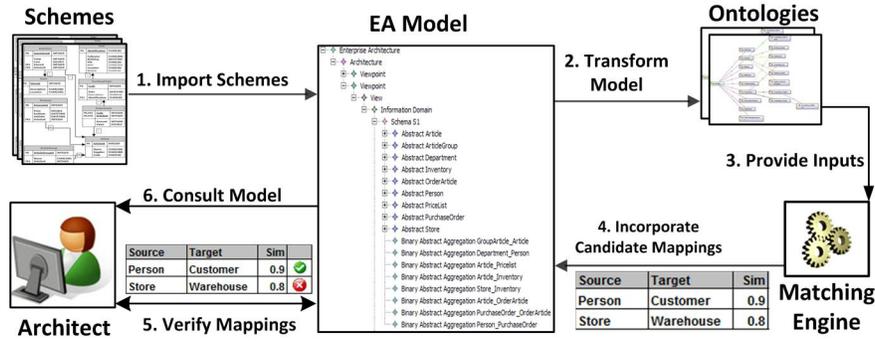


Figure 3. Solution Overview

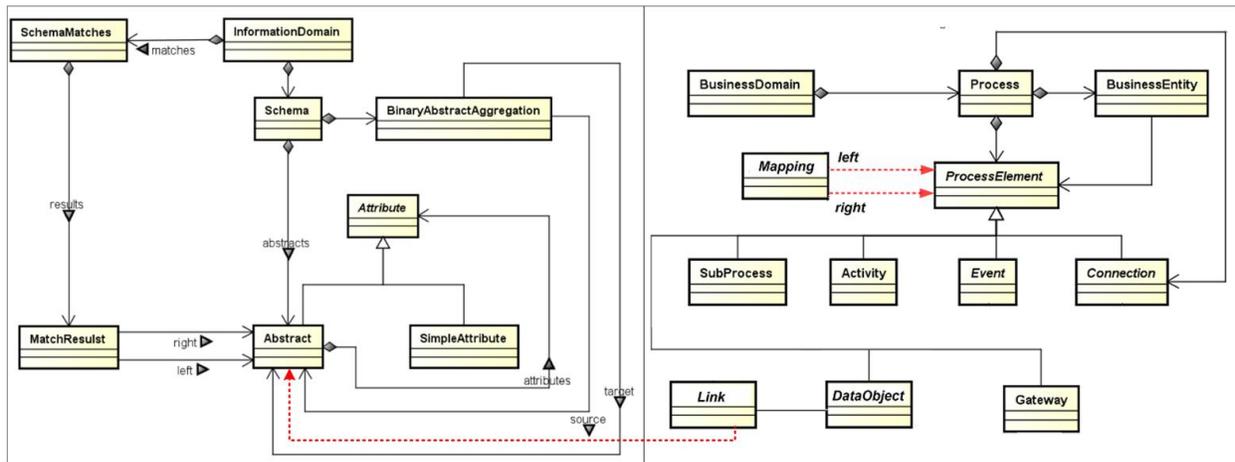


Figure 4. Information Domain and Business Process Domain Metamodels

are added after the first iteration. For example, an instance *MatchResult:S1.Person_S2.Customer* represents a mapping where, *left:S1.Person*, *right:S2.Customer*, *similarity:0.9* and *type:EQUIVALENCE*. As mentioned before, although this work focuses on the redundancy detection in the information dimension, the domain of business processes metamodel was also extended so the similarities between elements of processes (*Mapping* concept) could be expressed. Furthermore it is required to associate *DataObjects* with data entities (*Abstract*) that support them, through a *Link* metaclass. The automation of these comparisons between different domains is in progress at the time of writing this document.

B. Transform Model

Subsequently, we run a Tartarus-OWL transformation to bring all the concepts defined in the model to ontologies, see Figure 5. The decision of using OWL [29] was due to the greater expressiveness and vocabulary that provides for the description of knowledge, compared to other formats like RDF. An OWL file is generated for each schema defined in the model. Each *Abstract* object is translated

into an *owl:Class*. *SimpleAttribute* elements are mapped as *owl:DatatypeProperty* of the container OWL class. The data type of these attributes is redefined as primitive *XMLSchema* data. If the attribute is also marked as an identifier, an element *owl:FunctionalProperty* is included. Abstract type attributes are transformed into elements *owl:ObjectProperty*, where the domain is the container class and the range corresponds to the *Abstract* attribute. *BinaryAbstractAggregation* instances also become *owl:ObjectProperty*, where source and destination *Abstract* equated to domain and range respectively. Finally, the *Attribute.remarks* are transformed as *rdfs:comment* of OWL classes and properties.

C. Provide Inputs

The next step is to provide input ontologies to the alignment system. In order to do this, we made a combinatorial in which, from a universe of n ontologies we take a subset k of two elements (source and target ontology) to compare them. The total number of comparisons is given by the binomial coefficient $C(n, k) = \frac{n!}{k!(n-k)!}$. In a case with three schemes, whole aligning task requires three comparisons:

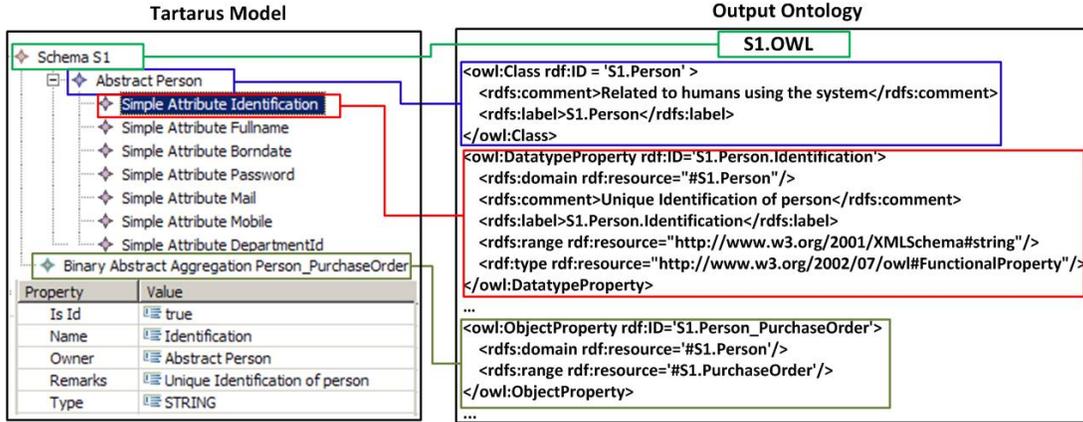


Figure 5. Tartarus-OWL Transformation

$S1xS2$, $S2xS3$ and $S1xS3$.

The matching engine, that is currently part of our solution, is AgreementMaker [13]. This decision was supported in terms of availability, documentation, algorithms variety, scalability and obtained results¹ in the last OAEI campaign [30]. For each pair of ontologies/schemas, we apply a matchers set that are already implemented within AgreementMaker. Each algorithm must be configured with parameters such as similarity threshold and cardinality. These techniques use names, comments, tags, data types and ontologies structures to determine degrees of similarity. With the outputs of each comparison, we conducted a post-processing that consists in applying transitivity relationship between different schemes alignments, to infer new associations.

To better explain the post-processing, we assume that we apply a set of algorithms $A = \{a_1, a_2, a_3\}$ over the schemes $S1$, $S2$ and $S3$. As a result we get three mappings $S1xS2 = \{S1.Person \simeq S2.Customer, S1.Store \simeq S2.Warehouse\}$, $S1xS3 = \{S1.State \simeq S3.Province\}$, $S2xS3 = \{S2.Customer \simeq S3.Client, S2.Bill \simeq S3.Invoice\}$. Note that the $S1xS3$ mapping did not include correspondence $S1.Person \simeq S3.Client$, but based on the mappings $S1 : Person \simeq S2 : Customer$ and $S2 : Customer \simeq S3 : Client$, and applying transitivity, we can infer: $S1 : Person \simeq S3 : Client$.

D. Incorporate Candidate Mappings

The results of each comparison are loaded back to the Tartarus model in a *Pending* state, which must be confirmed later by the architect. To bring the candidate mapping to the model, we implemented a tool using Eclipse Modeling Framework (EMF) which runs the model and the output alignments, to assign a similarity index and the type of relationship between each pair of objects. Thus the relation $\{S1.Person \simeq S2.Customer, sim : 0.9, type :$

$EQUIVALENCE\}$ is added to the model as an object $MatchResult:S1.Person_S2.Customer$.

E. Verify Mappings

Once the candidate mappings are calculated, they must be verified by the architect. The system presents a user interface with an inferred relationships and similarity index table, allowing him to approve, modify the relationship type or reject the mapping. After the expert confirmation, these checks become permanent (*Verified* state) in the model.

F. Consult Model

Once the alignment results are confirmed, a *Tartarus-dot* transformation generates a graphical report. *dot* is a domain specific language that allows drawing directed graphs as hierarchies [31]. Mappings are expressed as directed graphs, to provide the expert the view of correlations between the different model elements. Figure 6 shows the transformations of Tartarus elements to a dot elements: *Schema* to *Cluster*, *Abstract* to *Node* and *MatchResult* to *Edge*. Additionally some graphic conventions were added for ease of interpretation.

V. EXPERIMENTATION

In our experiment, we continue with the study case worked in Section II, extending it to all the elements of the scheme and including two additional SI. We have developed a proposal prototype as an Eclipse project. This project contains the metamodels, transformations and import and alignment tools. Using this prototype we have implemented a two phases experiment: First phase evaluates each of the transformations in terms of performance and accuracy of the alignment process. Second phase compares the results accuracy by applying transitivity between different alignments. In this section, we hope to evaluate the levels of precision achieved with this prototype. We also measured the contribution of post-processing, to the initial technics,

¹Top three in the Benchmark Track

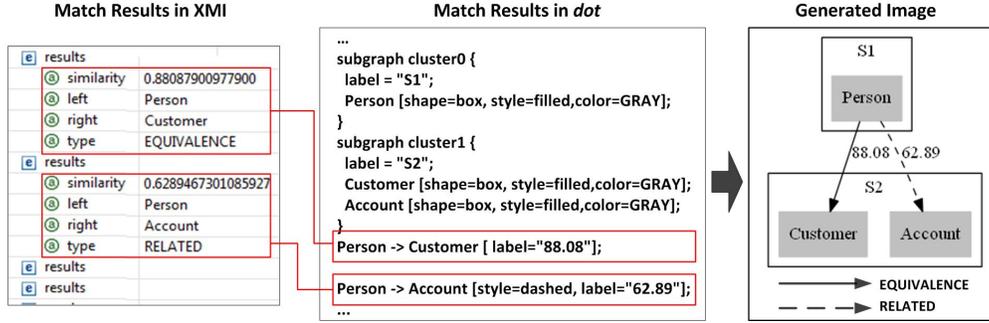


Figure 6. Tartarus-dot Transformation

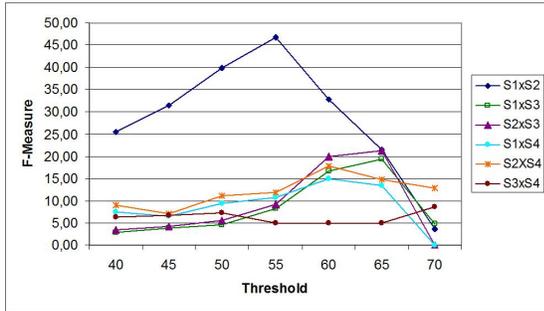


Figure 7. F-Measure for Threshold

in terms of F-Measure (see Section III-D). The machine on which the experiment was conducted is a laptop with dual core processor, 2.2 GHz, 64-bit and 4GB RAM.

A. Phase 1

In the initial test four schemes were loaded: ERP1 (S1) and ERP2 (S2) mentioned in Section II, together with a user and roles management scheme (S3) and a system CRM (S4). Through the JDBC-EMF importer, we access schemes to populate the Tartarus model. The tool loaded the structure in 922 ms from the database, comprised of: 48 tables and 168 fields distributed in 4 schemes. 72.7% of the items had comments, which were also included in the model to further support the process of alignment. The resulting model consists of 260 entities and 1,175 attributes. The next step was the execution of Tartarus-OWL transformations. The process output was 4 ontologies, one for each schema for a total of 48 *Class*, 168 *DataProperties* and 47 *ObjectProperties* in OWL format, with their respective annotations. The average execution time of these transformations was 490 ms.

The alignment was done programmatically in Java, invoking algorithms iteratively already incorporated in the AgreementMaker engine. Six comparisons were processed between different ontologies in 57,130 ms. The algorithms were the ones selected in the AgreementMaker proposal in the OAEI Benchmark Track 2010 Campaign: *Advanced Si-*

milarity uses a character comparison technique. *Parametric String* makes a pre-processing where it cleans, lemmatizes and tokenizes text for implementing syntax comparison. *Multi Words* is a technique based on vector space similarity where close concepts are taken into account. *Lexical Synonym* compares synonyms and hypernyms. *Partial Graph* makes structural comparisons based on graphs and *Linear Weighted Combination*, an algorithm that combines the results generated by the above, calculating an average of similarity. We vary the similarity threshold to evaluate the achieved accuracy in each of the six comparisons (S1xS2, S1xS3 ... S3xS4). The obtained results were compared against the reference alignments, which are golden rules defined by an expert. We use a reference alignment to evaluate the results accuracy. Figure 7 relates the levels of accuracy (F-Measure) achieved in each comparison for different similarity thresholds.

We could verify how in every comparison of ontologies pairs, different levels of accuracy are achieved under the same similarity threshold. Empirically we found that the ontologies/schemas that belong to distant domains, i.e. offered less common elements (i.e., S3xS4), have very low levels of accuracy (less than 10%). On the other hand, ontologies/schemas of closer domains (i.e., S1xS2) show better indicators (up to 46%). In general, these results show lower levels of accuracy in contrast to the 89% obtained by the AgreementMaker in the Benchmark Track 2010. Our hypothesis is that the distance between the compared domains affects significantly the levels of Recall and Precision. While in the Benchmark Track ontologies of related domain are compared (for example, ontologies of the Bibliography domain) that have a lot of concepts and the same or similar terms, our work compares not close and not exhaustive domains like an ERP or CRM systems.

B. Phase 2

The second phase evaluates the results incorporating the post-processing for verification of transitive relations. The same set of threshold values used in Phase 1 was taken. We estimate for the range of threshold values (40-70), the F-

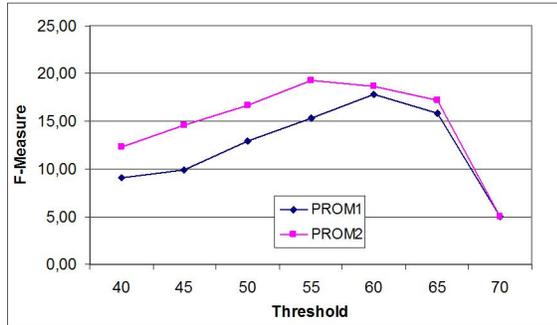


Figure 8. F-Measure Average for Threshold for each Stage

Measure average of the 6 comparisons. Figure 8 contrasts the averages obtained in Phase 1 (PROM1) and Phase 2 (PROM2).

We managed to increase the accuracy average of the alignment process for each of the evaluated levels of similarity threshold. The most significant increase occurred with a threshold of 55% where we got 4% more than what was obtained in Phase 1. We found that the best results were also obtained with thresholds close to 55%. Finally, the 2.138 mappings obtained with the best accuracy indicator, were incorporated into the model Tartarus. A total of 2.142 entities, 8.568 attributes and 4.284 associations were added to the model to store these inferences in 3.292 ms. Thus, the inferred relationships charted new links between EA elements to increase its completeness.

VI. RELATED WORK

Protégé [32] is a widely used environment for designing ontologies, and thanks to its extensible architecture, allows adding plug-ins for specific uses. Among them exists some plug-ins that let you import schemes to ontologies [33] and align ontologies [34].

Many studies have addressed the schemes alignment [6] [7] and implemented tools to automate it [10] [11] [12] [13] [14] [15]. These approaches align ontologies and export the resulting mappings to proprietary formats, spreadsheets, flat files, OWL or RDF. RiMoM [35] is a multi-strategy framework of ontology alignment that automatically combines different techniques. RiMoM was among the top three proposals of the OAEI Benchmark Track 2010 Campaign. This system allows you to configure database connections to compare schemes.

Finding similarities in business processes, has been a topic discussed by other authors. In [17] and [18] it is proposed the alignment application based on graphs and lexical to find similar activities in business process models. On the other hand, a proposal for expressing business processes with Petri nets on ontologies to realize semantic alignments is proposed in [16]. Also a framework for automatic derivation

of comparisons between two business processes is presented in [19].

Our proposal, unlike these works, integrates, centralizes and enhances the schemes alignment, since our metamodel defines the information domain and its entities, not in an isolated way but integrated with business process elements that make up the EA.

VII. CONCLUSIONS

We have presented a proposal to detect semi-automatically redundant elements in data schemas in an EA framework. Tartarus metamodel was extended to include explicit and implicit relationships between entities and business processes. We exploit a system to generate ontological alignments of these entities using different techniques. Then we incorporate the results of the EA alignment, to facilitate analysis processes in information architecture. Next we contrast the differences, in accuracy levels by comparing close and distant domain ontologies. We show evidence, such as applying transitivity, we improve the accuracy of an alignment process.

Given the accuracy rates in our experiments, we conclude that it is necessary an expert reviewing to verify and confirm the mappings in order to validate their quality. We have shown that it is possible to semi-automatic support the process of identifying potential redundancies, shortening the comparisons time and consolidating the resulting mappings. We are currently working on the inclusion of business processes domain within alignment tasks. This will enable semi-automatically aligning of business processes, and also infer links between processes and data elements.

REFERENCES

- [1] C. M. Pereira and P. Sousa, "Enterprise architecture: business and it alignment," in *Proceedings of the 2005 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2005, pp. 1344–1345.
- [2] X. Wang, X. Zhou, and L. Jiang, "A method of business and it alignment based on enterprise architecture," in *Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on*, vol. 1, oct. 2008, pp. 740–745.
- [3] A. Wegmann, P. Balabko, L.-S. Le, G. Regev, and I. Rychkova, "A method and tool for business-it alignment in enterprise architecture," in *17th CAiSE Forum, Porto, Portugal*, June 2005.
- [4] M. A. Cook, *Building enterprise information architectures: reengineering information systems*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.
- [5] P. Küngas and M. Dumas, "Redundancy detection in service-oriented systems," in *Proceedings of the 19th international conference on World wide web*. New York, NY, USA: ACM, 2010, pp. 581–590.

- [6] E. Rahm and P. A. Bernstein, "A survey of approaches to automatic schema matching," *The VLDB Journal*, vol. 10, pp. 334–350, 2001.
- [7] P. Shvaiko and J. Euzenat, "A survey of schema-based matching approaches," 2005, vol. 3730, pp. 146–171.
- [8] J. Madhavan, P. A. Bernstein, A. Doan, and A. Halevy, "Corpus-based schema matching," *Data Engineering, International Conference on*, vol. 0, pp. 57–68, 2005.
- [9] J. Euzenat and P. Shvaiko, *Ontology Matching*. Springer Berlin / Heidelberg, 2007.
- [10] J. Madhavan, P. A. Bernstein, and E. Rahm, "Generic schema matching with cupid." San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 49–58.
- [11] M. Ehrig and S. Staab, "Qom quick ontology mapping," in *The Semantic Web ISWC 2004*, ser. Lecture Notes in Computer Science, S. McIlraith, D. Plexousakis, and F. van Harmelen, Eds. Springer Berlin / Heidelberg, 2004, vol. 3298, pp. 683–697.
- [12] M. Ehrig, S. Staab, and Y. Sure, "Bootstrapping ontology alignment methods with apfel." Springer Berlin / Heidelberg, 2005, pp. 186–200.
- [13] J. Tang, J. Li, B. Liang, X. Huang, Y. Li, and K. Wang, "Using bayesian decision for ontology mapping," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 4, no. 4, pp. 243–262, 2006.
- [14] Y. R. Jean-Mary, E. P. Shironoshita, and M. R. Kabuka, "Ontology matching with semantic verification," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 3, pp. 235 – 251, 2009.
- [15] I. F. Cruz, F. P. Antonelli, and C. Stroe, "Agreementmaker: efficient matching for large real-world schemas and ontologies," *Proc. VLDB Endow.*, pp. 1586–1589, 2009.
- [16] S. Brockmans, M. Ehrig, A. Koschmider, A. Oberweis, and R. Studer, "Semantic alignment of business processes," *Proceedings of the Eighth International Conference on Enterprise Information Systems (ICEIS 2006)*, pp. 191–196, 2006.
- [17] R. Dijkman, M. Dumas, L. Garcia-Banuelos, and R. Kaarik, "Aligning business process models," *Enterprise Distributed Object Computing Conference, IEEE International*, pp. 45–53, 2009.
- [18] M. Dumas, L. Garcia Banuelos, and R. Dijkman, "Similarity search of business process models," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 32, pp. 23–23, 2009.
- [19] M. Weidlich, R. Dijkman, and J. Mendling, "The icop framework: Identification of correspondences between process models," in *Advanced Information Systems Engineering*, ser. Lecture Notes in Computer Science, B. Pernici, Ed. Springer Berlin Heidelberg, 2010, vol. 6051, pp. 483–498.
- [20] J. A. Zachman, "A framework for information systems architecture," *IBM Syst. J.*, vol. 26, pp. 276–292, September 1987.
- [21] D. A. F. W. Group, "Dod architecture framework version 1.0," 2005.
- [22] TheOpenGroup, *TOGAF Version 9*. Van Haren Publishing, 2008.
- [23] MoosasGroup, "Tartarus enterprise architecture meta-model," Jun. 2011. [Online]. Available: <http://moosas.uniandes.edu.co/doku.php?id=tartarus>
- [24] M. E. Rodriguez, F. Murcia, and D. E. Correal, "Tartarus una estrategia para construir y expresar arquitecturas empresariales," *Revista Avances en Sistemas e Informática*, vol. 8, pp. 61–70, 2011.
- [25] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology," Stanford Knowledge Systems Laboratory, Tech. Rep., Mar. 2001.
- [26] D. Gašević, D. Djuric, and V. Devedzic, *Model Driven Engineering and Ontology Development*. Springer Berlin / Heidelberg, 2009.
- [27] A. Gal and P. Shvaiko, "Advances in ontology matching," in *Advances in Web Semantics I*. Springer Berlin / Heidelberg, 2009, vol. 4891, pp. 176–198.
- [28] P. Atzeni, P. Cappellari, R. Torlone, P. A. Bernstein, and G. Giorgio, "Model-independent schema translation," *The VLDB Journal*, vol. 17, pp. 1347–1370, November 2008.
- [29] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, P. F. P.-S. Deborah McGuinness, and L. A. Stein, "Owl web ontology language reference," *W3C Working Draft*, 2004.
- [30] J. Euzenat, A. Ferrara, C. Meilicke, A. Nikolov, J. Pane, F. Scharffe, P. Shvaiko, H. Stuckenschmidt, O. Svab-Zamazal, V. Svatek, and C. Trojahn, "First results of the ontology alignment evaluation initiative 2010," vol. 10, pp. 334–350, 2010.
- [31] E. Koutsoos, E. K. Os, S. C. North, I. Compsparc, S. Sparccm, S. S. Sparcasemit, and I. I. Intnulled, "Drawing graphs with dot," 1993.
- [32] H. Knublauch, R. Fergerson, N. Noy, and M. Musen, "The protégé owl plugin: An open development environment for semantic web applications," in *The Semantic Web ISWC 2004*, ser. Lecture Notes in Computer Science, S. McIlraith, D. Plexousakis, and F. van Harmelen, Eds. Springer Berlin / Heidelberg, 2004, vol. 3298, pp. 229–243.
- [33] C. Nyulas and S. Tu, "Datamaster a plug-in for importing schemas and data from relational databases into protege," in *In Proceedings of 10 th International Protege Conference*, 2007.
- [34] N. F. Noy and M. A. Musen, "The prompt suite: interactive tools for ontology merging and mapping," *International Journal of Human-Computer Studies*, vol. 59, no. 6, pp. 983 – 1024, 2003.
- [35] J. Li, J. Tang, Y. Li, and Q. Luo, "Rimom: A dynamic multi-strategy ontology alignment framework," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1218–1232, 2009.