ORIGINAL ARTICLE

A Framework for Alignment of Data and Processes Architectures Applied in a Government Institution

Camilo Castellanos · Dario Correal

Received: 5 October 2012 / Revised: 28 November 2012 / Accepted: 20 December 2012 / Published online: 22 May 2013 © Springer-Verlag Berlin Heidelberg 2013

Abstract The alignment of business processes and information technologies (IT) is among the top concerns of IT management surveys, because it has a direct impact on the organization's agility and flexibility to change in response to business needs. Previous works have proposed frameworks and methodologies for alignment management; however, they suppose manual tasks such as applying surveys or comparing artifacts. Enterprise architecture is a valuable instrument to assess and achieve such alignment. In that direction, we introduce a framework to support the alignment of business architecture (BA) and information architecture (IA) via the comparison of their constituent components applying ontology matching techniques. Our key objectives are: (1) to support the process of evaluating alignment BA-IA and (2) to detect potential alignments and misalignments between BA and IA. We apply this approach in the Colombian Institute for the Evaluation of Education through the evaluation of misalignment heuristics.

Keywords Enterprise architecture alignment · Business—technology alignment · Alignment heuristic, Ontology matching

1 Introduction

The alignment of business and technology may be defined as a means to quantify the extent to which business needs are met by solutions provided by information technology (IT) [19].

C. Castellanos (⋈) · D. Correal
Department of Systems and Computing Engineering,
University of Los Andes, Bogotá, Colombia
e-mail: cc.castellanos87@uniandes.edu.co

D. Correal

e-mail: dcorreal@uniandes.edu.co

the agreement or coherence between the different enterprise dimensions like business process, information, applications and IT infrastructure.

Said alignment is a key issue in all organizations. Every

We understand the alignment of business and technology as

Said alignment is a key issue in all organizations. Every year, when technology directors are surveyed to identify their main priorities, the need for business and IT alignment consistently ranks among their top concerns [29]. Managing and evaluating business and IT alignment are not easy, neither in its conceptualization, nor in its accomplishment [25]. Outdated information, non-automated repetitive processes, information silos, as well as redundant processes and entities are common examples of the lack of alignment.

Enterprise architecture (EA) thus emerges as an important element in achieving the desired alignment of business and IT. However, in practice, it is found that the different domains of EA are not dealt in an integrated fashion. Each domain speaks in its own language, describing its own model and using its own tools and techniques.

In [30] some concerns are mentioned that face when a company tackles the challenge of aligning business and IT: (1) analyze the current situation and determine the future business strategy. (2) Document the state of current architecture and design the state of future architecture. These evaluations require an accurate and comprehensive 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 [31].

Previous studies [3,13,20,30] have set forth alignment models and methodologies centered on conducting surveys and tabulating the results. Furthermore, these methods provide no support for analysis based on automated tools. However, other studies [19,21,26] deal with alignment based



on the coherence among the elements of the different domains of EA, such as business (BA), information (IA), application (AA), and technology (TA) architecture. This concept of alignment is consistent with our sight presented at the beginning of this section. Therefore, this will be the road that we will follow to face the evaluating business—IT alignment.

Determining the degree of coherence requires that the components of each dimension be previously identified to compare and assess the various components through the use of heuristic rules that detect symptoms of possible faults in alignment. Some of these heuristics on business and data levels include: (1) redundancy in business processes and information assets, (2) processes that make no access to any entity, and (3) entities that are not accessed by any process.

1.1 Description of the Problem

The task of identifying alignments, or lack thereof, among the domains of EA (e.g., business processes and information) with traditional means entails the manual description, revision and comparison of a set of heterogeneous artifacts (e.g., diagrams, text documents, spreadsheets, images) that collectively describe an EA. The more elements there are in each domain of EA, the more complex the concept of alignment becomes, since more rules and heuristics have to be defined and applied to govern the relations among those elements [26]. Dealing manually with revision, comprehension, and association constitutes a time and resource-consuming approach, with a high likelihood of error. This is especially true in large organizations with complex EAs comprising hundreds of components. Therefore, this task is not only complex, but often unfeasible in practice.

The question currently driving our research is: what activities regarding process and data analysis alignment in EA can be automated, and how?

1.2 Objectives and Contributions

The main objectives of our proposal were: (1) to support the process of evaluating alignment BA–IA, (2) to automatically infer correspondences between elements in the business process and data domains, and (3) to detect potential alignments, and lack of alignment, among processes and data in an EA framework.

The fundamental contributions of this work are summarized as follows: we extend an EA metamodel using matching classes to formalize the associations among entities and business processes. We define a procedure for the alignment of business process and data elements based on ontology matching. We construct the Kalcas query language (KQL), a graphic domain-specific language (DSL) allowing the query of alignments and misalignments as found in an EA model.

This alignment query is the result of assessing alignment heuristics over a segment of BA and IA.

1.3 Document Structure

The remainder of this document is organized as follows: Sect. 2 provides a case study to motivate our approach. Section 3 describes the background on which our proposal is framed. In Sect. 4, we present our proposed solution. Our experimentation is presented in Sect. 5. Section 6 deals with related work, and finally, Sect. 7 draws conclusions.

2 Motivation

The framework under consideration has been applied at the Colombian Institute for the Evaluation of Education (ICFES) [14]. The ICFES's mission includes developing conceptual groundwork, design, construction and application of evaluation instruments aimed at students of all levels, from elementary to higher education.

We illustrate our proposal dealing with an ICFES's missional process, *Registration Process*. The corresponding business domain is described in business process management notation (BPMN) and shown in Fig. 1, and the entityrelation (ER) model illustrates the structure of the elements comprised in the information domain. To determine the extent to which the data provide support for this business process (i.e., BA–IA alignment), architects must manually compare these diagrams, using additional supporting artifacts such as a detailed description of the processes, and a data dictionary.

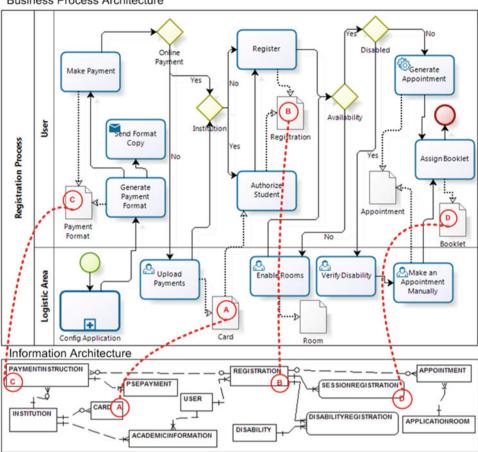
Given these descriptions, the goal is to find the correspondences, or maps, among IA and BA, for which a architect must apply a variety of techniques: *Textual comparison* is the most basic technique, pointing out relations based on the similarity of strings (see maps A and B). In turn, map C requires the use of *linguistic techniques* based on synonyms and hyperonyms. On the other hand, there are correspondences that pose a greater challenge to find, as is the case with Booklet (map D) that at first sight seems to be lacking entity support. However, a detailed inspection of the data dictionary reveals that it is found in a field SESSIONREGISTRATION. BOOKLET that stores the user's booklet number. This case implies a *textual and structural analysis*.

Furthermore, to infer redundancies in each domain we must contrast all of the elements in every process in the organization, as well as compare all of the entities in the data schemas. For example, there is another process regarding user sign up for examinations commissioned by corporate clients at ICFES, *Light Registration (P2)*. This process *P2* bears certain similarity with *Registration Process (P1)* since



Fig. 1 Diagram BPMN of registration process and underlying schema

Business Process Architecture



it deals with loading registered users, generating appointments and assign booklets. However, P2 is lighter, less restrictive and less automated than P1. We could evaluate P1 and P2, and their schemas (S1 and S2) to identify overlappings.

3 Background

3.1 Business-IT Alignment and Enterprise Architecture

The alignment of business and technology may be defined as a way to quantify the coherency level in relation to the business necessity and the solutions provided by IT [19]. Numerous works [21,22,29,30] have been concerned with alignment evaluation in terms of components in EA.

An EA provides a comprehensive and structured account of an organization, its information systems (IS), and the way in which these are integrated to meet business goals based on IT. This description is integrated by documents, diagrams, as well as other artifacts that formalize views of the organization from different points of view in a manner such that they support decision-making. Traditional frameworks for EA, such as [27,32], are similar insofar as they propose a dimensional disaggregation: (1) Business Architecture defines the strategy, governability, organization, and key business processes; (2) Data Architecture describes the structure of logical and physical information assets, as well as data management resources; (3) Application Architecture provides a model for the deployment of applications, specifying the interactions among them and their relations with the organizations' main business processes; (4) Technology Architecture describes the required software and hardware to deploy the necessary business services, data, and applications.

The problem of identifying BA-IA alignment can be formalized as a function between sets of the components comprised by these architectures, where a business component (C_i) is aligned with an information component (C_i) if there exists a correspondence above a similarity threshold (TH): $aligned(C_i, C_j) \Rightarrow C_i \in BA \land C_j \in IA \land sim(C_i, C_j) \ge$ TH.

On the other hand, we understand the definition of redundancy among the elements of each domain to be a similarity relation between components of the same domain (C_i y C_i) whose similarity index is greater than a given threshold (TH): redundant $(C_i, C_j) \Rightarrow (C_i, C_j \in BA \vee C_i, C_j \in BA \vee C_i)$ $IA) \wedge sim(C_i, C_j) \geq TH.$



The total number of alignment comparisons is given by the product set $(M \times N)$, where M is the number of elements in BA and N, the number in IA. The number of redundancy verifications is given by the binomial coefficient $\frac{n!}{2!(n-2)!}$, where n is the cardinality of the set under consideration in each domain.

To estimate how many comparisons must be conducted, let us consider a segment of ICFES's EA, comprising an IA composed of three schemas (220 tables) and a BA composed of three business processes (70 activities). A BA–IA alignment task requires the execution of 15,400 comparisons. To this we must add the redundancy evaluation, which entails 24,090 verifications in IA, and a further 2,415 in BA. This amounts to a total of 41,905 verifications required for the thorough assessment of the previously proposed alignment heuristics. These amounts reflect a great effort by a reviewing purely manual.

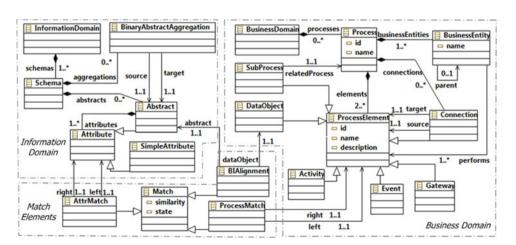
3.2 Tartarus Metamodel

Tartarus is a model-driven architecture (MDA) approach to EA analysis [17]. Tartarus originates as an option in response to the current variety of frameworks, standards, tools, and formats that integrate the definition of an EA [24]. The metamodel comprises five packages: *Enterprise, Continuum, Management, Environment* and *Architecture*. *Architecture* is divided into four domains of EA: *Business, Information, Application* and *Technology*. We shall now proceed with a description of the metamodel, detailing the Information (left) and Business (right) domains in Fig. 2.

3.2.1 Information Domain

Our information architecture metamodel is an adaptation of that presented in [1], enriched with the definitions of the inferred entity relations, table comments, and column comments. The Atzeni proposal includes a metamodel as a set of metaconstructs that can be used to define models in the most

Fig. 2 Extension of Tartarus metamodel of information and business domains



general version. Atzeni's work also explains how this metamodel is able to translate and wrap heterogeneous models [1]. Hence, this metamodel allows to describe heterogeneous data sources [e.g., XML schemas, object oriented (OO) models or relational databases (RDB)] in a unique and homogeneous repository. The Schema metaclass represents the schemas in the EA. The Attribute metaclass specializes into two subclasses: SimpleAttribute, defines the columns in the database or primitive data types in XML schemas (e.g., INTEGER, STRING), and Abstract, refers to entities in a relational model or to complex types in XML.

To explain how this information domain metamodel of Tartarus can hold other metamodels like RDB, XML, OO and ER, we present a comparison of IA metaclasses and their correspondences in the other metamodels in Table 1. These correspondences allow us to redefine in a generic way (using IA metamodel constructs) models described in those heterogeneous metamodels. Within the scope of this work we have developed an importer from RDB to IA, although IA metamodel is suitable to support other kind of models.

For instance, in our case study schema S1 becomes instance Schema: S1. USER entity in schema S1 becomes an Abstract: S1. USER and each one of its fields (e.g., NAME, DOCUMENT) are objects of the SimpleAttribute class, with their respective data types. Binary Abstract Aggregation defines the existing relations between each pair of Abstract elements. The relation between USER and REGISTRATION entities is represented by the Binary Abstract Aggregation: USER_REGISTRATION association.

3.2.2 Business Processes Domain

This domain defines the company's business processes. BPMN constructs like process elements, business entities, flow objects, and connections are highlighted in the metamodel. The metamodel deals with the different activities, events, and business process flows in the BPMN



Table 1 Correspondences between IA metamodel and other metamodels

IA metaclass	RDB	XML	00	ER
Abstract	Table	Complex type	Class	Entity
Simple attribute	Column	Primitive type	Property	Attribute
Binary abstract aggr.	Foreign key	Nested element	Class reference	Relationship

nomenclature. The DataObject concept associates data entities that are read and/or generated by activities.

In our case, the *Registration Process* corresponds to a Process-type element that contains 11 activities (Activity) connected by Connection and/or Gateway class elements. Data objects such as Payment Format and Card are stored as DataObject-type instances.

3.2.3 Tartarus Extensions

As part of our work, we have extended the business process and information metamodel to express the correlations that may arise among the different components. These correlations are traced through Match elements detailed in the bottom of Fig. 2. We designed a new metamodel package with metaclasses which define how BA elements and IA elements could relate to each other.

The Match superclass represents correspondences of elements within the same domain (potential redundancies), or across domains (potential alignments) by assigning a similarity index and an assessment state (either PENDING or VERIFIED). AttrMatch relates attribute pairs of different schemas, for instance, a coincidence between the S1.REGISTRATION and S2.REGISTERED. In a similar fashion, ProcessMatch represents potential coincidences between pairs of ProcessElement, for instance, among the P1.Register and P2.Migrate Registration. Finally, the BIAlignment subclass allows the alignment of the Information and Business domains by associating DataObject and Abstract elements.

3.3 Ontologies and Ontology Matching

An ontology is, basically, an explicit description of a specific knowledge domain, in terms of its concepts, properties, attributes, constraints, and individuals [18]. Formally we define an ontology as:

$$O = \{C, P, H^C, H^P, A^O, I, R^I\}$$
 (1)

where C is the set of concepts, P the set of properties. 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 c_i is a subconcept of c_j . Similarly, H^P defines the hierarchy of relationships between properties. A^O is the set of axioms. I comprises the set of individuals, it means, instances of concepts and properties which are associated among relational

instances R^I . One of the main advantages of ontologies is to provide useful characteristics for intelligent systems, knowledge representation and engineering [12].

Ontology matching consists in finding correspondence relations between separately designed ontologies with the goal of restoring semantic interoperability. An ontology alignment function can be defined formally: $f(O_1, O_2) = \{e_{i1}, e_{i2}, i_i, r_i\}$ [10,11], where O_1 and O_2 are input schemas/ontologies, commonly called source and target, respectively, e_{i1} and e_{i2} are the two compared entities, i_i corresponds to the index of similarity or confidence (measured between 0 and 1) and r_i is the relation (i.e., equality, specialization, generalization) that may exist between e_{i1} and e_{i2} . Detecting similar elements between different sources is also a central issue in processes assessment, migration, integration and evolution of SI, information sharing in P2P systems and web services composition [10].

Several methods exist for the automatic matching of ontologies (e.g., [16,23]), some of which have been integrated into our proposal. The main matching techniques are either schema-based, content-based, or of a combined nature. Those *schema-based* techniques deal exclusively with the structural information of the schema, disregarding its content. This group of techniques employs linguistic, textual, constraint, and structural comparisons. *Content-based* strategies deal with statistics, patterns, or even the data themselves to infer correspondences. *Combined* techniques apply the aforementioned approaches in search of better results. Combinations may be defined manually or automatically.

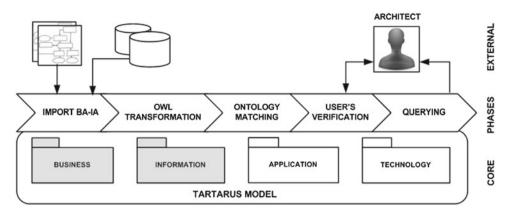
4 An Ontology Matching-Based Proposal

The core of our proposal is the Tartarus model of the organization, wherein the elements of the EA are formally expressed. The central objective is to define the BA and IA components and to apply alignment and redundancy functions supported in an engine of ontology matching to infer similarity indices among the elements in IA and BA.

Our proposal comprises five steps or phases, each of which is supported by a set of tools constructed as part of the present work. The user becomes involved by verifying the candidate mappings that the matching engine infers automatically (thus resulting in a semi-automatic approach), and by executing alignment queries on the model generated with our graphic DSL (Kalcas query language, henceforth KQL). Figure 3 provides a general overview of this proposal.



Fig. 3 Solution overview



4.1 Importing Business and Information Architectures

We must initially instantiate the organizations BA and IA models. To that end, they are imported with the use of a tool that populates the model from an XML process definition language file (XPDL) in the case of BA, or via JDBC (RDB) for IA. The final result is a model of the organization spanning the aforementioned dimensions, expressed by means of Tartarus concepts. Not only does this stage incorporate the elements of each set into the formal descriptions, it also incorporates their structure and related metadata, thus generating enriched models that favor inference making.

4.2 OWL Transformations

Subsequently, we conduct a Tartarus–OWL transformation to bring all the definitions in the model to the form of OWL ontologies. This transformation includes the models processing for organizing data in the way that is suitable for ontology matching. Figure 4 provides an example of such transformations. An OWL file is generated for every schema and process in the Tartarus model.

All ProcessElement-type elements (i.e., Activity, SubProcess, Gateway, DataObject and Event) found in the BA are transformed into OWL classes (owl: Class). On the other hand, Connection class elements become owl:ObjectProperty objects that convey relations among the ProcessElement.

In IA, each Abstract object is translated into an owl:Class. SimpleAttribute elements are mapped as owl:DatatypeProperty of the container OWL class. Their data type is redefined to be a primitive XML-schema type. BinaryAbstractAggregation instances are transformed into owl:ObjectProperties, with origin and destination of Abstract types being set into the domain and range, respectively. The comments of processes and entities are included in the ontologies in form of rdfs:comment.

After this phase, we obtain a set of BA ontologies containing the semantics for each process, and a set of IA ontologies

with the semantics for each scheme. It means, for our case study this phase produces two BA ontologies (i.e., *P1.owl* and *P2.owl*) and two IA ontologies (i.e., *S1.owl* and *S2.owl*).

4.3 Ontology Matching

This stage consists in processing the previously generated ontologies with a matching engine. The set of ontologies is processed by pairs, where each pair is an input to matching engine and each pair generates a mapping. We define two types of mappings when executing matching tasks: BA–IA alignments, BA and IA redundancies.

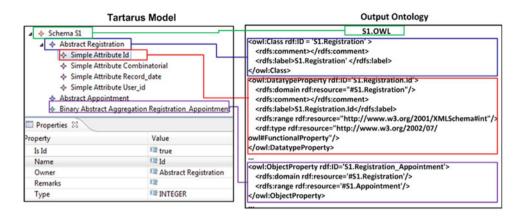
AgreementMaker [5] is the matching engine currently used in our solution. We apply a set of matchers already implemented in AgreementMaker. Each algorithm must be configured with parameters such as similarity threshold, and cardinality. These techniques make use of names, comments, labels, data types, and structures to draw a degree of similarity, which is a number between 0 and 1. This work does not contribute in ontology matching techniques; however, it exploits existing developments in this area and explores new areas of application of such techniques.

The total comparisons of alignment between IA and BA elements are given by the cartesian product of total number of elements in BA and IA. The number of checks required to find redundancies is given by the binomial coefficient $\frac{n!}{2!(n-2)!}$, where n is the number of elements in both domains (i.e., IA and BA). We can conclude that the algorithmic order of complexity of alignment function is of the form $O(m \times n) \subseteq O(n^2)$. On the other hand, the algorithmic complexity of the redundancy functions is of the form $O(\frac{n^2}{2}) \subseteq O(n^2)$. Hence, these functions have quadratic complexity.

Our implementation for ontology matching processing comprises the iterative execution of a matching task for each process ontology and each schema ontology generated on the previous phase. Firstly, to execute redundancy comparison between ontologies from the same domain, in our case study: P1.owl–P2.owl, and S1.owl–S2.owl. After, to run alignment comparison between ontologies from different domains, for



Fig. 4 Tartarus–OWL transformation



instance in our case study: P1.owl–S1.owl, P1.owl–S2.owl, P2.owl–S1.owl and P2.owl–S2.owl.

In each matching task between two ontologies, we start a matching engine instance, we assign input ontologies, we execute the set of matcher and we obtain the resulting mapping. Additionally to mapping generated directly by matchers, we implement a mapping propagation to replicate mappings found between an Entity and a Dataobject to Activities linked to DataObject. The propagation's objective is to describe how the activities are related with a DataObject, which is also indirectly related with the Entity, because these activities make use of the entity from the business level, and the DataObject serves as a bridge between these elements. To explain the mapping propagation using the case study, the mapping generated by matching engine between DataObject Registration and Entity Registration produces two additional mappings: Entity Registration with Activity Register and Entity Registration with Activity Authorize Student (see Fig. 5). This mapping addition describes, indeed, that activities Registration and Authorize Student are aligned with Entity Registration.

After the execution of the matching tasks, all candidate output mappings are loaded back into the Tartarus model as AttrMatch, ProcessMatch, or BIAlignment elements in a pending state (state=Pending), with the similarity index calculated by the engine. For instance, the inferred correspondence between the S1.Registration and S2.Registered entities is stored in the model as the AttrMatch:S1.REGISTRATION_S2.REGISTERED object with the attributes: {left: S1.Registration, right: S2. Registered, sim:0.9, state:PENDING}.

4.4 User Verification

Once the candidate mappings of alignments and redundancies have been calculated, these must de verified by the architect. To that end, we provide a graphical user interface (GUI) that presents a table with the inferred correspondences and their similarity indices, allowing the architect to approve or

reject the mapping. We developed this GUI using Java Swing library and EMF for accessing and exploring the Tartarus model. Figure 6 shows the GUI when user confirms or rejects the candidate mappings. After being verified, they become permanently set in the model, it implies setting to VERIFIED the Match.state.

4.5 Querying

As a complement to our proposal, we define the KQL, a graphical DSL that allows querying of a Tartarus model using the inferences that were confirmed in the previous stage. KQL allows the heuristics introduced in Sect. 1 to be expressed via queries. We shall now present the grammar, the graphical editor, and the type of responses that our tool generates. Figure 7 presents these elements in the KQL GUI.

KQL grammar contains the following elements: the sections of the domains (business-information) located in the work zone and the entry elements in the palette zone. The command zone contains the buttons to run the desired queries (alignment or redundancy queries). Queries are designed by dragging elements from the palette to the different domain's sections: Process, Activity (BA), Schema, and Entity (IA). These components allow us to query the system at different levels of granularity regarding business and information. Additionally, elements may take on a specific value in the model (Activity:Generate Appointment or Entity: Appointment), or else an undetermined value *All (any Activity or any Entity) when queries are being defined.

To arrange and structure these concepts, we defined KQL query metamodel using EMF with the different concepts shown in Fig. 8. The Query element contains the query inputs. These may be processes and activities in the business domains, as well as schemas and entities in the information domain. From this metamodel, and making use of the EuGE-Nia tool [9], we generate a graphical modeling framework (GMF) editor allowing queries in the KQL to be expressed graphically. This editor works as an eclipse plugin and it



Fig. 5 Mapping propagation

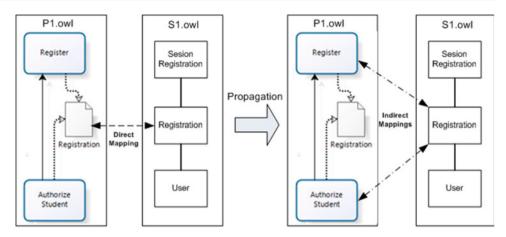


Fig. 6 User interface for mapping verification

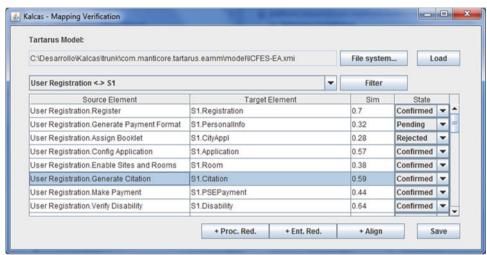
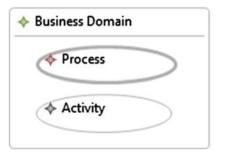
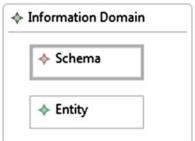


Fig. 7 Elements of KQL





allows user to design and execute alignment queries; for this the user must drag and drop components over each container.

Figure 9 illustrates a possible alignment query in our case study. We include the Registration Process and the S1 schema to determine how these two components are aligned. We select the desired level of detail for the response (Activity in BA, and Entity in IA), to finally click on the *Alignment Query* button. On the other hand, it is also possible to run redundancy queries. To that end, the user must place the desired elements of the Palette (e.g., Schema: S1 and Schema: S2) in the appropriate section, and then click on the *Redundancy Query* button.

Queries are processed over the Tartarus model, and responses are constructed by navigating the relations among components, which were identified and validated during previous stages of the process. The response is then presented using the GraphViz graph engine, which interprets *dot* source files (a graph description language) and displays them graphically. A Tartarus–dot transformation was developed to parse the result of the query into a *dot* graph.

Figure 10 presents the output of the query that whose design was previously described. The sections of the query editor representing the Business and Information domains



Fig. 8 KQL query metamodel

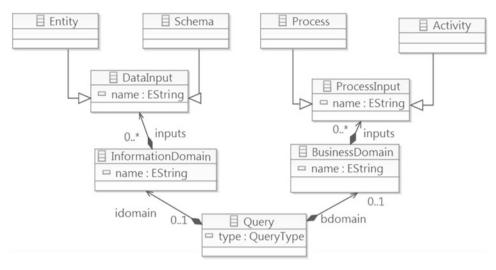
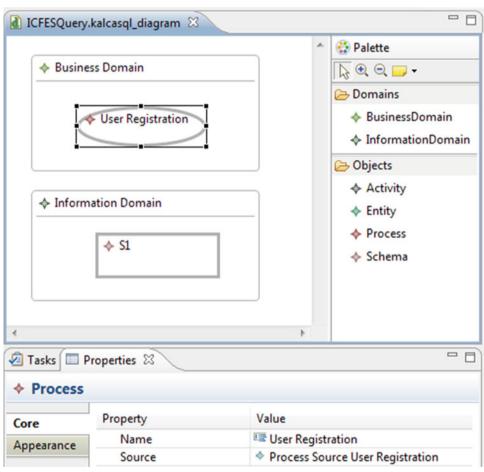


Fig. 9 Example of alignment query



are equally present in the output report, as also are the Activity, Process, Entity and Schema element conventions. Inferred relations are represented with dotted lines, while solid lines are reserved for relations given in the BPMN and ER models imported into the system. We identify three distinct categories used to identify alignments between components in the output format: *Aligned*: elements

that are supported by components of a different domain that is included in the query; *Omitted Aligned*: elements that are supported by components of a different domain that was not included in the query; *Misaligned*: elements that are not supported nor aligned with components of other domains. The output format for redundancy queries is similar, with the sole difference that only elements with duplicity associa-



tions indicated with dotted lines in each domain are processed (Fig. 11).

Potential misalignments, as defined in Sect. 1, include those objects deemed Misaligned as result of an *Alignment Query*, as well as those included in the result of a *Redundancy Query*.

5 Experimentation

This proposal has been applied to ICFES [14]. We dealt with two business processes: the *Registration Process* (*P1*) and *Light Registration Process* (*P1*) introduced in Sect. 2. We loaded the BPMN diagrams and the database schemas (*S1* and *S2*) on which the processes depend. Our objective was to test this approach over a real EA to obtain preliminary results and to analyze performance and findings. We are working in accuracy comparison and results will be presented in future works. For this purpose, we are including three business processes and three schemas from the ICFES EA, and we are building together with the ICFES architects the reference alignment of these models to apply a survey to evaluate recall and precision.

We developed a prototype of our proposal as an eclipse project. The machine used to run the experiment is a 64-bit, dual core 2.2 GHz laptop computer with 4 GB RAM.

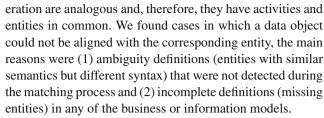
We initially imported the P1 and P2 business processes in XPDL format in 2,400 ms. The S1 and S2 schemas on which the processes are supported were also loaded to the Tartarus model using the JDBC-EMF importer. This operation was completed in 12,156 ms.

The next step consisted in executing the Tartarus–OWL transformations, thus generating two schema ontologies (*S1* and *S2*), and another two ontologies with business processes *P1* and *P2*. These transformations were carried out in 1.240 ms. The six matching tasks for the four ontologies were completed in 54,200 ms.

Once the candidate mappings were obtained, they were verified using the GUI and updated in the EA model. In this stage, we evaluated the accuracy of matching tasks and we achieved 55 % of average F-measure in the six tasks. We then performed alignment queries on the KQL editor (PI against SI, and P2 against S2). The output report of the PI-SI query corresponds to the graph shown in Fig. 10. We additionally performed redundancy queries (PI against P2 and P3 against P3), and its output can be seen in Fig. 11.

These queries provided means to express and assess the heuristics described in Sect. 1. The results of applying these heuristics to evaluate ICFES's processes and schemes (*P1*, *P2*, *S1*, *S2*) are presented in Table 2.

Following the analysis of these results, we find that components identified as redundant correspond indeed to overlapping processes or entities, since both processes under consid-



A more in-depth analysis was conducted in the case of misaligned objects, and three main situations were found: (1) non-automated manual activities; (2) entities that are not currently in use; (3) entities that are in reality accessed by the IS, but that are not referenced explicitly in the BPMN diagrams; (4) entities loaded to Tartarus, but whose processes were not included in our experiments. To summarize, we were able to identify potentially automatable activities (although not in every case) and unused entities, as well as BPMN diagrams needing further descriptors, and some false positives arising mainly from the design and scope of the experiment.

The work done in ICFES has offered a valuable information to EA development. It contributed with formalizing the relationships between processes and information assets whatever allow improving explicitness and understanding of EA. Additional to this, the alignment analysis performed has evidenced improvement points to align IT with business processes. Another point to consider, is redundancy discovering in processes and entities that offers possibilities of reuse, merge, synchronization and integration in both domains analyzed (i.e., IA and BA). At the end of this work, ICFES was starting to implement improvements supported in these findings. For example, they were designing a new process which merges features from the two similar processes to unify them. About duplicate entities, an ER version for a new IS include a unified entity that covers all their requirements. The ICFES' Business Intelligence project also has incorporated in its ETL procedures (extract, transform and load) information entities located in different schemas to analyze them as the same business entity.

6 Related Work

Cuenca et al. [6] have proposed a complete modeling framework for business—IT alignment, comprising life cycle stages, a maturity model, views and artifacts. It is implemented via expert surveys and interviews. Plazaola et al. [22] interprets business—IT alignment evaluation as the entry point for improving decision-making regarding alignment. A metamodel makes it possible to express quality criteria and evaluate them with the help of inference rules to quantify the organization's maturity level. Data for the input model are obtained via surveys. Their work is supported by the application of surveys to calculate said metrics. To summarize, the above models all make use of surveys and interviews



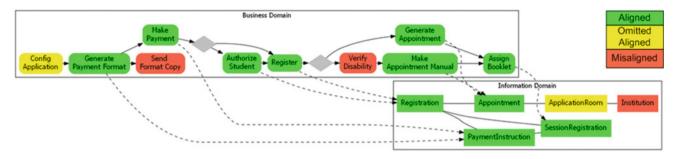


Fig. 10 Output of KQL alignment query

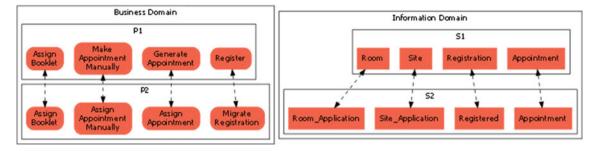


Fig. 11 Output of KQL redundancy query

 Table 2
 Results of heuristics

 evaluation

Assessed heuristic	Compared sets	Findings
Activities do not access at least one entity	P1-S1	2
Activities do not access at least one entity	P2-S2	4
Entities are not read at least by one process	P1-S1	1
Entities are not read at least by one process	P2-S2	2
Redundant entities	S1-S2	4
Redundant activities	P1-P2	4

to obtain their input data. Our approach, however, takes the formal definition of the EA as the starting point to find correspondences and assess alignment, so that we hope to get more objective and precise results. The scope of our work limited on BA and IA instead of all domains of EA, given its current focus. Nonetheless, this framework provides greater automation for inferring alignment and misalignment.

Aversano et al. [2] presents a strategy to be applied to detect misalignments between business processes and the IS that support them. Their strategy consists in identifying the objects that must be modified to restore alignment, considering a set of attributes that point to a possible misalignment based on object relations. This approach requires that a dependency graph between business processes and IS is defined previously to conduct the impact analysis. Such dependency graph is analogous to the alignment we seek to infer in our approach. We do not aim to indicate adjustments to restore alignment; it rather seeks to make explicit possible BA–IA misalignments in an automated way. In our case, the dependency graph is not given explicitly, but rather it is inferred via ontology matching.

Wegmann et al. [31] introduces the systemic enterprise architecture framework (SEAM) and an associated tool (SeamCAD) to verify alignment via the comparison of models given in terms of functional and organizational hierarchies. This proposal requires the explicit definition of relations among the different components. SEAM is applied while designing a architecture to promote alignment. Wegmann's work, like this proposal, makes use of reasoning based on domain-specific ontologies. Nonetheless, we focus on analyzing already existing BA and IA, and we do not attempt to create a new language to model data and process architectures. Furthermore, our work may be applied at both design and execution time, given that it can be used on current as well as future architectures.

A framework proposed by the CEO Group (*Centro de Engenharia Organizacional*) is described in [28] as a set of modeling primitives to express EA, IS architecture, Software Architecture and the dependencies among them. Additionally, they define a set of metrics which are assessed automatically to deliver a set of indicators about the impact of each of architectural decisions during the process of



constructing an IS Architecture. Our proposal is supported by Tartarus metamodel and thus also makes use of a specific language to express concepts in the EA domains. From that perspective, the use of UML gives CEO certain advantage, since additional transformations would be required to make compatible our framework with UML, and thus use UML compliant tools. Although this proposal contains the necessary information to calculate metrics, it does not generate alignment indicators in this version, but this issue is included in Sect. 7.1. On the other hand, this approach, being more than a modeler, makes use of reverse engineering on already existing models to infer relations among them.

ArchiMate [15] is one of the most popular EA integration modeling languages, providing notation to express the relations arising among the components of different domains of EA. Our approach makes no assumptions regarding the pre-existence of integration notations among architectural domains; it rather searches to infer such correspondence, starting with the definitions comprised in the EA. ArchiMate deals with interdependence among all domains of EA, while our current version is only concerned with BA and IA. And, as in previous cases, our aim is to offer a tool for the evaluation of an existing EA, as opposed to a modeling language.

Several studies [19,21,26] have evaluated alignment via the verification of proposed heuristics that must be accomplished across the domains of the EA. Regarding processes and data (for instance, all processes create, update and/or delete at least one entity), all entities are read by at least one process. Among processes and applications, consider the following: each business process should be supported by at least one application; business process tasks should be supported by a single application; critical business processes should depend on highly scalable and available applications. Even though their approach comprises heuristics for the IA, BA, AA and TA domains, it does not count with a tool to verify them, thus making it necessary to do it manually. Our misalignment definition is based on that heuristics, limited to the evaluation of BA and IA exclusively. Our work allows us to express and evaluate these heuristics with KQL, thus supporting the architect's work.

Other works have proposed different mechanisms to compare BA components. Dijkman et al. [7] present the application of lexical and graph-based alignments to detect overlapped activities in models of business process. On the other hand, Brockmans et al. [4] set forth an approach to express business processes with Petri nets on ontologies. Its objective is to realize semantic alignments to support the semiautomatic comparison of business processes. Further, Rodriguez et al. [24] point out to differences between two versions of the same business process using the delta between models. All of these works share a common feature with our proposal, namely that they all allow the detection of similarities between business processes via the use of enriched mod-

els. Additionally, we examine the use of ontology matching to find coincidences among information assets and alignment relations between BA and IA. Jointly with the work of Rodriguez et al. [24], we share the Tartarus metamodel as the core of our approach; however, our rationale makes use of ontologies in exchange of finding differences between models.

7 Conclusions

Initially, we described the context of alignment between business and IT, some existing approaches and their corresponding shortcomings. Then, we present a proposal to support the alignment evaluation between BA and IA using MDA and ontology matching. We implemented a set of tools that offer reverse engineering of BA and AI, inference, and query design. The automatic matching task presented average accuracy of 55 %. Further research on the techniques and algorithms used could provide greater accuracy in this stage.

We were able to use KQL to express business and information misalignment heuristics proposed in previous works. We applied these heuristics to a small segment of ICFES's BA and IA, and the results we obtained allowed us to detect some misalignments and shortcomings in the descriptions of the artifacts.

Generate output reports with processes and entities displayed like graphs, which facilitate the identification of alignments and misalignments. From our experience using KQL, we found that Graphical DSL facilitates the query design to analyze business–IT alignment, but a comparison with another mechanism (e.g., textual DSL) can provide new findings.

Experiments applied at ICFES allow us to state that it is possible to support alignment analysis with tasks such as inferring or discovering associations among business and data components, and evaluating misalignment heuristics via KQL queries. The automation of these tasks has been approached using ontology matching and analysis of EA models. We have shown that the automatic identification of misalignments among business processes helps cut back time and costs when compared to the manual execution of these activities. Our proposal is not intended to replace earlier methodologies based on interviews and surveys regarding the perceived business–IT alignment. Rather, we seek to complement this approach with a detailed assessment of the components of an EA.

We could not compare our approach with a similar proposal. Although we referenced other proposals around alignment evaluation, we did not find any tools that provided automatic alignment discovery. Our proposal is not intended to replace previous methodologies supported in interviews and surveys of perception about Business–IT alignment, but to



supplement them with a meticulous comparison on components contained in an EA.

7.1 Future Work

Future research could deal with the incorporation of alignment indicators or metrics to allow the evaluations of an EA with respect to previously presented maturity levels (e.g., [8]).

Including the other domains of EA, such as Applications, Technology, Services and Strategy (drivers, principles, objectives), shall improve the completeness of our proposal. At this point, we have important issues to consider, such as the fact of aligning elements at different levels of granularity (for instance, a business driver and an information entity).

The mappings generated by our framework could be exported to standard integration languages such as Archi-Mate [15]. Our proposal may be extended or modified to be used with other EA models. These in order that different tools can reuse the information inferred with this work. Other kinds of queries that solve different questions about the model can enrich KQL editor (for instance, getting all misaligned activities and entities).

To compare and improve the inference accuracy, we extended the experimentation using others matchers, engines, even different strategies like case-based reasoning or Petri nets.

We are currently working on an experiment to validate error reduction and decrease of time consumed using KQL in alignment analysis as opposed to other tools. Our results shall be included in future publications.

Our work uses inferred correspondence to support alignment evaluation, but new research may use this inferred correspondence to support other types of EA analysis (impact analysis). Our proposal may be extended or modified for use with other EA metamodels different to Tartarus.

References

- Atzeni P, Cappellari P, Torlone R, Bernstein PA, Giorgio G (2008) Model-independent schema translation. VLDB J 17:1347–1370
- Aversano L, Bodhuin T, Tortorella M (2005) Assessment and impact analysis for aligning business processes and software systems. In: Proceedings of ACM symposium on applied computing (SAC). ACM, New York, pp 1338–1343
- Bergeron F, Raymond L, Rivard S (2004) Ideal patterns of strategic alignment and business performance. Inf Manag 41(8):1003–1020
- Brockmans S, Ehrig M, Koschmider A, Oberweis A, Studer R (2006) Semantic alignment of business processes. In: Proceedings of the international conference on enterprise information systems (ICEIS), pp 191–196
- Cruz IF, Antonelli FP, Stroe C (2009) Agreementmaker: efficient matching for large real-world schemas and ontologies. Proc VLDB Endow 2:1586–1589

- Cuenca L, Ortiz A, Boza A (2010) Business and IS/IT strategic alignment framework. In: Camarinha-Matos L, Pereira P, Ribeiro L (eds) Emerging trends in technological innovation. IFIP advances in information and communication technology, vol 314. Springer, Boston, pp 24–31
- Dijkman R, Dumas M, Garcia-Banuelos L, Kaarik R (2009) Aligning business process models. In: IEEE enterprise distributed object computing conference, pp 45–53
- Elhari K, Bounabat B (2011) Platform for assessing strategic alignment using enterprise architecture: application to e-government process assessment. IJCSI Int J Comput Sci Issues 8(1):257–264
- Epsilon: Eugenia GMF tutorial (2012). http://www.eclipse.org/ epsilon/doc/articles/eugenia-gmf-tutorial/
- 10. Euzenat J, Shvaiko P (2007) Ontology matching. Springer, Berlin
- Gal A, Shvaiko P (2009) Advances in ontology matching. In: Dillon T, Chang E, Meersman R, Sycara K (eds) Advances in web semantics I. Lecture notes in computer science, vol 4891. Springer, Berlin, pp 176–198
- 12. Gaŝevic D, Djuric D, Devedzic V (2009) Model driven engineering and ontology development. Springer, Berlin
- Henderson JC, Venkatraman N (1989) Strategic alignment: a framework for strategic information technology management Center for Information Systems Research, Massachusetts Institute of Technology, Cambridge, MA
- ICFES: Instituto colombiano para la evaluación de la educación (2012). http://www.icfes.gov.co
- Jonkers H, Lankhorst M, Buuren RV, Bonsangue M, Torre LVD (2004) Concepts for modeling enterprise architectures. Int J Coop Inf Syst 13:257–287
- Madhavan J, Bernstein PA, Rahm E (2001) Generic schema matching with cupid. In: Proceedings of the international conference on very large data bases. Morgan Kaufmann Publishers Inc., Menlo Park, pp 49–58
- 17. MoosasGroup: Tartarus enterprise architecture meta-model (2011). http://moosas.uniandes.edu.co/doku.php?id=tartarus
- Noy NF, Mcguiness DL (2001) Ontology Development 101: a guide to creating your first ontology. Tech. rep., Stanford University, Stanford Knowledge Systems Laboratory
- Pereira CM, Sousa P (2003) Getting into the misalignment between business and information systems. In: The 10th European conference on information technology evaluation
- Pereira CM, Sousa P (2004) A method to define an enterprise architecture using the Zachman framework. In: Proceedings of the ACM symposium on applied computing (SAC). ACM, New York, pp 1366–1371
- Pereira CM, Sousa P (2005) Enterprise architecture: business and IT alignment. In: Proceedings of the ACM symposium on applied computing. ACM, New York, pp 1344–1345
- Plazaola L, Flores J, Vargas N, Ekstedt M (2008) Strategic business and IT alignment assessment: a case study applying an enterprise architecture-based metamodel. In: Proceedings of the international conference on system sciences (HICSS), p 398
- Rahm E, Bernstein PA (2001) A survey of approaches to automatic schema matching. VLDB J 10:334–350
- Rodriguez ME, Murcia F, Correal DE (2011) Tartarus una estrategia para construir y expresar arquitecturas empresariales. Revista Avances en Sistemas e Informática 8:61–70
- Scott GM (2005) Still not solved: the persistent problem of IT strategic planning. Commun Assoc Inf Syst 16(47):904–936
- Sousa P, Pereira CM, Marques JA (2005) Enterprise architecture alignment heuristics. Microsoft Archit J 4:1–53
- TheOpenGroup (2008) TOGAF version 9. Van Haren Publishing, Netherlands
- Vasconcelos A, Sousa P, Tribolet J (2007) Information system architecture metrics: an enterprise engineering evaluation approach. Electron J Inf Syst Eval 10(1):91–122



- 29. Wang X, Ma F, Zhou X (2008) Aligning business and IT using enterprise architecture. In: Proceedings of the international conference on wireless communications, networking and mobile computing, pp 1–5
- Wang X, Zhou X, Jiang L (2008) A method of business and IT alignment based on enterprise architecture. In: IEEE/SOLI, vol 1, pp 740–745
- 31. Wegmann A, Balabko P, son Le L, Regev G, Rychkova I (2005) A method and tool for business-IT alignment in enterprise architecture. In: CAiSE short paper proceedings
- Zachman JA (1987) A framework for information systems architecture. IBM Syst J 26:276–292

