Semantic Web Technologies in Business and IT Alignment

Multi-Model Algorithm of Ontology Matching

Ulf Seigerroth, Julia Kaidalova
School of Engineering
Jönköping University
Jönköping, Sweden
Ulf.Seigerroth@jth.hj.se, Julia.Kaidalova@jth.hj.se

Nikolay Shilov
Computer Aided Integrated Systems Laboratory
SPIIRAS
St.Petersburg, Russia
nick@iias.spb.su

Tomasz Kaczmarek
Department of Information Systems
Poznan University of Economic
Poznan, Poland
t.kaczmarek@kie.ue.poznan.pl

***Abstract*— The importance of using appropriate and effective IT means to facilitate business functions have been acknowledged and discussed by practitioners and scientists over the past two decades thus giving a rise to the Business and IT alignment (BITA) problem. In BITA, the issue of sharing and processing large amounts of information from distributed and heterogeneous sources is of significant importance. Ontologies have shown their usability for this type of tasks. However, they also bring new challenges. The paper proposes utilising of the Semantic Web technologies to assist in solving them. Namely, the paper describes application and adaptation of the ontology matching algorithm for the BITA problem. The algorithm integrates different matching models. Its operation is shown via an illustrative example.**

Keywords-ontology; matching; business and IT alignment

I. INTRODUCTION

A key issue in today's enterprise activity is information technology (IT) that supports business needs, processes, and strategies [1]. The importance of using appropriate and effective IT means to facilitate business functions have been acknowledged and discussed by practitioners and scientists over the past two decades [2]. The problem of Business and IT alignment (BITA) is even more complex due to the dynamic and evolving nature of both sides – business and IT [3]. Today, BITA is a crucial issue for enterprise success and it is often addressed as a top concern of IT and business practitioners [4, 5]. It is possibly caused by the recognition of organizational benefits that BITA can bring to the table.

Researchers and practitioners discuss and tackle BITA in various ways. Initially, from more general point of view, the problem was studied as linking business plan and IT plan or business strategy with IT strategy. Talking more particularly, one possible way to perceive it is the following: in order to align business and IT perspectives it is required to enable alignment of their representations [6]. Here, models come into play, since in this respect models are often used as a supportive means that are able to capture and represent different aspects and constructs of an enterprise. Subsequently, models are often used as a support in a transition process to take a business from one state to another and to create BITA.

In such tasks, the issue of sharing and processing large amounts of information from distributed and heterogeneous sources (information management systems, experts, electronic documents, real-time sensors, etc.) is of significant importance. Due to such factors, as different data formats, interaction protocols, etc. this leads to a problem of semantic interoperability.

Hence, the information sharing in a business network is highly important for BITA and should be achieved at both technical and semantic levels. The interoperability at the technical level is addressed in a number of research efforts. It is usually represented by such approaches as, e.g., Service-Oriented Architecture or SOA [7] and on the appropriate standards like WSDL and SOAP [8]. The semantic level of interoperability in the flexible supply network is also paid significant attention. As an example (probably the most widely known), the Semantic Web initiative is worth mentioning [10]. The main idea is to use ontologies for knowledge and terminology description.

Ontologies have shown their usability for this type of tasks (e.g., [11], [12], [13]). These are content theories about the sorts of objects, properties of objects and relations between objects that are possible in a specified knowledge domain. Ontologies provide potential terms for describing the knowledge about the domain [14]. An ontological model is used to solve the problem of heterogeneity of descriptions of different enterprise elements. This model makes it possible to enable interoperability between heterogeneous information sources due to provision of their common semantics [9].

However, in open or evolving systems, as in the case of BITA, different parties would, in general, adopt different ontologies. Thus, just using ontologies, like just using XML, does not reduce heterogeneity; it raises heterogeneity problems at a higher level.

Ontology matching is a promising solution to the semantic heterogeneity problem. It finds correspondences between semantically related entities of the ontologies. These correspondences can be used for various tasks, such as ontology merging, query answering, data translation, or for navigation on the Semantic Web. Thus, matching ontologies enables the knowledge and data expressed in the matched ontologies to interoperate [15].

The goal of ontology matching is basically solving the two major problems, namely:

- Identify ontology entities, which have different names but have the same meaning.
- Identify ontology entities, which have the same (similar) name but have different meaning.

In 2010, a multi-model approach for on-the-fly ontology matching to be used in smart spaces has been developed [16]. In this paper, the above mentioned concept is further developed and adapted for the BITA problem.

The paper is structured as follows. Section II introduces the problem. The background in the area of ontology matching is presented in Section III. Section IV describes the ontology matching algorithm and its components. The algorithm's adaptation is presented in Section V. The case study and its discussion is given in Section VI. Major results are summarized in the last section.

II. Problem Statement

An example of a BITA problem could be the following situation: there are certain IT systems deployed in an enterprise, which already support its operations. However, the support might be only partial (full would mean an enterprise requiring little human intervention). Second, the environment, in which the enterprise is placed, evolves (customers, suppliers, legal conditions, and general macroeconomic conditions - all change). IT systems have to be modified to reflect these changes. Third, enterprise managers or the owner might want to change the way the business is carried out (even if no particular external change enforces it) – again IT systems have to be modified. BITA therefore is the problem of changing the IT infrastructure and systems to the imagined future state of these systems.

Why is this hard? From a more general point of view considered previously [6], the problem lies in identifying what to change and in what way (how). Now, the modelling comes into the play. Having an "as-is" model and a "to-be" model of an enterprise, let us assume that we want to change a part of the "as-is" model that describes enterprise operations, to reflect the desired "to-be" situation of enterprise operations. The part of the model that describes IT systems is misaligned and has to be changed too.

A. Assumptions

The notations and terminology between models describing the "as is" and "to be" situations are the same. However, the interpretation of the same objects may differ depending on its connections with other objects.

At the moment, the RDF [17]-like formalism (consisting of triples "subject-predicate-object") is considered as the most generic (consisting basically of objects and relationships between them). In this case the models can be considered as ontologies.

B. Approach ideas

The main aim of the approach is to find a way to describe relationships between different models so that these relationships could be transferred between "as is" and "to be" sets of models. Model versioning techniques could be used if the "to be" set is built upon the "to be" set.

Ontology matching originating from Semantic Web could assist in solving the "as is" and "to be" alignment problem.

III. Ontology Matching

The existing ontology matching techniques has been analyzed by Smirnov et al. [16]. All the similarity metrics in the performed state-of-the-art review are based on the two information retrieval metrics, namely metrics of precision and recall. The balance between these is achieved via choosing the right threshold value. The possibility of choosing the right threshold value has to be taken into account in the development of the matching models.

The above mentioned approaches to ontology matching apply the following techniques in different combinations. The techniques include:

- Identification of synonyms
- Similarity metrics (name similarity, linguistic similarity)
- Heuristics (for example two nodes are likely to match if nodes in their neighborhood also match)
- Compare sets of instances of classes instead compare classes
- Rules: for example, if class A1 related to class B1 (relation R1), A2 related to class B2 (relation R2) and B1 similar to B2, R1 similar to R2 therefore A1 similar to A2.

As a result of matching, the following types of elements mapping proximity can be identified:

- One-to-one mapping between the elements (Associate-Professor to Senior-Lecturer)
- Between different types of elements (the relation AdvisedBy(Student, Professor) maps to the attribute advisor of the concept Student)
- Complex type (Name maps to the concatenation of First Name and Last Name)

All methods can be separated into the following four groups.

<u>Linguistic Methods</u>. These methods are focused on determining similarity between entities based on linguistic comparison of these entities (count of the same symbols estimation, estimation of the longest similar parts of words, etc.).

Statistical Methods (Instance-Based). These methods compare instances of the ontology entities and based on this estimation entities can be compared.

Contextual Methods. The aim of the contextual similarity is to calculate a measure of similarity between entities based on their contexts. For example, if parents and children of the ontology classes are the same consequently the classes also the same.

Combined Methods. These methods combine specifics of two or three of the above methods.

In the considered problem domain, the differentiation between instances is not an easy task. Because of this reason, the techniques and methods relying on instances were not considered for further development. Hence, the developed models presented below integrate all of the above techniques (except those dealing with instances) and propose a set of combined methods having features of the linguistic and contextual methods.

IV. Multi-Model Approach for On-the-Fly Ontology Matching

The below proposed approach allows matching of ontologies for the interoperability purposes and is based on the ontology matching model illustrated by Figure 1. The approach takes into account that matched ontologies are responsible for concrete and well-described tasks, which means that they generally should be small–to–medium size and describe only limited domains. A detailed description of the approach can be found in [16].

Ontology is represented as RDF triples, consisting of the following ontology elements: subject, predicate, object. Degree of similarity between two ontology elements is in the range [0, 1]. The approach consists of the following steps:

- Compare all elements of two ontologies and fill the matrix M'. Matrix M' is of size m to n, where m is the number of elements in the first ontology and n is the number of elements in the second ontology. Each element of this matrix represents the degree of similarity between two ontology elements.
  - The degree of similarity between equal elements or synonyms is set to 1 (maximum value of the degree of similarity).
  - WordNet or Wiktionary are used to calculate semantic distances based on the synonymy relationship or other relationships (with a lower degree of similarity).
  - The degree of similarity between other string terms of ontology elements is calculated using the fuzzy string comparison method.
- Update values in matrix M, where each new value of elements of M is the maximum value of (M, M')
- Improve distance values in the matrix M using the graph-based distance improvement model.

As a result, the matrix M contains degrees of similarity between elements of two ontologies. This allows determining correspondences between elements by selecting those for which the degrees of similarities are above the pre-selected threshold value.

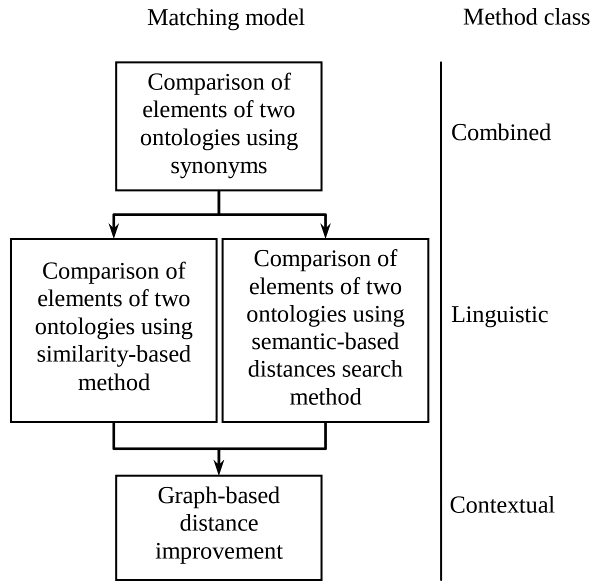

Figure 1. Multi-model approach to on-the-fly ontology matching.

V. Adaptation of the Algorithm and Case Study

The analysis of the possible application of the above ontology matching approach to the BITA domain has shown that the following elements of the approach that could be applied:

1. Comparison of elements of two models taking into account synonyms (e.g., a synonym can be considered as a description of the same aspect)
2. Comparison of elements of two ontologies using fuzzy string comparison.
3. Graph-based distance improvement (e.g., if an object A in one model is a super class of three other objects, and object B in the other model is a super class of the same three other objects, then it is likely that A and B are the same)

The analysis of the possibility to apply Wiktionary for the experimentation purposes has shown that in the business domain it does not have enough synonyms to produce any significant results in the ontology matching. The situation with the WordNet is better, but the amount of synonyms in the business domain is still not sufficient. This issue is still under research and will be addressed in the future work.

Below, the detailed description of the used techniques can be found.

A. Fuzzy string comparison

The basis of the fuzzy string comparison algorithm is the well-known conventional algorithm that calculates occurrence of substrings from one string in the other string.

1. Perform the comparison based on the above algorithm twice: $FC_1 = FuzzyCompare(Element_1, Element_2)$ and $FC_2 = FuzzyCompare(Element_2, Element_1)$.
2. Calculate the result as an aggregation of the above results in accordance with the following formula: $Re' = n*FC_1+(1-n)*FC_2$, where

n is a weight, $n \in [0;1]$; $n = 0.5$ sets the same weight to the both strings, $n = 0$ searches only $Element_2$ within $Element_1$, and $n = 1$ searches only $Element_1$ within $Element_2$. It is proposed to set $n = 0.5$.

B. *Graph-based distance improvement*

The graph-based improvement model for propagation similarities from one ontology elements to another is presented in Figure 2. The aim of this model is to propagate the degree of similarity between closely matching ontology elements to ontology elements related to them through RDF triples.

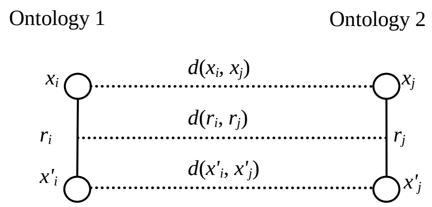

Figure 2. Graph-based distance improvement.

Set $X = (x_1, x_2, ..., x_n)$ is the set of subjects and objects in the ontologies of two knowledge processors, $D_x = (d(x_i, x_j), ...)$ is a degree of similarity between x_i and x_j. Set $R = (r_1, r_2, ..., r_n)$ is the set of predicates in the ontologies of two knowledge processors. Constant Tr is a threshold value which determines if two ontology elements mapped to each other or not.

The following algorithm allows propagating similarity distance to RDF subjects and objects:

$d(x_i, x_j)$ = maximum(D_x)
while ($d(x_i, x_j) > Tr$) do
  for each $d(x'_i, x'_j)$ as $x_i\, r_m\, x'_i$ and $x_i\, r_m\, x'_j$ do
    $d(x'_i, x'_j) = \sqrt[2]{d(x_i, x_j)\, d(x'_i, x_j)}$
  endfor
  Exclude $d(x_i, x_j)$ from D_x
  $d(x_i, x_j)$ = maximum(D_x)
endwhile

Today, modern research in the considered and adjacent areas [18] - [20] suppose finding threshold value by experts based on the statistical data. For every task family the threshold will be different but usually it is in range [0.5, 0.95].

VI. CASE STUDY

The ontologies that are used as examples of “as is” and “to be” are two ontologies taken from a real project. In this project two types of ontologies were used: base ontology and domain ontology. The examples in this article are domain ontologies. The domain ontologies were used to give a structure for conceiving organizational work performed by humans and artifacts on behalf of an enterprise. It was an important driver for adopting a multi-layered thinking and for creating alignment between and within different abstraction layers in an enterprise (strategy, practice, process, service, IS/IT infrastructure). This domain ontology served as an essential and necessary enabler for directing attention to what to conceive on different levels of abstraction (focal areas), how to interrelate different levels, and how to gradually increase the degree of formalism on more detailed levels in the framework (lower levels of abstraction).

A. *“As is” ontology*

In order to match the “as is” and “to be” ontologies the “as is” ontology had to be built out of two source domain ontologies. These ontologies are presented in Figure 3. To find the correspondences between classes of the domain ontologies the same ontology matching algorithm was applied. It produced the following results:

Similar objects:
Prerequisite / basis - Basis = 0.55 (the result of fuzzy string comparison)
Product - Product = 1

Matched objects:
Product - Product = 1

It can be seen that 2 pairs of similar classes were found and no similarity propagation has been made. With the threshold being set to 0.75 the pair “Prerequisite / basis” – “Basis” was dropped.

After an analysis of the ontologies, it was concluded that in this particular case, the classes “Prerequisite / basis” and “Basis” are the same. As a result, it was decided to explicitly mark these as synonyms. The second try taking into account the introduced synonymy relationship produced the following results:

Similar objects:
Prerequisite / basis - Basis = 1
Product - Product = 1

Matched objects:
Prerequisite / basis - Basis = 1
Product - Product = 1

It can be seen that 2 matching pairs of classes were found. All other classes are different. As a result the merged “as is” ontology have been built as a union of these two ontologies, where classes “Prerequisite / basis” - “Basis” and “Product” - “Product” have been merged into “Prerequisite / basis” and “Product” correspondingly. The resulting ontology is shown in Figure 4.

B. *“As is” – “to be” ontology matching*

In this subsection, the matching of the above built “as is” ontology and “to be” ontology shown in Figure 5 is performed. The “to be” ontology is outlined in such a way so that its differences from the “as is” ontology would be obvious.

The matching results produced by the multi-model algorithm were as follows:

Similar objects:
Result - Result = 1
Prerequisite / basis - Prerequisite / basis = 1
Activity - Activity = 1

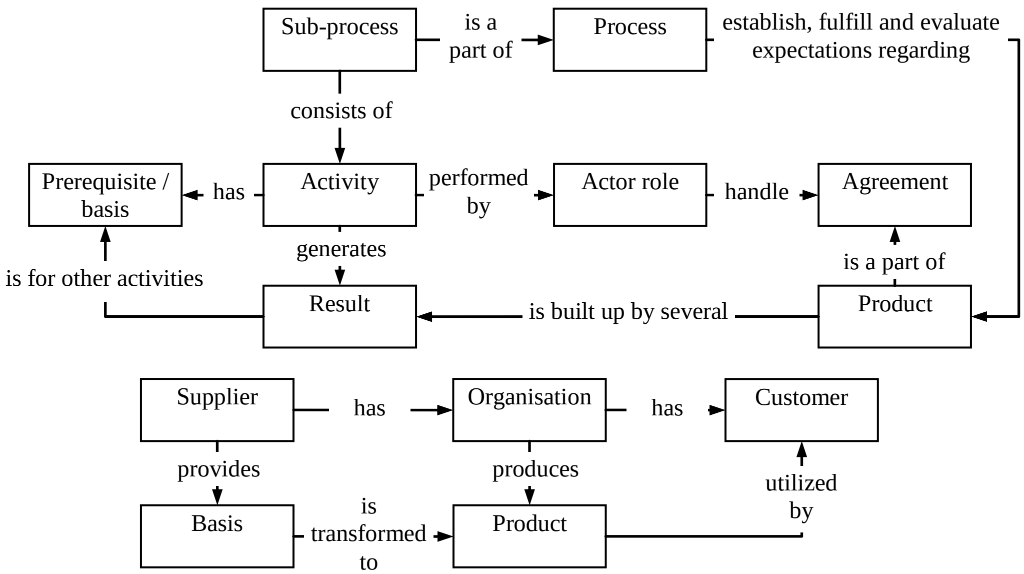

Figure 3. "As is" ontologies.

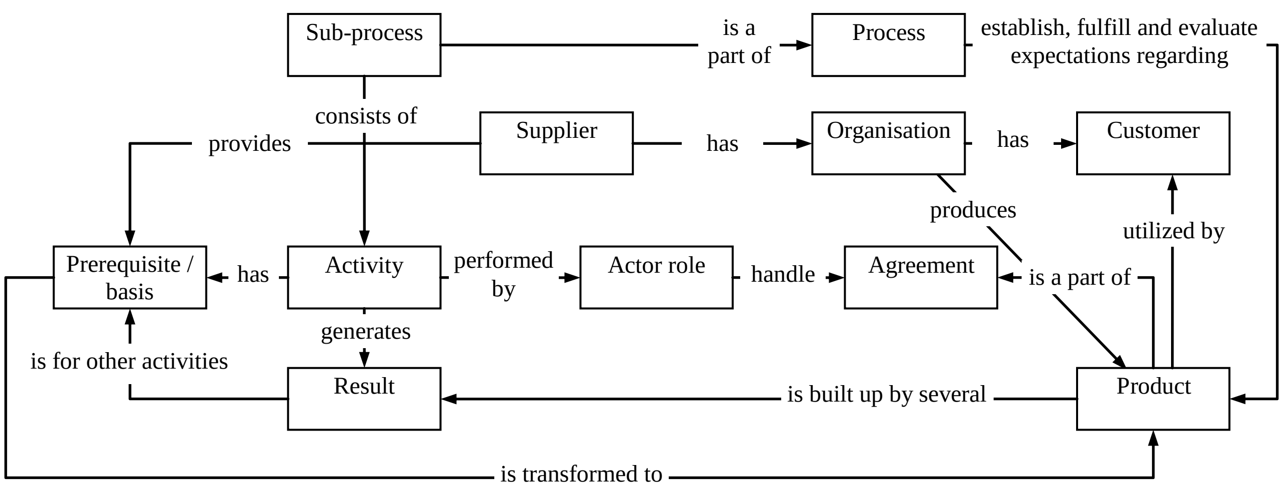

Figure 4. Integrated "as is" ontology.

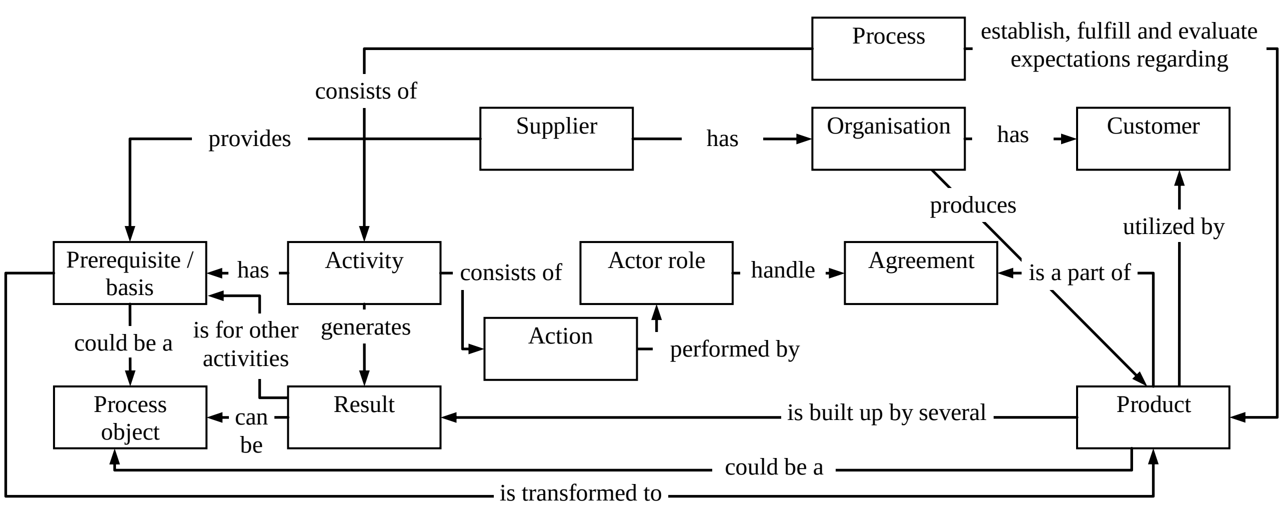

Figure 5. "To be" ontology.

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-300-1

Activity - Action = 0.40 (the result of fuzzy string comparison)
Sub process - Process = 0.72 (the result of fuzzy string comparison)
Process - Process = 1
Process - Process object = 0.65 (the result of fuzzy string comparison)
Actor role - Actor role = 1
Agreement - Agreement = 1
Product - Product = 1
Supplier - Supplier = 1
Organisation - Organisation = 1
Customer - Customer = 1

Similarity propagation:
Activity - Activity = 1
Sub process - Process: 0.72 -> 0.85;
The "Sub process" class of the "as is" ontology and the "Process" class of the "to be" ontology matched with the similarity of 0.72. Since they are both related to the classes "Activity" matched with the similarity 1, the graph-based distance improvement algorithm has propagated this fact via increasing the similarity between the "Sub process" and "Process" classes from 0.72 to 0.85 ($\sqrt[2]{1*0.72}$).

Actor role - Actor role = 1
Activity - Action: 0.40 -> 0.64;
The "Activity" class of the "as is" ontology and the "Action" class of the "to be" ontology matched with the similarity of 0.40. Since they are both related to the classes "Actor role" matched with the similarity 1, the graph-based distance improvement algorithm has propagated this fact via increasing the similarity between the "Activity" and "Action" classes from 0.40 to 0.64.

Product - Product = 1
Process - Process object: 0.65 -> 0.80;
The "Process" class of the "as is" ontology and the "Process object" class of the "to be" ontology matched with the similarity of 0.65. Since they are both related to the classes "Product" matched with the similarity 1, the graph-based distance improvement algorithm has propagated this fact via increasing the similarity between the "Process" and "Process object" classes from 0.65 to 0.80.

The final result is as follows:
Matched objects:
Result - Result = 1
Prerequisite / basis - Prerequisite / basis = 1
Activity - Activity = 1 or Activity - Action = 0.64
Sub process - Process = 0.85
Process - Process = 1 or Process - Process object = 0.80
Actor role - Actor role = 1
Agreement - Agreement = 1
Product - Product = 1
Supplier - Supplier = 1
Organisation - Organisation = 1
Customer - Customer = 1

The result is quite predictable. However, there are three interesting facts, which deserve some discussion.

The class "Process" of the "as is" ontology matched to the classes "Process" (with similarity 1) and "Process object" (with similarity 0.80) of the "to be" ontology. This could be interpreted as a sign to the expert to check if the new added class "Process object" is the same with the existing class "Process", which is not the case in our example.

The class "Activity" of the "as is" ontology also matched to two classes of the "to be" ontology: "Activity" (with similarity 1) and "Action" (with similarity 0.64). This means that the algorithm actually discovered that in the "to be" ontology the class "Activity" was split into two classes "Activity" and "Action".

Similarly, via finding the match between both the "Sub process" and the "Process" classes of the "as is" ontology with the class "Process" of the "to be" ontology. The algorithm informed the experts that the class "Sub process" was probably merged with the class "Process", what is the truth in the presented example.

VII. Conclusion and Future Work

The paper presented the adaptation of the ontology matching algorithm originating from research works in the area of Semantic Web to BITA. By far, there are no approaches aimed at comparison of ontologies representing "as is" and "to be" situations. The proposed algorithm is aimed at assisting experts in finding changes between the mentioned ontologies. The algorithm is based on the idea when the least computationally expensive operations are followed by more computationally expensive. As a result, the more computationally expensive operations work with less data resulting in a higher efficiency of the algorithm. The illustrative example presented shows some of the advantages of using this approach. It has been discussed how this algorithm can reveal and inform the experts about situations where classes are split or merged. For the presented example the algorithm works nearly instantly.

At the moment, the algorithm does not implement the matching based on the synonymy relationship. The Wiktionary does not have enough synonyms to produce any significant results in the ontology matching for BITA. The situation with the WordNet is better, but the amount of synonyms in the business domain is still not sufficient. This issue is still under research and will be addressed in the future work.

Acknowledgment

This paper is due to the project COBIT sponsored by the Swedish Foundation for International Cooperation in Research and Higher Education. Some parts of the work have been sponsored by grant # 12-07-00298-a of the Russian Foundation for Basic Research.

References

[1] A. J. G. Silvius, "Business and IT Alignment: What We Know and What We Don't Know," Proc. International Conference on Information Management and Engineering, IEEE, 2009, pp. 558-563.

[2] J. O. Vargas, "A Framework of Practices Influencing IS/Business Alignment and IT Governance," Thesis, School of Information Systems, Computing and Mathematics in Brunel University, 2011.

[3] N. J. Luftman, "Measure Your Business-IT Alignment," Optimize: Business execution for CIOs Magazine, iss 26, 2003.

[4] Y. E. Chan and B. H. Reich, "IT alignment: what have we learned?" Journal of Information Technology, vol. 22, 2007, pp. 297-315.

[5] J. Luftman and E. R. McLean, "Key issues for IT executives," MIS Quarterly Executive, vol. 3, no. 2, 2004, pp. 89-104.

[6] T. Kaczmarek, U. Seigerroth, and N. Shilov, "Multi-layered enterprise modeling and its challenges in business and IT alignment," Proc. International Conference on Enterprise Information Systems, 2012, pp. 257-260.

[7] SOA: Service-oriented architecture definition, 2007. URL: http://www.service-architecture.com/web-services/articles/service-oriented_architecture_soa_definition.html [retrieved June, 2013].

[8] Web Services explained, 2007. URL: http://www.service-architecture.com/web-services/articles/web_services_explained.html [retrieved June, 2013].

[9] M. Uschold and M. Grüninger, "Ontologies: Principles, methods and applications," Knowledge Engineering Review, vol. 11, no. 2, 1996, pp. 93-155.

[10] Semantic Web, 2006. URL: http://www.semanticweb.org [retrieved June, 2013].

[11] D. J. Bradfield, J. X. Gao, and H. Soltan, "A Metaknowledge Approach to Facilitate Knowledge Sharing in the Global Product Development Process, Computer-Aided Design & Applications, vol. 4, no. 1-4, 2007, pp. 519-528.

[12] E. C. K. Chan, and K. M. Yu, "A framework of ontology-enabled product knowledge management," International Journal of Product Development, Inderscience Publishers, vol. 4, no. 3-4, 2007, pp. 241-254.

[13] L. Patil, D. Dutta, and R. Sriram, "Ontology-based exchange of product data semantics," IEEE Transactions on Automation Science and Engineering, vol. 2, no. 3, 2005, pp. 213-225.

[14] B. Chandrasekaran, J. R. Josephson, and V. R. Benjamins, "What are ontologies and why do we need them?" IEEE Intelligent Systems, vol 14, no. 1, Jan/Feb, 1999, pp. 20-26.

[15] Ontology Matching, 2013. URL: http://ontologymatching.org [retrieved June, 2013].

[16] A. Smirnov et al., "On-the-Fly Ontology Matching in Smart Spaces: A Multi-Model Approach," Smart Spaces and Next Generation Wired/Wireless Networking, Proc. Third Conference on Smart Spaces, ruSMART 2010, and 10th International Conference NEW2AN 2010, S. Balandin, R. Dunaytsev, and Y. Koucheryavy (Eds.), Springer, LNCS 6294, pp.72-83.

[17] Resource Description Framework (RDF), 2004. URL: http://www.w3.org/RDF/ [retrieved July, 2013].

[18] A. Doan, J. Madhavan, P. Domingos and A. Halevy, "Learning to map between ontologies on the semantic web," Proc. of the 11th international conference on World Wide Web, 2002, pp. 662-673.

[19] D. AnHai, M. Jayant, D. Pedro, and H. Alon, "Ontology Matching: A Machine Learning Approach," Handbook on Ontologies in Information Systems (Staab, S., Studer, R., eds.), Springer-Velag, 2003, pp. 397-416.

[20] J. Euzenat, and P. Shvaiko, "Ontology matching," Springer-Verlag, 2007.