# Ontology Matching for Product Lifecycle Management

Alexander Smirnov, Nikolay Teslya

# Ontology Matching for Product Lifecycle Management

Alexander Smirnov and Nikolay Teslya[0000-0003-0619-8620]*

SPIIRAS, 14th Line 39, 199178, St.Petersburg, Russia
smir@iias.spb.su, teslya@iias.spb.su*

**Abstract.** Every product goes through many stages of the life cycle from manufacturing through usage to utilization. Some stages can change its properties, which in turn changes its description. To describe the product and its life cycle many ontologies have been created with varying levels of detail. Ontologies usage at different product lifecycle management (PLM) stages provides a better match to these stages since they provide properties of the product important only for this stage. During the transition between stages, the data should be integrated across all systems in manufacturing domain to provide semantic interoperability. Therefore, an issue arises of matching descriptions presented with ontologies of lifecycle stages. This is especially critical if ontologies for different stages are created by various specialists (for example, designer, technology engineer, retailer, maintainer, etc.). The paper proposes the method of matching ontologies for the formation of a common PLM ontology based on the automatic matching of ontologies referred to PLM stages. It allows to overcome heterogeneity and ensure interoperability in the process of tracking the product through the PLM stages. The matching process is based on the identification of common concepts by which ontologies will be combined into one. To identify common concepts, the ontology matching method is used, based on a combination of a context-based matching with neural network to find similarities of concepts (name, characteristics names and their string values) and study the ontology structure to identify common design patterns.

**Keywords:** Product, Ontology, Alignment, Lifecycle, Matching, Heterogeneity

## 1 Introduction

Most of knowledge management systems for the production network provide ontology-based description of products and lifecycle stages for semantic-based operability between participants on the various stages [1]. The management of product lifecycle can be viewed in two possible directions: based on product instances or product types lifecycles [2]. For the first direction the main attention is paid to product items that can be produced, used and recycled. The second direction corresponds to the product development process. In this work the main focus will be concentrated on the first direction – product instances.

Every product goes through many stages of the lifecycle from planning and producing through usage to utilization or recycling. Some stages can change its properties, which in turn changes its description. However, the data should be integrated across all systems in manufacturing domain. Production network as well as management systems have to be scalable and adaptable for their environment changes and variability of participants' knowledge. A method of ontology matching for participants of the production network has to be developed to automate matching of the ontologies of several participants for their interoperability. It allows to increase efficiency of the future interaction during joint operational management in product lifecycle.

During the product transition between lifecycle stages, the issue arises of matching descriptions presented with ontologies and the transferring of product properties between stages. This is especially critical if ontologies for different stages are created by various specialists (for example, designer, technology engineer, retailer, maintainer, etc.).

For supporting production network scalability and adaptability to changes in business environment the paper proposes an original method of ontology matching. Method is based on the combination of several matching strategies:

- Pattern-based matching. For the patterns the ontology design patterns are used.
- Context-based matching (including neural network usage [3]). The context of ontology concept includes neighboring concepts, relation types, semantic of concept term.
- Community-driven matching. To increase accuracy and quality of matching experts can be involved. They provide expertise on the stages of product lifecycle to find better correlations between product ontology and stage ontology.

This combination provides analysis on ontology concepts from the various points of view. Usage of multiple matchers is widely spread practice [4, 5]. The main advantages of such kind of approach is that it allows to take into account various aspects of ontology concept's semantics and increase accuracy of matching.

The rest of the paper is structured as follows. Section 2 provides overview of ontology matching methods, using for product lifecycle management. Section 3 provides architecture description of proposed matching method. Section 4 describes implementation from the technologies point of view. Section 5 shows an evaluation using the example of ontologies from the same project on the different

## 2 Overview of Ontology Matching for Product Lifecycle Management

In developing ontologies for managing the product life cycle, various techniques are used. As a rule, for each stage of the life cycle, its own ontology is developed that most fully describes the characteristics of the product that are important in this cycle. These private ontologies are designed by experts using top-level ontologies and using design patterns. To ensure the knowledge sharing between the stages of the life cycle in this case, the use of a mechanism for ontology matching is required. The purpose of this mechanism is to identify concepts that are common to different ontologies and to align or merge ontologies into one large ontology, based on general concepts.

Another technique is the formation of a large ontology, which describes the product at several or all stages of the life cycle [6, 7]. This approach requires combining knowledge from different problem domains and their reduction into a common knowledge base. This process is carried out by experts. In fact, experts manually or semi-automatically apply methods for ontology matching to form a common ontology from private ones. The quality of the alignment using this technique is higher due to the use of experience and implicit knowledge of experts in the subject area.

Automation of the process of creating and matching ontologies in both cases is possible using ontology matching methods. These methods are specially designed to identify concepts that are of similar meaning in different ontologies. The basis for matching may be the semantic meaning of terms describing ontologies, their context, the structure of the concept (general characteristics and domains of the value of characteristics), the structure of the ontology. In this regard, in the classification of methods, several groups are identified that correspond to the parameters on the basis of which a comparison is made [8]. Of this work, of all the available classes, context-based, pattern-based and community-based techniques are of particular interest.

While using the context-based ontology matching the relation between ontology with its environment is used [9]. This environment can be upper level ontology that is using as an external source of common knowledge about concepts of matching ontologies, as well as a problem area [10]. Also, the semantics of concept can be utilized for estimating concept similarity. It can be done by dictionaries, like WordNet that can compare concept by their meaning, or by neural networks [11]. Two approaches are existing on neural network utilization for ontology matching. The first one is a classification of ontology concept that allows to find similar concept based on the neural network training to classify concept between ontologies based on the set of parameters, i.e., characteristics, relations, and terms [12, 13]. The second approach is based on the linguistic analysis of concepts within the vector model of language. The vector model can be based on Word2Vec [14, 15] or BERT model [16]. It reflects the word meaning by concept vector. Words that describes similar concepts are described by vectors that are close to each other in vector space. It can be used in ontology matching by additional training the existing models on ontology concepts and further estimation of similarities between ontology concepts.

Using design patterns when creating an ontology can also help with their subsequent comparison. The ontology design patterns are meant as aligned groups of linked ontology concepts (ontology fragments) [6, 17, 18]. These patterns are used as templates to create ontologies describing similar concepts. Patterns can relate to separate connection of classes (relations between concepts) and classes with their relation, or to complex interconnection of classes, etc. When a pattern is detected in different ontologies, it can be stated with a high degree of certainty that concepts corresponding to the pattern will have similar meaning in ontologies.

When using the first two techniques, situations are possible where the algorithm cannot determine with high certainty the similarity of ontology concepts. In this case, the best way is to contact the experts to resolve the ambiguous situation. For each problem area the community of professionals is forming from the employees of production network participants [19]. They have different roles and competences that allows to engage experts of problem area to ontology matching process. It also allows to share alignment through the network, discuss it and reusing. All of that allows to increase alignment completeness and quality, but requires significant time consuming from the side of community members.

Professional community also resolves controversial situations that raises when alignment coefficient belongs to the controversial range. Recommendation of the professional community provides adjusting of the alignment coefficients ranges boundaries, creating of the new matching patterns or modification (adaptation, adjustment) of existing matching patterns.

## 3 Architecture of Ontology Matching for PLM

The proposed ontology matching method includes the following main steps.

1. The ontologies of problem domains are formed from the ontologies of production network and its' participants that have to be matched. Problem domain ontologies include abstract and operational contexts which are structure of domain and current situation correspondingly [20, 21].
2. Specific problem domains are defined using the abstract context. Problem domain defines set of ontology designing patterns that can be used in matching process in it. The result of pattern-based matching is a matrix containing align coefficients of ontologies concepts.
3. The context-based matching of operational context ontology models is performing. During the context-based matching, the instances of ontology concepts are matching. The result is matrixes that contain align coefficients of concept instances. The results are using to clarify coefficients in ontologies alignment matrix. For example, the value of align coefficient can be increased in the case of the value of align coefficient of concept instances is higher that the value of concept alignment coefficient.
4. Checking of the alignment coefficients is performed for each problem area. For this purpose, the values ranges are used. The ranges define are separate concepts aligned, not aligned or controversial. Concepts are controversial if the alignment cannot be defined automatically. Range boundaries are set by experts with highest ratings among users with similar roles from the professional community before the matching process.

To provide semantic interoperability between the entities of product lifecycle management system it is proposed to develop a special method for ontology matching. Architecture of the method is presented on the Fig. 1. It describes required modules and links between them to provide the ontology matching.

Architecture can be conditionally divided into three major components related to the ontology matching in the product lifecycle management system. The first one is related to the entities of the system, which needs to have support for semantic interoperability. Also, among these entities there are a *community of experts* who participate in the process of ontologies matching, providing their knowledge and experience to adjust the alignment obtained automatically. The entity of the system that request semantic interoperability support provides ontologies in form of OWL and the corresponding context to the *ontology matching method*.

The received ontologies and context are divided into abstract and operational contexts in the ontology matching method, according to the approach described in the previous section. Abstract contexts are matched in the module *"Patter-based matcher"*, which uses background knowledge in form of ontology design and correspondence patterns from public *pattern libraries*. Operational contexts are used in the *"Context-based matcher"*, which, in addition to ontologies' own context, refers to sources of background knowledge for additional information while calculating the correspondence. These sources are upper level ontologies and lexical databases.

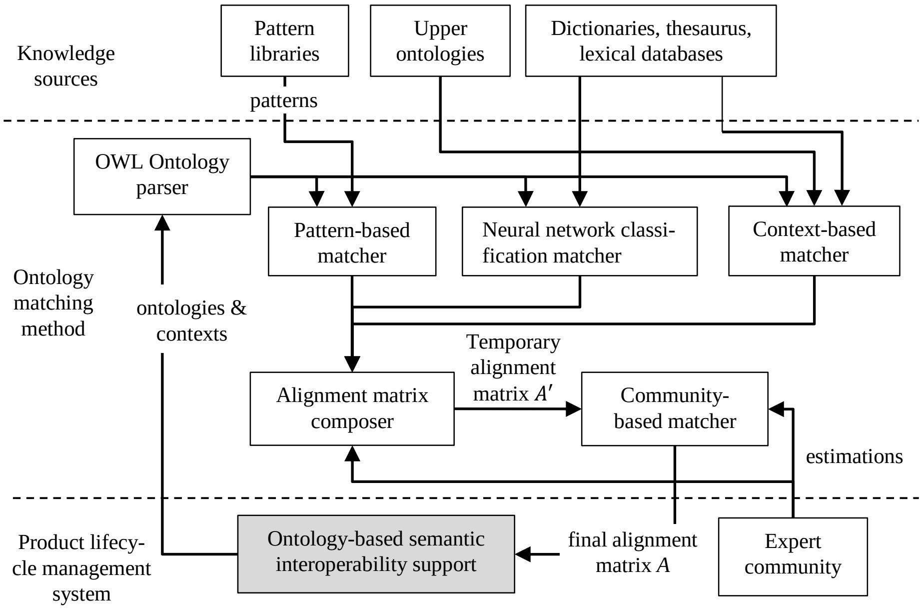


**Fig. 1.** Architecture of ontology matching method for product lifecycle management system

Also, ontology concepts are transferred to the input of the neural network, where the training and further classification is carried out. The results of the work of all matchers are combined according to the formula (1) in the module *"Alignment matrix composer"*. The intermediate result of the matching (temporary alignment matrix $A'$) is transferred to the *"Community-based matcher"*, where it can be adjusted by the community of experts that are part of the product lifecycle management system. Decisions of experts can also influence to the work of *"Alignment matrix composer"*, by changing the coefficients in formula (1). The result is transferred to the entity of the product lifecycle management system, which initiated the process of ontology matching.

### 3.1 Context-Based Matching

When considering comparisons based on contextual information, two stages are distinguished. The first stage involves comparing the names of ontology concepts and is based on the fuzzy string comparison algorithm presented in [22]. The result of this comparison is the matrix $A_{context}$, which contains similarity coefficients of ontology concepts, among which are concepts that exceed the lower threshold of concept comparison $b_1 = 0.6$, at which concepts can still be considered coincident. The rows of the matrix correspond to the concepts of ontology $O_1$, and the columns correspond to the concepts of ontology $O_2$.

The upper threshold, $b_2 = 0.9$, defines the boundary above which the concepts are considered to coincide. Pairs of concepts for which the similarity coefficient falls into the interval $[b_1, b_2]$ are transferred to the community for a final decision. The thresholds values $b_1$, $b_2$ can be defined manually by expert who match ontologies, or automatically based on the analysis of average values from be other matching cases from the problem domain.

The second stage of contextual matching is comparing concept instances. Here we will consider instances for fragments of ontologies limited by the classes organization (company), product, consumer (customer), supplier, process.

To compare operational contexts, the measure of inclusion of sets composed of instances of ontology concepts is analyzed. Each individual set includes instances of a separate concept. The inclusion measure is asymmetric and shows to what extent one set is included in another and vice versa. In this paper, the following measures of inclusion is used.

$$K(A,B) = \frac{n(A \cap B)}{n(A)} \quad (1)$$

$$K(B,A) = \frac{n(A \cap B)}{n(B)}, \quad (2)$$

corresponding to a symmetric measure of Sørensen similarity [23] (the latter is obtained by averaging the presented asymmetric measures), where A and B are the sets under consideration. To determine the same elements of a set, the fuzzy string comparison method used earlier to compare the names of concepts is used. We compare the elements of one set with the elements of another set using the fuzzy string comparison algorithm, summarize the results and divide by the number of elements in the first set. Repeat the procedure for the second set. We choose the maximum of two measures:

$$\mathrm{k} = \max(K(A,B), K(B,A)). \quad (3)$$

### 3.2 Neural Network for Concept Matching

In the proposed approach the neural network is used to analyze ontology concepts and cross-search for similar concepts in matching ontologies. The network is based on the Word2Vec model and uses pre-trained English language model. The neural network consists of three layers: the input, the hidden and the output layer. The input layer corresponds to the dimension of the word vector in Word2Vec model. The hidden layer of the neural network stores a vector model of titles, and can be represented by a matrix of size [m; n], where m is the number of words in the model dictionary, and n is the size of the word vector. The dimension of the output layer corresponds to the size of the dictionary. The neurons of this layer implement the sigmoid activation function [24], which activates neurons depending on the value of the vector in hidden layer. Neurons with the maximum value of the activation function will contain the names of the concepts that are as close as possible to the provided vector of the ontology concept.

Neural network can be trained with any pre-cleaned text corpora. Text can be obtained from encyclopedias articles, electronic libraries, and news or messages archives. A previously trained word model can be used instead of training. It can be general model of language or specific model for a problem area of the ontologies under consideration. The model used can be further trained on the context of ontology concepts (neighboring concepts, relationships between concepts, characteristics of concepts), to provide better reflection of the semantics for a specific ontology.

The method accepts the ontologies in the OWL format and returns the result as an alignment matrix for ontology concepts A. The basic scheme of the method includes use of entities labels and the context of the concepts of these ontologies (see Fig. 2).

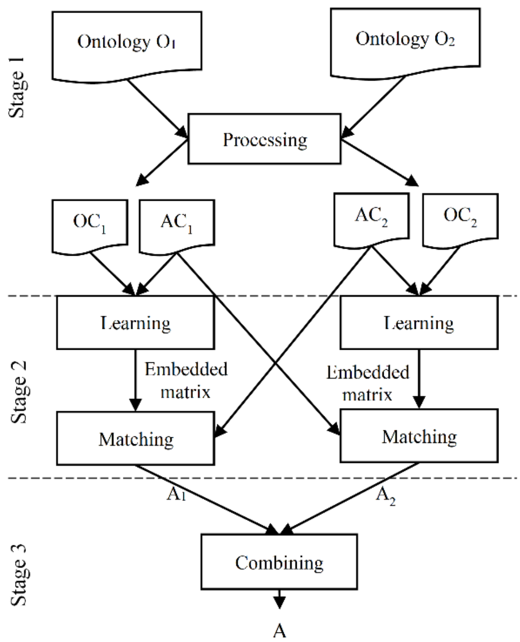


**Fig. 2.** Ontology matching scheme with neural network

The work of the method can be divided into three stages: processing of input ontologies; the mapping of processed ontologies; the composition of intermediate correspondence matrices. During the processing if input ontologies, abstract ($AC_1$, $AC_2$) and operational ($OC_1$, $OC_2$) contexts are identified, as well as the problem area of each ontology is extracted. The mapping is based on methods using matching patterns and concept contexts takes place. Through the composition the final matrix is forming, which is the result of the work of the ontology matching method.

### 3.3 Pattern-Based Matching

The proposed approach allows using the ontology design patterns to identify matching classes. It is assumed that the coincidence of classes and relations between them, according to the provided pattern, indicates the coincidence of specific instances of the concept class. The pattern can be selected from existing pattern libraries, for example, W3C, the ontologydesingpatterns.org portal and the Manchester ODP catalog according to the criteria for matching the problem area of the ontologies under consideration or specified by the community. It is proposed to search for a pattern in each ontology based on the method proposed in the article [22]. To obtain a result, a complete ontology fragment must be fully consistent with the presented pattern. The example of patter in presented on Fig. 3. This pattern can be used to describe relations between customer and supplier, that are actors, where supplier provides prerequisite and basis for creating product that further is utilized by customer.

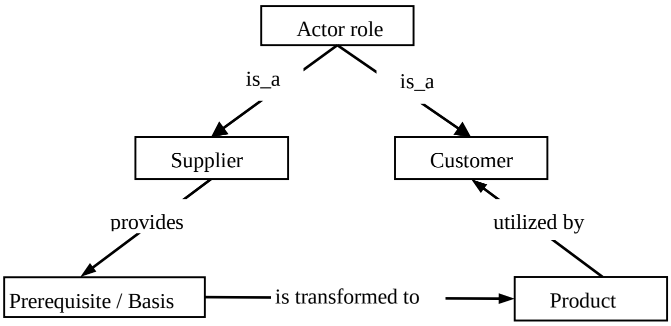


**Fig. 3.** Example of a pattern

The matrix $A_{patt}$ of pattern-based concept correspondence contains unambiguous correspondences between ontology classes that correspond to a given pattern. This result can be used to obtain the final comparison matrix. To do this, it is necessary to select from the matrix $A_{patt}$ the submatrix $A^2_{patt}$ (it is believed that the dimension of the correspondence matrix for the template cannot exceed the dimension of the correspondence matrix for the context method), the matrix $A_{context}$ corresponding in dimension and the rows and columns included in it, and apply the composition function

$$A' = f(A_{\text{context}}, A^*_{\text{patt}}) = \alpha_1 A_{\text{conntext}} + \alpha_2 A^*_{\text{patt}}, \quad (4)$$

where $\alpha_1, \alpha_2, \sum_{i=1,2} \alpha_i = 1$ are weight coefficients that allow changing the effect of the intermediate results of each of the comparison methods (based on the use of ontology templates and based on the analysis of the context of concepts) on the final result . Subsequently, the matrix $A'$ is combined with the matrix $A_{patt}$ with the replacement of the elements in the corresponding rows and columns and we obtain the final matrix A containing the ontology matching model.

### 3.4 Community-Based Matching

For elements of the matrix $A'$, threshold values $b_1, b_2$ are defined that define sets corresponding to three similarity states of the concepts $o_1, o_2$ of the ontologies $O_1$ and $O_2$, respectively.

$$f(o_{1i}, o_{2j}) = \begin{cases} 0, a'_{ij} \in [0, b1) - merging\ not\ found \\ 0.5, a'_{ij} \in [b_1, b_2] - merging\ questionable \\ 1, a'_{ij} \in [b_2, 1] - merging\ found \end{cases}$$

A couple of concepts for which the value of the function $f(o_{1i}, o_{2j}) = 0.5$ are sent for analysis to a community made up of experts - employees of the organization who are specialists in the problem domain to which the ontology belongs. Experts make the final decision in determining which of the ontology concepts are similar. In addition to working with specific pairs of concepts, experts can change threshold values $b_1, b_2$ and edit the templates used.

## 4 Implementation

To verify the proposed method, two ontologies were used that describe the process of creating and maintaining a product in different companies (Fig.4 and Fig. 5). When organizing a joint production cycle, organizations should exchange knowledge about the product life cycle, each providing information based on its own ontology. Accordingly, for the organization of interaction, life cycle ontologies for organizations should be compared.

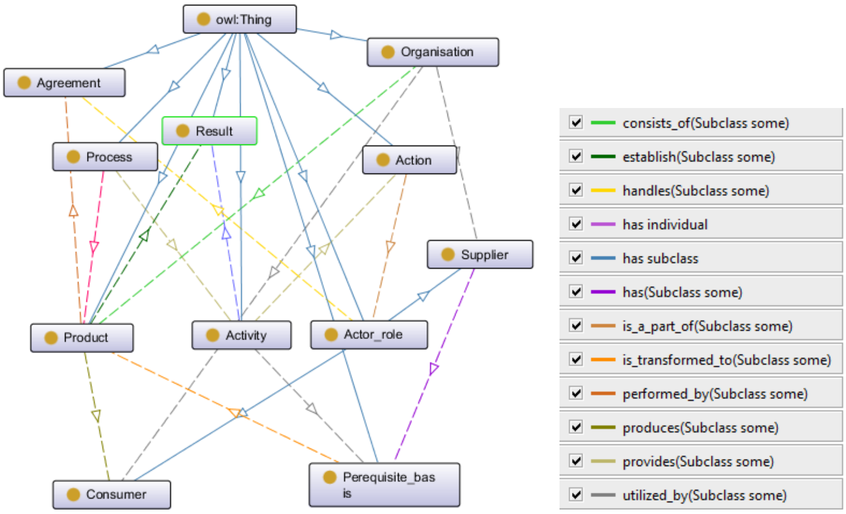


**Fig. 4.** Example ontology $O_1$

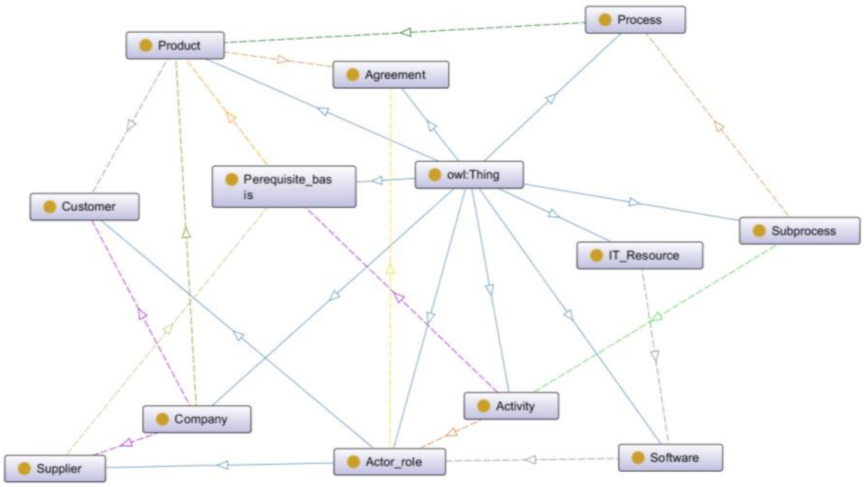


**Fig. 5.** Example ontology $O_2$

These ontologies have been initialized with the following instances of concepts (see Table 1 and Table 2). These operational contexts correspond to supply chain of three companies for car manufacturing. The first one, OC includes $O_1$ ontology concepts instances for supplying products for consumers. Two last ($OC_1$ and $OC_2$) are based on the same ontology $O_2$ and describe processes that corresponds to product creation in companies.

**Table 1.** Examples of OC instances for fragment of $O_1$ ontology

| OC | | | |
|---|---|---|---|
| Organization: | Consumer: | Product: | Supplier: |
| Lear<br>Bosch | GM<br>Volvo<br>Volkswagen<br>Subaru | Seating<br>Car multimedia<br>Steering system | WeSupply<br>DataStream |

First step is a context-based matching of ontology concept. After performing a matching, the following alignment has been obtained based on the names of the concepts that are stored in the matrix $A_{context}$:

- Prerequisite_basis – Prerequisite_basis = 1.00
- Product - Product = 1.00
- Activity - Activity = 1.00
- Actor role - Actor role = 1.00
- Agreement - Agreement = 1.00
- Supplier - Supplier = 1.00
- Process - Process = 1.00
- Thing - Thing = 1.00
- Process concepts - Subprocess = 0.76 => must be shared with the community.

According to pattern-based matching the following results was obtained. In the $O_2$ ontology, a complete match is found for the presented *P* pattern, including the connections between the concepts. Analysis of the $O_1$ ontology for matching with the *P* pattern allows us to determine the coincidence of the structure and, using the "Actor role" and "Product" concepts common to both ontologies, to assume the coincidence of the "Customer" and "Consumer" concepts, which ultimately gives the diagonal alignment matrix $A_{patt}$. So, the result is as follows:

- Company (1) - Organization: $k = 1$ (full match)
- Company (2) - Organization: $k = 1$ (match)
- Customer (1) - Consumer: $k = 0.5$ (less than the lower threshold, not considered further)
- Customer (2) - Consumer: $k = 0.5$ (less than the lower threshold, not considered further)

**Table 2.** Examples of $OC_1$ and $OC_2$ instances for fragment of $O_2$ ontology

| | | | |
|---|---|---|---|
| $OC_1$: | | | |
| *Company:* | *Customer:* | *Product:* | *Process:* |
| Lear | Ford<br>Toyota<br>GM<br>Volvo<br>BMW<br>Nissan<br>Porsche | Seating<br>Interior trim<br>Electrical power management system | Production |
| $OC_2$ | | | |
| *Company:* | *Customer:* | *Product:* | *Process:* |
| Bosch | Volkswagen<br>BMW<br>Ford<br>Toyota<br>Porsche<br>Subaru | Gasoline system<br>Diesel system<br>Electrical drive<br>Starter motor<br>Generator<br>Car multimedia<br>Steering system | Production<br>Design<br>Research |

As a result, a coincidence of ontology concepts was found, the coincidence of which was not revealed by other methods: Company and Organization, which is entered into the intermediate alignment matrix $A_{patt}$.

Moreover, the measure of inclusion defined for Customer - Consumer exceeds the value of the similarity coefficient in the matrix $A_{patt}$ for the corresponding concepts, which indicates the need to increase the similarity coefficient in the matrix $A_{patt}$ to 0.5, based on the similarity of operational concepts for the concepts.

The result of context-based and community-based alignment is combined using the equation (4). For the given example, the coefficients $\alpha_1 = \alpha_2 = 0.5$, which means the equal contribution of each of the methods to the result of the comparison.

Alignment matrix $A'$ is evaluated by the following threshold values: $b_1 = 0.6, b_2 = 0.9$. The interval $[b_1, b_2]$ includes pairs of concepts:

- Process - Subprocess = 0.76
- Consumer - Customer = 0.75

These pairs should be analyzed by the community, after which a joint decision is made whether the concepts under consideration coincide and a matrix A is formed from the matrix $A'$, containing coefficients only for the matching concepts.

## 5 Conclusion

One of the ways to support semantic-based interoperability in a product lifecycle management system is the ontology management technology. Development of efficient methods of ontology matching could significantly improve the knowledge processing and interoperability between various stages of production.

To utilize ontology matching in product lifecycle management system an approach has been proposed that combines four models of ontology matching: pattern-based, context-based, neural network based on natural language model, and community-driven matching. The results from first three methods are composed to temporal alignment matrix that is adjusted by experts of production lifecycle.

The architecture of service for ontology matching method has been also proposed in the paper. The architecture allows to combine various modules for ontology matching by adding or removing any of them and adjusting the function that calculates the temporal alignment matrix. The usage of community-based matcher after the automated matching allows to utilize knowledge and experience of the product lifecycle management system's experts if autonomous matchers cannot find correspondences for the problem domain. It is expected that community-based matcher will be used in the cases of strong uncertainty, when the most of alignment coefficients calculated by autonomous matchers will belong to the ranges of ambiguous alignment coefficient values.

The future work will be aimed on studying of method appropriateness for industrial practice for close integration of a new supplier or new stage to the product lifecycle. More of real cases will be gathered and tested in order to estimate performance and accuracy of the proposed method.

**Acknowledgments** The reported study was funded by RFBR, project number 20-07-00904 for section 3 of ontology matching methods and by Russian State Research No. 0073-2019-0005 in the other sections.

## References

1. Mohd Ali, M., Rai, R., Otte, J.N., Smith, B.: A product life cycle ontology for additive manufacturing. Computers in Industry. 105, 191–203 (2019). https://doi.org/10.1016/j.compind.2018.12.007.
2. Nyffenegger, F., Hänggi, R., Reisch, A.: A Reference Model for PLM in the Area of

Digitization. In: IFIP Advances in Information and Communication Technology. pp. 358–366 (2018). https://doi.org/10.1007/978-3-030-01614-2_33.

3. Teslya, N., Savosin, S.: Matching Ontologies with Word2Vec-Based Neural Network. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 745–756. Springer Verlag (2019). https://doi.org/10.1007/978-3-030-24289-3_55.
4. Otero-Cerdeira, L., Rodríguez-Martínez, F.J., Gómez-Rodríguez, A.: Ontology matching: A literature review. Expert Systems with Applications. 42, 949–971 (2015). https://doi.org/10.1016/j.eswa.2014.08.032.
5. Ngo, D., Bellahsene, Z.: YAM++: A multi-strategy based approach for ontology matching task. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 7603 LNAI, 421–425 (2012). https://doi.org/10.1007/978-3-642-33876-2_38.
6. Schönteich, F., Kasten, A., Scherp, A.: A pattern-based core ontology for product lifecycle management based on DUL. In: CEUR Workshop Proceedings. pp. 92–106 (2018).
7. Bruno, G., Antonelli, D., Villa, A.: A reference ontology to support product lifecycle management. Procedia CIRP. 33, 41–46 (2015). https://doi.org/10.1016/j.procir.2015.06.009.
8. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38721-0.
9. Raad, E., Evermann, J.: The role of analogy in ontology alignment: A study on LISA. Cognitive Systems Research. 33, 1–16 (2015). https://doi.org/10.1016/j.cogsys.2014.09.001.
10. Hecht, T., Buche, P., Dibie, J., Ibanescu, L., dos Santos, C.T.: Ontology Alignment Using Web Linked Ontologies as Background Knowledge. In: Studies in Computational Intelligence. pp. 207–227 (2017). https://doi.org/10.1007/978-3-319-45763-5_11.
11. Lin, F., Sandkuhl, K.: A survey of exploiting WordNet in ontology matching. IFIP International Federation for Information Processing. 276, 341–350 (2008). https://doi.org/10.1007/978-0-387-09695-7_33.
12. Manjula Shenoy, K., Shet, K.C., Dinesh Acharya, U.: NN based ontology mapping. Communications in Computer and Information Science. 296 CCIS, 122–127 (2013). https://doi.org/10.1007/978-3-642-35864-7_18.
13. Ganesh Kumar, S., Vivekanandan, K.: Odmm - An ontology based deep mining method to cluster the content from web servers. Journal of Theoretical and Applied Information Technology. 74, 162–170 (2015).
14. Jayawardana, V., Lakmal, D., De Silva, N., Perera, A.S., Sugathadasa, K., Ayesha, B.: Deriving a representative vector for ontology classes with instance word vector embeddings. In: 7th International Conference on Innovative Computing Technology, INTECH 2017. pp. 79–84 (2017). https://doi.org/10.1109/INTECH.2017.8102426.
15. Wohlgenannt, G., Minic, F.: Using word2vec to build a simple ontology learning system. In: CEUR Workshop Proceedings. pp. 2–5 (2016).
16. Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT 2019. pp. 4171–4186 (2019).

17. Hammar, K.: Ontology Design Patterns in WebProtégé. In: 14th International Semantic Web Conference (ISWC-2015). pp. 1–4. , Betlehem (2015).
18. Scharffe, F., Zamazal, O., Fensel, D.: Ontology alignment design patterns. Knowledge and Information Systems. 40, 1–28 (2014). https://doi.org/10.1007/s10115-013-0633-y.
19. Smirnov, A., Teslya, N., Savosin, S., Shilov, N.: Ontology Matching for Socio-Cyberphysical Systems: An Approach Based on Background Knowledge. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 10531 LNCS, 29–39 (2017). https://doi.org/10.1007/978-3-319-67380-6_3.
20. Levashova, T., Lundqvist, M., Sandkuhl, K., Smirnov, A.: Context-based modelling of information demand: Approaches from information logistics and decision support. Proceedings of the 14th European Conference on Information Systems, ECIS 2006. 171, (2006).
21. Teslya, N., Smirnov, A., Levashova, T., Shilov, N.: Ontology for Resource Self-organisation in Cyber-Physical-Social Systems. In: Klinov, P. and Mouromtsev, D. (eds.) Knowledge Engineering and the Semantic Web, Kesw 2014. pp. 184–195. Springer-Verlag Berlin (2014). https://doi.org/10.1007/978-3-319-11716-4_16.
22. Seigerroth, U., Kaidalova, J., Shilov, N., Kaczmarek, T.: Semantic Web Technologies in Business and IT Alignment: Multi-Model Algorithm of Ontology Matching. AFIN 2013, The Fifth International Conference on Advances in Future Internet. 50–56 (2013).
23. Sørensen, T.: A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and its Application to Analyses of the Vegetation on Danish Commons. Det Kongelige Danske Videnskabernes Selskab Biologiske Skrifter. 5, 1–34 (1948).
24. Elfwing, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. Neural Networks. 107, 3–11 (2018). https://doi.org/10.1016/j.neunet.2017.12.012.