

# Semi-automatic Ontology Matching Approach for Integration of Various Data Models in Automotive

Václav Jirkovský<sup>1</sup>(✉), Petr Kadera<sup>1</sup>, and Nestor Rychtycký<sup>2</sup>

<sup>1</sup> Czech Institute of Robotics, Informatics, and Cybernetics, Czech Technical University in Prague, Zikova 4, 166 36 Prague, Czech Republic

{[vaclav.jirkovsky](mailto:vaclav.jirkovsky@ctiirc.cvut.cz),[petr.kadera](mailto:petr.kadera@ctiirc.cvut.cz)}@ctiirc.cvut.cz

<sup>2</sup> Ford Motor Company, Dearborn, MI, USA

[nrychtyc@ford.com](mailto:nrychtyc@ford.com)

**Abstract.** All manufacturing companies need to be able to closely monitor the processes, labor, tooling, parts and throughput on the assembly plant floor. This might be a challenging task because of a large number of plant floor applications that operate using different hardware and software tools. In many cases, there are a large number of devices that need to be monitored and from which critical data must be extracted and analyzed. This situation calls for the use of an architecture that can support data from heterogeneous sources and support the analysis of data and communication with these devices. Ontologies can be developed to facilitate a proper understanding of the problem domain, and subsequently, knowledge from external sources can be shared through linked open data or directly integrated (mapped) using an ontology matching approach. In this paper, we demonstrate how ontological data description may facilitate interoperability between a company data model and new data sources as well as an update of stored data via ontology matching. The MAPSOM system (system for semi-automatic ontology matching) is introduced and described in this paper, and subsequently, an example of new data model integration is demonstrated using the MAPSOM system.

**Keywords:** Heterogeneity · Ontology · Ontology matching · Self-organizing map · Active learning

## 1 Introduction

All manufacturing companies need to be able to closely monitor the processes, labor, tooling, parts and throughput on the assembly plant floor. This is often complicated because of a large number of plant floor applications that operate using different hardware and software tools. In many cases, there are a large number of devices that need to be monitored and from which critical data must be extracted and analyzed. This situation calls for the use of an architecture

that can support data from heterogeneous sources and support the analysis of data and communication with these devices. Another factor to consider are the significant differences between the hardware/software at different manufacturing facilities even though they may be building the same product. This can be due to a variety of reasons including the availability of tooling at different locations around the world, local differences and the need to support different versions of hardware and software at many plants. In many cases, the data also needs to be localized to support a plant and textual data may require translation using either machine or human translation. Other issues that need to be addressed include different units of measurement between locations (imperial vs. metric) and even different formats for dates between plants around the world. All of these factors contribute to the difficulty of the problem in developing a solution for integrating manufacturing data on the plant floor.

There are a number of different solutions that can be currently applied to this data heterogeneity problem. A data warehouse can be built to include the various data sources that are present, but this will require the development of a data model that will represent all of the different data sources. This is a difficult process because the different variations and inconsistencies between disparate data sources need to be correctly represented in the common data model. In many cases, the same data element has different names and formats in separate databases which then need to be merged into a single data model. The data model needs to be maintained and modified as new data sources are incorporated into the production system. Commercial vendor solutions can also be applied but often require the use of proprietary data representation models that cannot be easily integrated with external systems.

An ideal solution would allow for the usage of a simplified data representation model that can support various data sources and uses an open standard that can exchange information easily between systems. This solution should also allow for easy maintainability as there will be frequent additions and modifications to the data model. It would also consolidate manufacturing data using a global open standard and would be able to represent and communicate with these different data sources. It is also important that the proposed solution supports knowledge expressiveness and reasoning as well as the ability to keep track of the source of each data item. These requirements led us to select the use of semantic technologies to develop a common architecture for the manufacturing data model.

Semantic technologies are built around common XML-based representation standards such as RDF/OWL and provide a framework for building applications that support heterogeneous data sources. Ontologies can be developed to facilitate a proper understanding of the problem domain, and subsequently, knowledge from external sources can be shared through linked open data or directly integrated (mapped) using an ontology matching approach. Within the framework of this work we utilized our previous experiences with the development of manufacturing ontologies and will be building upon those ontologies in this work [15, 16]. Other advantages for semantic technologies include flexibility,

standardization, expressiveness, provenance and a reasoning/inferencing capability. There are many vendors who have built tools to support these semantic web standards which can support manufacturing data integration and analysis.

The goal of this paper is to demonstrate how ontological data description may facilitate interoperability between a company data model and new data sources as well as an update of stored data via ontology matching. Furthermore, a user involvement in the ontology matching process is a very important feature within the automotive industry. Knowledge management and a matching of new data models are very important not only within automotive but also in every distributed environment including agent-based and SOA-based industrial systems.

This paper is organized as follows: first we provide a general overview of the heterogeneity problem. Then, we introduce the ontology matching problem including similarity measures aggregation and user involvement possibilities in the ontology matching problem. Next, we demonstrate an integration of the Ford supply chain ontology and MS Excel spreadsheet representing a list of spare parts together with many important details on MAPSOM system which utilize a self-organizing map, visualization methods, and active learning for ontology matching.

## 2 Heterogeneity

An essential prerequisite for an accurate integration is to reduce heterogeneity between data models—the shared ontology and a data source for integration in our case. Many different types of heterogeneity have been defined and discussed e.g. in [1,4,5]. The most obvious types of heterogeneity are as follows [6]:

- **Syntactic heterogeneity** represents the situation when two data sources are expressed in different representational language. In the case of ontologies, this situation happens when ontologies are modeled in different representation formalisms, e.g., OWL<sup>1</sup> and KIF<sup>2</sup>.
- **Terminological heterogeneity** stands for different names of the same entity in different data models. An example may be a usage of different natural language—Wing vs. Křídlo (Czech term); or usage of synonyms—Wing vs. Fender.
- **Semantic heterogeneity** (a.k.a. logical mismatch) represents differences in modeling the same domain of interest. This logical mismatch arises due to a utilization of different axioms for defining the same elements from data sources. Two different mismatches may be distinguished: 1. the conceptualization mismatch—differences between modeled concepts; 2. the explicitation mismatch—differences how the concepts are expressed as discussed in [19]. Moreover, [2] identifies and describes three essential reasons for conceptual differences:

<sup>1</sup> Web Ontology Language - <https://www.w3.org/OWL>.

<sup>2</sup> Knowledge Interchange Format - <http://www-ksl.stanford.edu/knowledge-sharing/kif>.

- Difference in coverage—two data models describe different (possibly overlapping) parts of the world at the same level of detail and from the same perspective.
  - Difference in granularity—two data models describe the same part of the world from the same perspective but with different levels of detail.
  - Difference in scope—two data models describe the same part of the world with the same level of detail but from a different perspective.
- **Semiotic heterogeneity** stands for a different interpretation of entities by various people. In other words, entities from two different data models with the same semantic interpretation may be interpreted by an interpreter (human, expert system, etc.) with regard to the context. The semiotic heterogeneity is difficult to detect and solve by computer and often by a human as well.

In general, more than one type of heterogeneity occurs at once. It is caused for example because of various ad-hoc tailored system integration, etc.

### 3 Ontology Matching

In this section, we introduce the ontology matching problem [6]. The term ontology is defined as an explicit specification of a conceptualization [7] sometimes extended with the requirement for a shared conceptualization. In other words, an ontology represents a conceptualization of some particular domain which is shared among users (if everybody has his unique ontology they cannot communicate to each other) and is expressed by using a particular explicit means.

The goal of ontology matching is to find correspondent entities expressed in different ontologies. The simplest possible relation between elements is a one-to-one relation, e.g., Person maps to Human. Furthermore, there are more complex types of a semantic relationship, e.g., Student maps to Undergrad-Student and Postgrad-Student as well.

Ontology matching systems are widely used especially in the Semantic Web domain where the systems are responsible for the integration of a lot of large ontologies. Thus, the techniques for finding relations have to be fully automatic. However, even though many researchers have been trying to develop fully automatic and faultless matching systems, there are many cases where faultless matching could be achieved only by means of a skilled user supervision.

The goal of this paper is to introduce a hybrid matching system prototype which is responsible for matching elements from an MS Excel<sup>3</sup> file (XLS) to an ontology. We assume an XLS is a general spreadsheet file, i.e., we are not limited for example to Parcelized Ontology Model [9]—the approach how to store an ontology in an XLS. This approach has several differences comparing to a matching of two ontologies. The first difference is the way of elements extraction for matching. Naturally, ontology elements to be matched are clearly given (strings representing concepts, object properties, etc.). On the other hand,

<sup>3</sup> Microsoft Excel <https://products.office.com/en-us/excel>.

we must consider what should be the element for matching within an XLS. Should it be a content of cells, column names, sheet names, or the name of the XLS file? The second difference is the process of the subsequent XLS and ontology mapping. In this case, it is more difficult to decide what is a concept, an instance (an individual), a property (data or object) in the source XLS and more in merged ontology as an outcome of ontology mapping. For example, an XLS could be in many cases decomposed as a table name—a concept name; table columns—concept properties; table rows—individuals belonging to the concept.

A problem of the ontology matching (i.e., find out related entities) may be expressed as a problem of finding the most similar entities. There are many various already implemented similarity measures for computing a similarity of entities. In the following paragraphs, essential types of similarity measures are shortly introduced.

### 3.1 Basic Similarity Measures

**String-Based Techniques.** These methods are based on comparing strings as the name indicates. They compare a name, labels or comments of entities (e.g., a concept represented specific cultivar of apple could be characterized by following strings: name—anton; label—Antonovka apple?; comment (1)—A popular small green culinary apple variety from Russia; comment (2)—It has ability to tolerate extreme cold). A prefix or suffix similarity measure tests if one string is a prefix or suffix of another. Next, very widely used similarity measure is n-gram. This method computes the number of common n-grams (sequences of n characters) between two strings.

**Language-Based Techniques.** This group of similarity measures rely on using Natural Language Processing (NLP) methods. NLP is used for facilitating an extraction of meaningful terms. NLP methods can be divided into intrinsic methods (i.e., linguistic normalization) and extrinsic. Extrinsic methods utilize external resources, e.g., WordNet [14]. WordNet is an electronic lexical database for English, based on the notion of synsets or sets of synonyms. Furthermore, WordNet provides hypernyms and meronyms as well.

**Structure-Based Techniques.** These techniques aim to compare a structure of entities that can be found in ontologies. Structure-based techniques can be divided into comparison of an internal structure of an entity or the comparison of the entity together with surrounding entities. An example of a structure based similarity measure is the structural topological dissimilarity on a given hierarchy [18]. **Extensional Techniques.** This approach is applicable when concept individuals are available. The idea is based on the fact that if two concepts have the same individuals then they should represent identical concept.

**Semantic-Based Techniques.** Semantic-based methods belongs to the deductive methods. These methods alone do not perform well when they are utilized for

an inductive task like the ontology matching. Thus, semantic based techniques are suitable for verification or amplification of pre-alignments (i.e., entities which are presupposed to be equivalent). Examples of semantic-based techniques are propositional satisfiability, modal satisfiability techniques, or description logic based techniques.

### 3.2 Similarity Aggregation

The basic similarity measures are suitable for different dissimilarity kinds. Therefore, the basic measures may be utilized as building blocks of some complex solution. There are several techniques how to use these blocks together for ontology matching. The most widely used method is to aggregate them.

There are several proposed and implemented methods for the similarity measures aggregation. We will provide short overview of these methods in the following paragraphs.

**Weighted Product and Weighted Sum.** Triangular norms are well-known as conjunction operators in the uncertain calculi and weighted product (belonging to the triangular norms) may be used for ontology matching. The weighted product between two objects  $x, x'$  from set of objects  $O$  is as follows:

$$sim(x, x') = \prod_{i=1}^n sim_i(x, x')^{w_i},$$

where  $sim_i(x, x')$  is the  $i^{th}$  similarity measure of objects  $x, x'$ . Analogously, the weighted sum can be considered for example as a generalization of the Manhattan distance with weighted dimensions.

**Multidimensional Distances.** This aggregation is suitable for independent basic similarity measures. An example of multidimensional distances is Minkowski distance:

$$sim(x, x') = \sqrt[p]{\sum_{i=1}^n sim_i(x, x')^p},$$

where  $sim_i(x, x')$  is the  $i$ th similarity measure of objects  $x, x'$ .

**Machine Learning Approaches.** There are several proposed approaches for utilizing machine learning methods for the ontology matching problem. A similarity measures aggregation may be converted into a supervised machine learning problem with the help of training data containing a set of similarity measure values corresponding to every matching pair together with a value representing positive or negative mapping as described in [8]. Thus, general machine learning methods can be utilized for ontology matching problems, e.g., support vector machines (SVM), decision trees and neural networks.

### 3.3 Semi-automatic Ontology Matching

A fully automatic ontology matching systems are not suitable for all application domains. A system with the highest possible precision and recall is needed for communication among experts and systems from different domains, e.g. in manufacturing or in medicine. One of the possible examples for such a problem is described in [11]. Semi-automatic or manual ontology matching solutions overcome previously mentioned deficiencies. However, these solutions are usually more time consuming.

In other words, semi-automatic solutions are based on a user involvement in ontology matching process. There are three areas in which users can be involved in a matching solution: (i) by providing initial alignments (and parameters) to the matchers, (ii) by dynamically combining matchers, and (iii) by providing feedback to the matchers in order for them to adapt their results [6].

Furthermore, historical records of the prior matching may be used for improving a precision and a recall of an ontology matching. Existing matches positive/negative and a user action history can enhance a matching process to be more interactive and personalized [3].

## 4 Validation Study

In this section, we introduce our solution of the hybrid ontology matching problem. This solution is based on semi-automatic ontology matching system named MAPSOM and its corresponding extension for processing MS Excel files. Next, the approach is demonstrated on the integration of the Ford supply chain ontology and MS Excel file containing spare part information.

### 4.1 MAPSOM

We have extended our previously proposed and developed semi-automatic ontology matching system MAPSOM [10] to be able to compute possible matching pairs between the Ford supply chain ontology and MS Excel file containing spare parts items. This system combines a machine learning approach for a similarity measures aggregation and a user involvement into the ontology matching problem.

The similarity measure aggregation is based on a self-organizing map also known as Kohonen self-organizing maps (SOM/KSOM) [13]. In general, self-organizing maps are a type of neural networks with unsupervised training algorithm. The basic functionality of a SOM is an ability to assign similar input vectors to the same neuron of a SOM output layer.

The user involvement is represented by verification of computed matching values—by means of SOM visualization (see Fig. 4); and next by the active learning process—used for tuning of classified data.

The overall matching process consists of following steps:

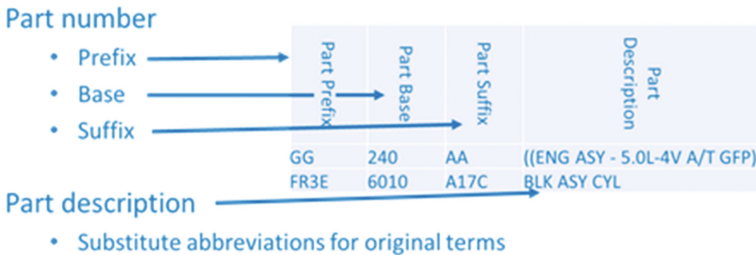
1. Compute desired similarity measures for element pairs
2. Train SOM
3. Compute clusters by means of a hierarchical clustering
4. Compute initial classification (positive or negative) of all neurons as well as of clusters.
5. A user may verify classified neurons and clusters in this step.
6. Conduct active learning process—the most probably badly classified neurons of each cluster are put forward a user.

After these steps, a user has a set of corresponding entities from the both data models (ontology and XLS file) and is ready for subsequent mapping.

## 4.2 Data Models Matching

**Data Models for Subsequent Matching.** The Ford supply chain ontology captures the risk management in the Ford global supply chain. Every car model depends on many different suppliers, and important capability is to be able to determine which vehicles at which plants would be impacted by a potential shortage, e.g., a limited supplier plant operation, a disaster (e.g. tsunami), etc. The Ford ontology captures all needed knowledge about vehicles, manufactures, and processes, and therefore the ontology can infer required information. Furthermore, the ontology would be able also to identify if Ford is dependent on one supplier plant for multiple vehicles.

The second source of items for matching is an Excel file (XLS) containing Ford spare part records. The XLS file has about 62 various columns identifying particular parts. A predominant number of columns contain specific numerical codes or strings composed of abbreviated labels. Obviously, a manual integration of such a data would be very time consuming and because of big volume of records probably impossible. Furthermore, a data preprocessing is needed for enabling an automatic model matching. The data preprocessing is described in the following paragraphs. An example of spare part records is illustrated in the Fig. 1.



**Fig. 1.** A segment of Ford spare parts MS Excel file



**Data Preprocessing.** The essential step preceding matching of models is data preprocessing. Data for matching could be enriched with additional and valuable information during this step.

Part numbers conceal a lot of important information which may make the matching more precise. Thus, we need to parse and decode these items. Part numbers are divided into three categories—regular parts (e.g., a cylindrical block); hardware and utility parts (e.g., machine screws); special service tools. Furthermore, two different part coding notations may be distinguished—before and after 1998.

In this paragraph, we provide a detail description of regular parts. Regular parts consist of tree part—prefix, base, and suffix. The prefix is represented by a four-digit alphanumeric character and denotes year, model, and engineering office of a given part. The base part has four or five digits and indicates a part. For example, base part number series 2000–2874 represent brakes. The suffix indicates change level, i.e., A: original design; B: changed once, etc. An example of spare part number decoding is illustrated in Fig. 2.



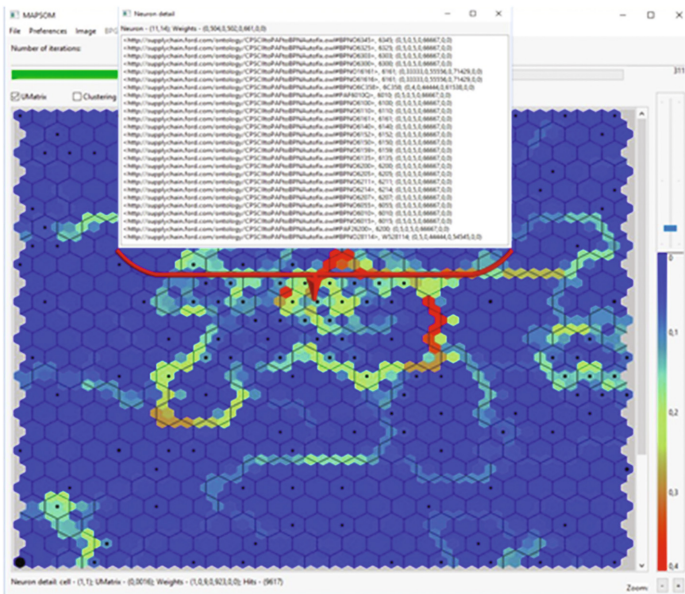
**Fig. 2.** An example of a spare part decoding

Next, every spare part has a description. This description is formed from abbreviations and therefore it could be hardly utilized for data models matching. We used Ford Speak for decoding a part description. The Ford Speak is database of acronyms, definitions, and terms originally designed for facilitating data exchange between manufacturers. Items from the Ford Speak may have more than one value, and we cannot decide which is the correct one. This fact decreases accuracy and increases complexity of a matching but a utilization of a part description is probably impossible without this preprocessing step. The decoding of a spare part description is illustrated in Fig. 3.



**Fig. 3.** An example of the part description decoding

**Models Matching.** After preprocessing, the data models matching by means of MAPSOM system may be conducted. Steps of the matching task and their order are stated in the Sect. 4.1. A SOM had a hexagonal topology and 25 neurons in both dimensions in our experiments. A training algorithm had following parameters [12]—a neighborhood function: Gaussian (parameter: 0.5); a neighborhood size: 5; an adaptation of learning rate: linear; an initial value of learning rate: 0.4. First, we trained self-organizing map. Training data were pairs composed of ontology elements from Ford supply chain ontology and preprocessed elements from MS Excel spare part list. A number of iteration was set to 1000 but after 300 iterations there were no evident changes of the output layer neuron weights. Thus, we could stop the learning algorithm after 300 iterations. The trained SOM within MAPSOM system is illustrated in Fig. 4. Next, we have to conduct initial classification of neurons. We used a Boolean conjunctive classifier [8] for an initial neuron classification as well as for a subsequent cluster classification.



**Fig. 4.** The trained SOM visualization by means of U-Matrix together with demonstration of data pairs represented by a particular neuron

Subsequently, pairs with the help of SOM visualization could be analyzed. We have several ways how to process it:

- **Clusters and their classification**—a cluster classification is computed according to its center of gravity. Clearly, the cluster classification is dependent on a count of clusters (centroids are translating). Therefore, MAPSOM system offers an option for varying different numbers of clusters according to a given data.

- **U-Matrix visualization**—important neurons as well as neuron clusters may be recognized by means of U-Matrix (unified distance matrix) visualization [17]. U-Matrix displays distances between neurons (blue color - a short distance; red color - a long distance), and thus we can recognize an important neuron in the middle of trained SOM in Fig. 4. In general, neurons with decidedly positive and negative classification have a longer distance to remaining neurons in many cases. It enables a recognition of positive matching even without initial classification.
- **Hit histogram**—this additional information denotes how many pairs are represented by a particular neuron. Hit histogram may be combined with U-Matrix visualization as well as with visualization of clusters.

The last step of models matching is an active learning process. A utilization of this process has several benefits. First, the found positive matching pairs may be improved during this step (the least probable matchings are presented for verification and user can change a classification of a corresponding neuron or a cluster). In other words, a user should be capable of verifying correctness of discovered neuron classifications. We used active learning mainly for verification of a given matching in our case.

## 5 Discussion

The knowledge management task is a difficult task even in the case of one data model. Furthermore, an integration of various data models together with a maintenance of their consistency is very complex task which is essential in many applications and domains including agent-based and SOA-based industrial systems. Semantic Web technologies may offer solution for these tasks. There are many already proposed and implemented systems for ontology matching which could be used for integration of various data sources.

In many cases, manual matching could be very time consuming or even impossible due to huge number of matching entities. Thus, many researchers and developers try to develop fully-automatic matching systems. These systems are capable to process very big number of entities. On the other hand, the precision of the matching has to be taken into account in many domains, i.e., healthcare, industrial domain, etc. A user involvement in semi-automatic matching system is the best solution how to process huge data amount and ensure a satisfactory precision of matching.

A user should be involved not only within the matching process itself. A user may provide additional valuable information for the matching—mainly in the preprocessing phase. In this paper, we have shown that the preprocessing phase could be essential for enabling matching in many applications. Here, available information in XLS file are not sufficient for any reasonable matching. Thus, a user is able to extend knowledge about matching items by decoding part numbers and abbreviated part description. Apparently, a user is not involved for example in converting all part numbers manually but in providing a definition for the

system how to convert part numbers and corresponding descriptions. A blind automatic matching approach cannot achieve such outcomes.

A user-friendly visualization of matching data during the matching process is essential for proper understanding of data as well as a matching process itself. A suitable visualization method is strongly dependent on methods and technologies which are used for the matching. However, visualization methods have to reflect several assumptions because of ensuring usable and efficient user interaction, i.e., offer a capability to manipulate with a whole set of similar entities (for example change a classification of the proposed matching), provide a mechanism which recommends suitable data for user verification, etc.

## 6 Conclusions and Future Work

In this paper, we introduced the approach how to utilize ontology matching for semi-automatic various data model integration and how important is the user involvement in this process together with the preprocessing phase. The approach is demonstrated on the matching task where spare parts are matched to the Ford supply chain ontology.

In this article, we focused on spare part matching to the ontology elements. In future work, we will be aimed at a utilization of preprocessed (extended) XLS data for a derivation of new concepts, properties, and relationships and how to conduct the most precise mapping between the original ontology and the new derived ontology segments. This extension of the ontology should offer a better interoperability as well as efficiency for supply chain management. We would like to automate the process of ontology management (e.g., adding new concept into an existing ontology) by means of a utilization of ontology learning methods and cover a creation of following ontology parts step by step—terms, concepts, concepts hierarchy, relations, relations hierarchy, axioms. We especially will emphasize a user involvement in previously mentioned research directions for achieving the best outcomes required within the automation domain.

**Acknowledgment.** This work is supported through the Ford Motor Company University Research Proposal (URP) program and by institutional resources for research by the Czech Technical University in Prague, Czech Republic.

## References

1. Batini, C., Lenzerini, M., Navathe, S.B.: A comparative analysis of methodologies for database schema integration. *ACM Comput. Surv. (CSUR)* **18**(4), 323–364 (1986)
2. Benerecetti, M., Bouquet, P., Ghidini, C.: On the dimensions of context dependence: partiality, approximation, and perspective. In: Akman, V., Bouquet, P., Thomason, R., Young, R. (eds.) *CONTEXT 2001*. LNCS, vol. 2116, pp. 59–72. Springer, Heidelberg (2001). doi:[10.1007/3-540-44607-9\\_5](https://doi.org/10.1007/3-540-44607-9_5)

3. Bernstein, P.A., Melnik, S., Churchill, J.E.: Incremental schema matching. In: Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 1167–1170. VLDB Endowment (2006)
4. Bouquet, P., Ehrig, M., Euzenat, J., Franconi, E., Hitzler, P., Krotzsch, M., Serafini, L., Stamu, G., Sure, Y., Tessaris, S.: Specification of a Common Framework for Characterizing Alignment (2005)
5. Euzenat, J.: Towards a principled approach to semantic interoperability. In: Proceedings of IJCAI 2001 Workshop on Ontology and Information Sharing, pp. 19–25 (2001). (No commercial editor)
6. Euzenat, J., Shvaiko, P., et al.: *Ontology Matching*, vol. 18. Springer, Heidelberg (2007)
7. Gruber, T.R., et al.: A translation approach to portable ontology specifications. *Knowl. Acquis.* **5**(2), 199–220 (1993)
8. Ichise, R.: Machine learning approach for ontology mapping using multiple concept similarity measures. In: Seventh IEEE/ACIS International Conference on Computer and Information Science, ICIS 2008, pp. 340–346. IEEE (2008)
9. Standardized product ontology register and transfer by spreadsheets – Part 1: Logical structure for data parcels. International standard, International Electrotechnical Commission, Geneva, CH (2014)
10. Jirkovský, V., Ichise, R.: MAPSOM: user involvement in ontology matching. In: Kim, W., Ding, Y., Kim, H.-G. (eds.) JIST 2013. LNCS, vol. 8388, pp. 348–363. Springer, Cham (2014). doi:[10.1007/978-3-319-06826-8\\_26](https://doi.org/10.1007/978-3-319-06826-8_26)
11. Jirkovský, V., Obitko, M.: Ontology mapping approach for fault classification in multi-agent systems. *IFAC Proc. Vol.* **46**(9), 951–956 (2013)
12. Kohonen, T.: Essentials of the self-organizing map. *Neural Netw.* **37**, 52–65 (2013)
13. Kohonen, T.: The self-organizing map. *Neurocomputing* **21**(1), 1–6 (1998)
14. Miller, G.A.: WordNet: a lexical database for English. *Commun. ACM* **38**(11), 39–41 (1995)
15. Ostrowski, D., Rychtycky, N., MacNeille, P., Kim, M.: Integration of big data using semantic web technologies. In: 2016 IEEE Tenth International Conference on Semantic Computing (ICSC), pp. 382–385. IEEE (2016)
16. Rychtycky, N., Raman, V., Sankaranarayanan, B., Kumar, P.S., Khemani, D.: Ontology re-engineering: a case study from the automotive industry. In: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 3974–3981. AAAI Press (2016)
17. Ultsch, A.: Self-organizing neural networks for visualisation and classification. In: Opitz, O., Lausen, B., Klar, R. (eds.) *Information and Classification*, pp. 307–313. Springer, Heidelberg (1993). doi:[10.1007/978-3-642-50974-2\\_31](https://doi.org/10.1007/978-3-642-50974-2_31)
18. Valtchev, P., Euzenat, J.: Dissimilarity measure for collections of objects and values. In: Liu, X., Cohen, P., Berthold, M. (eds.) *IDA 1997*. LNCS, vol. 1280, pp. 259–272. Springer, Heidelberg (1997). doi:[10.1007/BFb0052846](https://doi.org/10.1007/BFb0052846)
19. Visser, P.R., Jones, D.M., Bench-Capon, T.J., Shave, M.J.: Assessing heterogeneity by classifying ontology mismatches. In: *Proceedings of the FOIS*, vol. 98 (1998)