Domain dependant matching of MES knowledge and domain independent mapping of AutomationML models

Miriam Schleipen, Dirk Gutting, Franziska Sauerwein
Fraunhofer IOSB
Fraunhoferstr.1
D-76131 Karlsruhe
miriam.schleipen@iosb.fraunhofer.de

Abstract

Knowledge is one of the most valuable goods within production planning. It can be divided into domain dependant knowledge and domain independent knowledge. An example for domain dependant knowledge is Manufacturing Execution System (MES) specific knowledge. An example for domain independent knowledge is the XML-based data exchange format AutomationML. Both types of knowledge are necessary.

This contribution shows the application of ontology mapping and matching methods in the context of domain dependant and domain independent knowledge to support the engineer in working with the data and improve the usage of such information processing.

1. Introduction

MES [1] describe a sort of production management software with direct access to the field controllers e.g. PLCs. The acquired knowledge about the actual production state is aggregated within the MES and key performance indicators (KPI, see [2] are derived and calculated. MES are one example for a system type with specific domain language within the heterogeneous IT landscape in production environment.

Ontologies [3] try to enable the usage of domain specific concepts. They are used to simplify the exchange of knowledge und introduce semantics into the modeled data. Therefore, they include domain-dependant concepts which can be processed by systems as well as humans.

For MES as supervising and controlling systems it is evident to collect and to use data and information from different heterogeneous data sources. In an ideal environment, a homogeneous data format for the exchange of information is used. This format has to be as generic as possible, but as detailed as required. MES therefore require two specific paths in information management to increase efficiency.

On the one hand, tools and software components have to be developed which are able to process and fusion data from heterogeneous sources. Ontologies and ontology mapping techniques are one possible solution to this problem. The MES ontology in [4] takes specialties of the MES domain into account.

On the other hand, the usage of one homogeneous data and modeling format for the exchange between different tools would help. AutomationML [5] and the sub-format CAEX [6] and corresponding fusion and mapping methods are one example for such a format in the production environment (see [7]).

The outline of the paper is as follows: In chapter 2 and 3, the MES ontology and AutomationML are introduced. Chapter 4 summarizes different matching and mapping methods. In chapter 5 the authors show a possibility for domain dependant matching of MES knowledge. In Chapter 6 the domain independent mapping of AutomationML models is explained. In chapter 7, summary and outlook finalize the paper.

2. MES ontology

A working group within the German association of engineers (VDI) developed a common MES domain dependant concept – the MES ontology (see [8]). As common model and method for comprehension, it includes the most frequently used data items. The developers decided to use the web ontology language (OWL) as XML-based ontology data format to model and store the ontology. The MES ontology is structured on different hierarchy levels: It differs in folders and data items. The MES domain is divided into different main areas e.g. product, process, resource. Each of the main areas is structured into prerequisites which are communicated from MES to automation level and results which are transferred from automation level to MES. An additional division is made between single values and curves. Examples of concrete concepts referring to the product section within the MES ontology are a product type, product batch, product quality status, test schedule, product test value. The meaning of a each concept is explained in further detail within the ontology. The example of the test schedule shows this. Test procedures are planned for quality assurance purposes and for monitoring the production. The test schedule describes the procedure to be followed for testing a part or a product. Furthermore, the MES ontology defines that test schedules can contain only the ID and designation or also the specifications which shall be exchanged between MES and machine.

The MES ontology defines which data type can be assigned to an ontology concept. Possible data types are integer, boolean, real, string and datetime. It differs in mandatory and optional data.

Each involved research or industry users defines the correspondences between the MES ontology and the individual interface by means of a manual mapping. One possible tool to assist this mapping is OWLTreePrint from the Fraunhofer IOSB. It imports the user-specific interface as instances of the ontology which are mapped to the concepts of the MES ontology by the user. One example is the mapping of a test schedule of the concrete individual interface to the test schedule of the MES ontology.

3. AutomationML

AutomationML is an independent standardized data format for the exchange of data between different engineering tools used during the plant planning process. It represents a domain independent concept due to the possibility to describe information for all different tools and disciplines involved in the plant planning process.

AutomationML is not yet another data format, but relies on existing data formats and defines additional rules and restrictions for their usage. This saves specification and testing time and makes the usage of proved concepts possible.

CAEX (Computer Aided Engineering Exchange) was chosen as top-level format for AutomationML (see [9]). Due to object orientation, it is suitable for modeling complex interlinked planning information. COLLADA (COLLAborativ Design Activity) was chosen for geometric and kinematic information and PLCOpen XML was chosen for logics and behavior (see ([9], [10]). Additionally, the usage of further XML based standard data formats is possible.

Main task of AutomationML is the usage of the existing formats and the intelligent interlinking between them by links and references. The data remains therefore within the correspondent data formats and are stored separately. This simplifies the exchange and adaption of complex plant models. Additionally, if users already possess data in the existing data formats, they can reuse them via an introducing by means of linking them into the higher-level structure.

CAEX as top-level format is of interest for this contribution. The XML structures of CAEX are divided into different libraries and a hierarchical plant model (InstanceHiarchy). The libraries consist of interfaces (InterfaceClassLibrary), type description (SystemUnitClassLibrary) and semantic descriptions of plant components (RoleClassLibrary). It consists of object-oriented modeling techniques and provides different types of possible referencing and interlinking methods.

4. Matching and mapping methods

In the following section, the authors explain methods for a domain dependant matching of MES knowledge and a domain independent mapping of AutomationML models.

4.1. Matching methods

The usage of abstract domain dependant knowledge in terms of ontologies, the concrete interface has to be matched to the concepts of this ontology e.g. the MES ontology. This means that the elements of the concrete interface become individuals of specific concepts within the ontology. It can be difficult to find a matching between concepts and individuals where individuals have identifiers or names different to the concepts identifiers or explanations.

Therefore different matching techniques exist. Ontology matching can be divided e.g. into terminological, structural, instance-based or global techniques. [11] and [12] summarize these methods.

Terminological methods compare character strings with each other. One example for such a method is the Damerau-Levenshtein distance to calculate the edit distance between two character strings and find matching or equal strings. The Damerau-Levenshtein distance allows therefore ddition, deletion, substitution, and the transposition of two adjacent characters. To achieve best results, the strings to compare are normalized by means of different methods, e.g. deletion of excess blanks, punctuation marks, filler words and numbers, substitution of umlauts and special characters, and transformation of characters into lower-case.

Structural matching methods compare elements of an ontology and find equalities within the structure of an ontology. One example for such a method is to regard the ontology as graph and consider that similar elements have similar neighbors. This is called graph matching and does not take hierarchical relations into account.

Instance-based matching techniques find equal classes within the ontology. Classes are assumed to be equal if they posses many common instances.

Global matching techniques take into account the results of other matching techniques or user-defined matching results (user feedback) and combine them to reach an optimal matching result.

4.2. Mapping methods

As CAEX is very similar to ontologies, ontology mapping techniques can be used in the context of CAEX models. The data format CAEX was chosen as top-level format of AutomationML, therefore the mapping methods will be used for the CAEX part of AutomationML models.

A Mapping consists of a list of assignments between the elements of two structures. The mapping describes transformation rules to convert data from one structure into the other. This transformation is hard to find due to different identifiers, structures or even a similar wording, but different meaning of concepts. To find such a mapping between two given structures or schemas, there are different methods which can be distinguished by the type of the given data or the way to interpret the given data (see [13]).

String-based mapping techniques calculate the similarity between two given strings by means of different distance functions. They fail in case of different identifier for the same objects, but can be calculated very efficiently.

Language-based mapping techniques normalize the input to be able to compare it. Linguistic methods use thesauri to find matches. They are able to discover abbreviations, synonyms or translations.

Graph and taxonomy based algorithms try to discover similarities within the structure of objects. This makes it possible to map differently named objects.

Beside these examples for base mapping techniques, there are also hybrid forms of mapping techniques which combine base mapping techniques to an overall result (see [13], [14]).

5. Domain dependant matching of MES knowledge

The MES ontology simplifies the communication between MES significantly. To utilize the powerfulness of the MES ontology, two MES with proprietary interfaces can easily exchange data and information by means of the ontology as broker. The ontology consists of defined and described concepts whereas proprietary interfaces often lack descriptions of their elements.

For an exchange between two proprietary interfaces, each interface has to be matched one time to the MES ontology. Starting point of the matching between MES ontology and proprietary interface is a list of interface element which can be structured hierarchically. These elements consist of an identifier and optional additional descriptive information.

This means that the interface user creates a matching between the elements of his interface and the concepts of the ontology by assigning the type of a concept to the individuals of the interface. After this step an information exchange can be done with every other interface which has such a matching (see [8]).

The matching between the MES ontology and a proprietary interface description will normally be done manually by an expert of the proprietary interface who is aware of the MES ontology elements or who is assisted by a person with knowledge about the MES ontology.

This manual matching process can be assisted by software tools. One idea is to validate the manual matching and to improve it. For this validation, mistakes and failure of the user must be recognized and highlighted. An automatic matching can provide clues for necessary corrections.

Graphics and charts shall describe the quality of the processed matching task.

Due to the circumstance that the matching describes only whether or not an element of the interface was assigned to an ontology concept, this is not enough for a qualitative evaluation of the matching.

More information about the quality of one assignment between interface element and ontology concept or reasons for not assigned elements can only be achieved by a human expert.

[15] defines an expert system as program which almost reaches the specific problem solving strategies in a narrow application area. Normally, the evaluation of an expert system is done by the comparison with the manual work. For the given task, an expert system can be used to find good criteria for the evaluation of the matching. The manual work will be analyzed and evaluated by means of the automatic work of the expert system. The whole process is depicted in Figure 1. First, a preprocessing step must be done to normalize the ontology and the interface for a terminological matching. This results in working ontologies V and S and hast to be done one time for the ontology and one time for each proprietary interface.

The next step is to create the automatic matching. Therefore, the weighted Damerau-Levenshtein-Distance as terminological method and a hierarchy comparison as structural method are used. Pairs of elements which were found during the automatic matching process will be presented to the user as new recommendations. The user is able to give reasons why these recommendations do not match (see Figure 2), e.g.

- Coherence between ontology and interface elements with regard to content
- Unambiguousness of coherence between ontology and interface elements
- Complete coverage of ontology class by interface element
- Complete coverage of interface element by ontology class
- Generality of the interface element outside of company context
- Adequacy of ontology class in comparison to other ontology classes
- Level of detail of the ontology class description within the [4]

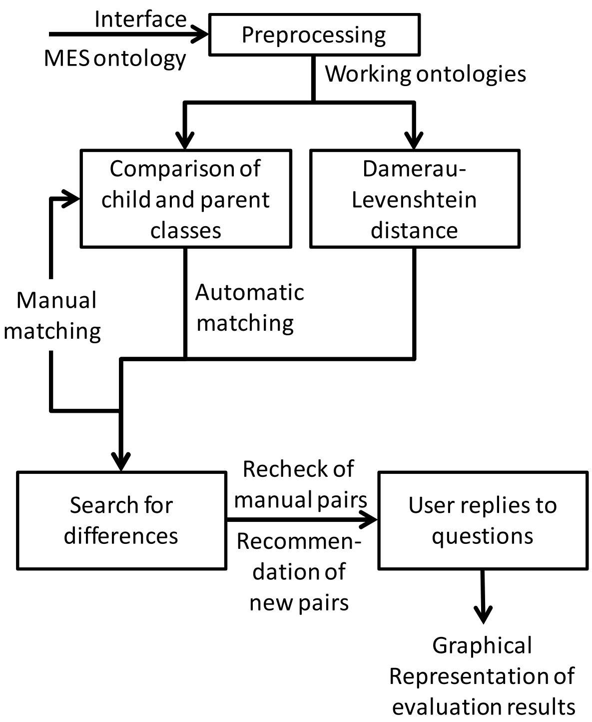

Figure 1. MES matching evaluation process

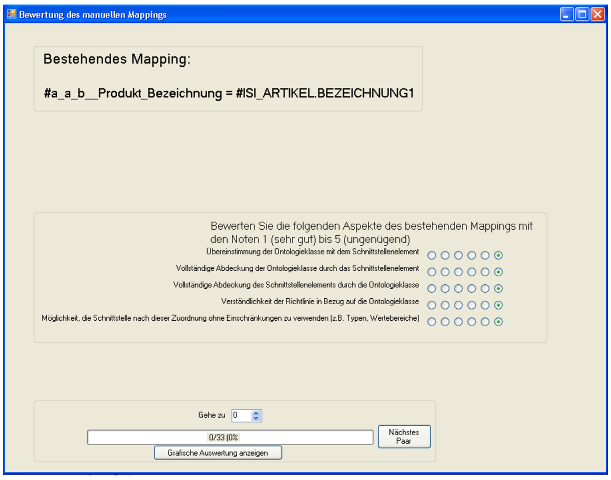

Figure 2. User input for rating

Pairs of elements which were assigned during the manual matching process will be presented to the user for a re-evaluation. The evaluation consist of different question concerning the matching criteria to be rated by the user, e.g.

- Complete coverage of ontology class by interface element
- Complete coverage of interface element by ontology class
- Comprehensibility of the ontology class description within the [4]
- Possibility to use interface element without restrictions (e.g. types, range of values) according to the assignment

Another detailed evaluation of the collected data will be created. Hence, the user (and his valuable expert knowledge) is included in the evaluation. The user answers to questions concerning the pairs of elements. The result will be prepared and depicted graphically (see Figure 3) and leads to newly created pairs of elements which will be adopted. The expert system based evaluation of the manual matching lead within the application examples to a amelioration of the created manual matching. Furthermore the recommendations for additional matching increased the number manual results by 20% on average.

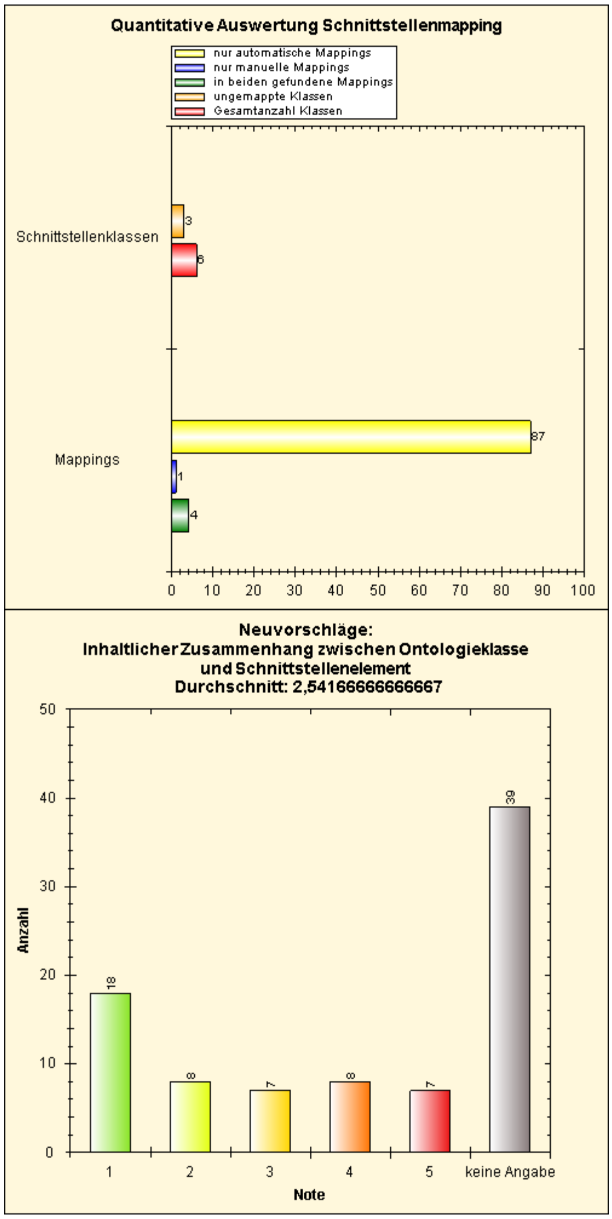

Figure 3. MES matching evaluation results

6. Domain independent mapping of AutomationML models

AutomationML as standardized modeling language can be used to full capacity if information from different sources is combined within the models. Hence, the knowledge of different domains is unified. To consolidate different AutomationML models describing one production plant, a mapping tool was developed which helps eliminating redundant information. This mapping can be done manually or by means of different comparison criteria. In the case of manual mapping, the user selects matching elements of source and target model. The mapping tool can check recursively if a mapping of sub nodes makes sense. As checking algorithm, a recursive function searching for name equality of the elements within the two models can be used (see [7]). This doesn't lead to satisfying results because the element names out of different domains are rarely similar to each other.

Therefore, the authors developed a combined mapping algorithm which works automatically. It combines different ontology mapping methods to find matching elements within two AutomationML models. This mapping algorithm focuses the top-level format of AutomationML models – CAEX – and searches for correspondences within the CAEX models. The combined mapping bases on the name equality mapping method in [7], but replaces the check of name equality by the combination of other mapping methods. The process of the combined mapping is depicted in Figure 4.

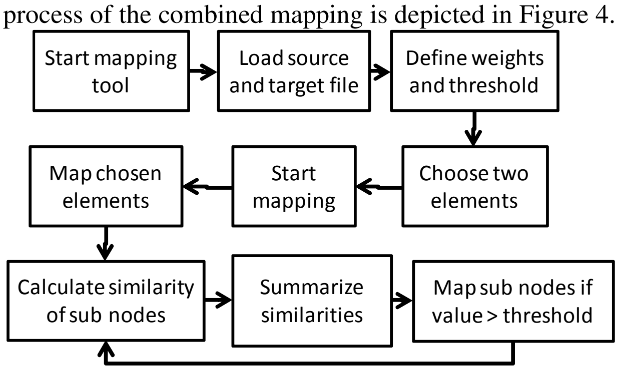

Figure 4. CAEX mapping process

For the mapping, string-based, graph-based and linguistic-based similarities are calculated.

String-based similarity (sim_{String}) is realized by the edit distance between two strings, the Damerau-Levenshtein distance. It describes the minimal number edit operations (add, delete, replace, swap) which are necessary to transform one string into another (see [16]).

The linguistic-based similarity ($sim_{Lexicon}$) is calculated by means of a domain specific dictionary which includes abbreviations, synonyms and translations. An example for such a dictionary is given in Figure 5 which includes German and English translations of domain-specific concepts. If two terms are included in the dictionary, they get the similarity value 1, otherwise 0.

turntable	Drehtisch
conveyor	Transportband
resource	Ressource
product	Produkt
control panel	Bedienpult

Figure 5. Example dictionary

To recognize structure-based similarity (sim_{Graph}), a graph algorithm is used. This is realized by means of the bottom-up maximum common sub tree isomorphism algorithm (see [17]). Therefore, source and target CAEX model have to be transformed into a tree structure, but this is very easy due to the fact that CAEX is an XML-based format which is structured hierarchically.

Further methods or algorithm can be included as well.

The three mapping algorithms are combined by means of weights (w_{String}, $w_{Lexicon}$, w_{Graph}) chosen by the user. The user can choose (via a dialog) the weights for each similarity type and a threshold for the combined mapping. The similarities are calculated and summarized with the user-given weights. The values are normalized to a value between 0 and 1. If the summation (sim_{Total}) exceeds the threshold, the two given elements of source and target model are mapped. The formula to calculate the total similarity value is given in Figure 6.

$$sim_{Total}(u,v) = \frac{w_{String} * sim_{String} + w_{Lexicon} * sim_{Lexicon} + w_{Graph} * sim_{Graph}}{w_{String} + w_{Lexicon} + w_{Graph}}$$

Figure 6. Total similarity calculation

Unmapped elements are added to the result model separately. Finally, the result is shown to the user. Figure 7 shows the user interface where on the left hand the source and on the right hand the target model is shown.

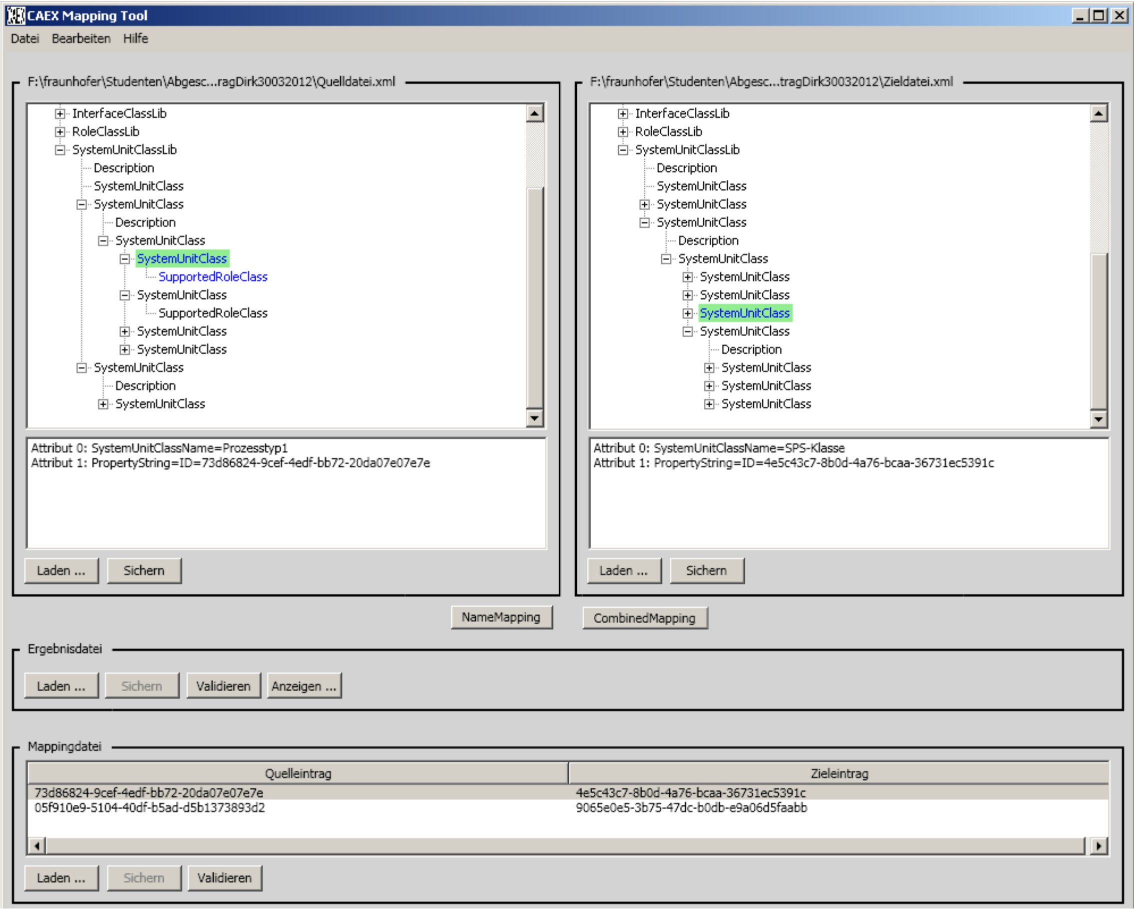

Figure 7. MES matching evaluation results

On the bottom of the user interface the mapping elements are shown as list. The user interface provides the possibility to review the found mappings and delete them.

Test with typos (see Figure 8), structural anomaly (see Figure 9) and synonyms (see Figure 10) were included into the evaluation data.

```
<SystemHierarchyElement SystemUnitInstanceName="Ressourcen">

<SystemHierarchyElement SystemUnitInstanceName="Resourrcen">


Ausgabe: damLev: 0,8 dictSim: 0 graphSim: 1
```

Figure 8. Test with typos

```
<SystemHierarchyElement SystemUnitInstanceName="Produkte">
  <SingletonClassDescription>
    <InternalElement localElementName="Karosse-xyz" description="Produkt, ...
      <PredefinedRealisation [...]
    </InternalElement>
    <InternalElement localElementName="Motorblock-xyz">
      <PredefinedRealisation [...]
    </InternalElement>
  </SingletonClassDescription>
</SystemHierarchyElement>

<SystemHierarchyElement SystemUnitInstanceName="Produkte">
  <SingletonClassDescription>
    <InternalElement localElementName="Karosse-xyz" description="Produkt, ...
      <PredefinedRealisation [...]
    </InternalElement>
    <InternalElement localElementName="Motorblock-xyz">
      <PredefinedRealisation [...]
    </InternalElement>
    <InternalElement localElementName="Fahrgestell-xyz">
      <PredefinedRealisation [...]
    </InternalElement>
  </SingletonClassDescription>
</SystemHierarchyElement>

Ausgabe: damLev: 1 dictSim: 1 graphSim: 0,631578947368421
```

Figure 9. Test with structural anomaly

```
<SystemHierarchyElement SystemUnitInstanceName="Produkte">

<SystemHierarchyElement SystemUnitInstanceName="Artikel">


Ausgabe: damLev: 0,142857142857143 dictSim: 1 graphSim: 1
```

Figure 10. Test with synonyms

Even with equal weights, one of the methods compensates the weakness of the others.

Furthermore, the combined mapping method was tested with different example models of production plants. The evaluation phase showed that a combined mapping method leads to better results than the implementation of a simple mapping method. But it cannot replace the manual and time-intensive mapping by an expert with knowledge within both domains. The implementation of an extended dictionary specific for the use case could improve the combined mapping.

7. Summary and Outlook

This contribution exemplifies the application of ontology matching and mapping methods in the context of data modeling for production planning.

A domain dependant matching of knowledge was explained by means of the MES ontology. A domain independent mapping of AutomationML models was done by means of a combined ontology mapping algorithm.

The domain dependant matching could be improved by the introduction of a thesaurus. Furthermore an amelioration of the manual matching could help. The combined mapping method for AutomationML (respectively CAEX) models can be improved by an automatic optimization of the user-defined weights. Furthermore the introduction of further mapping methods could help. In both cases, the assistance for the user by extended interaction can be a future topic.

Furthermore, the combination of the knowledge (domain dependant and domain independent) is possible by introducing the MES specific knowledge into AutomationML models. This integration can be realized by the creation of a MES specific CAEX-RoleClassLibrary including the ontology concepts. This makes the usage of MES terms within AutomationML models possible.

References

[1] VDI 5600-1, Manufacturing Execution Systems (MES), Beuth Verlag, 2007.

[2] DRAFT INTERNATIONAL STANDARD ISO/DIS 22400-2, ISO/TC 184/SC 5 Secretariat: ANSI, Automation systems and integration — Key performance indicators for manufacturing operations management — Part 2: Definitions and descriptions.

[3] M. Uschold, M. Grüninger, “Ontologies: Principles, Methods and Applications”, *Knowledge Engineering Review 11(2) (1996)*, pp. 93-155, 1996.

[4] VDI 5600-3, Manufacturing Execution Systems (MES) - Logic interfaces for machine and plant control, Beuth Verlag, 2011.

[5] AutomationML. http://www.automationml.org/, date of access: March, 2012.

[6] IEC 62424. Representation of process control engineering - Requests in P&I diagrams and data exchange between P&ID tools and PCE-CAE tools, 2008.

[7] M. Schleipen, M. Okon, The CAEX Tool Suite – user assistance for the use of standardized plant engineering data exchange, ÍEEE ETFA, 2010.

[8] Miriam Schleipen, Olaf Sauer, Lidmilla Fuskova, “Logical interface between MES and machine - semantic integration by means of ontologies”, *Proceedings of: CIRP ICME ’10 - 7th CIRP International Conference on INTELLIGENT COMPUTATION IN MANUFACTURING ENGINEERING, Innovative and Cognitive Production Technology and Systems*, 978-88-95028-65-1. 23 - 25 June 2010, Capri (Gulf of Naples), Italy, 2010.

[9] Miriam Schleipen, Rainer Drath, “Three-View-Concept for modeling process or manufacturing plants with AutomationML”, *IEEE International Conference on Emerging Technologies and Factory Automation*, 2009.

[10] Arndt Lüder, Elisabet Estévez, Lorenz Hundt, Marga Marcos, „Automatic transformation of logic models within engineering of embedded mechatronical units”, The International Journal of Advanced Manufacturing Technology Volume 54, Numbers 9-12 (2011), 1077-1089, DOI: 10.1007/s00170-010-3010-y, 2011.

[11] P. Shvaiko, J. Euzenat, “Ontology matching: state of the art and future challenges”, In: *IEEE Transactions on Knowledge and Data Engineering,* 2012.

[12] Z. Aleksovski, *Using background knowledge in ontology matching*, SIKS dissertation series, 2008-24, 2008.

[13] P. Shvaiko, J. Euzenat, “A survey of schema-based matching approaches”, http://disi.unitn.it/~p2p/RelatedWork/Matching/JoDS-IV-2005_SurveyMatching-SE.pdf, date of access: March 2012.

[14] H. Paulheim, *Skalierbarkeit von Ontology-Matching-Verfahren,* http://www.ke.tu-darmstadt.de/lehre/arbeiten/master/2008/Paulheim_Heiko.pdf. date of access: march 2012.

[15] K. Kurbel, *Entwicklung und Einsatz von Expertensystemen,* 2. verb.Aufl., Berlin: Springer Verlag, 1992.

[16] Vladimir I. Levenshtein, *Binary codes capable of correcting deletions, insertions, and reversals*, Soviet Physics Doklady, 1966.

[17] G. Valiente, “An efficient Bottom-Up Distance between Trees”, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.102.7958&rep=rep1&type=pdf, date of access: march 2012.