Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Semantic Digital Factory – Using Engineering Knowledge to Create Ontologies for Virtual Training

Dominic Gorecky*, Matthias Loskyll*,
Christian Stahl*

* *German Research Center for Artificial Intelligence, Kaiserslautern, Germany*
(e-mail: {dominic.gorecky, matthias.loskyll, christian.stahl}@dfki.de)

Abstract: The following paper investigates the design, implementation and practical usage of ontologies for virtual assembly training in the automotive industry. On the example of the VISTRA training system it is demonstrated, how ontologies can facilitate the (re-)usage of heterogeneous enterprise data for computer-based applications in the field of knowledge management and training. The VISTRA training system consists of several components and is embedded in existing enterprise data structures of the *Digital Factory*. Ontologies have proven to be appropriate for complex data integration scenario, since they facilitate merging of heterogeneous data by defining a common, comprehensive and computer-understandable information structure. In the VISTRA scenario the ontology comprises information related to products, assembly processes, team and plant structures as well as training plans and results. The VISTRA ontology represents the core element of a semantic knowledge platform (including semantic repositories and web services interfaces), which aims at establishing an interoperable, unambiguous and highly flexible data exchange within the IT-infrastructures of the *Digital Factory*.

1. INTRODUCTION

To ramp-up production of complex, mass customized products, such as automobiles, represents both a technological and organizational challenge. The *Digital Factory* framework holds promise to support and to accelerate all essential processes of the production life cycle by enabling “a holistic planning, evaluation and on-going improvement […] of the real factory” (VDI-Guideline VDI 4499). Under its umbrella various computer-aided tools for design and engineering of product and production structures (CAD, CAE) as well as for the planning of processes (CAPP, CAM) and layouts (CALP) are subsumed. Although product development and production planning are performed to a large extent within virtual environments today, the virtual methods and tools for the product ramp-up are still in their infancy. For a successful ramp-up of manual or semi-automated production processes, not only the view point of design and planning is required, but also the perspective of training and qualification. This especially applies to the automotive industry, in which operators must be acquainted perfectly with the products and the respective assembly processes before they are allowed to perform the assembly in the production line. Giving the fact that automobiles are produced in hundreds of different configurations, there is a pressing need to deploy CAX for training purposes, which are – until now – not explicitly addressed within the framework of the *Digital Factory*. In compliance with the goals of the *Digital Factory*, virtual training can be introduced as a tool to train complex, manual assembly processes at an early stage by interactive involvement of operators. It has shown a high potential to complement or even replace physical setups for the training of assembly processes in – and potentially beyond – the automotive industry (Gorecky *et al*., 2012). Although the idea of virtual training is one of the traditional ones in the field of virtual reality, real world experience shows that it has not found its way into the daily practice of the automotive industry. The main reason behind this reluctance lies in the extensive manual preparation effort (*authoring*), which current training solutions require to set up the virtual training scenarios. The precondition for the breakthrough of virtual training is training scenarios generated with little or no authoring effort at all. Although data from the product and manufacturing design (e.g. structural and geometrical data on products and production structures, process descriptions) is available and could be used for this purpose, a methodology for integration and re-use of existing data structures is missing. To integrate and re-use existing data structures, a semantic modelling approach based on ontologies was chosen due to its advantages over other information modelling approaches in terms of integration, verification and reuse of knowledge.

The next sections first give an overview about related work and the technological background. After that, the VISTRA system as a whole and especially its ontology architecture is presented, which could serve as a basis for similar applications in the production domain. Following this, some important advantages of using ontologies are discussed and the problem of integrating this technology into reliable industrial applications is illuminated.

2. RELATED WORK

Several commercial solutions exist, which are supposed to support a worker performing maintenance tasks by reviewing and exploring the product structure virtually, like “nGrain”, “DiSTI”, or Aerosim’s, Virtalis’, and Simdustrial’s virtual maintenance training systems. Vizendo's virtual training suite, as well as the work of (Brough *et al.*, 2007) offer virtual training in manufacturing processes; however, these solutions require a certain effort for training creation out of heterogeneous data sources. This could notably be reduced in the long run by an ontology-based data integration approach, like it is presented in this work. The VISTRA training system tries to train workers on their everyday tasks and thus concentrates on standard assembly sequences. The creation and management of these sequences, as well as the incorporation of product and plant structures as well as training plans and results into an overall homogeneous data structure, based on ontologies, are thereby the main differentiators in regard to other available systems.

As already stated in (Lin *et al.*, 2007), many different data formats and standards are used by different companies to describe the same data about products and their production processes. This emphasizes the need to create a unified data model for this domain before beginning to build applications on it, such as virtual assembly training (Stork *et al.*, 2012). Using ontologies for this purpose, rather than purely syntactic data formats like PLM XML (Ding *et al.*, 2007), has several advantages, which have not remained undetected in the research community: The application of technologies from the semantic web for the homogenization of various data sources has advantages in terms of integration, verification and reuse of knowledge (outlined in detail in the following sections). Most importantly, ontologies pave the way for knowledge sharing across company boundaries, which is a non-straightforward process at the moment (Young *et al.*, 2007). Additionally, it is easy to introduce a common vocabulary while at the same time maintaining company-specific terminology (Lin *et al.*, 2007), which improves the acceptance of the approach in industrial reality. Also, one research topic of the "VRCIM" laboratory is the application of ontologies in different scenarios within the product lifecycle management domain (Kim *et al.*, 2010) (Zhu *et al.*, 2012). In (Fiorentini *et al.*, 2007), an ontology for product lifecycle management has been created out of already available UML models. It seems to be quite detailed and also incorporates semantic reasoning; however, it focuses on the structure of already assembled products and not on the assembly process itself. VISTRA's process ontology instead defines additional concepts, specifying how the manufacturing process takes place. An interesting upper ontology for the manufacturing domain, called "MASON", is presented in (Lemaignan *et al.*, 2006). They define three head concepts: entities, operations and resources. In VISTRA, we first identified the demands on the process ontology with our industrial partners and then decided to introduce the operation as our central concept, around which further concepts are expanded. Also, MASON contains many details about its three head concepts, which would not have been of any use for VISTRA. Since we wanted to avoid computational overhead, we decided to build our own ontology from scratch, which now aligns with our scenario perfectly. Especially because of its modular structure, our ontology can even serve as an upper ontology for similar applications dealing with production processes.

3. VISTRA SYSTEM OVERVIEW

The VISTRA system architecture consists of three fundamental components (see Fig. 1).

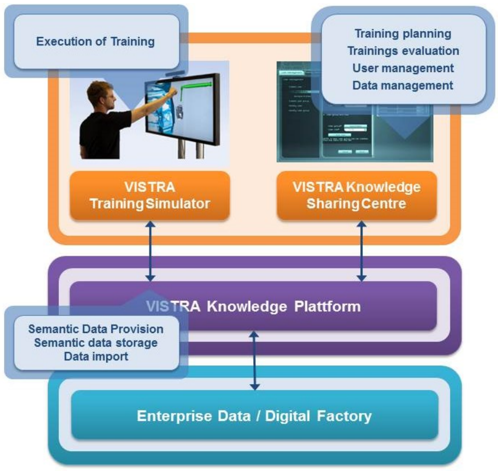

Fig. 1. Overview of the VISTRA system components.

The first and central component of the VISTRA system is the *VISTRA Knowledge Platform* (VKP). It is responsible for receiving enterprise data in various formats, and for mapping it to the *Semantic Core Model*, which represents a semantic-rich information model implemented as a collection of several loosely coupled ontologies, which define a data structure combining information about processes, products, plant structures, users, teams as well as results and feedback from the training. The second component of our system is called *VISTRA Training Simulator (VTS)*. This is an interactive virtual assembly simulation, where the virtual training is actually performed. Training scenarios are loaded dynamically from the VKP, considering available products, stations and assembly sequences, trainer guidelines and trainee profiles. Training results and feedback are written back to the VKP for further review and for improving training scenarios in future sessions. Planning training sessions and reviewing training results are important tasks of the third and last system component, the *VISTRA Knowledge Sharing Centre (VKSC)*. Furthermore, imported enterprise and user data can be checked, edited, and manually extended where necessary.

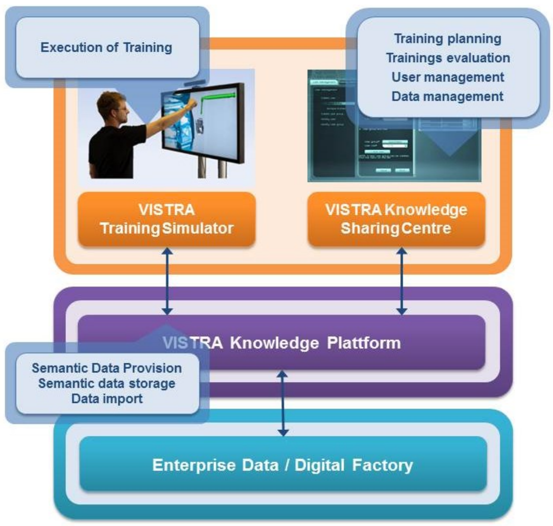

Fig. 2. Internal Architecture of the VKP.

All data is finally written back to the VKP. Inside the VKP, which acts as the central information hub in our system, all data is saved in triple stores which provide efficient access times and allow for semantic reasoning. Data delivery and reception is implemented mostly as REST web services (see Fig. 2). The data structure for process, plant, user, and statistics data is thereby defined in OWL ontologies, which enable a system-internal understanding of assembly processes. The architecture and the content of these ontologies are described in detail in the next section.

4. ONTOLOGY ARCHITECTURE

To build up a common data model enabling the homogeneous description of manufacturing process data, relational, object-oriented, as well as semantic modelling techniques seem to be suitable in general. The selection of ontologies as the preferred modelling approach for our system is justified by their distinct advantages over other approaches concerning especially integration, consistency checking, and reusability of knowledge.

To meet the demands of reusability, (Guarino, 1998) has introduced classification criteria to distinguish ontologies according to the area of validity. The highest level of abstraction and generality are given by top level ontologies, representing general concepts, such as time or space. Domain ontologies, however, describe a specific viewing area without being limited to specific applications or tasks. In contrast, one defines tasks ontologies to describe specific tasks or workflows in a generic, domain-independent manner. Application ontologies are relevant only to a specific application, which severely restricts the level of reusability. For virtual training, an application ontology is needed, which combines both domain-specific and task-specific concepts. However, the knowledge about the automotive manufacturing domain is not only relevant for the virtual training, but it is useful, for example, for knowledge management, documentation, or assistance tasks. A model of the "training" task can, in principle, be transferred to other domains such as medicine. Therefore a two-stage modelling approach is suggested, which differentiates between domain and task knowledge to satisfy the claim of maximum reusability: Initially the relevant manufacturing knowledge is conceptualized in a domain ontology (DO) and the training activity is described in a separate task ontology (TO). Both ontologies are thus independent and abstract from the concrete target application. In a second step, both ontologies are merged to an application ontology for virtual training in automotive manufacturing – as a specialization of the domain and task (see Fig. 3).

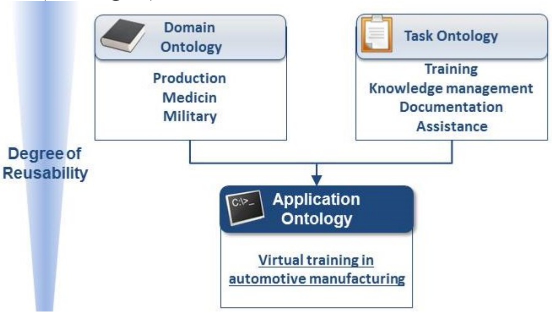

Fig. 3. Classification of ontologies regarding area of validity.

Both ontologies are again structured in modular models – so called sub-ontologies – as depicted in Fig. 4. This approach follows a semantics-driven modularization strategy, where each sub-ontology describes a logically independent aspect of the area of knowledge (Stuckenschmidt *et al.*; 2009, Loskyll *et al.* 2012). The main purpose of this modular structure is to simplify the reuse of single components in other projects and to make it easy to maintain the defined data structure. Four ontologies have been created, which together specify the data structure used throughout VISTRA in a machine-understandable way: e.g. *Assembly Process Model (DO)*, *Plant Model (DO)*, *User Model (TO)* and *Statistics Model (TO)*. The flexible aggregation of the individual models is thereby established by respectively importing other models and matching the corresponding concepts of the sub-ontologies. For example, as defined in the *Assembly Process Model*, each *Operation* needs to be performed by at least one *User*. However, the concept *User* is not defined in the *Assembly Process Model*, but in the *Users Model*, where it is also further described with appropriate properties.

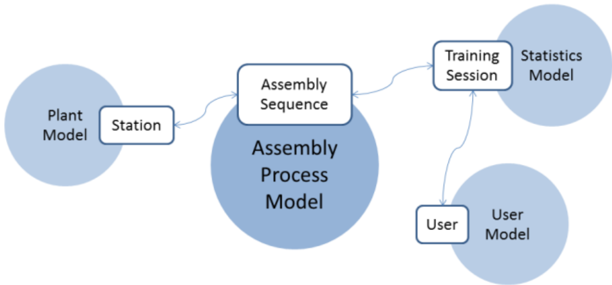

Fig. 4. High-level overview of the modular ontology structure inside the *Semantic Core Model* of VISTRA (exerpt).

As already mentioned, the central concept in the *Semantic Core Model* is the *Operation*, defined in the *Assembly Process Model*. An *Operation* is the smallest single element in an assembly sequence, and therefore should be looked at independently. Each *Operation* has several properties like *Tools*, *Parts*, *Activities,* and *Success Criteria*. Chains of *Operations* are called *Assembly Sequences*. Connections to the other ontologies are e.g. drawn by the fact that each *Operation* has to be performed by at least one *User* (defined in the *User Model*), and each *Assembly Sequence* is located at a specific *Station* inside some *Plant* (both defined in the *Plant Model*).

For illustration, Fig. 5 shows an exemplary operation “OP_123”, which is part of some assembly sequence called “AS_123A”, together with its most important properties and the respective relations connecting them. The figure also explains the simplified structure of the *Plant Model* ontology: Basically it describes how plants are organized in multiple stations. The *Assembly Process Model* then connects to the *Plant Model* by associating *Assembly Sequences* to the respective *Stations*. There is also the possibility to setup more fine-grained plant structures using the provided concepts *Plant*, *Shop*, *Line*, *Sub Line,* and *Station*.

The *User Model* defines the possible user properties, and, more importantly, his/her name, login information, role, and the specific user groups they belong to. The different roles, defined by a fixed set of instances in the *User Model* ontology, specify the privileges of a specific user. For example, only instances of the *Trainer* concept are allowed to change assembly sequences and to review training results in the VKSC, but not instances of the *Trainee* concept.

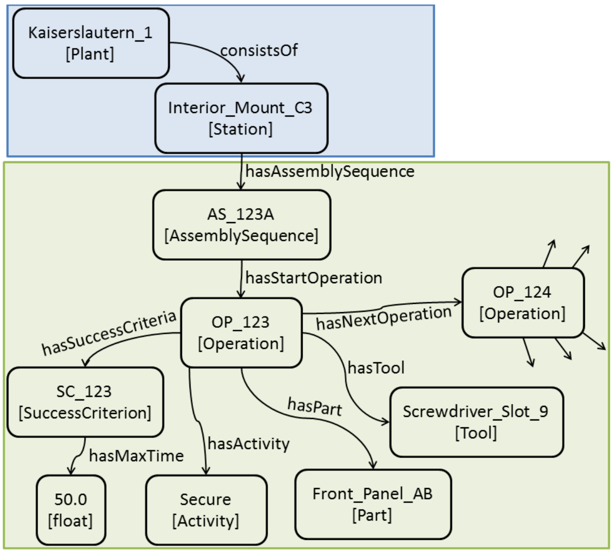

Fig. 5. Simplified view on an exemplary operation instance with its most important connections and properties. The upper part (blue) uses concepts defined in the *Plant Model*, whereas the lower part (green) uses concepts from the *Assembly Process Model*. The respective types of the instances are given in box brackets.

Another component of our modular *Semantic Core Model* is the so-called *Statistics Model*, which combines concepts defined in the *Assembly Process Model* and in the *User Model,* and introduces some additional concepts like *Time*, *Score,* and *Comment*. The main task of the *Statistics Model* is to save information about past training sessions to enable trainers to review and comment on training results. Another important aspect is the re-usage of knowledge offered by the *Statistics Model* for a better selection of future training scenarios, or to deduct implications and improvements for the design of assembly processes. Since all data structure definitions are realized with our machine-understandable ontologies, it is easy to define appropriate rules to automate even this process.

Once the described modular ontologies have been set up, many benefits can be derived, which would mostly be quite hard to realize if only purely syntactic data descriptions had been used. These benefits can be grouped into two main areas, namely usability and intelligence. Since the manual creation of valid OWL ontologies is very hard due to the cryptic and error-prone RDF syntax (Knublauch *et al.*, 2004), the expectations on the improved usability could be lowered at first sight. However, thanks to graphical ontology editors like e.g. Protégé, which supports the user with a simple syntax and built-in validation mechanisms (Knublauch *et al.*, 2004), the generation of classes, instances, and their relations is quite comfortable and intuitive. The model creation and editing process is further simplified by the modular ontology structure and its loose internal coupling, which allows editing specific knowledge areas independent from each other. The improved usability is primarily revealed when integrating additional concepts or maintaining the data structure at a later point in time. Additional benefits lie in the resulting system intelligence. By adding many quite simple rules, which are mostly self-evident for a human being, the system becomes more and more intelligent, meaning that it is aware of relationships and rules defining how the single concepts work together. Using a semantic reasoner even new knowledge can be inferred.

One example for the possible applications of a semantic reasoner are the relations *hasNextOperation* and *hasSubsequentOperation*, which are defined in the *Assembly Process Model*. By connecting them as illustrated in Fig. 6, it is very easy to draw conclusions about what tools are needed in a certain assembly sequence. Therefore, the appropriate question to ask (e.g. using SPARQL) would be: "Give me all tools belonging to all operations which are subsequent to the start operation of assembly sequence AS_123A." Using OWL reasoning, this will return the correct answer, even if the relation *hasSubsequentOperation* has never been used explicitly to connect operations to all of their subsequent ones during data import. The concepts *TransitiveProperty*, *FunctionalProperty,* and *isSubPropertyOf* are thereby defined in the OWL language to enable this kind of semantic reasoning. Of course, the investigation of semantic constructs like this takes some time at the beginning; however, they ease the modelling process in the long term: In the example above, only directly neighboured operations need to be manually connected by the user (using *hasNextOperation*) – the system still knows how to interpret these single connections when being asked for "subsequent" operations. The retrieved information can then e.g. be used by the VTS at the beginning of training sessions to load all needed tool geometry for the selected assembly sequence in advance.

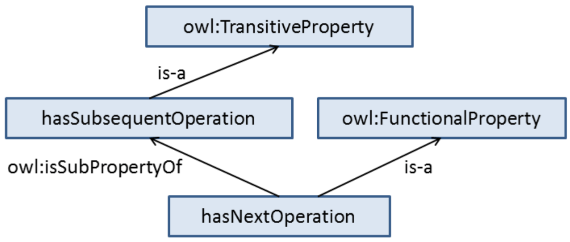

Fig. 6. Due to the suitable definition of the shown properties, every *Operation* may only have exactly one "next operation", whereas all of its "subsequent operations" in the actual assembly sequence are assigned to it automatically.

Another advantage of the definition of rules in our ontologies is the automatic detection of inconsistent data. For example, it can be checked that an operation with the activity *Screw* also needs to have a suitable *Tool*, which must be some instance of the *Screwdriver* concept. If this is not the case for some screwing operation, a semantic reasoner is able to automatically warn the user about data inconsistencies.

The use cases of semantic reasoning in our system described above are only two examples of many. Thus, the main advantage of using ontologies throughout the system is probably the ability to extend the models whenever needed and to add more and more rules to assure that the system's overall intelligence gradually increases.

5. DATA INTEGRATION USING ONTOLOGIES

Ontologies are a well-suited technology for integrating information coming from several heterogeneous data sources.

In VISTRA, we have to deal with this problem at the point where the original data from our different industrial partners enters the VKP. Instead of setting up the same rules and data structures for every distinct set of terms, we transform each data input into a consistent representation using concepts defined in our ontologies. Of course, this requires an implementation of several mappings from the respective data format and terminology to the structure and terminology of our system. However, this effort pays off since only one set of semantic rules is needed in the end, regardless of the actual number of heterogeneous data input sources. This way, a consistent and easy-to-upgrade data structure can be created, which will thus remain valid over a very long period of time, thereby effectively reducing future data integration efforts.

Nonetheless, being independent from specific terms, which can actually describe the same object in reality, is a core idea of ontologies. Due to this, OWL already contains certain concepts like “owl:equivalentClass”, which can help to integrate diverse information conceptually into a “neutral” model, like our modular ontology structure. This model then provides the ability to have a consistent overview of the data and to possibly find interesting relations, even across the eliminated borders of the previously different data structures and terms (Stuckenschmidt *et al.* 2009).

6. APPLICATION-INDEPENDENT CONSISTENT DATA

As already mentioned above, describing data structures semantically with the help of ontologies, as realized in the VISTRA project, provides the possibility of an automatic data consistency check. This is very useful, especially in industrial applications, and a clear advantage over common systems dealing with very many different data formats and extensions, which usually make it quite hard to keep track of overall data consistency.

Since the data structures are clearly defined at one point in the whole system (here the *VISTRA Knowledge Platform*), other system components which are built upon (here the *VISTRA Training Simulator* and the *VISTRA Knowledge Sharing Centre*) have a consistent, unambiguous overview of the data and its structure. This makes it easy to maintain and to alter the structure later on, since in most cases the changes only have to be made at one specific point in the system. Also, if some changes should become necessary at other parts in the system, the usage of ontologies is once again advantageous: Since they are not only machine-understandable but also easily readable by humans, they also support the software developers in having a consistent overview of the data.

A good example for a system component that makes use of this central semantic data description is the data exchange between different system components via semantically enriched RESTful web services, which are further described in the next section.

7. EMBEDDING ONTOLOGIES IN INDUSTRIAL APPLICATIONS

In our system, it is a crucial point that each and every component always uses the definitions made by the ontologies of the VKP, instead of building up a duplicated data structure definition in a syntactically different form. Only if this is the case, the central semantic data structure definition can play off its advantages and different components may reliably work together.

One example for this is the data exchange mechanism between different system components in VISTRA. Almost all communication there happens via RESTful web services, which exchange XML documents. The structure of the data delivered by these web services is strictly oriented on the definitions of the ontologies; suitable XML schema files are even automatically created from the respective OWL files. Due to the resulting semantic annotation of the transmitted data with concepts defined in the ontologies, a clear, unambiguous data transfer is possible.

Another important aspect for the industrial application of semantic technologies besides reliability is speed. Early query and reasoning systems for OWL ontologies have been quite slow, especially with an increasing number of statements and rules. However, in the meantime, solutions exist which reach a reasonable access time on knowledge bases containing millions of statements, even for queries which incorporate semantic reasoning. In VISTRA, we make use of the semantic triple store *OWLIM*, which stores all statements in a big database and is able to infer new knowledge based on OWL semantics during data retrieval.

To also account for the usability of our system, we designed all OWL ontologies using the graphical ontology editor *Protégé*. Also ontology maintenance and possible extensions are managed in *Protégé*. Once a final OWL file has been produced, it is loaded into the semantic triple store (realized with *OWLIM*) to build the basis for data input, which is realized via RESTful web services. Also the data output is implemented with RESTful web services and uses internal SPARQL queries to retrieve data from the triple store. Since the OWL models with their semantic rules and the corresponding data are located in the same semantic-aware *OWLIM* triple store, consistency checks and the inference of implicit knowledge thereby become an automatic process, invoked whenever the triple store is accessed.

8. IMPLEMENTATION AND EVALUATION OF THE SYSTEM PROTOTYPE

The complete VISTRA training system has been developed and implemented as a prototype (see Fig. 7) at two automotive end-user sites, and has been tested in extensive user studies with more than 50 people. The technical evaluation has proven that the storage and provision of semantic data using triple stores and web services can efficiently be realized in an industrial environment. The ingestion of the semantic data structure can be realized in an automatic or manual manner. If the input and target data structure is understood – and sufficiently formalised – it is even possible to create an automatic import mechanism, which automatically transforms heterogeneous data structures (e.g. PLM XML) to OWL format (Stork *et al.*, 2012). In this way, it became possible to reuse existing data from the Digital Factory and to drastically reduce the effort and time for setting-up virtual training scenarios. The results from the evaluation are very promising and will help to support further system development. The *VISTRA Knowledge Platform* as a semantic data interface shows a high potential for use in new kinds of computer-based applications (e.g. to automatically create interactive virtual work instructions).

Fig. 7. Screenshot of the VISTRA virtual assembly trainer.

10. CONCLUSIONS

In the paper, we showed the design, implementation and practical usage of ontologies in the industrial context. The VISTRA training system served as a vivid example of how ontologies can facilitate the (re-)usage of enterprise data for various computer-based applications, e.g. in the field of knowledge management and training. Furthermore we demonstrated how industrial applications can benefit from semantic technologies by:

- facilitating the integration of heterogeneous data
- automatically ensuring data consistency
- enabling query and reasoning capabilities

For the design of the ontologies a modular modelling approach was taken, which consists of two main ideas: (1) to separate between domain and task knowledge and (2) to split the ontologies in logically independent modules, which then can be flexibly aggregated and (re-)used for various applications.

Although the potential of ontologies for the industrial usage has been proven, it must be stated that the awareness of the usage and benefits of semantic technologies is still widely missing in industries, and must further be propagated with best practise examples, such as the VISTRA project.

In the future, we will investigate a more comprehensive, rule-based mapping approach, which allows the automatic transformation of different, heterogeneous enterprise data into the OWL format and to ingest it in the triple store. In this way the adaptation of semantic technologies in industry should be further facilitated.

ACKNOWLEDGEMENT

This research was conducted in the context of the VISTRA project, which is co-funded by the 7th Framework Programme of the European Union under theme “ICT-2011.7.4 Digital Factories: Manufacturing Design and Product Lifecycle Management”, project number ICT-285176. More information can be found on the project website http://www.vistra-project.eu.

REFERENCES

Brough, J.E., M. Schwartz, S.K. Gupta, D.K. Anand, R. Kavetsky and R. Pettersen (2007). Towards the development of a virtual environment-based training system for mechanical assembly operations. In: *Virtual Reality*, 189-206. Springer, London.

Ding, L., A. Ball, J. Matthews, C.A. McMahon and M. Patel (2007). Product representation in lightweight formats for product lifecycle management (PLM). In: *4th International Conference on Digital Enterprise Technology*. University of Bath.

Fiorentini, X., I. Gambino, V.-C. Liang, S. Rachuri, M. Mani and C. Bock (2007). An Ontology for Assembly Representation.

Gorecky, D., G. Lawson, K. Mura, S. Hermawati and M. L. Overby, (2012). User-centered Design of a Game-based, Virtual Training System. In: *Conference Proceedings. International Conference on Applied Human Factors and Ergonomics* (AHFE-2012), July 21-25, San Francisco, CA, United States, AHFE.

Guarino, N. (1998) Formal ontology in information systems. In: *Proceedings of the first international conference (FOIS’98)*. June 6-8, Trento, Italy, Bd. 46. Ios Press Inc,

Kim, O., U. Jayaram, S. Jayaram and L. Zhu (2010). iTrain: Ontology-Based Integration of Information and Methods in Computer-Based Training (CBT) and Immersive Training (IMT) for Assembly Simulations. ASME.

Knublauch, H., R. W. Fergerson, N. F. Noy and M. A. Musen (2004). The Protégé OWL plugin: An open development environment for semantic web applications. In: *The Semantic Web–ISWC 2004*, 229-243. Springer Berlin Heidelberg.

Lemaignan, S., A. Siadat, J.-Y. Dantan and A. Semenenko (2006). A proposal for an ontology of manufacturing domain. In: *Distributed Intelligent Systems: Collective Intelligence and Its Applications*, 195-200. IEEE.

Lin, H. K. and Jenny A. Harding (2007). A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration. In: *Computers in Industry 58.5,* 428-437.

Loskyll, M., I. Heck, J. Schlick, M. Schwarz (2012). Context-Based Orchestration for Control of Resource-Efficient Manufacturing Processes. Future Internet, 4(3), 737-761.

Stuckenschmidt, H., C. Parent, S. Spaccapietra (2009). Modular Ontologies: Concepts, Theories and Techniques for Knowledge Modularization. Springer.

Stork, A., D. Gorecky, N. Sevilmis, D. Weber, C. Stahl, M. Loskyll and F. Michel (2012): Enabling virtual assembly training in and beyond the automotive industry. In: *18th International Conference on Virtual Systems and Multimedia*, 347-352. IEEE.

VDI-Guideline VDI 4499 Part 1 02.2008: Digital factory: Fundaments

Young, R. I. M., A. G. Gunendran, A. F. Cutting-Decelle and M. Gruninger (2007). Manufacturing knowledge sharing in PLM: a progression towards the use of heavy weight ontologies. In: *International Journal of Production Research, 45(7)*, 1505-1519.

Zhu, L., U. Jayaram and O. Kim (2012). Semantic Applications Enabling Reasoning in Product Assembly Ontologies – Moving Past Modeling. In: *Journal of computing and information science in engineering, 12.1*.