AN ONTOLOGY FOR LEARNING SERVICES ON THE SHOP FLOOR

Carsten Ullrich
Educational Technology Lab, DFKI GmbH
DFKI Projektbüro Berlin, Alt-Moabit 91c, 10559 Berlin, Germany

ABSTRACT

An ontology expresses a common understanding of a domain that serves as a basis of communication between people or systems, and enables knowledge sharing, reuse of domain knowledge, reasoning and thus problem solving. In Technology-Enhanced Learning, especially in Intelligent Tutoring Systems and Adaptive Learning Environments, ontologies serve as the basis of adaptivity and personalization. For mathematics learning and similarly structured domains, ontologies and their usage for adaptive learning are well understood and established. This contribution presents an ontology for the industrial shop floor (the area of a factory where operatives assemble products) and illustrates its usage in several learning services.

KEYWORDS

adaptivity, domain model, workplace learning

1. INTRODUCTION

An ontology expresses a common understanding of a domain that serves as a basis of communication between people or systems. It enables knowledge sharing, reuse of domain knowledge, reasoning and thus problem solving (Chandrasekaran, 1999). In Technology-Enhanced Learning, especially in Intelligent Tutoring Systems and Adaptive Learning Environments, ontologies serve as the basis of the domain model, which is the basis for adaptivity and personalization. In highly structured domains such as mathematics, physics and computer science, ontologies with the aim of supporting learning are well understood and several examples exist. This has enabled a large body of research. For instance, ActiveMath (Melis, et al., 2001) is a web-based learning environment for Mathematics that creates on the learners' demand courseware adapted with respect to their knowledge state and learning goals. Similarly, the physic tutor Andes generates problems specifically targeted to the individual learner in order to achieve the best possible learning gain (VanLehn, et al., 2005). Without a domain ontology, such advanced adaptive features would not be possible.

In the domain of the shop floor (the area of a factory where operatives assemble products) no such ontology exists, despite the high relevance of technology-supported learning for industry. Today's workplace on the shop floor is highly demanding (Mavrikios, Papakostas, Mourtzis, & Chryssolouris, 2013). The employee is under constant pressure to solve problems occurring on the shop floor as fast as possible, and simultaneously to improve his work-related knowledge, skills, and capabilities. Yet, research on the potential of adaptive learning on the shop floor is rare. Given the progress in other domains, we suspect that one reason is that the groundwork, in the form of descriptions of the central concepts of the domain to be learned, and how they are interrelated, is still missing. This paper addresses this short-coming by describing an ontology that represents entities on the shop floor relevant from a pedagogical perspective. The following section describes related work in this area, followed by an account of the central concepts and their relations (Section 3) and an illustration of its benefits by examples of adaptive services that make use of it (Section 4). The conclusion (Section 5) discusses the problem of scalable application of the proposed ontology and suggest a possible solution.

2. RELATED WORK

The relevance of educational technology in supporting workers in manufacturing (Mavrikios, Papakostas, Mourtzis, & Chryssolouris, 2013) as well the potential of smart and adaptive environment for workplace-based learning (Koper, 2014) have been clearly recognized. However, most existing work has focused on very specific areas, such as assembly, in order to increase process quality (Stoessel, Wiesbeck, Stork, Zaeh, & Schuboe, 2008) (Stork, Stößel, & Schubö, 2008), collaboration between machine and operator (Sebanz, Bekkering, & Knoblich, 2006) (Lenz, et al., 2008), control (Bannat, et al., 2009) or monitoring (Stork genannt Wersborg, Borgwardt, & Diepold, 2009). Recent work investigated how to use data from factory-wide sensor networks to control information flow so that cognitive overload of employees can be avoided (Lindblom & Thorvald, 2014) or how to display the data in a way that workers' satisfaction is increased (Arena & Perdikakis, 2015).

Research in the Semantic Web has led to formalizations that enable distributed ontologies, easily linked and reused (Allemang & Hendler, 2008). In the area of manufacturing, prior work has investigated informal classifications of universal, shop-level, and machine-level knowledge (Shah & Mäntylä, 1995) and ontologies for very restricted areas such as toolpath planning (Xu, Wang, & Rong, 2006) or fixture design (Ameri & Summers, 2008). The SemProM project has taken wider approach, representing all information collected during the lifespan of a product (Wahlster, 2013). In contrast to the limited work on ontologies stands a plethora of standards and specifications. The technical committee ISO/TC 184 for instance, has published more than 800 ISO standards that describe different aspects of automation systems and integration. These, however, have a very technical focus, and none we have seen is on the required level of abstraction for learning on the shop floor. What is required is a focused domain description, which focuses on the learning needs of the operators. The potential of such descriptions has been shown for generating assembling instructions automatically from product lifecycle data (Stoessel, Wiesbeck, Stork, Zaeh, & Schuboe, 2008), for supporting the transfer of practical knowledge (Blümling & Reithinger, 2015), as well as for providing manufacturing assembly assistance (Alm, Aehnelt, & Urban, 2015). However, none of this work has described the employed ontologies in sufficient detail to judge their general applicability nor to enable reuse.

3. DESCRIPTION OF THE ONTOLOGY

The ontology serves as a blueprint to describe the general concepts (also called types, in the following highlighted in bold font) relevant for teaching and learning on the shop floor. Figure 1 gives an overview of the type hierarchy of the ontology.

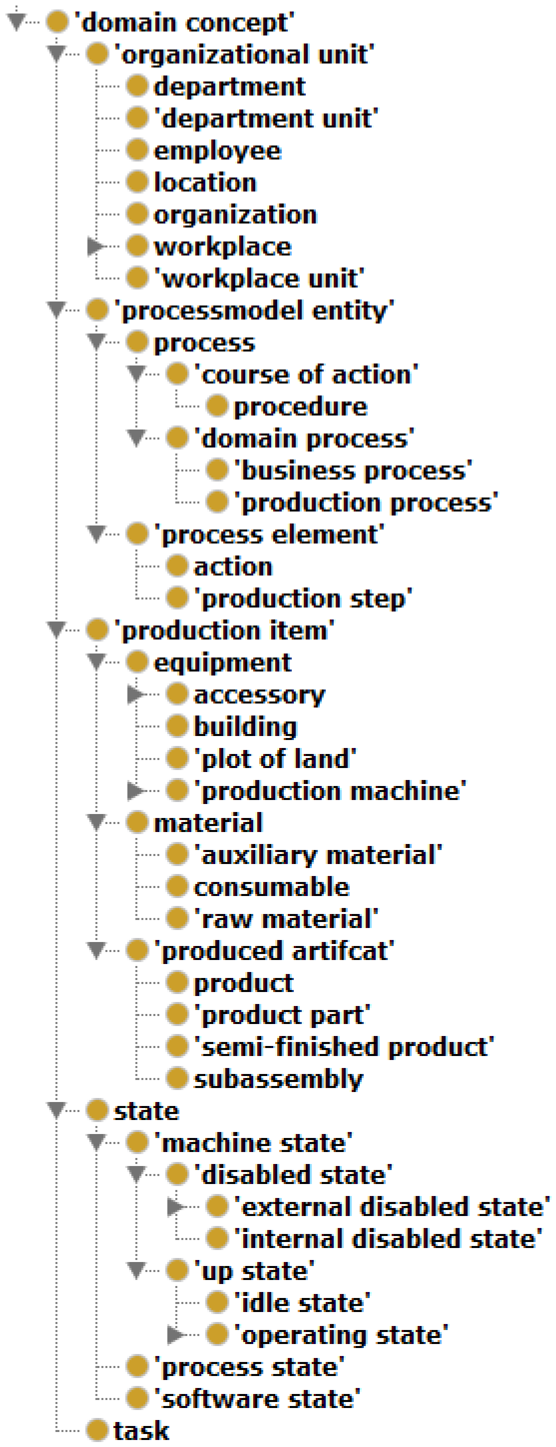
The highest level of the ontology distinguishes between the types **state**, **organizational unit**, **production item**, **function**, and **process model entity**.

The **state** describes the state of a domain entity. It distinguishes between **machine state**, which describes the state of a **production machine** (defined below), and **software state**, which describes the state of a software. Both types of states have a priority, indicating their severity (10, highly critical, to 1, not critical) and can be further divided into specific sub-states (adapted from (Kasikci, 2010)):

- **Up state** is the state of a machine characterized by the fact that it can perform a required function, with the specializations **operating state**, which describes when a machine is performing as required, and **idle state**, which describes a machine which is in an up-state and non-operating, during non-required time.
- **Disabled state** is the state of a machine characterized by its inability to perform a required function, for any reason. It has the specializations **external disabled state**, for describing a machine in an up-state, but lacking required external resources or is disabled due to planned actions other than maintenance, and **internal disabled state**, the state of a machine characterized by an inability to perform a required function.

The type **organizational unit** and its specializations allows to describe the structure of a company, from the top-most unit **organization**, to **location**, **department**, **department unit**, **workplace unit**, **workplace**, down to **employee**.

Figure 1. Overview of the type hierarchy of the ontology



The type **production item** describes entities used for or created during the production of goods. Its specializations are:

- **Equipment**: Equipment includes the entire technical apparatus used during production as well as property (plot of land), buildings, etc. In contrast to materials, equipment is not consumed. It has the specializations **building**, **plot of land**, **accessory** (a mechanical device that supports the production process, but does not perform the production itself.), and **production machine** (a stationary device directly involved in the production process.).

- **Produced artifact**: The marketed object that is the result of the production process, with the specializations **subassembly**, **semi-finished product**, **product**, and **part.**
- **Material**: Materials are physical entities consumed in the production of the finished product, with the specializations **consumable**, **auxiliary material**, and **raw material**.

The type **processmodel entity** represents types required for describing processes that occur in the target domain. Its subtypes are

- **Process**: A sequence of states, which is triggered by an event and leads to a final state. The ontology distinguishes between **domain process** (overarching process performed for the realization of the product, such as the **production process** and **business process**) and **course of action** (a process that is performed by a human, such as a work **procedure**).
- **Process element:** The states of the process, such as an **action** (for **course of action)** or a **production step** (for the production process).

The final type is **task**, which describes the task of human operator such as maintenance, repair, and operations or management.

Relations (also called properties) specify how the instances interrelate. Some of the properties only apply to specific types (written in bold font below), while others apply to several of the types (non-bold font). The following list describes the intended meaning of the relation (all nouns refer to types of the ontology):

- **requires: a machine state requires a procedure.**
- **produces: a production step produces a produced artifact**
- **executes: a production machine executes a production step**
- **has part:** expresses that some entity is part of another entity. This relation can be applied to many of the types.
- **has task: an employee has tasks**
- **has step: a process 'has the step' process element**
- **has state: a production machine has a machine state**
- **has procedure: a task has a procedure** (meaning that a task can be achieved by following a series of actions).
- **is part of**: the inverse relation of **has part**
- **realizes**: **a production machine realizes a production process.**

4. USAGE OF THE ONTOLOGY

This section illustrates how the ontology is used by several software services to support workplace-integrated learning. The services have been implemented in the APPsist architecture, which consists of mobile, context-sensitive and intelligent-adaptive assistance systems for knowledge and action support on the shop floor (Ullrich, et al., 2015). APPsist provides assistance to the human operator in solving problem by guidance through step-by-step instructions and provision of relevant documents. It also suggests content (learning materials, user manuals, other documents) relevant for medium- and long-term development goals to the operator.

4.1. Domain Modeling

The foremost usage of the ontology is to build up a domain model, i.e., a formal representation of the shop floor at a specific site of a company. This requires to specify instances (in the following highlighted in bold font) of the types. For example, if we were to say that Lenovo in its production department P1 in Guandong produces the laptop X, then we had to create the instance **Lenovo** of the concept **organization**, the instance **Guandong** of **location**, the instance **P1** of **department**, and the instance **X** of **product**. Using the property **produces**, one can specify that **X** is produced in **P1**.

In APPsist, the ontology was used to define the domain models representing three different companies, ranging from small- over medium- to large-sized: The small-sized company produces complex customer-specific tools and devices for car manufacturers and their suppliers. The APPsist pilot learning scenario focuses on installation and use of devices (milling machines). The medium-sized company produces customer-specific welding and assembly lines for car manufacturers. The pilot scenario focuses on error diagnosis and correction in the customer-specific machines. The large-sized company produces pneumatic and electric controllers for the automation of assembly-lines, which are used in customer-specific products as well as in their own production. The pilot scenario focuses on maintenance and repair, in particular outages (replacement of adhesives).

If a domain model is stored in a database, then it can be queried to retrieve information. In APPsist, the ontology and its instantiation are stored in a semantic database that allows queries involving reasoning. For instance, by following the appropriate relations, a query might return all produced products in a given location, even if that information is not specified directly. This functionality is a prerequisite of several of the following services.

4.2. Content Metadata

The types and instantiations of the ontology can be used to describe available content, such as learning materials, documentation (user manuals, programming handbooks), and order-related information (parts lists, wiring diagrams). Specifically, it allows to specify the production items a piece of content refers to, the target-audience, etc. Other services, as described below, will use this information to perform information retrieval tasks in different contexts.

4.3. Machine Information

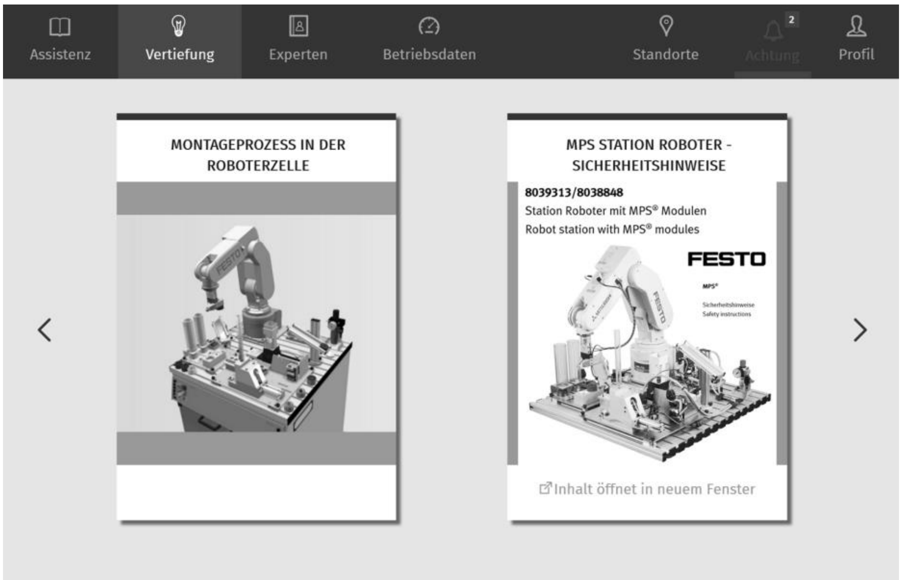
The machines on the shop floor use a multitude of sensors for managing the production process. Some of the sensor values are relevant for triggering actions of the human operators, especially in case of machine states that result in stopping the production process. However, the low-level senor data is too specific to be interpreted by support software. In APPsist, a machine information service translates the sensor data into instances of the type **machine state** and propagates the instances to the supporting services. This way, the learning supporting services can be set up to react to high-level, abstract machine states and thus can be more easily reused with new machines.

4.4. Content and Procedure Selection

If available content and the situation of the shop floor are described using the terms specified in the ontology, then that information can be used to select content suited for both assistance and knowledge building processes. For instance, the content selector service in APPsist uses a set of rules that are applied when a production machine gives an error state. The rules inspect the **organizational units** to determine which **employees** are assigned to the **production machine**, and uses information specified about the relevant **production items** to find content relevant for the **employee** in the specific situation. Analogous to the content selection, the procedure selection service recommends work **procedures** to be performed by the **employee** based on the current situation on the shop floor.

The rules used by the two services are abstract in the sense that they perform their reasoning on the ontology and do not encode information about the specific company in which they are used (the instances of the domain). Thus, they are transferable between companies. They thereby implement knowledge analogous to what a trainer or instructor possesses: given a specific set of circumstances, a trainer knows how to help the learner. Figure 2 contains a screenshot in which the APPsist system shows a selection of content relevant to the current context and user. The top row contains the main menu showing the available tabs, with the currently opened tab ("Vertiefung" meaning "consolidating content") being highlighted. The main screen below shows two work content items the system determined to be relevant to the employee in the current situation ("Montageprozess in der Roboterzelle" meaning "assembling process of the roboter station" and "MPS Station Roboter - Sicherheitshinweise" meaning "MPS station robot – security notes"). If the employees selects one of the work procedures, she will see the corresponding item. The other relevant tab is the first from the right ("Assistance" meaning "assistance"), which provides access to work procedures relevant to the current situation.

Figure 2: Screenshot showing recommended content



5. CONCLUSION

The contribution of this paper is an ontology that captures the entities and their interrelationships relevant for describing and reasoning about the shop floor from an educational perspective. The ontology is of general value as illustrated by its usage a) to represent the shop floor of three different companies and b) as a basis for a number of learning services, i.e., functionally highly specialized software applications that aim at supporting problem solving and learning of human operators. By referring to the terms defined in the ontology, these services become reusable, i.e., applicable in other companies than the three they were originally developed for.

It is a lightweight ontology, meaning that it does not use features such as axioms that would allow even more advanced reasoning but which come with the cost of higher complexity and difficulty of understanding.

The ontology was designed to focus on the production items, states, and processes. Still unclear is a generally applicable structure of the **tasks** of human operators for the shop floor. Currently, each of the industry partners has its own set of tasks, e.g., **installation**, **commissioning**, **maintenance**, **repair**, **operations**, or **assembly**. Due to the very different, company-specific views on the relevant tasks, we were not yet able to devise a suitable abstract representation.

Highly relevant is the problem of scalability. Architectures supporting problem solving and knowledge acquisition will only find widespread application if the cost of applying them to a new setting is reasonably low. Currently, integrating new content, processes and production machines into APPsist, requires manual input of the metadata, machine instances, etc., resulting in significant costs. Here, methods of information extraction that analyze existing documents might allow automating the ontology creation, instance creation and metadata annotation, and thus enabling low-cost, scalable support of human operators.

ACKNOWLEDGEMENT

This publication is a result of work performed in the context of the project “APPsist – Intelligente Wissensdienste für die Smart Production”, funded under the initiative “Autonomics for INDUSTRY 4.0” at the German Federal Ministry for Economic Affairs and Energy (BMWi), number 01MA13004C and supervised by the DLR. The authors are solely responsible for its content.

REFERENCES

Allemang, D., & Hendler, J. (2008). *Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL.* San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Alm, R., Aehnelt, M., & Urban, B. (2015). Processing Manufacturing Knowledge with Ontology-based Annotations and Cognitive Architectures. *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business* (pp. 25:1-25:6). New York, NY, USA: ACM.

Ameri, F., & Summers, J. D. (2008). An Ontology for Representation of Fixture Design Knowledge. *Computer-Aided Design and Applications, 5*(5), 601-611.

Arena, D. N., & Perdikakis, A. (2015). *DELIVERABLE D2.2 Knowledge model for human resource optimization.* Tech. rep., SatisFactory project.

Bannat, A., Gast, J., Rehrl, T., Rösel, W., Rigoll, G., & Wallhoff, F. (2009). A Multimodal Human-Robot-Interaction Scenario: Working Together with an Industrial Robot. In J. Jacko (Ed.), *Human-Computer Interaction. Novel Interaction Methods and Techniques. 5611*, pp. 303-311. Springer Berlin Heidelberg.

Blümling, S., & Reithinger, N. (2015). Aufbereitung und Speicherung von Erfahrungswissen im PLuTO-Projekt. *8. AAL-Kongress.* Frankfurt/Main: VDE Verlag.

Chandrasekaran, B., Josephson, J. R., & Benjamins, V. R. (1999). What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems, 14(1), 20-26.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies, 43*(5-6), 907-928.

Henze, N., & W.Nejdl. Knowledge Modeling for Open Adaptive Hypermedia. In *Adaptive Hypermedia and Adaptive Web-Based Systems: Second International Conference*, pp. 174-183. Springer Berlin Heidelberg.

Kasikci, I. (2010). *DIN EN 13306 (2010-12): Instandhaltung - Begriffe der Instandhaltung.* Tech. rep., Beuth Verlag.

Koper, R. (2014). Conditions for effective smart learning environments. *Smart Learning Environments, 1*(1), 5.

Lenz, C., Nair, S., Rickert, M., Knoll, A., Rosel, W., Gast, J., . . . Wallhoff, F. (2008). Joint-action for humans and industrial robots for assembly tasks. *Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on*, (pp. 130-135).

Lindblom, J., & Thorvald, P. (2014). Towards a framework for reducing cognitive load in manufacturing personnel. In K. Stanney, & K. S. Hale (Ed.), *Advances in Cognitive Engineering and Neuroergonomics* (pp. 233-244). AHFE Conference.

Mavrikios, D., Papakostas, N., Mourtzis, D., & Chryssolouris, G. (2013, 06). On industrial learning and training for the factories of the future: a conceptual, cognitive and technology framework. *Journal of Intelligent Manufacturing, 24*.

Melis, E., Andrès, E., Büdenbender, J., Frischauf, A., Goguadze, G., Libbrecht, P., . . . Ullrich, C. (2001). ActiveMath: A Generic and Adaptive Web-Based Learning Environment. *International Journal of Artificial Intelligence in Education, 12*(4), 385-407.

Melis, E., Büdenbender, J., Goguadze, G., Libbrecht, P., Pollet, M., & Ullrich, C. (2003). Knowledge Representation and Management in ActiveMath. *Annals of Mathematics and Artificial Intelligence, Special Issue on Management of Mathematical Knowledge, 38*(1-3), 47-64.

Sebanz, N., Bekkering, H., & Knoblich, G. (2006). Joint action: bodies and minds moving together . *Trends in Cognitive Sciences , 10*(2), 70-76.

Shah, J. J., & Mäntylä, M. (1995). *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications.* John Wiley and Sons, Inc.

Stoessel, C., Wiesbeck, M., Stork, S., Zaeh, M., & Schuboe, A. (2008). Towards Optimal Worker Assistance: Investigating Cognitive Processes in Manual Assembly. In M. Mitsuishi, K. Ueda, & F. Kimura (Ed.), *Manufacturing Systems and Technologies for the New Frontier* (pp. 245-250). Springer London.

Stork genannt Wersborg, I., Borgwardt, F., & Diepold, K. (2009). Real-time cognitive technical systems, a learning material processing system, social and future aspects. In IEEE (Ed.), *Advanced Robotics and its Social Impacts.*

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

Stork, S., Stößel, C., & Schubö, A. (2008). The Influence of Instruction Mode on Reaching Movements during Manual Assembly. In A. Holzinger (Ed.), *HCI and Usability for Education and Work. Proceedings of the 4th Symposium of the Workgroup Human-Computer Interaction and Usability Engineering of the Austrian Computer Society, USAB 2008. 5298*, pp. 161-172. Springer Berlin Heidelberg.

Ullrich, C., Aust, M., Blach, R., Dietrich, M., Igel, C., Kreggenfeld, N., . . . Schwantzer, S. (2015). Assistance- and Knowledge-Services for Smart Production. In S. N. Lindstaedt, T. Ley, & H. Sack (Ed.), *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business* (pp. 40:1-40:4). ACM.

VanLehn, K., Lynch, C., Schulze, K. G., Shapiro, J. A., Shelby, R., Taylor, L., . . . Wintersgill, M. (2005). The Andes Physics Tutoring System: Lessons Learned. *I. J. Artificial Intelligence in Education, 15*(3), 147-204.

Wahlster, W. (Ed.). (2013). *SemProM: Foundations of Semantic Product Memories for the Internet of Things.* Springer.

Xu, X., Wang, L., & Rong, Y. (2006). STEP-NC and function blocks for interoperable manufacturing. *Automation Science and Engineering, IEEE Transactions on, 3*(3), 297-308.