Knowledge Communication for Intelligent, Mechatronic Agents

W. Dangelmaier, Prof. Dr.-Ing. habil, whd@hni.upb.de
G. Leichtnam, Dipl. Info., gleicht@hni.upb.de
P. Scheideler, Dipl.-Wirt. Ing, peter.scheideler@hni.upb.de
Andreas Schmidt, Dipl.-Wirt. Ing, Andreas.Schmidt@hni.upb.de

Heinz-Nixdorf-Institut, University of Paderborn
http://wwwhni.upb.de/cim/
Fürstenallee 11
33102 Paderborn; Germany

KEYWORDS

Knowledge Communication, Ontology, Multi Agent System, Mechatronic System

ABSTRACT

The objective of the Collaborative research centre 614 “Self-optimising Concepts and Structures in Mechanical Engineering” is the ability to assemble comprehensive future mechanical systems from single intelligent units. These units have specific abilities assuming autonomously the solution of a problem. The individual systems are integrated into one overall system being characterised by an active communication and cooperation between the units. If these units exchange information (data), from the IT point of view, we refer to an agent system. An agent of this system is an autonomous, active, and intelligent entity striving to optimise its own behaviour either autonomously or in cooperation with other agents. In case the system is optimised by a cooperation process, it is necessary for the agents to communicate with each other. Therefore, a multi agent system acting in more than one world of topics requires a precise notional concept for the receiver and the sender to understand the exchanged message. The solution for achieving this state are ontologies. They structure and describe unambiguously the environment. Without this unambiguity, misunderstandings and therefore the failure or collapse of a complex system might be the consequence. Thus, a system environment is to be developed maintaining the unambiguousness for all situations.

1. INTRODUCTION

With the information technology penetrating the domain of mechanical engineering and the consequential success potential, future systems of mechanical engineering will consist of configurations of intelligent system elements and therefore can be regarded as distributed systems of cooperating agents. The Collaborative research centre 614 relates an agent to an autonomous, proactive, and highly adaptive function module. A function module is regarded as a heterogeneous subsystem with electronic, mechanical, and IT components. A function module (agent) controls itself independently, initiating activities on its own, but is embedded in an overall goal. It has a behaviour generic to runtime, which can for example comprise learning abilities or self-optimisations. The communication and cooperation abilities of the single intelligent function modules characterise the behaviour of our overall system.

In this paper we will present an approach to overcome communication problems in a hierarchical mechatronic system. Here communication is important between the various units of one level and among the levels. If words or signs are misunderstood the multiple units of a mechatronic system cannot understand each other and thus cannot reach an efficient collaboration of the individual units.

In section 2 we will provide a basic overview of our problem scenario. Section 3 follows with the hierarchical concept of our Collaborative research centre 614 based on a cross-linked mechatronic shuttle system. In section 4 we will explain the relevance and importance of ontologies for establishing an unambiguous communication process. In section 5 we will show an example how ontologies work in our system. Section 6 will conclude the paper.

2. PROBLEMS IN COMMUNICATION

The agent technology provides models for the communication and coordination of autonomous entities. There are numerous scientific studies and treatises about Agent Communication Languages

(ACL), e.g. FIPA-ACL and KQML, Content Languages (CL), e.g. KIF and FIPA-SL, and Ontology representations (OR), e.g. DAML, OIL and others *(Gmytrasiewicz and Huhns 2000, Reed et al. 2002, Steels 1998)*, *Willmott et al. 2001)*.

The basic approach of communication in Multi Agent Systems is on the one hand the establishment of a common protocol standard defining the syntax and partially the semantics of communication, and on the other hand the creation of a common understanding for the environment and vocabulary or terminology defining the terms for the communication and its semantics. *(Wang and Gasser 2002)*

Even if two agents speak the same language, terms can be described differently. However, the meaning of the exchanged message has to be understood by each agent. A reason for the semantic heterogeneity might be the application of the same term to different things or different terms refer to the same thing *(Uschold 2001)*. Davenport summarised this phenomenon in one sentence: „people can't share knowledge, if they don't speak a common language" (*Davenport et al. 1997*). Moreover, different evaluation systems could be used or naming schemes of information differ significantly (homonyms and synonyms) *(Goh 1997).*

Another problem of the communication between individuals is the existence of various points of views and assumptions being relevant in a common field. Furthermore, a different vocabulary or various overlapping or clashing concepts, structures, and methods could be used. This leads to a missing common understanding and therefore to an inadequate communication within or between groups. *(Uschold and Gruninger 1996)*

That's why a common understanding of the exchanged information is important, meaning a common world view. Reducing or even eliminating the confusion of concepts and terminologies and replacing it by a uniform, understandable terminology is one solution of the problem. *(Guarino 1998)*

3. HIERARCHICAL MECHATRONIC SYSTEMS

Intelligent mechatronic systems consist of a multitude of individual components with a variety of tasks and functionalities. These components can be arranged hierarchically, comparable with the subdivision of a main problem into smaller sub-problems. The Collaborative research centre 614 subdivides the structure of a mechatronic system into three levels *(Oberschelp et al. 2002, Gausemeier 2002)*. The "Mechatronic Function Modules" (MFM) represent the first level. They basically consist of actuators, sensors, and information processing required for the operation performance. A MFM may either include other sub-MFMs or be itself a part of another MFM. At this level information is exchanged at a sub-symbolic level and therefore is not in the focus of our research. At a higher level of the hierarchy there are "Autonomous Mechatronic Systems" (AMS) consisting of various MFMs and serving as a linking element of the individual MFMs and therefore have a superior IT-based point of view. AMSs must be able to interact with the environment and therefore must be able to react to its changes. The ability to communicate, to coordinate, and to plan is required for such a behaviour (and for the exchange of information). Functional processes assigned to multiple AMSs are represented on the highest level "Cross-linked Mechatronic Systems" (CMS). Thus a CMS results from the IT-coupling of the involved AMSs. Here, it is not referred to a physical coupling, but to the exchange and processing of information. Such a hierarchy is represented in figure 1.

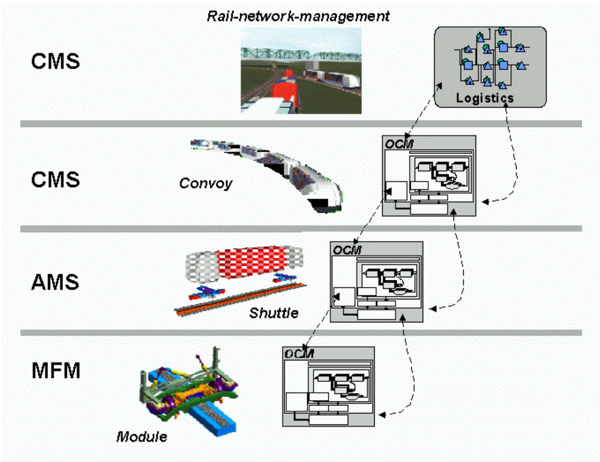

Figure 1: Hierarchical Structure

4. SOLUTION OF COMMUNICATION PROBELMS

Ontologies are to contribute to the solution of communication problems. Ontologies are models of an application domain, serving to facilitate the exchange and sharing of knowledge. In the literature, a multitude of varying definitions of the understanding of ontologies can be found. *(Guarino and Giaretta 1995, Guarino 1996)*

The term ontology is used when referring to a common understanding of a field of activity or interest serving as a framework to the solution of communication flaws and to the prevention of misunderstandings. This framework typifies a view expressed by a “Set of concepts” (e.g. entities E, attributes A, and processes P), the appropriate definitions and relations on a certain domain. *(Uschold 1998, Uschold and Gruninger 1996)*

Some definitions refer to the often-used definition of Gruber. According to him, an ontology is an explicit specification using formal language of a commonly used conceptualisation of reality phenomenon. The term conceptualisation is understood as an abstract and simplified point of view on phenomenon of a reality sector being of interest to predetermined insight purposes. *(Gruber 1993-1)*

The additional requirement of the explicit specification using formal language of this conceptualisation is a particular research interest. However there is scepticism, if the “basic meaning” or “semantics” of reality experience expressed with natural language can be reconstructed completely and unaltered by means of a representation using formal language.

When communicating, an ontology is used to provide a consistent vocabulary with determined terminologies. Furthermore, an ontology serves the interoperability, i.e. as a “translator” or “mediator” between the systems. Here, interrelations between processes and procedures are modelled with the help of ontologies or they even serve as a representation of the specification of the system. *(Uschold and Gruninger 1996)*

Ontologies can be classified into domain-, task-, and common ontologies. The domain-ontology represents the conceptualisation and the framework for knowledge bases of certain domains (e.g. rail traffic, medical science) and consists of objects and axioms. The Task-ontology contains methods or procedures to solve certain problems (e.g. acceleration of a car). The common ontology (or top-level-ontology) defines general concepts being found in several types of ontologies (e.g. for states, time, geometrical, physical or topological presentations). (*Guarino 1997, Guarino 1998)*

General guidelines and models for the construction phase exist in order to develop ontologies (similar to software development). The definitions described are to be objective and independent from social or technical context. A definition should be done in logical axioms – if possible – and documented in a natural language. Ontologies should be coherent, i.e. the inferences are not contradictory to the definitions (*coherence of ontology*). Furthermore, the defining of terms via existing vocabulary should be possible without examining and correcting the existing definitions anew (*extensibility of ontology*). The concept development of ontologies should not be made on a partial coding dependent on a subsequent implementation (*minimum use of implementation details*). In addition ontologies should have minimal, but sufficient propositions of a domain being modelled (*minimum description of ontology*) *(Gruber 1993-2).*

In a multi agent approach three fundamental architectures are described in order to represent and exchange information semantics: single ontology approaches, multiple ontology approaches, and hybrid approaches.

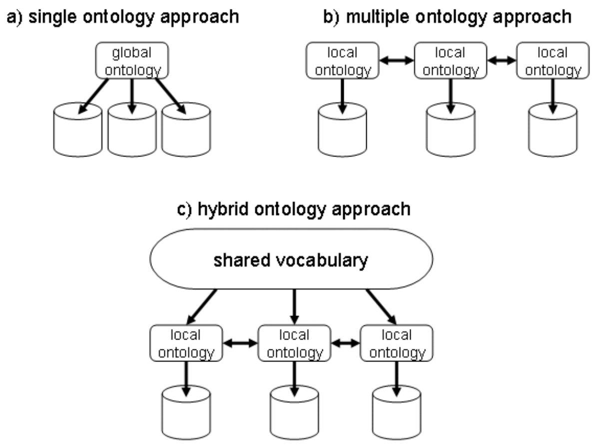

Figure 2: Three main ontology architectures *(Wache et al. 2001)*

Single ontology approaches use a single global ontology for the specification of the semantics. Thereby the whole system is described by one single domain model, causing the problem that a minimal ontological predefinition of different point of views on a domain are to be found. In multiple ontology approaches the semantics of each source of information is described by its own ontology. This is advantageous, since no minimal ontological predefinition has to be found. However, the lack of a common vocabulary makes the comparison of various ontologies difficult. The various advantages and disadvantages of single or multiple ontology approaches are to be overcome by the hybrid approach which we pursue in our example. Comparable with the multiple ontology approaches, the semantics of each source of information is described by its own ontology. However, the ontologies are based on a common vocabulary. *(Wache et al. 2001)*

In our research, where we follow the hybrid approach, we accept a semantic heterogeneity between the various agents. Ceri and Widom (*Ceri and Widom 1993*), for instance, list four categories of semantic heterogeneity (conflicts): (1) naming conflicts (different databases use different names to represent the same concept), (2) domain conflicts (different databases use different values to represent the same concept), (3) meta-data conflicts (same concepts are represented at the schema level in one database and at the instance level in another database), and (4) structural conflicts (different databases use different data organization to represent the same concept). In the research project KRAFT (*Visser et al. 1997*) they summarized those 4 categories down to 2: (1) *conceptualisation mismatches*, and (2) *explication mismatches*. We will demonstrate in one example from each category how to cope with such conflicts.

In the development process of the ontologies (Figure 3) we involved all aspects being relevant to their development and the aspects of maintenance within the application of the knowledge management. In that the Ontology $O:= \{C,R,H^C,rel,A^O\}$ is a 5-Tupel (*Maedche 2003*) which consists of a number of concepts C, a number of relations R, a concept hierarchy H^C ($HC \subseteq C \times C$), a function rel: $R \rightarrow C \times C$, and a number of axioms A^O. Thereby $\forall$ R: $rel(R) \not\subseteq H^C$, represents all non-taxonomic (is_a) relations. With rel any kind of relation can be formulated like employed_at, bigger_than, belongs_to. Based on the linguistic multifariousness the function rel is mentioned many times in connection to the Semantic Web.

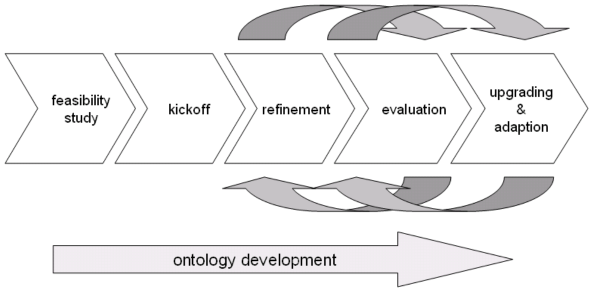

Figure 3: Ontology Development *(Staab 2002)*

5. EXAMPLE

During the development of our ontology a feasibility study served as the identification of the problems to be solved and the analysis of available possibilities to solve those problems. During the kick-off phase a first requirement specification was established by using texts, graphs, and interviews with the engineers. This first description of ontology was then refined by checking its consistency and completeness considering the requirements. Thereby the description of ontology is extended until all relevant concepts (C), relations (R, rel) and axioms (A^O) have been included. The description was extended during an iterative process, being formalized by the help of Protégé-2000 (a tool developed by the Stanford Medical Informatics).

The area to demonstrate the modelled environment mainly consists of a decentral, demand-driven shuttle-system. The shuttles have to communicate and cooperate within this system in order to reach their destinations. In the following, the exchange of information and knowledge between two autonomous intelligent shuttles will be presented and illustrated based on a possible scenario (figure 4).

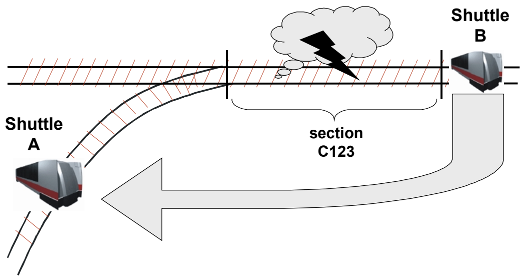

Figure 4: Exchange of Information

The exchange of information is as followed:

```
(inform
:sender (agent-identifier :name B)
:receiver (set (agent-identifier :name A)
:content
        "weather (Section C 123, thunderstorm)
         reduce_to (velocity, 100)"
:ontology Common Ontology
:language Prolog)
```

With the help of a common ontology (kind of "shared or basic vocabulary") the shuttles are permitted to communicate with other shuttles or autonomous entities (e.g. train stations or users). The common ontology (figure 5) comprises fundamental terms of the system, for example the physical basic parameters and their terminology, the possible conditions within the system or the fundamental rules and relations. However each shuttle has its individual Task Ontology describing the techniques of how to proceed when solving problems. In addition, each agent posses its own Domain Ontology (seeing the world with his "eyes"). At the beginning, every shuttle may have the same Domain Ontology though it might be changed subsequently by the shuttle's experiences, its evolution status, and updated knowledge.

Superclass (abstract): *PhysicalQuantity*			
Slot	**Type**	**Cardinality**	**Reference**
Magnitude	Float	=1	
Unit	String	=1	

Class (abstract): *FunctionalQuantity* *Sub-Class of PhysicalQuantity*			
Slot	**Type**	**Cardinality**	**Reference**
Formula	String	= 1	
Parameter	Instance	≥ 1	*PhysicalQuantity*
Magnitude	Float	= 1	
Unit	String	= 1	

Class (concrete): *MomentOfInertia (solid cylinder)* *Sub-Class of RotationQuantity* *Sub-Class of DynamicQuantity* *Sub-Class of SectionQuantity*			
Attribute	**Type**	**Card.**	**Reference**
Formula	String	= 1	Value={$½*m_{ges}*r$^2 }
Parameter	Instance	≥ 1	*Kilogram, Radius*
Magnitude	Float	= 1	
Unit	String	= 1	Value={kg m^2}

Figure 5: Classes of the Common (Mediator) Ontology

In our scenario we assume that shuttle A and shuttle B have different domain ontologies. Shuttle A is only able to identify the terms of a message from shuttle B with the help of the Common Ontology. Let's assume that we have an attribute-assignment mismatch (belongs to the category *conceptualisation mismatches*) meaning that the common ontology represents that the weather condition X is a "thunderstorm". However, the ontology of the shuttle A only knows the attributes "thunder" and "heavy rain" in association with the word weather condition and shuttle B only knows the attributes "storm" and "heavy shower". In this case the mapping between these terms can be done by a mapping function mf (Ont(A) denotes the ontology of the shuttle A, Ont(B) denotes the ontology of the shuttle B and Ont(C) denotes the ontology of the Common Ontology):

mf1 (Ont(C): thunderstorm (x)   Ont (A) thunder (x) ∧ heavy rain (x))
mf2 (Ont(C): thunderstorm (x)   Ont (B) storm (x) ∧ heavy shower (x))
mf3 (Ont (A) thunder (x) ∧ heavy rain (x) ⇔ Ont(B) storm (x) ∧ heavy shower (x)

Another mismatch can be occur with *explication mismatches,* e.g. if two systems use the same definitions to denote a concept but refer to it with different terms. Then the terms have to be mapped onto each other. Let's assume that the shuttle A wants to communicate that it needs an ID-Number from shuttle B. Shuttle B however does not know this term but uses the term ID-code. Again, we can and have to map those terms via our common ontology with the same method we have used above:

mf1 (Ont(C): passenger-Number(x)   Ont(A) ID-Number (x))
mf2 (Ont(C): passenger-Number(x)   Ont(B) ID-Code (x))
mf3 (Ont(A) : Id-Number (x) ⇔ Ont(B): ID-Code (x))

However, the Common Ontology should not only try to map different Ontologies, but also tries to build links between concepts to demonstrate their interdependencies. Figure 6 shows a small sector of the Common Ontology to demonstrate their storage of knowledge. The figure shows that if a passenger selects a luxury shuttle then this selection has some implication on the driving module. So, only shuttles which carry a special module with this driving standard are selected for the passenger.

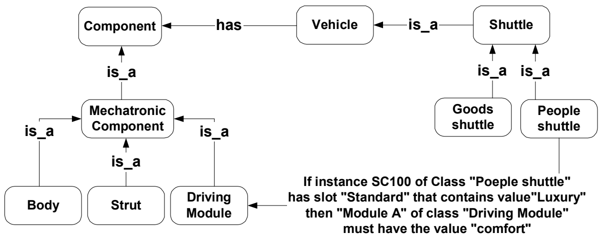

Figure 6: Example of an axiom in the Common Ontology

6. CONCLUSION

First, we have shown that it is necessary to use ontologies if you want to establish an unambiguous communication between individual units. Communication is needed within one level and between the levels to come to the best possible solution. In case that we have different ontologies operating in one system we need to map them. That means that we need to establish a mapping from the individual ontology of the one agent to the Common Ontology of the system. Once we have established a full integration of the individual ontology into the common one, the individual agent can communicate with all other agents even though they have a different understanding of the respective domain. Second, we have mentioned the aspect that ontologies carry knowledge which can be necessary to draw the right conclusion. If a passenger wants to have a deluxe shuttle, a shuttle must understand what exactly that means. The word "Deluxe" in an industrial country can mean something completely different than in a developing country.

ACKNOLEDGMENTS

This work was developed in the course of the Collaborative Research Centre 614 – Self Optimising Concepts and Structures in Mechanical Engineering – University of Paderborn, and was published on its behalf and funded by the Deutsche Forschungsgemeinschaft.

REFERENCES

Ceri, S., and J. Widom (1993). Managing Semantic Heterogeneity with Production Rules and Persistent Queues. Proceedings of the 19th VLDB Conference, Dublin, Ireland, pp. 108-119.

Davenport, T.H., L. Prusak, and L. Prusak (1997). Working Knowledge: How Organizations manage what they know. Harvard Business School Press, Boston.

Gausemeier, J. (2002). Von der Mechanik zur Selbstoptimierung. 20th CAD-FEM Users Meeting, Friedrichshafen, Germany.

Gmytrasiewicz, P., and M. Huhns (2000). The emergence of language among autonomous agents. IEEE Internet Computing, 4(4), pp. 90-92.

Goh, C.H. (1997). Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources. PhD, MIT.

Gruber, T.R. (1993-1). A Translation Approach to Portable Ontology Specifications. Knowledge Systems Laboratory, Technical Report KSL 92-71.

Gruber, T.R. (1993-2). Towards Principles for the Design of Ontologies used for Knowledge Sharing. Knowledge Systems Laboratory, Technical Report KSL 93-04.

Guarino, N. (1996). Understanding, Building, and Using Ontologies. Proceedings of the 10th Knowledge Acquisition for Knowledge based Systems Workshop, Banff, Canada.

Guarino, N. (1997). Semantic Matching: Formal Ontological Distinctions for Information Organization, Extraction, and Integration. In Lecture Notes in Computer Science 1299, Springer, pp. 139-170.

Guarino, N. (1998). Formal Ontology and Information Systems. Proceedings of FOIS 98, Trento, Italy.

Guarino, N. and P. Giaretta (1995). Ontologies and Knowledge Bases – Towards a Terminological Clarification. In N. J. I. Mars (ed.), Towards Very Large Knowledge Bases, IOS Press 1995, pp. 25-32.

Meadche, A. (2003). Ontology Learning for the Semantic Web. Kluwer Academic Publisher, Dordrecht, 2nd edition.

Oberschelp, O., T. Hestermeyer, B. Kleinjohann, and L. Kleinjohann (2002). Design of Self-Optimising Agent-Based Controllers. CFP Workshop 2002 – Agent Based Simulation 3, University of Passau 2002.

Reed, C., T. Norman, and N. Jennings (2002). Negotiating the semantic of agent communication languages. Computational Intelligence, 18 (2).

Staab (2002). Wissensmanagement mit Ontologien und Metadaten, Habilitation, University of Karlsruhe AIFB.

Steels, L. (1998). The origin of ontologies and communication conventions in multi-agent systems. Autonomous Agents and Multi-Agent-Systems, 1(2), pp. 169-194.

Uschold, M. (1998). Knowledge level modelling: concepts and terminology. The Knowledge Engineering Review, 13(1), pp. 5-29.

Uschold, M. (2001). Barriers to Effective Agent Communication, Position Statement. Workshop on Ontologies in Agent Systems, Montreal, Canada.

Uschold, M., and M. Gruninger (1996). Ontologies: Principles, Methods and Applications. Knowledge Engineering Review, 11 (2).

Visser, P.R.S., D.M. Jones, T.J.M. Bench-Capon and M.J.R. Shave (1997). An Analysis of Ontology Mismatches; Heterogeneity versus Interoperability. Kraft report KP6, also appeared in AAAI 1997 Spring Symposium on Ontological Engineering, Stanford University, California, USA, pp. 164-172.

Wache, H., T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hübner (2001): Ontology-Based Integration of Information – A Survey of Existing Approaches. In Stuckenschmidt, H., (editor). IJCAI-01 Workshop: Ontologies and Information Sharing, 2001, pp. 108-117.

Wang, J. and L. Gasser (2002). Mutual Online Concept Learning for Multiple Agents. AAMAS’02, Bologna, Italy.

Willmott, S., L. Constantinescu, and M. Calisti (2001): Multilingual Agents: Ontologies, Languages and Abstractions. Workshop on Ontologies in Agent Systems, Montreal, Canada.