INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITE DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

USAGE OF ONTOLOGIES AND SOFTWARE AGENTS FOR KNOWLEDGE-BASED DESIGN OF MECHATRONIC SYSTEMS

Ewald G. Welp[1], Patrick Labenda[1] and Christian Bludau[2]
[1]Ruhr-University Bochum, Germany
[2]Behr-Hella Thermocontrol GmbH, Lippstadt, Germany

ABSTRACT
Already in [1, 2 and 3] the newly developed Semantic Web Service Platform SEMEC (SEMantic and MEChatronics) has been introduced and explained. It forms an interconnection between semantic web, semantic web services and software agents, offering tools and methods for a knowledge-based design of mechatronic systems. Their development is complex and connected to a high information and knowledge need on the part of the engineers involved in it. Most of the tools nowadays available cannot meet this need to an adequate degree and in the demanded quality.
The developed platform focuses on the design of mechatronic products supported by Semantic web services under use of the Semantic web as a dynamic and natural language knowledge base. The platform itself can also be deployed for the development of homogenous, i.e. mechanical and electronical systems. Of special scientific interest is the connection to the internet and semantic web, respectively, and its utilization within a development process. The platform can be used to support interdisciplinary design teams at an early phase in the development process by offering context-sensitive knowledge and by this to concretize as well as improve mechatronic concepts [1].
Essential components of this platform are a design environment, a domain ontology mechatronics as well as a software agent. The developed domain ontology mechatronics contains basic knowledge for the early development phases of mechatronic systems. Relevant product knowledge is implemented on the two levels of a system as well as a behavioural model. This knowledge provided can be processed by a software agent. Embedded in the Semantic Web Service Platform, both, ontology and software agent, play a major role to more effectively support engineers and developers by an improved knowledge-based design process. The contribution at hand focuses semantic web technology and especially an ontology mechatronics and its use in the development of mechatronic systems.

Keywords: Mechatronics, knowledge-based design and development, ontology, software agent

1 INTRODUCTION
In the development of mechatronic systems both the high interdisciplinarity as well as the complexity involved mean a special demand to efficiently tap the full potential offered by the solution spectrum. Mechatronic systems take a trend-setting key position and make the conception of products possible with a steadily improved functionality and efficiency as well as an optimized behaviour [4]. But the innovative possibilities are contrasted by increased requirements in the development process. These are caused by the variety of knowledge domains involved, by the interactions of different engineering disciplines [5] as well as by the complexity of mechatronic systems which can be explained by the products' interdisciplinary character. The development of mechatronic systems therefore represents a knowledge-intensive process [6].
The challenges in the development process are the reason for the conception, development and use of methodologies as well as information technology systems which aim to support the product developer in the acquirement of domain integrated solutions. Most of the tools focus on a domain integrative model exchange [7] and structured element repositories [8] to enable the design of mechatronic systems. In contrast to these passive knowledge sources especially knowledge-based product development which aims at an efficient exposure to the resource knowledge is of emphasised importance. It aims at an active processing and provision of knowledge [9 and 10]. This includes

applications and systems which make it possible to collect and to organize knowledge as well as to provide it to the user in a structured form. Within the last few years knowledge-based systems have received increasing attention in the area of knowledge management [11] in form of PDM (Product Data Management) and PLM (Product Lifecycle Management) systems [12 and 13] as well as in the construction [14, 15 and 16] and design [17 and 18] of technical systems particularly. The existing approaches have clear deficits in the field of context-sensitive and suitable for a problem context processing as well as provision of data and information, though.
In connection with knowledge-based systems, a clear progress is achieved by Semantic web technology. It means the deposit of machine-readable, semantic information which is appended invisibly for the human user in the background of a document. The Semantic web itself represents the future Internet [19]. The contents of internet pages are enriched with meaning (semantics) so that finding, evaluating as well as processing of data and information by computer applications are improved fundamentally.
On the internet information is put down in the form of XML (eXtensible Markup Language) documents. In the Semantic Web these documents not only contain for humans interpretable information but also machine-readable semantics which is formatted in a RDF (Resource Description Framework) standard. Software systems accessing this technology can answer natural-language enquiries appropriately and work on even more complex tasks in a more efficient way.

2 SEMANTIC WEB SERVICE PLATFORM SEMEC

The aim of the project Semantic web services mechatronics has been the development of a methodical and technological platform in the area of services in the knowledge-based product development. A platform illustrated in Figure 1 as a Semantic web service has arisen which supports the design of mechatronic systems under application of software agent technology. The system itself deploys the Semantic web as a dynamical and natural language knowledge base.

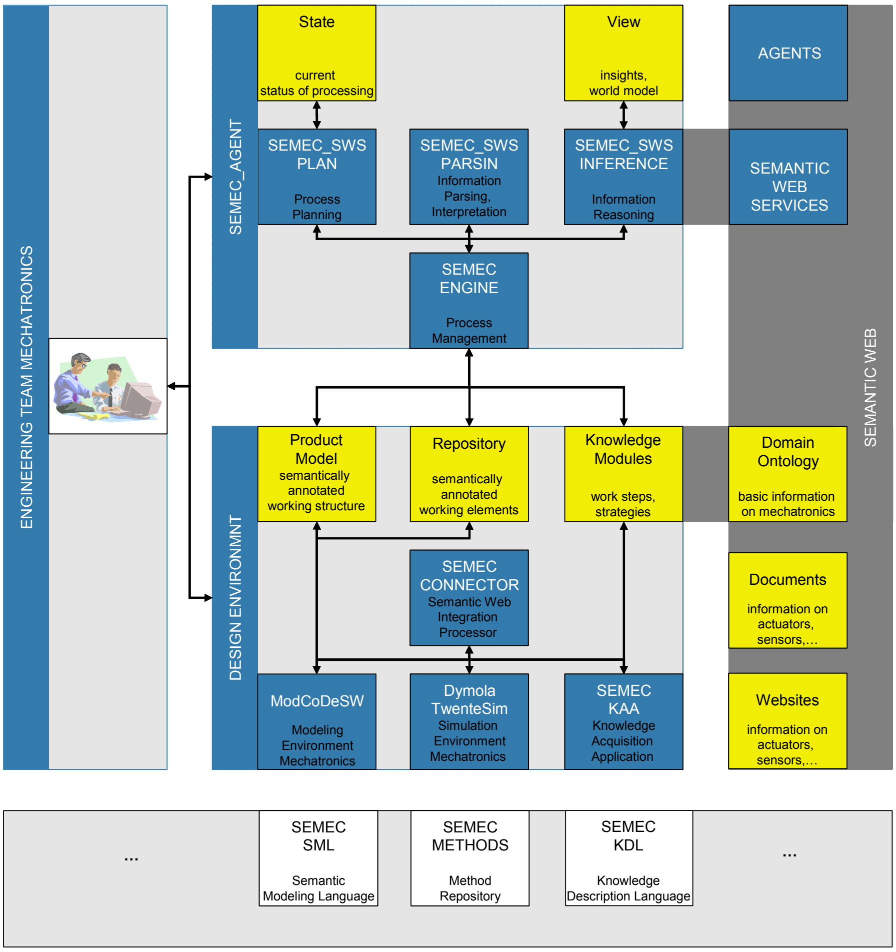

Figure 1. Semantic Web Service Platform SEMEC [1]

The platform's major functionality has already been described in [1 and 3] and is summarized in Figure 2. The platform can be used by engineers. For example, a mechatronic concept at different abstraction levels can be worked out by a team within a design environment. In the production of working structures element libraries and repositories are at the developers' disposal. The individual elements can be concretized by features and properties. The created product model is provided to a software agent for the analysis of the design context. This is made possible by the fact that the system elements are semantically annotated. With the help of ontologies, web pages and documents the created concept can be processed by the agent context- and problem-sensitively. Exemplarily a mechatronic system can be produced by the engineers that further on shall be used by the agent to the effect to generate an alternative concept. Furthermore the developed platform can be connected to commercial software systems, e.g. Dymola, to enable a system's simulation and behavioral evaluation.

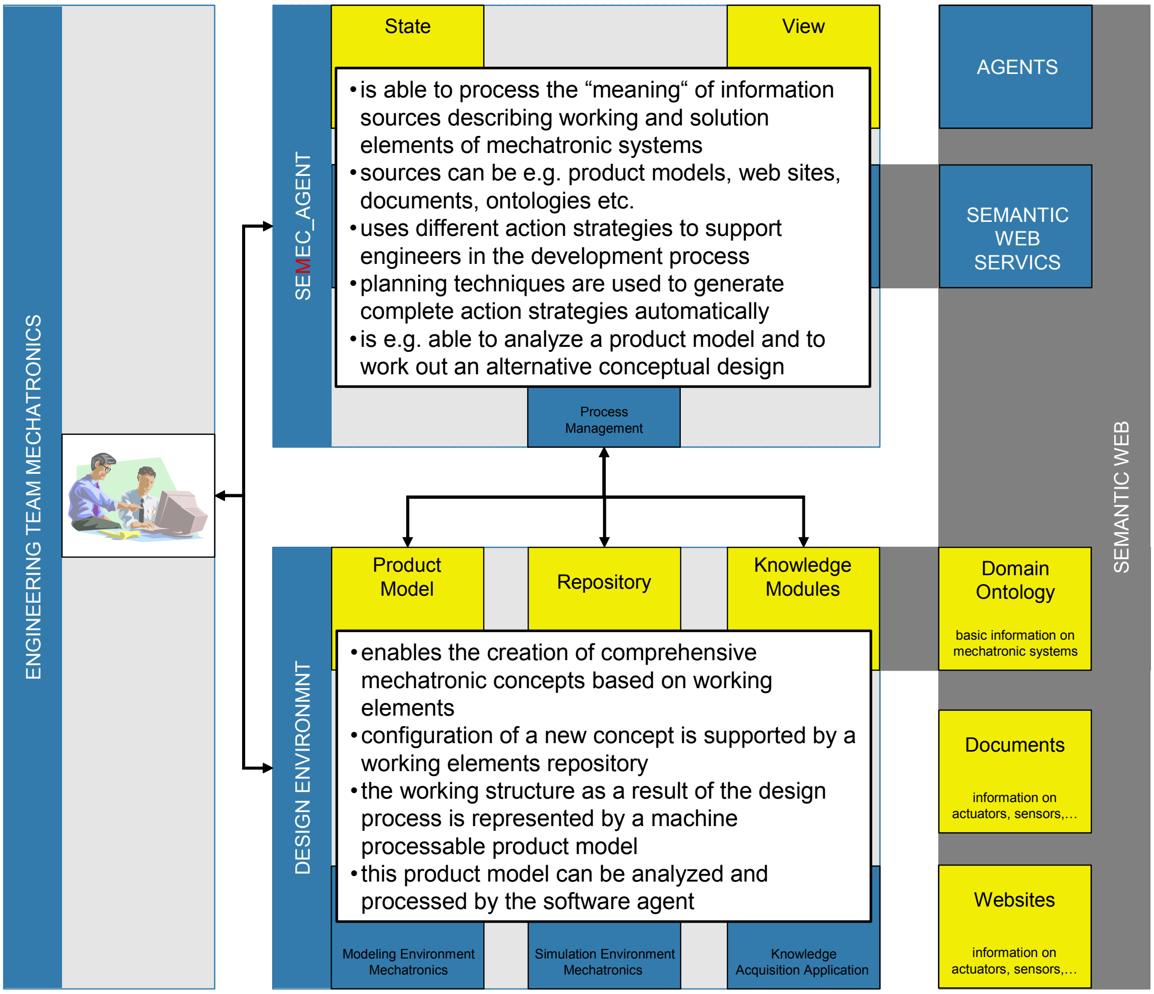

Figure 2. Semantic Web Service Platform SEMEC; major functionality

The software agent's actions are affected and defined by strategies and working steps contained in a knowledge module. Framework conditions, restrictions and the engineers' instructions are taken into account. The engineers can communicate with the software agent by configuration of strategies and work steps as well as by the definition of plans and meta-plans. The processing of system elements and systems is carried out with the help of information from the knowledge module, ontologies, web documents, web services as well as other software agents [1].

Further on the software agent consists of four essential parts. First of all it includes a process management for coordination of activities. Second, input data is analysed and interpreted by a parser. Third, there is a planning device for the definition of action strategies, their configuration, combination of plans as well as meta-plans, incorporation of other web services and software agents just to reach a predefined aim under the engineers' supervision. Last, the software agent contains an inference machine to reason about different forms of information, to draw logical conclusions as well as to generate implicit knowledge.

3 SEMANTIC WEB TECHNOLOGY

The growth of the World Wide Web makes progress steadily. Since many years it is the greatest existing knowledge base. It is of major importance also for product development. Internet pages more and more also contain technical contents in form of web documents which in particular are of interest for developers of technical as well as mechatronic products. Due to a steadily increasing data and information flood on the internet the extraction of information by search enquiries turns out noticeably more difficult. The mostly unsatisfactory search results always require a time effortful interpretation by the human user.

3.1 Semantic Web

Semantic web means intelligent dealing with web contents. It is a relatively young field of research but more and more components are getting standardized. The idea is to expand the existing World Wide Web with machine processable information. This is done by means of annotating web documents with metadata. Metadata means data about data. These metadata further on can be brought into a semantic context according to its meaning and can afterwards be analogously processed by a machine or software system [20]. It means that information can be processed by computer applications context-sensitively.

Nowadays the contents of a web page are graphically represented in HTML (Hypertext Markup Language) and structured in XML and XML(S) (eXtensible Markup Language Schema). Within XML no relations can be fixed between the produced objects and constructs. XML(S) gives the opportunity to add restrictions but still the contents of the web pages do not have any semantics. Right until now the information on a web page really efficiently can only be interpreted by a human user.

In the Semantic web metadata are added to a web page in the background not visible for the user. These metadata is implemented in RDF and RDF(S) (Resource Description Framework Schema). RDF and RDF(S) are also called metadata description languages. Extending XML, both, RDF and RDF(S) make it possible to put resources into relation with each other. The base element of RDF is a triple consisting of object, attribute as well as value. Further on RDF is extended by RDF(S) to enable the definition of characteristics as well as their restrictions and the development of hierarchies of classes and properties.

But RDF and RDF(S) are limited in their expression ability. One important deficit is the missing opportunity to distinguish between classes/objects and instances [21]. It is said that only a semantic basic skeletal structure [22] can be built up by means of RDF and RDF(S). This skeletal structure has further on to be completed by the contents of ontologies and by this to enable the creation of entities.

3.2 Ontologies

Metadata implemented in RDF and RDF(S) is brought into a semantic context by means of ontologies. These can be characterized as meaning defining knowledge bases. They classify the information of a specific field of knowledge. But, more important, ontologies are used to define the semantics of data. With a semantic annotation of information as well as the help of references to contents of an ontology the information get an analogous context and a processing according to its meaning is made possible.

Ontologies are said to be backbone [23] as well as semantic basic constituent of the Semantic web [20]. They have their origin in the Artificial Intelligence as one of the numerous possibilities of knowledge modelling as well as for the production of knowledge systems. Therefore, an ontology is suitable to build up an understanding model of a knowledge domain [20].

Essential part of an ontology's development is the conceptualisation of a certain field of knowledge. In this conceptualisation, single objects or reality parts are identified and described as concepts or entities. Further on the relations between the concepts are identified and described as well. The term 'concept' means a mental picture which is represented by a term and with which persons associate a real fact. Furthermore the development of an ontology demands a shared conceptualisation. It means an obligatory specification and acceptance of the contents by a user group as large as possible. Only by this an efficient (global) knowledge exchange [24] as well as a durable reuse of the stored knowledge [25] will be possible.

Just to make a machine or software system processing of knowledge possible it requires an appropriate representation formalism. Therefore the shared conceptualisation is specified and structured in a formal and explicit form [26]. To make it possible not only for a human user but also for a computer-

based system to process an ontology's contents it is essential to deploy a formal description language. To avoid ambiguities and misinterpretations the statements are formalised in an explicit form.
Ontologies are knowledge bases with a structured order of a number of concepts. These concepts are connected with each other by rules and relations. The deployment of such a knowledge base by a computer application leads to a better processing of tasks applied on this application. This is due to the possibility of the use of ontology contents for the drawing of logical conclusions. Implicit knowledge can be generated with the help of inference machines by means of rules- and logic-based definitions.
Essential components of ontologies are classes, properties and individuals. Any independent concept can be represented by a class. The specified classes are structured hierarchically in taxonomies. Every class or object gets properties which describe them more precisely. After the creation of instances these properties can be occupied with a value depending on the type of property. Object properties, e.g., can put classes and individuals into connection with each other. An example of an ontology is shown in Figure 3.

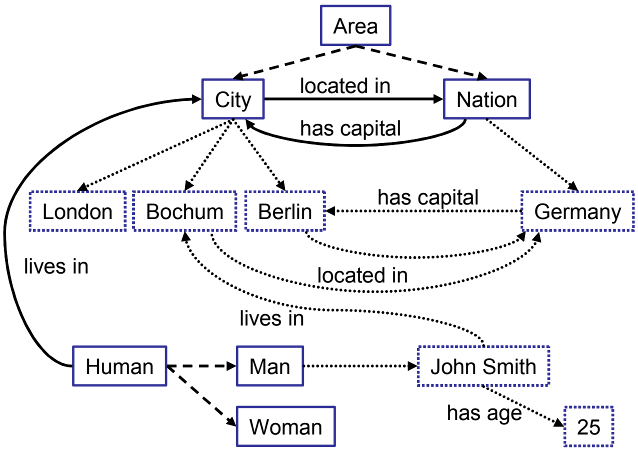

Figure 3. Example of an ontology

To make it possible to process ontology contents by a computer application a formal description language has to be used. Furthermore the integration of knowledge stored in an ontology as well as the Semantic web justifies the use of a web-compatible ontology specification language. Right now the Web Ontology Language OWL turns out to be the most expressive ontology language. It is an object-oriented language and already established standard of the World Wide Web Consortium (W3C). The ontology presented in Figure 3 is shown in OWL source code in Figure 4.

```
<owl:Class rdf:ID=„City">
  <rdfs:subClassOf>
    <owl:Class rdf:ID=„Area"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID=„Nation">
  <rdfs:subClassOf rdf:resource="#Area"/>
</owl:Class>
<owl:Class rdf:ID="Man">
  <rdfs:subClassOf>
    <owl:Class rdf:ID=„Human"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID=„Woman">
  <rdfs:subClassOf rdf:resource="#Human"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="lives_in">
  <rdfs:domain rdf:resource="#Human"/>
  <rdfs:range rdf:resource="#City"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="has_capital">
  <rdfs:range rdf:resource="#City"/>
  <rdfs:domain rdf:resource="#Nation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="located_in">
  <rdfs:range rdf:resource="#Nation"/>
  <rdfs:domain rdf:resource="#City"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="has_age">
  <rdfs:domain rdf:resource="#Human"/>
  <rdfs:range rdf:resource=
      "http://www.w3.org/2001/XMLSchema#int"/>
</owl:DatatypeProperty>
<City rdf:ID="London"/>
<City rdf:ID="Bochum">
  <located_in>
    <Nation rdf:ID=„Germany">
      <has_capital>
        <City rdf:ID="Berlin">
          <located_in rdf:resource="#Germany"/>
        </City>
      </has_capital>
    </Nation>
  </located_in>
</City>
<Man rdf:ID="HansSchmidt">
  <lives_in rdf:resource="#Bochum"/>
  <has_age rdf:datatype=
    "http://www.w3.org/2001/XMLSchema#int"
  >25</has_age>
</Man>
```

Figure 4. OWL source code for the ontology illustrated in Figure 3

Basic knowledge and language levels in the Semantic Web as already described in chapters 3.1 and 3.2 are shown in Figure 5.

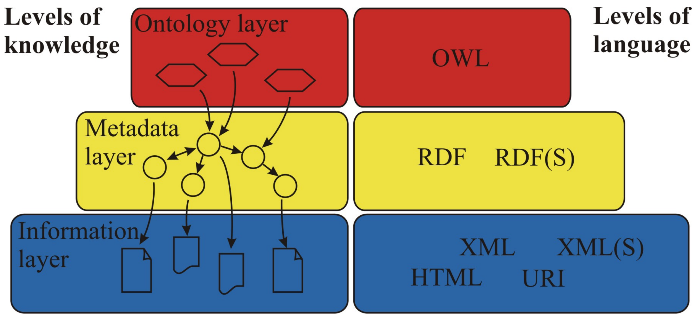

Figure 5. Levels of knowledge and language in the Semantic web [27]

The information level contains documents and web pages. Information given at this level is understandable only for a human user depending on the formulated premises. Every document and resource on the internet gets an URI (Uniform Resource Identifier). In the background of the information level there is the metadata level. It contains concepts of a knowledge domain which are put into relation with each other. Information in a certain document on the information level gets a meaning by assigning certain concepts of the metadata level to the information. The ontology level contains ontologies to the most different knowledge domains. These ontologies and the included concepts are specified formally. Concepts from the metadata level and concepts from the ontology level are put into relation with each other. Therefore also the information at the information level is put into relation with the ontology level and the information gets a semantic context.

3.3 Semantic Web Services and Software Agents

Semantic web technology has the aim to access knowledge put down in a system more efficiently. This can be achieved by a semantic annotation of documents as well as by putting it into a context by means of ontologies. Overall especially the number of hits as well as the precision of output during an enquiry for an interesting knowledge source can be improved significantly. By a meaningful provision of documents, on the one hand, an acceleration within the use of already existing knowledge will take place. By this, on the other hand, also the processing time will accelerate and development costs can be reduced.

One simple consequence resulting from these explanations leads to the fact that it is necessary to expand already existing PDM and PLM systems with semantic technology. In principle it means to equip these systems with more intelligence. An efficient provision of archived knowledge and an improved interoperability between heterogeneous computer applications can be achieved by an obligatory specification of a common understanding e.g. by means of ontologies. Furthermore a semantic association as well as the linking of documents of different formats is necessary. Fundamental idea is the integration of knowledge of the most different information sources [28].

An intelligent processing of (technical) information in the Semantic Web can be realized by Semantic web services and software agents. Semantic web services represent service applications in the Semantic web. They can be used for simple enquiries as well as for complex tasks. Furthermore it is possible to deploy Semantic web services by other web services and software agent systems.

At this agents represent software programs which can carry out tasks assigned to them as autonomously as possible. They reach back for this end on the knowledge provided to them by the user or their surroundings. According to [29] a software agent can be defined as follows: “A software agent is an autonomous, proactive, cooperative and adaptive functional module. Autonomous means self-control enabling the initiation of actions (proactively). Agents are described as functional modules which are further able to cooperate or to compete with each other. Their adaptive behaviour describes a generic process including also the ability to learn.” More easily spoken a software agent is a more or less intelligent client application which can be deployed to solve a special task.

One possible scenario in the Semantic web is illustrated in Figure 6:

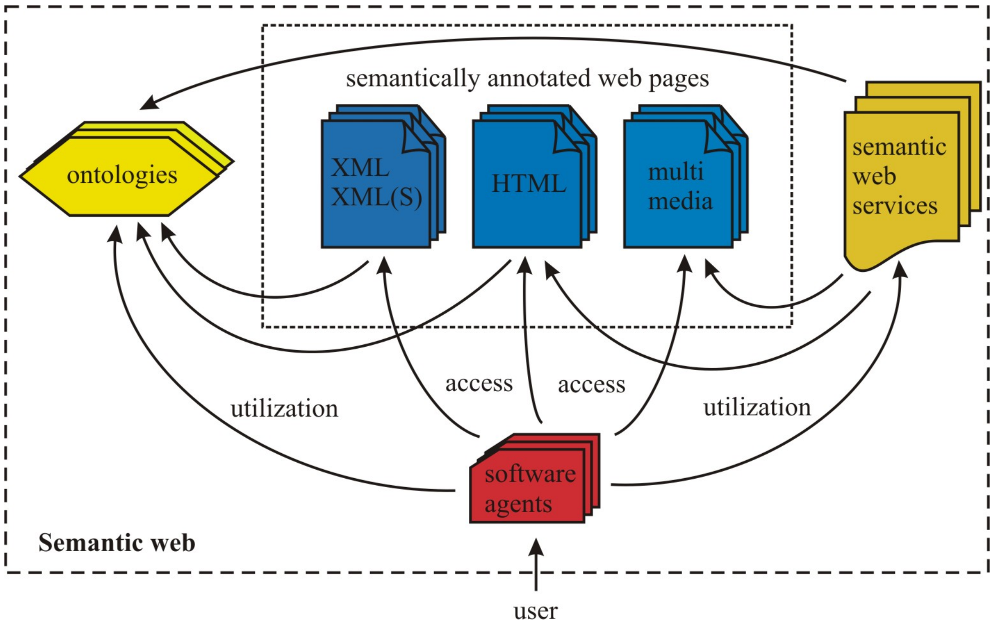

Figure 6. Components and possible interactions in the Semantic web [27]

4 ONTOLOGY ENGINERING

The field of science dealing with the development of ontologies is called "ontology engineering". It is a relatively young discipline and focuses on the development of methodical approaches as well as computer-aided tools for the generation of ontologies.

4.1 Ontology development process

An ontology development process is enormously time-consuming and labour-intensive. Although the process nowadays is supported by software tools an ontology's elaboration still is manual work. Some aspects are important. First, it must be guaranteed that it is always possible to adapt an ontology to new and modified content of knowledge. This aspect always depends on the dynamics of a special field of knowledge. Second, an ontology's generation should always be developed by persons with a funded knowledge as well as experience in connection with a special domain.
The most important ontology development methodologies are the following ones:

- Cyc methodology [30]
- Uschold, King [31]
- Grüninger, Fox [32]
- KACTUS method [33]
- METHONTOLOGY [34]
- SENSUS method [35]
- On-To-Knowledge Methodology [36]

An intense analysis of these methodologies has led to essential components of an ontology development process as illustrated in the following Figure 7.

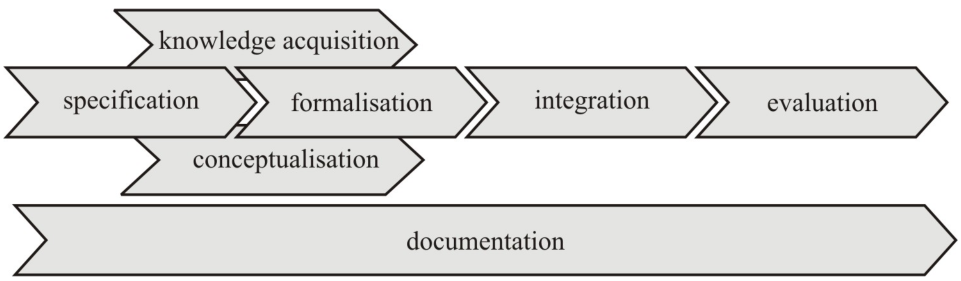

Figure 7. Major steps in an ontology development process

The specification phase includes all activities for the definition of the application area of interest as well as the incorporated requirements. Further on also an ontology's intent, knowledge domain as well as operational environment are of importance. These aspects can be called an ontology's scope.
According to the stated methodologies knowledge acquisition is of minor importance. Often it is part of the specification activities. But we have recognized it as a major part in the development of a completely new ontology. Knowledge acquisition can be supported by numerous methods such as interview of experts, text and data mining as well as a direct and active input by experts.
In Figure 7 the conceptualization phase is illustrated as a parallel process to knowledge acquisition. Conceptualization means a gathering of relevant concepts of a knowledge domain as well as the relations between these concepts. By this a conceptual model is generated. Concepts and relations are described in an explicit form which is different from a pure knowledge acquisition. To define concepts as well as relations constructs are used in the form of classes, objects, individuals and properties as has already been explained in chapter 3.2. Further on it can be said that both knowledge acquisition and conceptualization should happen simultaneously. The acquisition of data and information should always be accompanied by a direct conceptualization. By this missing content can be avoided more efficiently.
As a next step, formalization describes a process of transfer of a conceptual model to a formal model. A formal model, further on, is a model which can be processed by machines and software systems. A formalization of a conceptual model happens by means of a formal ontology language as well as a modeling software tool. Major steps during formalization of ontology content are illustrated in Figure 8.

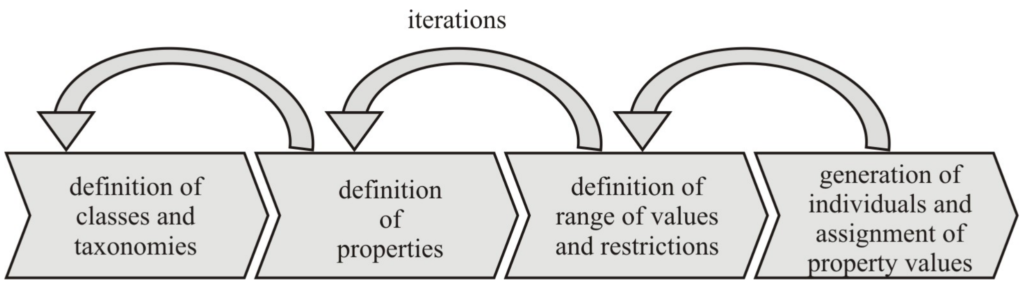

Figure 8. Ontology modeling process

This modeling procedure also has to be executed with regard to special criteria and rules. The following criteria are of major importance:

- Clarity: Unambiguousness in the definition of concepts
- Coherence: Unambiguousness in structure and organization of an ontology
- Extendibility: Possibility to easily vary and extend the structure and content of an ontology
- Minimal encoding bias: implementation of an ontology must be independent from its use
- Minimal ontological commitment: Ontological content must be without any judgmental aspects

These criteria are meant to help the development of distinct as well as complete ontologies and to prevent or minimize development faults.
Following formalization also the integration of already existing ontologies can be important. But this development phase has gained only little attention until now. This is due to a very high effort to combine two existing ontologies, which is called mapping [37] and inter-ontology mapping [38], or even to insert content from one ontology into another. Until now there is no recommended procedure for the integration process. Furthermore software tools, too, do not support this phase adequately.
Concluding, the developed ontology has to be evaluated and potentially worked over. The following five criteria are necessary preconditions for a verified as well as validated ontology:

- Consistency: ontological content must be free of contradictions in the context of interpretation
- Completeness: knowledge to a specific domain must be included explicitly or implicitly
- Conciseness: redundant declaration must be avoided
- Extensibility: it must be easy to extend an ontology with additional knowledge
- Stability: even extended and modified ontologies must stay consistent and feasible

The whole ontology development process has to be accompanied by the documentation of essential steps and results. This can be especially of interest for an efficient use as well as reuse of ontological content.

4.2 Languages for ontology engineering

Already in chapters 3.1 and 3.2 the most important description as well as metadata description languages have been introduced. Just to be able to use ontologies also in the Semantic web, a web compatible ontology language is needed. At the beginning of the 1990s several languages for ontology modeling have been developed but have not been suitable for the use in the Semantic web. These languages are e.g. FLogic (Frame Logic), KIF, Ontolingua and CycL. As a next step numerous languages emerged which could be used in the Semantic Web. The most important ones are SHOE, XOL, OML, OIL, DAML and DAML-OIL. But all these languages have not been standardized.
The Web Ontology Language OWL forms the end of the development of ontology languages for the time being and is based on DAML+OIL. OWL has the greatest expression strength of all ontology languages at the present time and permits an intensive use in the development of ontologies. The language has three different language levels which differ primarily in the expression strength and are explained in the following table.

Table 1. Language levels of OWL

OWL language level	Properties
OWL Lite	• contains the most essential language constructs for the definition of class hierarchies and simple relations • low expressional strength • insufficient for complete ontologies
OWL DL	• includes all important language elements • medium expressional strength • compromise between OWL Lite and OWL Full
OWL Full	• includes all language elements • maximum expressional strength • due to the complexity, perhaps all conclusions cannot be calculated

OWL Lite is suitable for the production of very simple ontologies in which the class hierarchy and the most essential relations between the concepts are included. The language level OWL Full represents the opposite to OWL Lite. With it the complete linguistic area of OWL can be used what permits "complete" ontologies with a maximum expressive power. However, this has the consequence that maybe not all decisions can be taken with the use of these ontologies. It is not guaranteed that a computer application, e.g. a software agent, can draw or calculate all logical conclusions.
With OWL DL (Description Logics) the ontology developer has a suitable combination from OWL Lite and OWL Full. The language level OWL DL represents a compromise between expressive power and processibility in finite time and provides the ontology developer with the use of the language constructs of OWL Full as well as OWL Lite. However, their use is set to limits. Major restrictions are:

- a class cannot be an individual or a property at the same time
- a property cannot be a class or an individual at the same time
- a property can be either only an object property or a data type property

Due to the standardization of OWL by the W3C, the language has been used also in the work at hand for the development of an ontology for the domain mechatronics. The language level OWL DL has already been characterized as a language, which enables on the one hand a development of a complex ontology and on the other hand a processing of the contents of the ontology by means of software agents in finite time.
For a computer-aided ontology development the choice of a modeling environment has been carried out mainly into dependence of the possibility of the use of OWL DL.

4.3 Software tools for ontology modeling

Computer-aided tools for ontology modeling make it possible to put knowledge down to a certain domain in a hierarchically structured form within an ontology. This process is supported by means of a graphic surface so that no direct ontology construction must be carried out in form of a source code. At present, numerous software tools are available. The most famous ones are the KAON Tool Suite, OntoStudio, Ontolingua, Protégé and WebOnto. The software tool Protégé is freely available and enables the development of ontologies with OWL. It has therefore been used for the elaboration of a domain ontology mechatronics.

5 DOMAIN ONTOLOGY MECHATRONICS

The domain ontology mechatronics [27] has been developed for the reason to contain basic knowledge to the domain mechatronics for the early phases in the development process. In general, there are decisive limits for the storage of design relevant knowledge within an ontology [39]. It can be said that only declarative or product and factual knowledge can completely be modelled ontologically. Due to the intuitive and subjective nature of informal, vague and incomplete knowledge it is not possible to include it into an ontology. Most important is the fact that it is not possible to model operative and process knowledge due to the "impossibility of algorithmisation of design processes" [39].

5.1 Basics

Within the domain ontology mechatronics product knowledge in the case of mechatronic systems is put down at two essential levels – system as well as behavioural model [27]. On the one hand the system model describes the system structure as a component dominated topology. This one results from the components involved as well as the relations between them. This is illustrated in Figure 9.

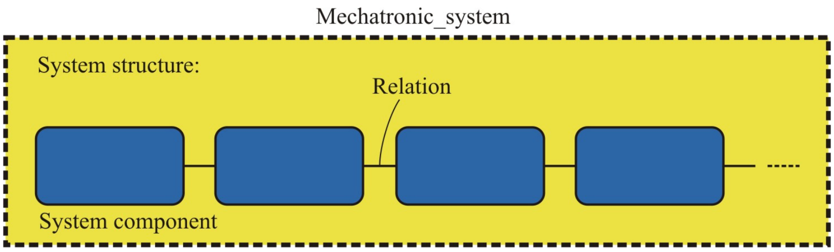

Figure 9. A system's structure in the domain ontology mechatronics

On the other hand the behavioural model is mathematically dominated. It is modelled in form of condition and differential equations including characteristic parameters. This modelling level is shown in Figure 10.

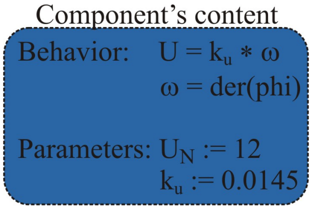

Figure 10. Behavioural model

By chaining all behavioural descriptions of the included components it results the behaviour of the complete system. To define the transmission of input and output quantities from one component to another, all system elements are split up into the two parts "contents" and "interface" already proposed in [40] as shown in Figure 11.

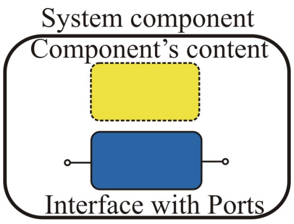

Figure 11. Parts of a system component

As already mentioned above the contents serve to describe an element's inner features. Further on the interface with ports enables the integration of single elements to a whole system. The two basic forms of interfaces as well as transmitted quantities are illustrated in Figure 12.

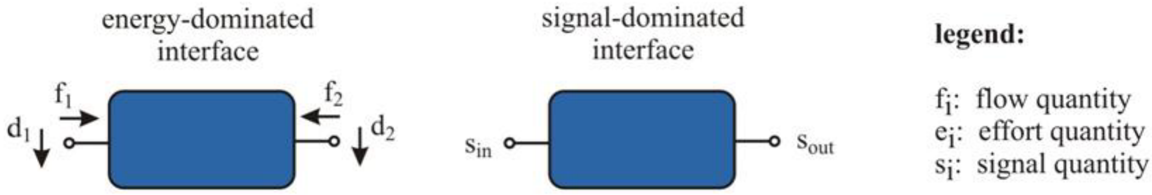

Figure 12. Interfaces and defined quantities

The interfaces as well as ports and transmitted quantities are based on the physical analogies between technical disciplines, the bond graph theory as well as numerous citations (e.g. [41, 42 and 43]). Also in the domain ontology mechatronics the generally accepted approach of effort and flow quantities of a component is pursued.

5.2 System model

Within the ontology a modeling is carried out at a working structure-like level. For the description of a mechatronic system basic elements are sensors, actuators, information processing into combination with a mostly mechanical basic system, necessarily consisting of so-called machine elements. The structuring of the subclasses of the class Actuator is carried out according to working principles points of view, that one of the classes Sensor and Machine_element from user view. A taxonomy for working and wirk elements, respectively, is given in extracts in Figure 13.

Figure 13. Working elements in the domain ontology mechatronics

Regarding the classification and hierarchical structuring of relevant working elements it can be stated that there are numerous possibilities. A knowledge domain can be modeled as well as formalized in many different forms and a final decision whether an ontology is good or not will never really be possible.

5.3 Modeling of mechatronic systems

The modeling of mechatronic systems in the domain ontology mechatronics is illustrated in extracts in Figure 14. It shows relevant knowledge to the domain mechatronics exemplified with the help of a direct current motor as part of an arbitrary mechatronic system.

Shown are classes (black), individuals (red) and properties (blue). Regarding a direct current motor it has been represented as a subclass of an electro dynamical converter which itself is a subclass of an electromechanical actuator. Furthermore this one is subclass of an actuator and further of working and wirk element, respectively. Additional necessary classes like interface, port and connection are shown, too.

A DC_motor_variant1 as individual of the class direct current motor is part of a Mechatronic_system1. The assembly of a whole system is achieved by means of object properties. A direct current motor has got an electrical to rotational interface with two ports. In this case it is an electrical energy port and a rotational mechanical energy port. These ports are especially responsible for the transmission of dimensions in form of effort and flow quantities. The electrical energy port e.g. enables the transmission of current and voltage as illustrated in Figure 14. The dimensions are further part of equations describing a working element's behaviour and are represented by constants as well as symbols and have a unit as well as abbreviation.

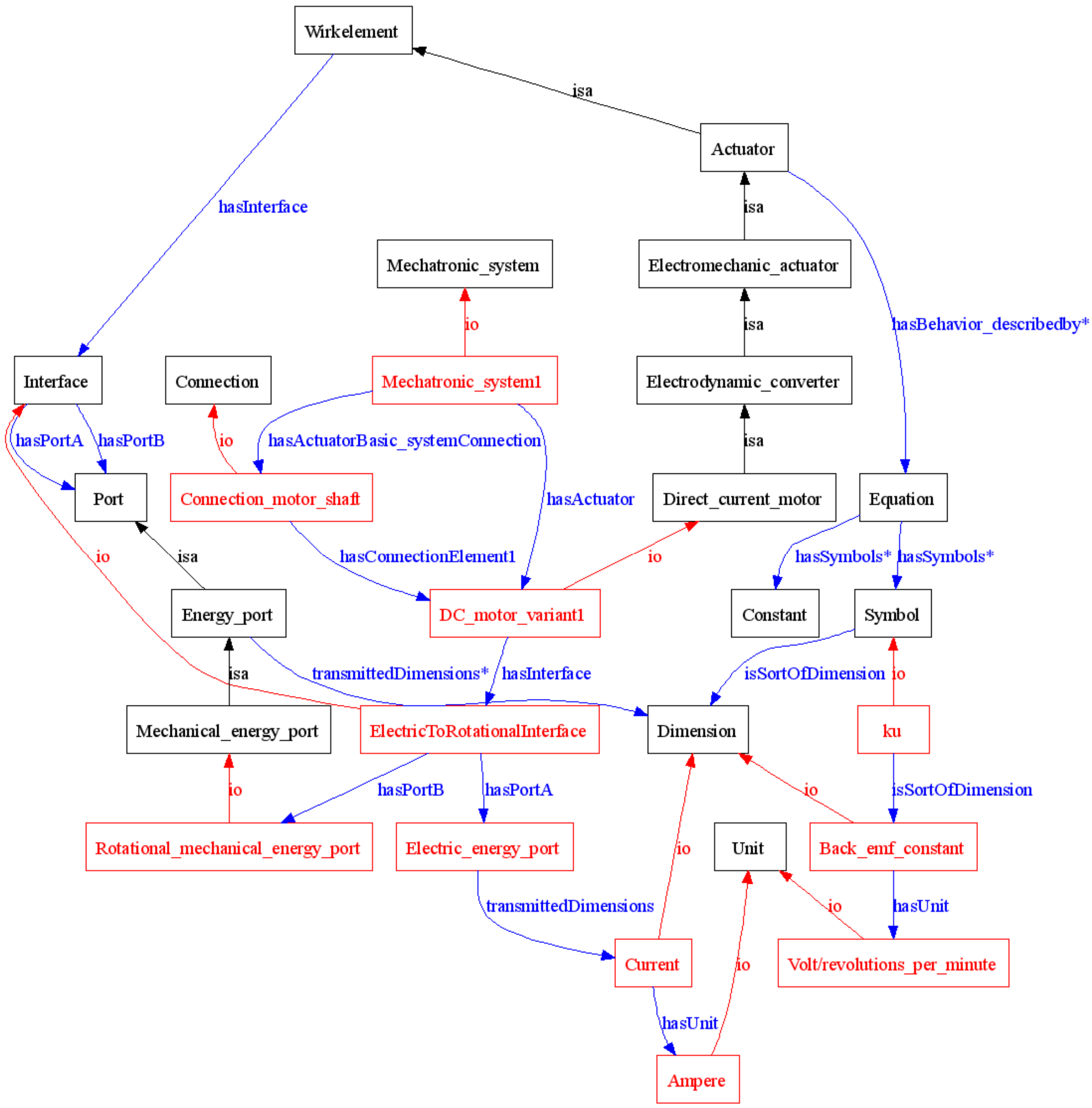

Figure 14. Extract from the domain ontology mechatronics

6 CASE STUDY

The functionality of the Semantic Web Service Platform SEMEC, its potential as well as its 'cooperation' with the domain ontology mechatronics shall be explained with the help of a case study already introduced in [1 and 3]. A team of developers creates a first concept of an actuator unit of a mechatronic system. The platform's task is to support this team in such a way that an alternative concept shall be developed by the software agent. It is only one possibility for the usage of the developed platform.

To specify a first concept the developers use the platform's design environment. They build up a first working structure consisting of different elements from diverse technological domains. This can be done with the help of working elements stored in a repository or by the creation of new ones. These working elements are annotated semantically. As a consequence a whole system results which, too, is annotated semantically. This first concept is at the agent's disposal. The agent can further on process the system according to its design context due to the fact that it contains metadata which is linked to the developed domain ontology mechatronics.

The agent is able to analyze the model and acts according to general action strategies put down in the platform's knowledge module. First of all it gains further information about mechatronic systems as well as the components involved in the model at hand. The most important information is given in the domain ontology mechatronics. Due to existing work steps, plans and meta-plans as well as their configuration the agent also takes restrictions and constraints into consideration. Further on given web pages and documents are analyzed for additional information. The generated and provided results can be optimized by an intensive dialog between SEMEC and developers as well as the cooperation of SEMEC with other Semantic web services and software agents.

In the case study at hand the acquirement of an alternative concept is carried out in the simplest case by the use of the physical input as well as output quantities of the individual components. Firs of all a system's components as well as relations are determined. This is illustrated in Figure 15 in case of involved components.

```
<Mechatronic_system rdf:ID="System">
<hasComponents rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
    DC_motor
    Shaft
    Gear
    Spur_rack
    Sled
    Linear_guides_without_lubricant
</hasComponents>
</Mechatronic_system>
```

Figure 15. Working structure in XML

Further on these components' interfaces as well as ports are of interest as illustrated in Figure 16.

```
<Direct_current_motor rdf:ID="DC_motor">
    <hasName rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >DC_motor</hasName>
    <hasInterface rdf:resource="#ElectricToRotationalInterface"/>
```

Figure 16. Determination of an element's interface

By this, an alternative concept is generated by the analysis of input and output quantities of alternative components. One possible new concept is introduced to the developers within the design environment. This can look as follows:

```
PROCESSOR:
D://Projekte//Agent//Agent_G1_0004//plan//alternative_1onN.xml
-> Plan alternative_1onN.xml started!
-> result:    Linear_motor    Sled    Linear_guides_without_lubricant
-> Plan alternative_1onN.xml ended!
```

Figure 17. Result achieved by means of SEMEC

7 SUMMARY

Computer application for knowledge-based design with an improved interoperability with the internet/ Semantic web will have a promising future. To enable a better interaction of software systems and the Semantic web especially ontologies are of major importance. Their use in the integration of data, information as well as knowledge throughout the complete design cycle of a product has the potential to improve both the products and the processes substantially.

REFERENCES

[1] Bludau C. and Welp E.G. Semantic web services for the knowledge-based design of mechatronic systems, In *International Conference on Engineering Design, ICED 2005,* Melbourne, 2005.

[2] Welp E.G., Bludau C. and Labenda, P. Nutzung von Ontologien zum Wissens- und Produktdatenmanagement in der Entwicklung mechatronischer Systeme, *Tagungsband Informations- und Wissensdrehscheibe Produktdatenmanagement,* GITO-Verlag Berlin, 2005.

[3] Bludau C. and Welp E.G. Semantic Web Services für den wissensbasierten Entwurf mechatronischer Systeme, *Tagungsband 3. Paderborner Workshop „Intelligente mechatronische Systeme",* Heinz Nixdorf Institute, University Paderborn, 2005.

[4] VDI 2206. *Entwicklungsmethodik für mechatronische Systeme, Design methodology for mechatronic systems*, Berlin: Beuth Verlag, 2004.

[5] Gausemeier J., Ebbesmeyer P. and Kallmeyer F. *Produktinnovation - Strategische Planung und Entwicklung der Produkte von morgen*, Hanser Verlag, 2001.

[6] Isermann R. *Mechatronische Systeme – Grundlagen*, Springer - Verlag, Berlin, 2002.

[7] Hahn M. CAMeL-View: Ein Werkzeug für den Entwurf mechatronischer Systeme, *4. Sitzung des Arbeitskreises Mechatronik an der FH Kiel*, 2001.

[8] Otter M. and Olsson H. New Features in Modelica 2.0, In *Proceedings for 2nd International Modelica Conference*, Oberpfaffenhofen, DLR, pp. 7-1 – 12, 2002.

[9] Welp E.G. and Bludau C. Knowledge-Based Development of Mechatronic Products, In *Proceedings of DETC'02*, ASME, Montréal, Paper-No. CIE-34459, 2002.

[10] Porter I., Counsell J. and Shao J. *Schemebuilder Mechatronics – Modelling and Simulation*, Lancaster, Brunel University, 1998.

[11] Probst G., Raub S. and Romhardt K. *Wissen managen*, Gabler-Verlag, 3. Auflage, 1999.

[12] Abramovici M. and Sieg O. PDM-Technologie im Wandel – Stand und Entwicklungsperspektiven, In *Industrie Management 5*, Berlin, GITO-Verlag, 2001.

[13] Hahn A. Kollaboratives Wissensmanagement in der Produktentwicklung, In *Tagungsbeiträge zur 4. Oldenburger Fachtagung Wissensmanagement*, Bremen, 2003.

[14] Rude S. *Wissensbasiertes Konstruieren*, Karlsruhe University, Faculty for Mechanical Engineering, Habilitation, Shaker-Verlag, Aachen, 1998.

[15] Klein R. and Gero J. S. (ed.) Knowledge Modeling in Design – the MOKA framework, In *Proceedings of the International AI in Design Conference Worcester*, MA, Kluwer, 2000.

[16] Hochgeladen R. *Beherrschung komplexer assoziativ-parametrischer Strukturen im CAD*, 2001.

[17] Franke H. J. and Lux S. Gadgets, wissensbasierte aktive Entwurfselemente, In *VDI Berichte zur VDI-Tagung „Verkürzte Entwicklungsprozesse durch Integration von Gestaltung und Berechnung"*, VDI Berichte 1487, VDI-Verlag, 1999.

[18] Vajna S. Approaches of Knowledge-based Design, In *Proceedings of the International Design Conference – Design 2002*, Dubrovnik, Croatia, 2002.

[19] Berners-Lee T., Hendler J. and Lassila O. The Semantic Web, *Scientific American,* 2001.

[20] Stollberg M. *Ontologiebasierte Wissensmodellierung – Verwendung als semantischer Grundbaustein des Semantic Web*, Master Thesis, Free University Berlin, 2002.

[21] Roth S. Ontologien im Information Retrieval, *Seminar Angewandtes Information Retrieval*, University Basel, 2004.

[22] Broekstra J., Klein M., Decker S., Fensel D. and Horrocks I. Adding Formal Semantics to the Web: Building on Top of RDF Schema, *Proceedings SemWeb*, 2000.

[23] Bruijn de J. *Using Ontologies – Enabling Knowledge Sharing and Reuse on the Semantic Web*, DERI – Digital Enterprise Research Institute, Technical Report, 2003.

[24] Musen M.A. Dimensions of knowledge sharing and reuse, *Computers and Biomedical Research 25*, pp. 435 – 467, 1992.

[25] Borst W.N. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*, Doctoral Thesis, Centre for Telematics and Information Technology, Netherlands, 1997.
[26] Fensel D. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*, Springer Verlag, 2000.
[27] Labenda P. *Zusammenstellung des im Kontext mechatronischer Entwurfsprozesse relevanten Basiswissens zur Domäne Mechatronik sowie dessen Überführung in eine Ontologie*, Diploma thesis, Institute of Engineering Design, Ruhr-University Bochum, 2005.
[28] Angele J. and Oppermann H. *Semantic PLM – Next Generation Product Lifecycle Management*, ontoprise, Whitepaper Series, 2005.
[29] Gausemeier J. Von der Mechatronik zur Selbstoptimierung, In *20th CAD-FEM Users' Meeting*, 2002.
[30] Lenat D.B., Guha R.V. *Building large knowledge-based systems,* Addision-Wesley Publishing Company, Inc. 1990.
[31] Uschold M., Grüninger M. Ontologies: Principles, Methods and Applications, In *Knowledge Engineering Review 11(2)*, 1996.
[32] Grüninger M., Fox M.S. Methodology for the design and evaluation of ontologies, In *Workshop on Basic Ontological Issues in Knowledge Sharing*, Montreal, Canada, 1995.
[33] Bernaras A., Laresgoiti I., Corera J. Building and Reusing Ontologies for Electrical Network Applications, In *Proceedings of the European Conference on Artificial Intelligence*, ECAI 96, 1996.
[34] Fernández M., Gómez-Pérez A., Juristo N. METHONTOLOGY: From Ontological Art Towards Ontological Engineering, In *Symposium on Ontological Engineering of AAAI*, Stanford, California, 1997.
[35] Swartout B., Ramesh P., Knight K., Russ T. Toward Distributed Use of Large-Scale Ontologies, In *Symposium on Ontological Engineering of AAAI*, Stanford, California, 1997.
[36] Staab S., Schnurr H.-P., Studer R., Sure Y. Knowledge Processes and Ontologies, In *IEEE Intelligent Systems, 16(1)*, 2001.
[37] Schmaltz R.: *Semantic Web Technologien für das Wissensmanagement,* Georg-August-Universität Göttingen, Institut für Wirtschaftsinformatik, Arbeitsbericht Nr. 1, 2004.
[38] Wache H.: Ontology-Based Information Integration – A Survey of Existing Approaches, In *Proceedings of the Workshop Ontologies and Information Sharing*, IJCAI, 2001.
[39] Braun P. *Objektorientierte Wissensarchivierung und -verarbeitung in modellassoziierten Gestaltungs- und Berechnungssystemen*, Doctoral thesis, Ruhr-University Bochum, 1999.
[40] Lippold C. *Eine domänenübergreifende Konzeptionsumgebung für die Entwicklung mechatronischer Systeme*, Doctoral thesis, Ruhr-University Bochum, 2000.
[41] Schönfeld R. Leistungsfluss und Signalfluss in mechatronischen Systemen, *VDI-Berichte Nr. 1842*, 2004.
[42] Böhme S. and Haase J. Netzwerkmodellierung für heterogene Systeme, *VDI-Berichte Nr. 1842*, 2004.
[43] Otter M. and Schweiger C. Modellierung mechatronischer Systeme mit Modelica, *VDI-Berichte Nr. 1842*, 2004.

Contact: Patrick Labenda
Ruhr-University Bochum
Institute of Engineering Design
Universitätsstr. 150
44801 Bochum
Germany
Tel.: Int +49 234 32-24054
Fax: Int +49 234 32-14159
E-mail: labenda@lmk.rub.de
URL: http://www.lmk.ruhr-uni-bochum.de/