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

COMPARISON OF KNOWLEDGE REPRESENTATION IN PDM AND BY SEMANTIC NETWORKS

J. Conrad[1], T. Deubel[1], C. Köhler[1], S. Wanke[1] and C. Weber[2]

[1] Institute of Engineering Design/CAD, Saarland University, Germany
[2] Engineering Design, TU Ilmenau, Germany

ABSTRACT

"Nowadays, computer-aided tools have enabled the creation of electronic design documents on an unprecedented scale, while determining and finding what can be reused for a new design is like searching for a 'needle in a haystack'. (…) The availability of such extensive knowledge resources is creating new challenges as well as opportunities for research on how to retrieve and reuse the knowledge from existing designs." [1] If the requested knowledge is implicit (which means that it is only in the minds of the employees of a company) the retrieval and reuse of knowledge is even more complicated.
By representing the (engineering) data backbone of a company, PDM systems are the software implementation which should support the designer to retrieve information about existing and successful design projects. This paper shows that the known data classification approaches of common PDM systems are not applicable to represent implicit (tacit) knowledge. Furthermore a new approach to knowledge representation is introduced by using Semantic Networks.
The feasibility of the presented work is shown by a use-case scenario in which the conventional PDM system supported product development process is compared with the proposed way by using the software "The Semaril" – a software tool developed at the Institute of Engineering Design/CAD based on Semantic Networks [2].

Keywords: PDM, Semantic Networks, Knowledge Representation, The Semaril

1 INTRODUCTION

"Time to Market" is one of the key factors in product development and manufacturing [3]. In this context the reuse of validated solutions is one way to shorten the development time by avoiding redundant work. This implies yet fast and efficient data, information and especially knowledge retrieval. But the reuse of knowledge is associated with a high complexity and hard traceable reasoning structures. Additionally, it is often required to "read between the lines" for retrieving desired knowledge. The main issue of the traceability is the way how knowledge is structured and kept. So the reusability of knowledge fluctuates with the underlying knowledge representation.
Focus of this paper is to compare the knowledge representation in PDM systems and by systems based on Semantic Networks. Aim is not to substitute PDM systems with all their functionalities by Semantic Networks but to show that Semantic Networks are more appropriate from the human point of view to represent knowledge.
Even though this topic is touched by the way in projects (for example the *n*-dim project [4], [5]) with other emphases, this paper shows focused how a semantic tool like The Semaril can be utilised to represent extended product knowledge which allows an almost context-independent retrieval of knowledge, which is impossible in hierarchical architectures[1]. It provides the opportunity to store so far inexplicit data explicitly.

2 PDM -SYSTEMS AS KNOWLEDGE BASE – STATE-OF-THE-ART

Since the product development process is increasingly computer-supported, the amount of engineering data and information generated in the product development process and at the same time the number

[1] This doesn't mean the information technology point of view.

and complexity of relations between data elements are continuously increasing [6]. So there is a need to store the data and information elements and their relations systematically and to manage this data and information base in an effective way. For this task Product Data Management (PDM) systems have been developed and have become state-of-the-art in industry.
One of the core functionalities of a PDM system is to make (product and process) information transparent within the whole company. "Transparent" means that stored information should be retrievable and reusable. "One reason for reusing information is to avoid the cost and the time of repeating a task that has already been carried out. When information is created, it goes through a process of specification, development, production, test, modification, use, storage, etc. The process usually includes several quality checks to ensure that the information is correct. (…). The overall process takes time, and costs money." [7]
By introducing classification systems, the developers of PDM-software try to reach this transparency with focus on the reuse of information. Proper classification should help the product developer to find, recombine or to adapt already existing and approved solutions. The classification of objects within a company is desirable but the complexity (acquisition, classification, value coding and maintenance) is highly rising depending on the number of objects to be classified.
Below we show and describe which kinds of classification systems exist and how they can help to retrieve information.

2.1 Classes of classification systems

The aggregation of objects to groups by using their similarities affords a classification. Classification systems support the process to transform an unordered group of objects to an ordered. There are three classes of classification systems which will be described and evaluated below (according to [8], [9]):

- Ahierarchical Systems:
  Ahierarchical systems are independent combinations of information. The terms are coequal and without an inherent structure but they can be assorted e.g. alphabetically. Examples are alphabetically sorted subject catalogues, random, scheduler composition objects or the direct access on parts of text within documents (full text retrieval). The advantage of this classification system is the complete acquisition of objects. The disadvantage is the need of enormous memory requirements and computing time.
- Partial-hierarchical systems:
  By adding order characteristics partial-hierarchical systems provide the opportunity to make a first grouping of objects according to content-related criteria. So there is the possibility to select groups by needed properties. One example for this category of systems is a Thesaurus.
- Hierarchical systems:
  Hierarchical systems possess a strict hierarchical structure. By the way of offering information two kinds of systems are distinguishable: manually and automatically coding systems. Numbering systems such as parallel numbering system are the most common manual coding systems. Here an alphanumerical coding of objects is given by a certain scheme. This system is advantageous for a short, clear representation of information, but in consequence of the limitation of digits there is just a poor significance of number coding possible.

In industry the hierarchical systems are the most widely spread.

2.2 Criteria for information classification

There are different criteria on how engineering objects can be classified. The most common ones are introduced below:

- Functional classification:
  Objects are classified by their functional reference, e.g. the chassis in vehicle development. This way of classification has a high usability because the chosen terms are simple to interpret and easy to associate with a company's units. The main disadvantage is that it is not possible to associate most of the objects clearly with just one function. In consequence it is inescapable that there are redundancies.
- Form oriented (geometrical) classification:
  The criterion for classification is geared to the form and to the production technology. Due to the

multiplicity of different forms there is a very abstract form key, which leads to a very general classification. Due to this and the fact that most objects of business units like sales, product management or buying are function oriented, the usability and the acceptance of this classification approach is low.

- Manufacturing oriented classification:
  In this case objects are grouped by the manufacturing method they are generated with. This approach is used in business sections right before production.

2.3 Data- and knowledge-providing in PDM systems

"Several (…) studies have found that accessibility is the factor that most influenced the engineers' selection of information sources. Accessibility is understood as a subjective measure of the effort that a designer needs to make in order to access such an information source" [10]. So the representation of information is very important for the retrieval of information and the acceptance of tools to support the information retrieval process.

Current PDM software represents the product data typically in static and hierarchical structures which are often generated at the beginning of a development project (see Figure 1).

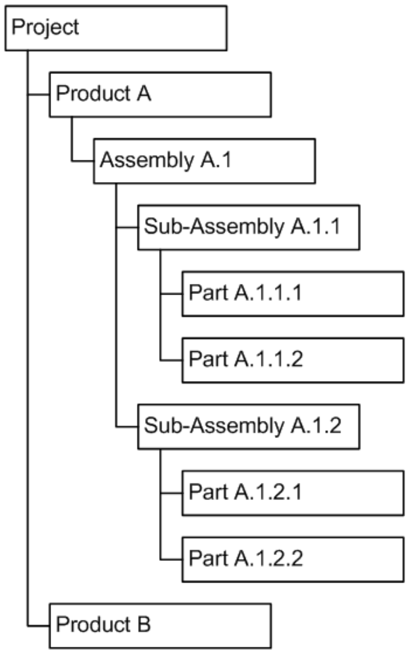

Figure 1: General provision of project structures in a PDM-System

The classification methods shown above are advantageous for the implementation in computer systems. They are computer-oriented and not user-centred. Thus, there is a gap between the present system architectures – or rather information representations – and the applicability (information providing) of the systems. Humans do not think exclusively in hierarchical structures, they think in an associative way [11], [12], [13]. Hierarchical PDM systems force the user to store documents and other data context-specifically. This causes redundancies in the hierarchy tree when the same object is used in different contexts, inhibits an efficient information retrieval and, in consequence, the reuse of information.

Another disadvantage of common PDM-systems is the general lack of representation of (product) knowledge. Knowledge contained in product data is not captured explicitly and so hard to retrieve. So a reuse of (implicit) knowledge is really difficult. This could be connected to the criticism mentioned above, as there is even evidence that hierarchical structures are not at all suitable to capture knowledge [14].

Nevertheless, common PDM-software is suitable to prevent information islands and it is established as data backbone in a lot of companies.

3 KNOWLEDGE REPRESENTATION

3.1 Fundamentals

"Knowledge is distinguished in implicit and explicit knowledge. Implicit (tacit) knowledge exceeds (…) clearly the amount of explicit, codable knowledge which can be extracted as subset from the implicit knowledge" [15]. According to Ogden et al. [16], the process of representing the meaning of objects can be pictured in a triangle of meaning (see Figure 2). Physical objects are connected with its context and a representing model. The context of an object is more than the object itself and includes all senses that are connected with it [17].

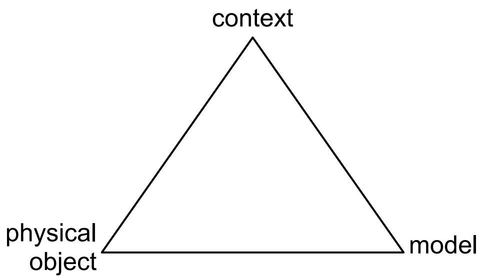

Figure 2: Triangle of meaning (according to [17])

For a company acquired knowledge is only profitable if it is used in the value-added process [18]. To realize this, knowledge created once should be kept and be available in the entire company. Knowledge Representation (KR) serves for modelling knowledge, to allow exchange, storing and re-use. This is done with common knowledge bases like the introduced PDM systems. But generally, most of a company's knowledge is distributed in the minds of the employees and therefore hard to catch with conventional computer systems. Personal knowledge, i.e. knowledge that is not electronic or paper-based, is only implicitly available (out of the company's view). The personal knowledge of a person X is not available for a person Y and it is also not available for the company at all. This fact is visualized in Figure 3 where implicit knowledge is especially shown by the areas A and B.

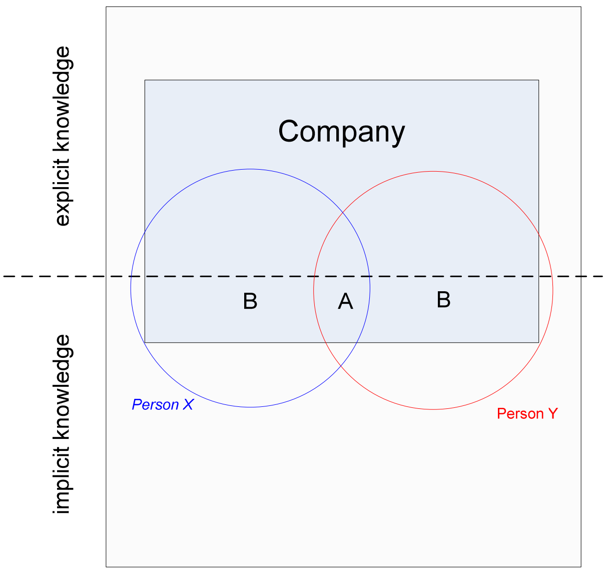

Figure 3: Classification of personal knowledge (according to [19])

Therefore, the representation of implicit knowledge (which is in literature sometimes also called "unprintable mental knowledge", [17]) is the main challenge of knowledge representation.
The recording of models whose counterparts in the real world are known, is according to Brachmann and Levesque [20] the quintessence of the KR. There exist a lot of approaches to represent knowledge.

The easiest ones are catalogues, glossaries or taxonomies [1]. More powerful are bases like thesauri or classifications. Semantic Networks, ontologies, frames or first order logics are even more powerful.

3.2 Semantic Networks

As in this paper an approach is introduced to represent implicit knowledge by Semantic Networks, the fundamentals of Semantic Networks are introduced below.
Semantic Networks were firstly introduced by Charles Peirce. He proposed a graphical notation of nodes and arcs, called existential graphs, which were later renamed to "Semantic Networks" [21]. Peirce's research results were, among others, picked up by Tim Berners-Lee in 1994 [22] when he introduced the Semantic Web (based on Semantic Networks) as an evolution of the current world wide web. Bernes-Lee wants to represent knowledge in a more associative, at the same time computer-processable way. The burgeoning Semantic Web comprises newly created and/or transformed web data sources endowed with computer processable meaning (semantics) which should help the humans to find and reuse data and knowledge easier.

3.2.1 Structure and Attributes of Semantic Networks

There are many variants of Semantic Networks, but all are capable of representing individual objects, categories of objects, and relations among them. According to Helbig et al. [23] a Semantic Network is "the mathematical model of a conceptual structure consisting of a set of concepts and the cognitive relations between them". It can be represented by a generalised graph where concepts or objects are visualised by nodes and the relations by labelled edges (Figure 4).
The nodes do not wear any knowledge. The knowledge is represented by the links, which start from the nodes.

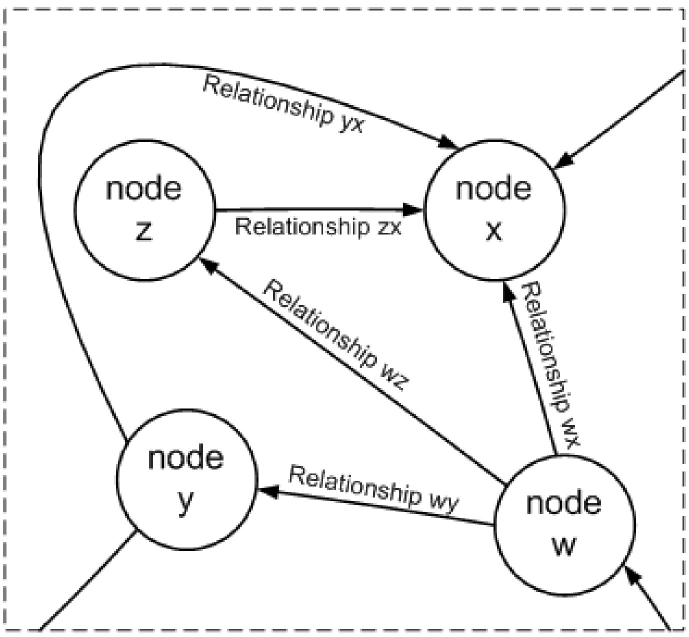

Figure 4: Structure of a Semantic Network

The main difference between simple networks and Semantic Networks is the kind of relations which combine the individual nodes. The relations bear a meaning by itself. This fact is a huge advantage of this data architecture, because software applications, e.g. search engines, are able to understand and process the relations. Especially abstraction relations (e.g. "is part of" or "is sub-assembly of") or inheritance mechanisms known from object oriented programming (e.g. "is subclass of") are very easy to represent.

3.2.2 Possible field of application and realisations

By the universality of the concept of Semantic Networks they can be used everywhere where an amount of data can be classified by meaningful relations.
According to [24] Semantic Networks combines the cognitive psychology with computer science. So, Semantic Networks are especially a means to design user interfaces between computers and users

more intuitively. The ease-of-use can be raised fundamentally. Particularly queries are supported very efficiently, because the query criteria are focused on the human thinking not on the computer processing.
There are already some software applications based on the Semantic Network technology like The Semaril (see section 4.2), Internet search engines like the Getess project [25], the Conweaver project [26] or the TAP system from Stanford [27], just to call some representatives.

4 AVAILABLE SOFTWARE TOOLS

To clarify the benefits of knowledge retrieval and reuse by Semantic Networks compared with PDM-systems, two software solutions were chosen which will be introduced here briefly.

4.1 PDM-system

To instance a widespread PDM-system, SmarTeam has been selected. SmarTeam has a common architecture like other PDM-systems using a vault server to store documents (user data) and a database to store the metadata (part of which is the relations between the data entries). So in principle an extrapolation to other PDM-systems seems to be possible. The provision of data is ordered in hierarchical structures as mentioned in section 2.
SmarTeam provides integrations to office products and software tools specific for the product development domain like CAD-systems. Thus, SmarTeam offers all core functionalities of a modern PDM-System (according to [6]):

- Master and structure data management
- Document management
- Group technique, classification
- Project management
- Workflow management
- Release and access authorisation mechanisms
- Change and version/revision management
- Viewing and redlining
- Input/output management
- Publishing
- Archiving

4.2 The Semaril

The software “The Semaril” [2] is a knowledge management tool based on Semantic Networks. It has been developed at the Institute of Engineering Design/CAD of Saarland University. The name is a made-up word and has no further meaning or translation. The General User Interface (GUI) and the main work areas are described in [2].

The data handling in The Semaril is based on five elements:

- Terms:
  A term can represent everything. It is similar to an entity in a conventional database. Technically, it consists of a name and an optional text. This text can also contain hyperlinks to other terms, files on a hard disk or websites. An “is-a” relation is attached to each term telling something about its meaning. So every term is part of an abstraction level hierarchy.
  A term can be linked to other terms by various relation types (“is prerequisite for“, ”conflicts with“, ...)
- Relations between terms:
  Terms can be linked to each other by means of relations. The Semaril offers some important relation types by default; the types are freely editable and extendible.
- Notes:
  Notes are highly similar to terms, but they can be attached only to (one or more) terms.
- Groups of terms (“bags“):
  Bags are used for mass operations on terms. Different Bags can be combined by the set operations unification, subtraction and intersection. The combinations of these few basic mechanisms allow very complex queries. These queries can be recorded with a macro recorder

and can be re-evaluated when the database contents change. Bags also enable the user to cognize and manipulate relations all elements in a bag have in common. This offers a very comfortable and quick way for data maintenance. The contents of bags can be used to create situation-specific reports in form of HTML files containing notes and relations.

- Filters:
  Filters are used for mass operations on terms. They can filter those terms which has a defined set of relations. Filters can be used to search the whole Semantic Network or a special bag.

"Because of its flexibility in relation to the uncommitted definition of entities and relations as well as to the possibilities of establishing links to other data, The Semaril is a powerful tool for the modelling and the management of knowledge" [2].

5 USE-CASE SCEANRIO

The following section compares the conventional way of representing product data with a tool based on Semantic Networks. The use-case should exemplify the difference of a conventional and a semantic-oriented knowledge representation.

5.1 Initial situation

A company develops pens since several years. The main series are ball pens with retractable points. To make the use of this mechanism comfortable, the company did a lot of ergonomic research to level the optimal hand force for optimal handling.
Several years later, the company decides to develop and produce flush-mounting light switches. During the development project the product developers have to design the core mechanism to operate this switch. It shall be manually operated, and so it is necessary to define (and later realise) the corresponding force for it.

5.2 The conventional situation with a common PDM-system

At the beginning the developers of the switch have the problem to invent what is the appropriate hand force to operate the switch comfortably. There are different ways of information retrieval to reach a solution for this problem like searching for ergonomic norms. By a stroke of luck someone of the developing team remembers that something similar was considered already at the pen project. They search the project data in the PDM-system and open the pen project. If the project is classified by a hierarchical classification system, especially by a numbering system, it is hard to find the right project if there is no acknowledgment about the taxonomy of the numbers used.

If the query for the project succeeds it is possible to scan the product data to find the desired information (Figure 5). A lot of knowledge is not explicitly represented in the PDM data stock (for instance the hand force) and is therefore hard to extract. In this case it could be possible to recognize that the spring (by the spring constant and the spring travel which are at best stored in additional documents) somehow contains – considering the respective mechanism – the not explicitly represented hand force. Or in the PDM-system they find the name of somebody who was responsible for the ergonomics at the pen project. If this person is traceable, the desired information could perhaps be obtained out of his or her personal knowledge in a few seconds.
So in the existing landscape of PDM-systems the main addressed function, the reuse of already acquired knowledge, is just fulfilled in case of part or, at least, geometry reuse. In case of non-geometrical information like behavioural aspects, PDM-systems store it in best case implicitly.

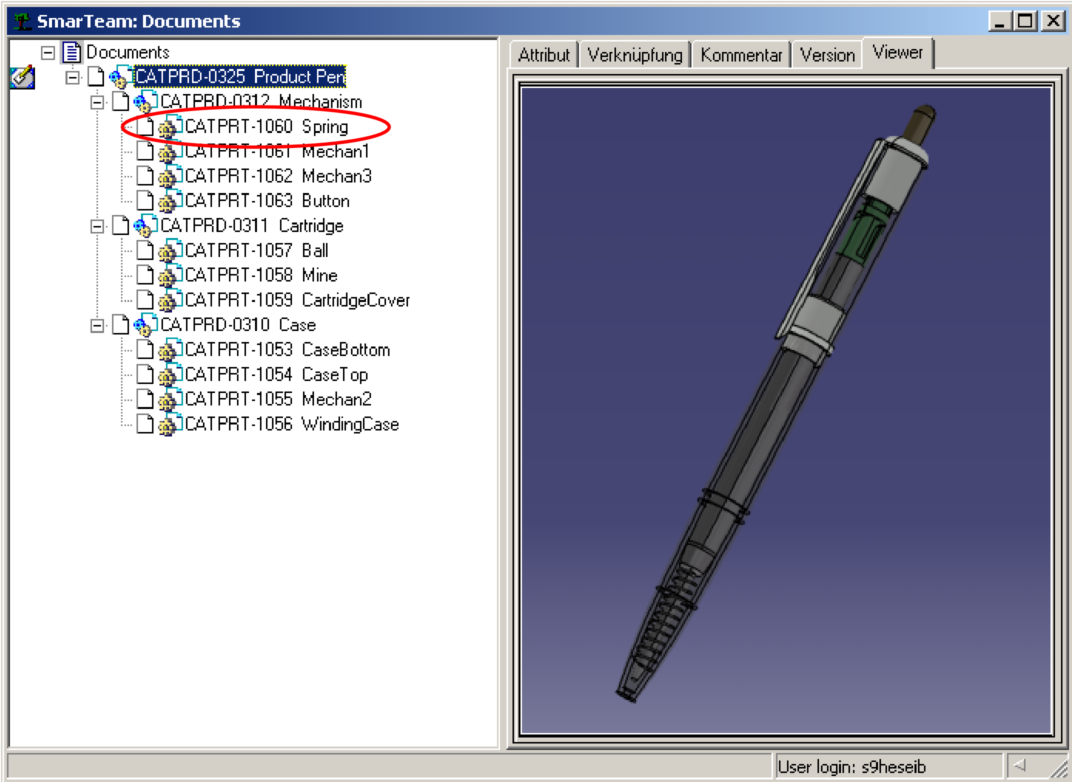

Figure 5: Representation of the pen project in a PDM-system

5.3 Opportunities offered by Semantic Networks

If the same company manages its knowledge by a Semantic Network, it develops a net similar to the one shown in Figure 6 for the situation mentioned above. This visualisation of the network is just an abstract of a whole product development project, shown here simplified to clarify the essential relations. The visualisation itself is not essentially part of the designing of Semantic Networks and only serves illustration purposes.

In a Semantic Network hierarchies can be easily modelled by using the relation "is a", "is part of" or "is subassembly of" (see figure 7). Additionally there is the possibility to attach nodes with further information. In this use-case knowledge about the ergonomic reasonable hand force is modelled into the Semantic Network and is connected by meaningful relations. The relation "depends of" is quite general but there is no limitation of the granularity and the level of abstraction. This arc could also be divided into a sub network with nodes and meaningful relations.

For using Semantic Networks it is necessary to evolve the product data in a semantic tool. The Semaril enables the filing of (product-) structures and hyperlinks to files. Every item inserted in The Semaril is assigned to a core category. By developing the product structure, more and more items and relations are recorded in the Semantic Network. Relation types are free to define; in this use-case the implemented relation types correspond to those of Figure 6. The final network is a very complex, hardly human-readable construct but The Semaril by its powerful filter-mechanism enables to provide desired extracts of the whole net.