Development of a Knowledge Mapping Approach for Independent Knowledge Elicitation and Representation

Helena Hashemi Farzaneh[1], Alexander U. Reik[1] and Maik Maurer[1]

[1]*Institute of Product Development, Technische Universität München,
Boltzmannstr.15, 85748 Garching, Germany
hashemi@pe.mw.tum.de*

Abstract

For the development of innovative technical products specialised knowledge is becoming increasingly important. Knowledge can be transferred from external sources and company-internally between engineers. Internal knowledge transfer is gaining importance due to a growing fluctuation of engineers possessing specialised knowledge. To enable knowledge transfer between engineers, the knowledge has to be elicited and represented. However, knowledge elicitation and representation are time-consuming processes, since the engineers are often unaware of their knowledge and have to be guided through the process by a moderator. The aim of this work is to develop a knowledge mapping approach which enables the engineer to independently elicit and represent his knowledge. We analyse results from a case study and verify findings from psychology, knowledge management and technical product development. Based on this analysis, we combine elements from existing knowledge elicitation and representation methods to a 3-phases concept for independent knowledge elicitation and representation which does not necessitate a moderator.

1. Introduction

Knowledge transfer is a driver of innovation [1]. This is especially true for technology companies which profit from the evolvement of new technologies to develop innovative technical products. In addition to knowledge transfer from external sources such as research partnerships with universities, internal knowledge transfer represents a competitive advantage for companies [2]. This expertise is often stored "in the heads" of a few engineers. Still, there is a growing fluctuation of engineers due to job changes and retirements [3]. Additionally, increasing international product development projects require the transfer of expertise between engineers working for the same company in different locations. Both fluctuation and international collaboration necessitate an intensified knowledge transfer. It has to be efficient and effective to so that it can be integrated into the companies' processes [4].

To be able to transfer knowledge, engineers first have to elicit their knowledge. In knowledge management, a number of methods exist for knowledge elicitation. However, they are often time-consuming and require a moderator (see section 2). According to [5], knowledge management methods do not ensure the usefulness and validity of the captured knowledge and could profit from theories from psychology on knowledge organization in the human brain.
In order to visualise the "knowledge about knowledge", knowledge maps showing different types of elements and relations can be used [6, 7]. They enable the communication and transfer of knowledge to other persons [6].

The aim of this work is to develop a knowledge map for representing an engineer's relevant knowledge. Furthermore, we develop a concept for a knowledge elicitation method which an engineer can use independently without the support of a moderator to "fill" his knowledge map.

To start with, we give an overview on existing knowledge elicitation and representation methods. Then, we develop the knowledge map based on domains and relations in product development, psychology and knowledge management research and the analysis of knowledge elements acquired in a case study.
In the following, we conceptualise a 3-phase method for the elicitation of knowledge by combining elements from several knowledge elicitation methods. To conclude, we discuss the developed knowledge map and elicitation method and provide an outlook on future steps.

2. Literature review: Knowledge elicitation and representation methods

Knowledge elicitation methods are used to capture a person's knowledge. In the following, we present a number of knowledge elicitation methods. The majority is performed with the support of a moderator who guides the person through the method.

Different types of **interviews** and different types of questions address different types of knowledge. For example an "episodic" interview aims at stimulating the interviewed person to bring specific situations to mind [5]. Similar to the interview, a **questionnaire** poses questions, but it has predefined questions which leave less freedom for the person to express additional thoughts [8].
In **brainstorming** a person is asked to elicit as many ideas as possible without evaluation. However, according to [9] it is less useful for knowledge elicitation than for knowledge creation. **Sorting** of cards with concepts predefined by the moderator or the person himself is used to structure knowledge and to compare the structure of different persons [5].
In **storytelling** or lecturing, the person is asked to phrase knowledge by means of a metaphor or remembered situation. Storytelling is also used for direct knowledge transfer to other persons. However, storytelling requires expertise and experience from the "storytelling" person [9].
A method, which simulates the working routine of several persons, is the use of **role games**. The moderator needs outstanding capabilities, because he has to design the scenario and roles of each participant and analyse the outcomes [9].
To finish, a person's knowledge can be inferred from **observation** of his working routine. He can be asked to **think-aloud** to provide an "insight" into his thoughts. However, thinking aloud contains irrelevant knowledge and not all relevant knowledge. In addition, when analysing the captured "thoughts", there is a high deviation depending on the analysing person [5].

To conclude, existing knowledge elicitation methods provide a number of elements facilitation the elicitation, such as focussed questions (interview, questionnaire), free flow of ideas (brainstorming) structuring the knowledge (sorting) and putting the person into his daily routine (role game, observation and think-aloud method).

Except sorting of cards, none of the elicitation methods integrate the visualisation or representation of the knowledge elements into the elicitation process. We define knowledge elements as the "knowledge about knowledge" according to [6]. For example, "manufacturing techniques" in Table 1 is a knowledge element. The knowledge element points towards the knowledge content, in this case the knowledge about manufacturing techniques such as welding, drilling etc. Knowledge elements can be visualised in **structured text and tables**, **sketches and images** as visual metaphors and **graphs** [7]. A particular type of graph is the knowledge map. According to [6] many knowledge maps are based on models of the human memory in cognitive psychology and evolved to practical knowledge management tools. Knowledge maps represent knowledge elements belonging to certain domains and their relations. This structure facilitates their implementation in software systems such as graph editors or ontologies,which enable reasoning and analysing with software tools. Domains and relations are defined according to the type of knowledge they represent. For example **knowledge source maps** map knowledge elements to persons providing the knowledge [10, 6]. **Concept maps**, **mind maps** [6], **topic maps** [11] or **knowledge structure maps** [10] represent knowledge elements with hierarchical relations. **Causal maps** link knowledge elements with a cause-and-effect relationship [6]. **Knowledge development maps** [10] and **knowledge flow maps** [6] represent processes and their procedural relations. [7] states that interactive visualisations foster the interaction with information which generates new "insights". This effect can support the person to elicitate his knowledge.

Therefore, the goal of this work is to use a knowledge map as an interactive visualisation in the phase of knowledge elicitation instead of separating the elicitation from the representation phase as in [3, 12, 13, 14, 15]. In section 3 we develop the adequate domains and relations for representing the relevant engineer's knowledge. In section 4, we combine the interaction with the developed knowledge map and elements from the described elicitation methods to conceptualise a knowledge elicitation method. It seeks to enable the engineer to elicit his knowledge more efficiently without a mentor.

3. Development of a knowledge map

To develop a knowledge map which can be used by the engineer independently, the domains and relations have to be adequate for representing his relevant knowledge elements and their relations. Therefore, we analyse definitions of knowledge domains and relations from psychology, knowledge management and product development literature and verify if they are applicable on the knowledge elements from a case study. In the case study, a semi-structured interview was performed with six engineers of a product development department in a medium-sized engineering company. From the interviews, common knowledge elements were deduced. An example is shown in Table 1.

3.1 Knowledge domains

In technical product development, researchers have analysed the knowledge of engineers and its relation to specific engineering and product development tasks [3, 12, 13, 14, 15, 16, 17]. In several studies, the statements of engineers and designers were analysed and single knowledge elements were extracted [3, 12, 13, 14, 15, 17]. These knowledge elements were categorized into varying domains [3, 12, 13, 14, 15, 17]. Comparable domains are defined in literature on knowledge management [7].The different fields of engineering provide for comprehensive domains and a variety of specific domains. For example, [15] interviewed service engineers and found specific knowledge domains for their in-service information needs. As the knowledge map approach aims at capturing knowledge from engineers working in different industries and on different tasks related to the product life-cycle, including all possible specific knowledge domains leads to too high complexity of the knowledge map. The comprehensive domains are more adequate.

In this section, we consider comprehensive domains from literature and apply them on knowledge elements from the case study. As an example, Table 1 lists alienated knowledge elements from the case study. Based on literature and on the case study we define domains for the knowledge map approach.

Table 1: Exemplary knowledge elements from the case study (alienated)

Knowledge element	Domain
design of sheet metal parts	task
integration of all parts into the system	task
manufacturing techniques	expert knowledge
materials for sheet metal parts	expert knowledge
company-specific manufacturing techniques	process knowledge
product history	product knowledge
design methods for sheet metal parts	methods
CAD software: standard application	software
CAD software: sheet metal module	software
CAD software: technical drawings	software
controlling department	department/company
production planning department	department/company
manufacturing unit	department/company
product data management system	information storage

A central aim in technical product development is to understand which knowledge is needed for performing specific tasks [3, 13, 15] or which knowledge the engineer or designer uses when performing his tasks [12, 14]. In addition, to elicit knowledge from a person, tasks provide a facilitated access to the related knowledge as tasks are explicit, i.e. an engineer is aware of his/ her tasks. Therefore, in the case study, the engineers were asked about their tasks at the start of the interview. In the next step, they named the knowledge elements they needed for the tasks. For example, Table 1 shows the task "design of sheet metal parts" and a number of knowledge elements, the interviewed person named in consequence. In conclusion, *task* is defined as a central domain for the knowledge mapping approach*.*

To facilitate the elicitation of *tasks* and especially the related knowledge elements from other domains, a concept from the knowledge elicitation and transfer method storytelling can be adopted: In learning psychology it has been observed that in order to learn a general activity such as the "visit to the cinema", a person first internalizes a specific event such as the "visit to the cinema to watch the film Titanic" [18]. This observation is transferred to the domain *tasks*, and the domain *situation* is defined as can be seen in Figure 1.

Several researchers name *product* and *process knowledge* as knowledge domains [12, 13, 14, 16, 17]. *Product knowledge* includes knowledge such as the product's components [14] or the product's history [16]. *Process knowledge* refers to all processes and procedures related to the product's life cycle, such as the development process [16], manufacturing, operation or maintenance [15]. Both *product* and *process knowledge* are company-specific. The analysis of the case study shows that the interviewed persons additionally related company-unspecific expert knowledge to product and process knowledge. This is illustrated in Table 1: The knowledge element "manufacturing techniques" is *expert knowledge* independent of the company. It serves as a basis for the company-specific *process knowledge* element "company specific manufacturing techniques". In this case, an example for *product knowledge* is the knowledge element "product history". We therefore define *product*, *process* and *expert knowledge* as domains for the knowledge map.

Methods as knowledge domain are used by [3, 14, 16]. In contrast to *process knowledge*, *methods* refer to the working process of the person whose knowledge is captured. In the example in Table 1, the person with the task "design of sheet metal parts" needs to apply "design methods for sheet metal parts". The engineer has to know about the "company specific manufacturing techniques", but does not manufacture the sheet metal part.

Software is named as a knowledge domain by [14]. In the digital age, software is used by every engineer and can therefore also be found as a knowledge element in the case study. In Table 1, "CAD software standard application" and the "sheet metal module" as well as derived "technical drawings" are exemplary knowledge elements for this domain.

Knowledge about the information source [16] or the personal network providing information [3] are relevant for a number of *tasks*. In knowledge management, [7] call this knowledge *know-where* and *know-who*. The relevance of the source of information and the personal network is illustrated in Table 1: The controlling department, the production planning and the manufacturing unit provide information which is necessary for the *task* "design of sheet metal parts". In most cases, an engineer has specific contact persons from these departments. Another information source is the "product data management system". In addition, information about the output of the *task*, the designed sheet metal part, has to be stored in the PDM system: it is an information storage. Therefore, not only the information sources or networks are relevant, but also the knowledge about their information input as well as the information output of the task. Consequently, the domains *information storage*, *persons* and their *department/company*, *input information* and *output information* are defined for the knowledge map.

3.2 Knowledge relations

In the knowledge map, it has to be possible for the engineer to model the relevant relations between knowledge elements. In section 3.1 we observed that, in a number of cases, to acquire certain knowledge elements, previous knowledge elements are required. Therefore, we define relations of the type "requires" between knowledge elements belonging to different domains:

- The knowledge elements of the domain *tasks* can be linked to knowledge elements of the domains *expert knowledge*, *process knowledge*, *product knowledge*, *methods* and *software* with a "requires"-relationship as depicted in Figure 1.
- . For example, the *task* "design of sheet metal parts" *requires* the *expert knowledge* "manufacturing techniques".
- The knowledge elements of the domain *software* can *require* the knowledge elements from the domains *expert knowledge*, *process knowledge*, *product knowledge* and *methods*. In case of the example in Table 1, for understanding and using the "sheet metal module" of the CAD software, the engineer requires knowledge about "design methods for sheet metal parts" (*method*).
- The knowledge elements of the domain *methods* can require knowledge elements from the domains *expert knowledge*, *process knowledge* and *product knowledge*. To acquire knowledge about "design methods for sheet metal parts" the engineer requires *process knowledge* about "company-specific manufacturing techniques" (for sheet metal parts).
- The knowledge elements of the domain *product knowledge* can require knowledge elements of the domain *process knowledge* and vice versa. In the example, for acquiring knowledge about the product history (*product knowledge*) the engineer requires *process knowledge* about "company-specific manufacturing techniques" to understand why a previous product has a specific design.
  Knowledge elements from both domains can require knowledge elements of the domain *expert knowledge*. For example, for acquiring the *process knowledge* "company-specific manufacturing techniques" the engineer requires the *expert knowledge* "manufacturing techniques" as a general basis.

The definition of further types of relations between the knowledge elements of specific domains is closely linked to the concept of knowledge storage and retrieval in the human brain:

[19] distinguish between explicit knowledge and implicit knowledge. A person is aware of his explicit knowledge and can retrieve it directly from his memory to communicate it. On the other hand, a person is not aware of his implicit knowledge and therefore cannot communicate it. Activation is necessary to turn it into an explicit state [5, 19]. In psychology of learning, two basic types of knowledge are defined: declarative knowledge (mostly explicit) and procedural knowledge (mostly implicit) [18]. Both types of knowledge have different relations which are important for the storing of the knowledge. We use these relations for the knowledge map as explained in the following:

Declarative knowledge incorporates concepts, their description and explanation [18]. For example, the concept "car" has four wheels (description) and is a vehicle, because it transports people (explanation). The explanatory part ("a car is a vehicle") is characterised by hierarchical relationships between the concepts: The concept "vehicle" is superordinate to the concept "car". Other subordinate concepts for "vehicle" are lorries, bikes, trains for example. These hierarchic relations can be observed in the case study between knowledge elements. As to the example in Table 1, the knowledge element "CAD software: standard application" is superordinate to the knowledge element "CAD software: sheet metal module" and "CAD software: technical drawings" because "CAD software: standard application" is more general knowledge. It includes the specific knowledge elements.

Consequently, the hierarchic relation *"is superordinate to"* is defined for knowledge elements from the domains *tasks, expert knowledge, process knowledge, product knowledge, methods and software*. This type of relation is also defined between knowledge elements of the domains *department/company* and *person*.

Procedural knowledge includes concepts for sequences of activities [18], for example driving a car includes activities such as unlocking and opening the car door, sitting at the wheel, turning the ignition key and so on. The sequence of activities are linked by procedural relations. This can be transferred to the knowledge elements from the domain *tasks*. For example, the *task* "integration of all parts into the system" *requires* the previous *task* "design of sheet metal parts". Consequently, in the knowledge map, *tasks* can be linked by the procedural relationship *requires*.

By this means, relations between knowledge elements from the domains tasks, information input, information output, person, department/company and information storage are defined. These relations are shown in Figure 1:

- a *task requires* an *information inpu*t which *requires* an *information storage*, a *person* or a *department/company*.
- an *information storage*, a *person* or a *department/company requires* an *information output* which *requires a task.*

With regards to the domain *situation*, three additional types of relations are defined between *tasks* and *situations*. To capture the corner cases, the relations *challenge in*, *routine in* and *ideal in* are defined as illustrated in Figure 1. By this means the engineer is reminded of challenging, routine and ideal *situations* for performing a specific *task*.

Development of a Knowledge Mapping Approach for Independent Knowledge Elicitation and Representation
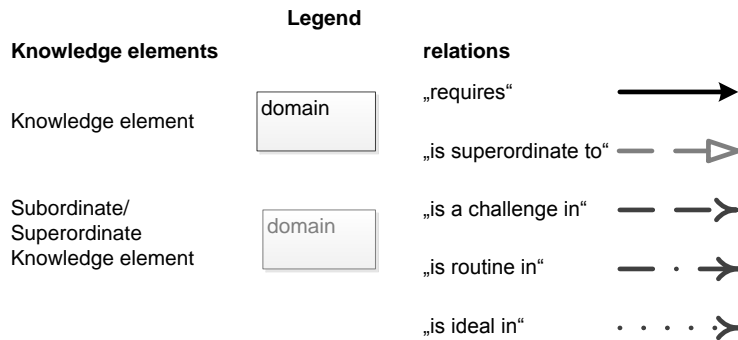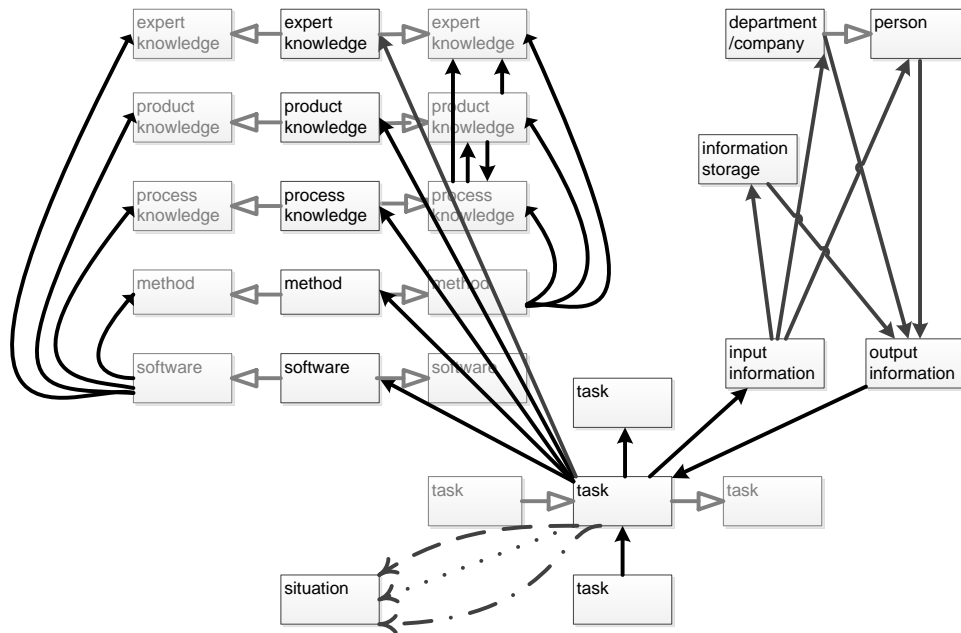Helena Hashemi Farzaneh, Alexander Reik, Maik Maurer

Figure 1: Developed Knowledge Map

4. Development of a concept for independent knowledge elicitation and representation

The developed knowledge map has to be "filled" with content. As the goal is to enable an engineer to do this independently without the support of a moderator, the moderator is replaced by a software implementation which guides the engineer through the knowledge elicitation. We develop an approach consisting of three phases which combines elements from the elicitation methods presented in section 2.

50

4.1 Phase 1: First filling of the knowledge map

In the first step, the engineer who is not familiar with the knowledge map has to elicit a first draft of his knowledge map. The aim of phase 1 is that the engineer can insert his knowledge elements in a "free flow of ideas", an element adopted from *brainstorming*. In order to enable this, the software has to guide the engineer through the knowledge map and explain the domains and relations while he is filling in his first knowledge elements and establishing relations. This is implemented via an interactive question dialogue using focussed questions, an element from the elicitation methods *interview* and *questionnaire*. As explained in section 2, the interaction with the knowledge map fosters the concentration and interest of the engineer [7]. In combination with the instantaneous visualisation of the knowledge elements, this can facilitate the elicitation process.

4.2 Phase 2: Revising and structuring of the knowledge map

The result of phase 1 is a first draft of a knowledge map and it is probable that implicit knowledge is still missing. Therefore, the software has to support the engineer to revise and structure his knowledge map and to discover additional implicit knowledge. The engineer can freely add knowledge elements and relations. In particular, the engineer can introduce more hierarchic relations for structuring, an element from the elicitation method *sorting*. In addition, the software supports him with the recommendations shown in Table 2. The recommendations are partly based on simple structural characteristics such as leaves (a knowledge element related to a single other knowledge element), end nodes (knowledge element without outgoing relations) and activity sum (number of outgoing relations).

Table 2: Implementation of software recommendations for revising and structuring the knowledge map

recommendation	implementation
add forgotten elements and relations	missing knowledge elements and relations, leaf, end node
unify equal knowledge elements	equalities
consider similar hierarchic levels for all knowledge elements of one domain	(degree of) hierarchy / distance
if a knowledge element has many subordinate knowledge elements, consider to introduce an additional hierarchic level	high active sum of the knowledge element
if a knowledge element has a few subordinate knowledge elements, consider to add knowledge elements	low active sum (>0) of the knowledge element

4.3 Phase 3: Integration into the working routine

In phase 1 and 2 the engineer has developed, revised and structured his knowledge map. However, his working routine can provide further insights to necessary knowledge and its structure. Particularly the previous development of the knowledge map can increase the engineer's attention to verify the plausibility of the knowledge map. Therefore, in phase 3, the engineer returns to working on his task. At the same time, the knowledge map is visible and can be modified without major effort. This corresponds to the *observation* and *think-aloud* elicitation methods which put the person into the daily routine to facilitate knowledge elicitation.

5. Discussion

For the knowledge map, we simplify the possible relations and name both procedural and causal relation "requires". For example a task requires expert knowledge (causal relation) and requires a previous task (procedural relation). This is strictly speaking incorrect, but does not imply difficulties in case of this knowledge map, because the knowledge elements linked by a causal relationship cannot have procedural relations. Therefore, the simplification reduces the amount of different relations the engineer has to use and can facilitate the use of the elicitation method.

6. Conclusion and outlook

In this work, we developed a knowledge mapping approach for independent knowledge elicitation and representation for an engineer. The approach uses a knowledge map for representing knowledge. It is developed based on psychology, knowledge management and product development research. The indications for knowledge domains and relations were verified with the analysis from knowledge elements acquired in a case study. The elicitation process is supported by interactive visualisation of the knowledge map and combines elements from a number of knowledge elicitation methods.

In a next step, the knowledge map and the conceptualized elicitation method will be implemented in software and tested in case studies with engineers. The engineers will use it independently to elicit and represent their knowledge. During the case studies, they will be observed and asked for their feedback. This will enable further improvement of the developed knowledge mapping approach.

6. References

1. Howells, J. R. L. Tacit knowledge, innovation and economic geography. Urban studies Vol 39: 5-6, pp. 871-884 (2002)
2. Argote, L.; Ingram, P.: Knowledge transfer: a basis for competitive advantage in firms. Organizational behaviour and human decision processes Vol 82: 1, pp. 150-169 (2000)
3. Maurer, M.; Kesper, H.: Knowledge transfer applying the structural Complexity Management approach, International Conference on Information Retrieval & Knowledge Management (CAMP'10), Shah Alam, Selangor, Malaysia, (2010)

4. Alavi, M.; Leidner, D. E. Knowledge management and knowledge management systems: Conceptual foundations and research issues. MIS quarterly Vol 25: 1, pp. 107-136 (2001)
5. Jetter, A. Elicitation - Extracting Knowledge from Experts. In: Jetter, A.; Kraaijenbrink, J.; Schröder, H.-H.; Wijnhoven, F. (Eds.): Knowledge Integration: The Practice of Knowledge Management in Small and Medium Enterprises, Heidelberg: Physica, pp. 65-76 (2006)
6. Jetter, A. Codification - Knowledge Maps. In: Jetter, A.; Kraaijenbrink, J.; Schröder, H.-H.; Wijnhoven, F. (Eds.): Knowledge Integration, Heidelberg: Physica, pp. 77-90 (2006)
7. Eppler, M.; Burkhard, R. A. Visual representations in Knowledge management: framework and cases, Journal of Knowledge Management Vol 11: 4, pp. 112-122 (2007)
8. Mayer, H. O.: Interview und schriftliche Befragung, 4 ed., München: Oldenbourg (2008)
9. Gavrilova, T.; Andreeva, T. Knowledge elicitation techniques in a knowledge management context, Journal of Knowledge Management Vol 16:4, pp. 523-537 (2012)
10. Lehner, F. Wissensmanagement. München: Carl Hanser Verlag (2012)
11. Weber, H. Erstellung nutzer-individueller Dokumente für die Vermittlung von Produktentwicklungswissen durch den Einsatz von Topic Maps. phd thesis, Technical University Darmstadt, Darmstadt (2010)
12. Ahmed, S. Encouraging reuse of design knowledge: a method to index knowledge, Design Studies Vol. 26, pp. 565-592 (2005)
13. Vianello, G.; Ahmed, S.: Knowledge transfer between service and design phases in the oil industry, International Conference on Engineering Design (ICED'09), Stanford (2009)
14. Trevelyan, J. P.: Reconstructing engineering from practice. Engineering Studies 2 Vol 3, pp. 175-195 (2010)
15. Jagtap, S.; Johnson, A.: In-service information required by engineering designers, Research Engineering Design Vol. 22, pp. 207-221 (2011)
16. Koller, R.; Berns, S.: Strukturierung von Konstruktionswissen, Konstruktion Vol. 42, pp. 91-96 (1990)
17. Wickel, M. C.; Schenkl, S. A.; Schmidt, D. M.; Hense, J.; Mandl, H., Maurer, M.: Knowledge structure maps based on Multiple Domain Matrices, International Conference on Innovation through Knowledge Transfer (Innovation-KT 2013), Derry-Londonderry, Northern Ireland (2013)
18. Edelmann, W.; Wittmann, S. Lernpsychologie, Weinheim: Beltz Verlag (2011)
19. Nonaka, I.; Takeuchi, H.: The Knowledge-Creating Company - How Japanese companies create the dynamics of innovation, New York: Oxford University Press (1995)