INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

COSECO (CONTEXT-SENSITIVE-CONNECTOR) – A LOGICAL COMPONENT FOR A USER- AND USAGE-RELATED DOSAGE OF KNOWLEDGE

S. Weiß, B. Berger, J. Jänsch, H. Birkhofer

Abstract

An all knowing database is useless, as long as it is not possible to find specific contents. This paper describes a situational approach, trying to extract knowledge from the database in a context sensitive way. By integration of different dimensions this approach tries to consider several aspects, such as types of users, aims and processes. Threads of implementation are predominant. Main intention is to discuss the gap between user related profiles (describing target requirements in respect of contents depending on situation) and modularised knowledge objects memorised in a database, carrying both, content and descriptive attributes (metadata used for internal processing).

Keywords: knowledge-based engineering, VDI 2221, databases, e-learning

1 Introduction

1.1 Factor *Knowledge*

Lets put ourselves in the place of a business company offering products to their customers in the market. The reaction of these customers to this offer depends on various product related parameters, such as quality, price and image. But also the customers individual situation is an aspect of relevance (e. g. employment market, propensity to consume, propensity to save). These companies' existence is highly dependet on their products [1]. But products have to be *developed* first, so the effort of employees plays an important role. Precisely, products are generated by the effort of product developers transforming their available knowledge to products in a creative way. Thereby, the factor knowledge is of central importance. The situation companies face is marked by rapid change in the market and a high speed of innovation. The predominant consequences of this are a steady drop off in prices, the need for individual adaptation to customer wishes, shorter product lifecycles and new business segments. Therefore, companies are forced to develop more rapidly: knowledge, especially specialised knowledge, is becoming more quickly outdated [2].

A good deal of total time involved in product development accounts for sourcing, processing and transfering information and knowledge. So, the *availability* is of predominant importance. Within the conception discussed in this paper, knowledge is stored in so called *knowledge-units*. A knowledge-unit is a well-defined and modularised object, constituting a specific formal or semantic content, also containing a set of descriptive attributes.

1.2 Knowledge Management and Competitiveness

Now lets assume, that several product developing companies are competing in the market. This happens not merely in the national market, but rather international. The increasing pressure of globalisation is one reason that only companies which effectively manage the knowledge factor will sustain their competitiveness. So managing knowledge in an efficient way becomes a factor of central importance. This means that the presence, preservation and continuous update of knowledge are the key resources for gaining a sustainable competitive advantage in the long run [3].

As a first approximation, the important requirements are: fast acquisition of knowledge, extensive availability (existence **and** location!) of knowledge and presentation in a suitable mode (e. g. complexity, level of detail, language).

2 State of the Art

Not only product developers are highly dependent on knowledge. The exposure to knowledge is also predominant to teachers and students. Dependent on their students, teachers have to process knowledge practically to put it across in a didactical adequate way.

2.1 The Pinngate Approach

In this regard, one way to satisfy the requirements of product developers, teachers and students is the pinngate-approach (Fig. 1).

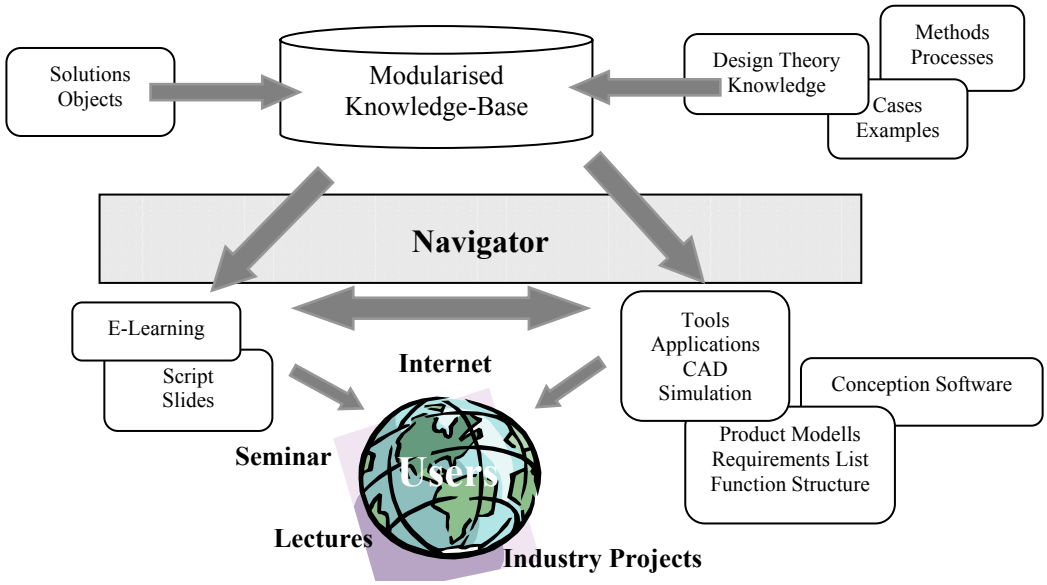

Figure 1 Schematic representation of the pinngate-conception.

Pinngate has primarily to be understood as a broad conception and consists of different parts.

A central part of pinngate is a knowledge base. This is a database, memorising modularised contents. The modularisation results from the elements-modules-containers-conception (for a detailed discussion refer to [4], [11]). Elements, modules and containers are knowledge-units. These knowledge-units contain knowledge about design theory, design methods, cases und a collection of concrete solutions as well as objects for design tasks. The import of contents is done by experts, called *authors*. Authors generate contents and request for a memorisation in the database via an organ attended to quality assurance, called *quality board* (QB). The QB is

a commission of experts (near-pinngate), controlling most database activities. In cases of doubt, a memorisation of contents is denied. A specific software tool has been developed to render the building of a modularised database possible (*EMC-Manager*; paragraph 4.5).

As a second part of pinngate, a teaching and learning environment has been implemented, providing documents for students and developers. These documents can be derived and configured from knowledge-units out of the database for specific purposes. Through this, it is possible to satisfy specific interests of potential user groups.

A third part of pinngate provides application tools especially for the conceptual phase. These are concrete tools (e. g. evaluation, morphological box) supporting developers in practice. Thereby, database-related contents are processed.

Among other things, globalisation and modern information technology have caused an exponential growth of available and potential accessable knowledge. Ehrlenspiel differs between quality and quantity of knowledge [5]. Nevertheless, in this context the focus is the *quality* of knowledge, because the quantity of knowledge *can* be managed. This is correct because all knowledge-units resides within our system boundary. In this context it is necessary to differ between *duplicates* and *redundancies*: duplicates are documents with same content and same character, redundancies are documents with same content but different character. Within the conception of knowledge-units (which are objects), duplicates are reduced and redundancies are integrated by objectivisation. All redudancies may be considered as one. Because of that it is possible to get a grip on erratic growing knowledge-bases and to give a special emphasis on quality.

A component of central importance is the *navigator*. This is a central interface imparting knowledge-units and processing data between the different parts mentioned before. A sub-system of the navigator, called *context-sensitive-connector* (COSECO), is discussed is this paper. The pinngate-system is split into two levels (Fig. 1): firstly, the *level of contents* (beyond the navigator), secondary, the *level of application* (below the navigator). The navigator transfers knowledge-units from the level of contents to the level of application. Knowledge-units shall be transferred only according to users intention. This is done by COSECO.

Moreover, the navigator controls other tasks such as logging user-performances (e. g. to arrange undo) or establishing a base of data in terms of adaptivity.

2.2 Deficits

The relation between knowledge-units and classes of users is an escrow issue. This papers' aim is to discuss and to reconstruct the transfer of knowledge-units from the level of contents to the level of application. Thereby, an approach shall be considered, separating relevant from less-relevant knowledge-units. In this regard the needs of users are of central importance.

Another problem arises from links, that can be composed between knowledge-units. Assumed that between all database-contents links can be made (however) and that every content can be brought in by different persons, the *degree of disorder* in the database would increase rapidly. Pinngate uses the QB to control quality (the QB is of central importance), but a total control is nearly impossible. The results are dead links, unused knowledge-units and destroyed structures. Moreover, a continuously increasing number of memorised knowledge-units justifies an increasing number of links that may be composed potentially. Thereby, a total control of quality affects an exceeding of costs over profits. So control of quality is reasonable merely to a certain extent. Under economic aspects its incremental costs shall equal its marginal utility.

This is quite complex: lets assume an existing database contains e. g. 20.000 knowledge-units and another unit is brought in. This unit has to be interlaced. This is done by linking it to other units concerning the new one. The number of possible links increases as the amount of contents rises. In the pinngate-system, this is solved by an integrated, intelligent linking-editor in conjunction with the QB.

Arising from further discussions regarding the pinngate project a motivation has resulted, to find an approach satisfying different aspects, such as general aspects of pinngate, aspects of highly stuffed databases, aspects of content-quality and aspects of recovery.

Within the scope of pinngate, general aspects are system-immanent. Moreover, the quality is guaranteed by the QB. Assuming databases are always stuffed, the aspect of recovery is left. Thus, this shall be specified.

3 A Situational Approach

3.1 Who Needs When, Wherefore, Which Content?

On the one hand, a knowledge-unit may be of special interest concerning a defined situation. On the other hand, the same knowledge-unit may be exactly the opposite concerning another situation. So we have to establish a context-relationship between contents or knowledge-units and needs of users.

To extract knowledge-units out of the database in a well directed way, two options may be selected: firstly, ask yourself "What do I need?" and have a look for it; secondly, ask yourself "What do I need *not*?", exclude this set and have a look for the rest. Here, we follow the second way. It is an analogue to the processes "Selection" and "Evaluation" [6].

The answer to the question "What do I need?" (and therewith the question "What do I need not?") is highly dependent on the aim we want to achieve. But different types of users have different aims. But the main question results from modifying the initial question: "Who needs when wherefore what?". The diversity of possible answers requires a reference to context.

Lets assume there is a formalised and functional relation between this four dimensions *Who*, *When*, *What* and *Wherefore*. Then one dimension is interdependent determinable by a specification of the three other dimensions. This thread shall be described in the following.

3.2 Context and Situation

The term *context* stands for surrounding relation. The term context-sensitivity stands for sensitiveness of relations [7]. In the framework of a context there are different concrete *situations*.

Concerning the pinngate-project, an important aspect is the implementation. So it is necessary to formalise this because formalised structures are easier to implement as verbal descriptions. Therefore, the terms *context* and *situation* shall be expressed in a different way.

Lets assume there are different attributes m and an unknown number of dimensions i, resulting in i attributes m_i (one attribute per dimension). Lets assume furthermore: each attribute m_i has $k(i)$ values w. Then, a set of values W_i concerning attribute i is defined by $W_i = \{w_{i,1}, w_{i,2}, \ldots, w_{i,k(i)}\}$.

So it is possible to integrate knowledge-units in a context according to different attributes. Moreover, every dimension may be allocated by a concrete value out of a set of values. This concretises a situation.

Thus, a context is defined by descriptive attributes and a situation is defined by concrete values: *context* $c = f(m_1, m_2, \ldots, m_i)$ and *situation* $s = g\left(w_{1,x_1}, w_{2,x_2}, \ldots, w_{i,x_i}\right)$ and $1 \le x_y \le k(y)$ for each $y = 1 \ldots i$.

Dependent on fineness in respect of descriptive attributes, an endless number of dimensions would theoretically result. This is not functional. So it is necessary to define some important dimensions. By this, it is possible to reproduce at least approximately a real situation.

3.3 Dimensions of Context

Certainly, the context dimensions are extensive, such as type of user, aim, process, progress in processes, branch, degree of innovation, business company or kind of manufacturing. Main intention of this paper is not to create a complete overview of all attributes constituting a context-relation. How these attributes can be found and how variable they are, is not discussed herein also. Main intention is, on assumption of known dimensions, to use them for the extraction of specific knowledge. Therefore, it is useful to chose those attributes being important respective seeming to be important. So we can gain a first access to this theme. So the following dimensions shall be selected as dimensions of context: *process / progress of process*, *type of user* and *aim*.

A *process* describes a timed change of state, transforming an initial state into an ending state. The states before and after the transformation are real, observable, measurable and describable [8]. During the change from initial state to ending state different sub-processes are passed through. This is the *progress* of a process (Figure 3 in paragraph 3.4). A *type of user* is a given collective of persons defined by similar profiles of interests. In this context, an *aim* stands for an intention of a specific type of user.

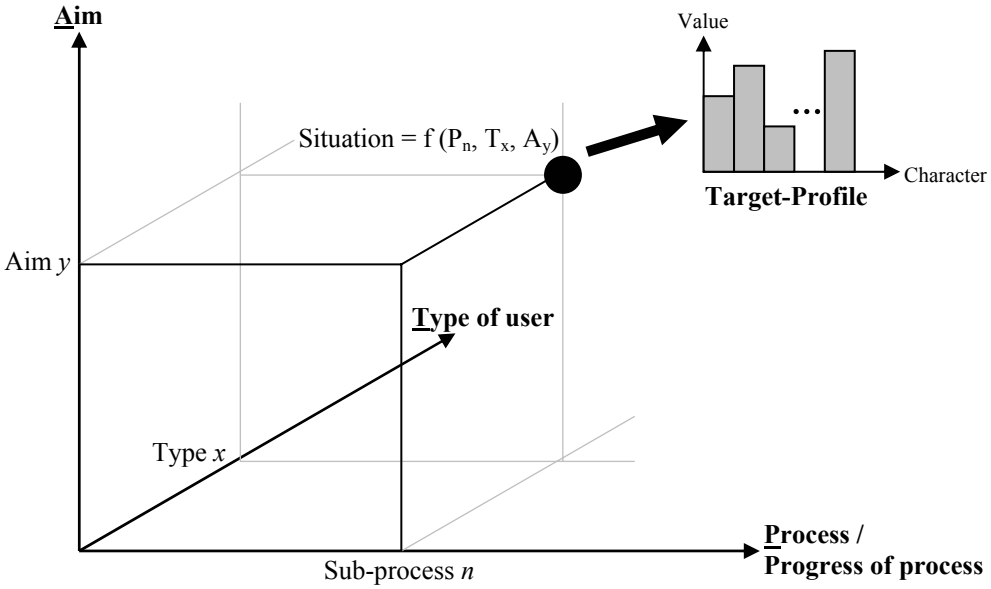

Figure 2 Context and situation defined by 3 dimensions (process, type of user, aim).

Now, a three-dimensional *situation-space* is spanned (we may also call it *context*). Each *situation* is defined by exactly one concrete value for each attribute. In Fig. 2 the three selected dimensions are illustrated: m_1 = progress of process, m_2 = type of user and m_3 = aim. Appropriate values, for attribute 2 are for example: $w_{2,1}$ = product developer, $w_{2,2}$ = Professor / Research Associate and $w_{2,3}$ = Student. The process bases in all cases on VDI 2221 [10],

[12]. For each defined situation, a target-profile is derivated. This profile describes the requirements for relevant knowledge-units.

3.4 Process (When?)

Considering different types of users and therewith associated aims in respect of pinngate, many processes are imaginable. Particularly such as the processes of product development, teaching and learning.

The (iterative) process of product development has to be understood according to VDI 2221. The processes are formalised and transformed in a data-model processable by computers (due to space, this is not explained here). It is possible to inform a computer about a considered process and its progress (Fig. 3).

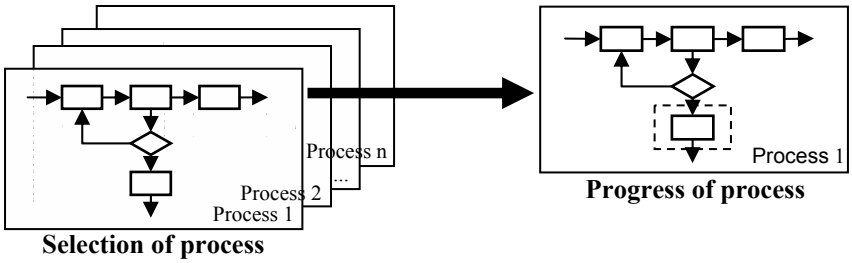

Figure 3 Selection of process and progress of process.

But looking at a process and its progress is an insufficient determinant to narrow down the information overload.

3.5 Types of Users (Who?)

Within the scope of pinngate, there are different types of users, such as product developer in industry, Professor / Research associate and Student. Every type has specific interests. So the question “When?” is extended by integration of user-types with a “Who”-component. This is disposed by the fact, that the same progress of process requires different knowledge because of different intentions. Of course, this depends on the main aim (every type of user can e. g. learn or develop). E. g. within the scope of VDI 2221 one step is to “clarify the task”. In this context, a developer in industry is interested in a form sheet to complete his task while a student trying to understand the steps needs a low level explanation at first (e. g. requirement types). But process and type of user as determinants for their own will still result a high number of knowledge-units.

3.6 Aim (Wherefore?)

A declaration of an aim integrates the users intention in this structure. The questions are: “Why does the user request for knowledge?” and “What is the background?”. In doing so, it is not functional to work with an endless set of aims, rather classes of aims have to be built. This aims should have a simple structure, such as “Gain Introduction” or “Apply Method”.

4 Implementation

As a first approximation, these three dimensions (“When”, “Who” and “Wherefore”) suffice to isolate knowledge-units of importance. Now we have a look at the steps we have to pass through to isolate knowledge by using these dimensions.

4.1 The Modularised Knowledge-Base

The knowledge-base contains all the available knowledge memorised in a modularised structure (concerning modularisation: [4], [11]). But it is necessary to extend the set of attributes used for modularisation by attributes used for providing knowledge in a context-sensitive way. I. e., every knowledge-unit has to contain an actual-profile, describing the potential suitability of this unit. Moreover, further attributes allow general conclusions by their rule-character (e. g. "is not qualified for…"). The definition of these attributes has to be done while creating a knowledge-unit. This is provided by the EMC-Manager (paragraph 4.5). While definition a concept-space is created, too, enabling an access by a semantic level. By using semantic networks, it is possible to map relations between concepts [9]. Aspects concerning the stage of implementation are mentioned later (paragraph 4.5).

4.2 Rules

The following procedure corresponds to the processes "Selection" and "Evaluation" [6].

Lets imagine a broad field of knowledge-units. At first, a rule-based selection is made to exclude irrelevant knowledge-units temporary. The result is a number of units we have to evaluate. By this selection, we have reduced the quantity of units we have to evaluate, so specific algorithms are faster. Moreover, we have a shortened set of units, which may be scanned manually by the user in case of doubt.

Rules should be built in a way, that whole categories can be excluded concerning dimensions of context. Concerning units, attributes are provided storing Boolean values in respect to rules.

Examples: Rule A – "is qualified for user-type…"; Rule B – "is not qualified for user-type…"; Rule C – "is qualified for learning-process". Logical relations between rules can be established (e. g. $A = \overline{B}$). These conclusions are utilised to exclude knowledge-units by logic.

4.3 Adjustment of Profiles

In the following step, an adjustment of target-profile and actual-profiles is made. The target-profile is generated by its various determinants, such as process / progress of process, type of user and aim. Furthermore, for each knowledge-unit an actual-profile is stored. The adjustment of these profiles (1:n) is problematic.

a) To compare an applied target-profile with a describing actual-profile a transformation is necessary because of different profile-structures. The target-profile describes a situation and is generated on context. The unit-describing actual-profile is attribute-based and usually different from a target-profile. So a comparison in a direct way can not be drawn. In the pinngate-project simple structures of profiles are installed to analyse required transformations. Gradually, more complex structures of profiles are planned. Through this, it is possible to acquire an access to the problem of this transformation. Additional, first experiences can be made.

b) Another problem is to define the value referencing attributes. To prioritise knowledge-units, an evaluation and a ranking has to be made (Fig. 4). Topological instead of metrical scales are reasonable, because it is not accurate to say: unit one is 2.67 times more relevant in a specific situation than unit two. But it can be determined that unit one is more relevant in a specific situation than unit two. Metrics may be used for internal processing but results have to be interpreted in a topological context.

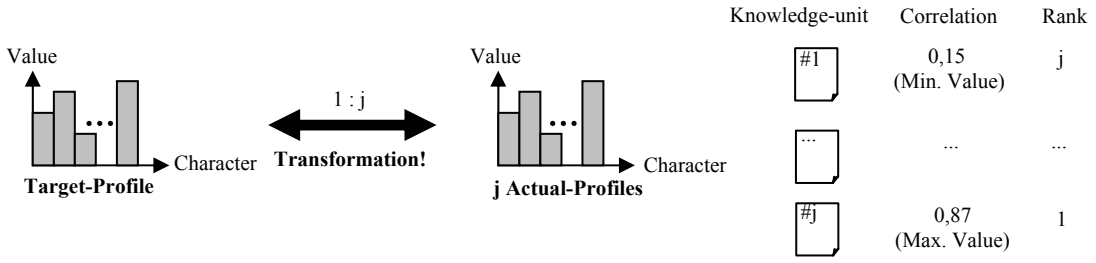

Figure 4 Adjustment of target-profile and actual-profiles with resulting rank.

4.4 Pinngates „Context-Sensitive-Connector“ - COSECO

The implementation of context-sensitivity in pinngate bases on the thoughts explained before. This is summarised in (Fig. 5). A rule-based filter, dependent on process / progress of process, type of user and aim, encloses *k* knowledge-units out of the modularised database. *j* units are left. Each unit has an actual-profile. Context-related requirements on these knowledge-units are centralised described by these profiles. The transformation mentioned before bypasses the gap between target-profile and actual-profiles. A comparison follows, finalised by a derivated ranking-list arranging the knowledge-units being left. This set of units is the base for further processing.

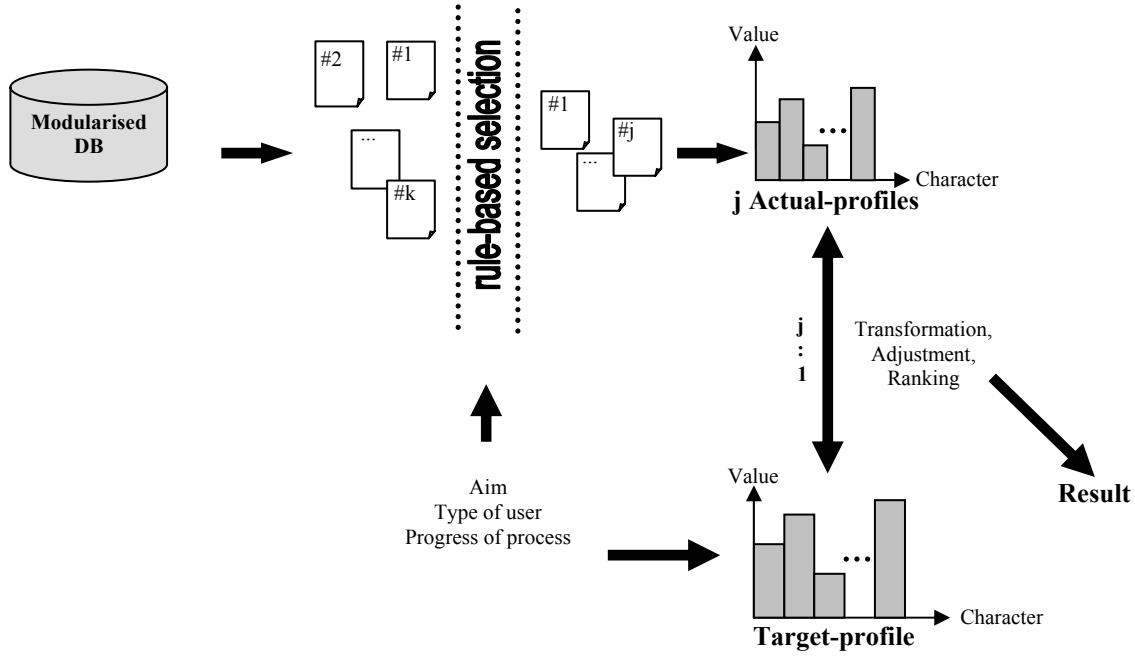

Figure 5 Schematic representation of COSECO-conception (simplified).

The term *connector* was chosen, because the conception is an intermediary between pinngate-sub-systems. COSECO is a sub-system of the navigator.

4.5 Stage of implementation and experimental results

To run a modularised database we need three levels according to the concept of elements, modules and containers [11]. A concept space, here built up by a semantic net (level one): this

is implemented by a tool called S-Net-Manager. This tool creates and manages semantic networks and is able to link down to elements, modules and containers created with another tool called EMC-Manager. The S-Net-Manager provides all its information realtime and online in the WWW as well. The contents of the database (level two and three) are built by a tool called EMC-Manager. This is a drag & drop tool to create and manage modularised structures. S-Net-Manager and EMC-Manager are interactive. Deficits mentioned in 2.2 like dead links, unused knowledge-units and destroyed structures are handled by the tools too.

Currently, we gain experience by the use of these tools concerning the contents. In this context we use material concerning design theory and useful material to perform industrial development projects. The tools mentioned before are proprietary developments by the author. By an iterative process of programming and testing (runtime behaviour and usefulness in practice) these prototypes will be improved. To satisfy the needs of different user-types (and to feed COSECO) rules have to be built. For this purpose, empirical material is currently analyzed, such as [13], [14]. Moreover, we use our experience concerning course activities at the university and industrial cooperation projects.

5 Summary and Outlook

The factor *knowledge* plays a large part in the creation of value. Consequently, an exposure to knowledge is of central importance. A main aspect is that availability consists of existence and location. Knowledge has to be available in high quality and extensive and should be provided situational by low-effort. Therefore, an *efficient* knowledge-management is necessary. But different problems have to be solved such as mastering the quantity of knowledge-units and a well directed situational access. Using a context-relation is one approach. But each result has to be considered as a recommendation. The final decision has always to be made by the developer. So a system has to consider this aspect but it has to be highly flexible too.

The main problem is the transformation which is necessary to compare knowledge-units (described by attributes) and target-profiles (situational generated, describing requirements). By incremental approximation first results were achieved. There is a gap between a widely implementation-related approach of attributing knowledge-units and a widely socio-psychological-based human requirement for knowledge. By finding a suitable transformation this gap can be closed. Therefore it is necessary to integrate an adequate number of parameters whereby the complexity rises. So suitable parameters have to be found by scientific research. This applies especially concerning the profiles because knowledge-unit attributes are widely known (derivated by argumentations like [4], [11]).

Context sensitivity and modularisation were combined by this approach. A user may browse the contents by using the semantic net or the user may be guided by system suggestions. But it is very important to remark: such a system always has to *support* and never to *automate* a process! More over: from the authors point of view it will never be possible to automate a developing process totally. Because artificial intelligence can never substitute the human factor. This would hypothesize a strong artificial intelligence – a copy of the consciousness implemented in a machine (sentiments included!). But this is an (unrealistic) future vision.

6 Literature

[1] Birkhofer, Herbert, Kloberdanz, Herrmann, Berger, Benjamin, „Skript zur Vorlesung Produktentwicklung I, Wintersemester 2000/01“, Fachgebiet Maschinenelemente und Konstruktionslehre, TU Darmstadt, Kap. 1, S. 1.

[2] Schüppel, Jürgen, „Wissensmanagement: organisatorisches Lernen im Spannungsfeld von Wissens- und Lernbarrieren“, DUV, Wiesbaden, 1996.

[3] Bürgel, Hans Dietmar, Zeller, Andreas, „Forschung und Entwicklung als Wissenscenter“, in: Bürgel, Hans Dietmar (Hrsg.), „Wissensmanagement: Schritte zum intelligenten Unternehmen“, Berlin u. a., 1998, S. 53 – 65.

[4] Birkhofer, Herbert, Berger, Benjamin, Walter, Stephan, „Modularisation of Knowledge – A New Approach in the Field of Product Innovation“, Proceedings of the International Design Conference 2002, Dubrovnik, Croatia, Volume 1, S. 289.

[5] Ehrlenspiel, K., „Knowledge-Explosion and ist Consequences“, Proceedings of the 11th International Conference on Engineering Design ICED 1997, Vol. 2, S. 477 – 484, WDK 25.

[6] Pahl, G., Beitz, W., „Engineering Design – A Systematic Approach“, Springer, 1996, S. 99.

[7] Storath, Elmar, „Kontextsensitive Wissensbereitstellung in der Konstruktion“, Dissertation, Lehrstuhl für Konstruktionstechnik, Erlangen, 1996, S. 74.

[8] Heidemann, Bernd, „Trennende Verknüpfung – Ein Prozessmodell als Quelle für Produktideen“, Dissertation, Fachbereich Maschinenbau, Darmstadt, 2001.

[9] Seeberg, Cornelia, “Modulare Wissensbasen zur Erzeugung adaptiver und kohärenter Lehrdokumente“, Dissertation, Fachbereich Informatik, Darmstadt, 2001, S. 52 ff.

[10] VDI-Richtlinie 2221, „Methodik zum Entwickeln und konstruieren technischer Systeme und Produkte“, Düsseldorf: VDI-Verlag, 1993.

[11] Birkhofer, H., Berger, B., “Modularization of Design Knowledge as a Basis for a High Quality Competence Pool in Product Development”, Proceedings of DETC’02: ASME 2002 Design Engineering Technical Conferences and Computers Information in Engineering Conference, Montreal, Canada, 2002.

[12] Sauer, T., Kloberdanz, H., Walter, S., Birkhofer, H., “Describing solutions for the conceptual phase – structured and user orientated”, expected ICED 03.

[13] Frankenberger, E., “Arbeitsteilige Produktentwicklung”, Dissertation, Fortschr.-Ber. VDI Reihe 1 Nr. 291, Düsseldorf: VDI-Verlag, 1997.

[14] Wallmeier, S., “Potentiale in der Produktentwicklung”, Dissertation, Fortschr.-Ber. VDI Reihe 1 Nr. 352, Düsseldorf: VDI-Verlag, 2001.

Sascha Weiß

Product development and machine elements, pmd

Darmstadt University of Technology

Magdalenenstrasse 4

64289 Darmstadt, Germany

Phone: +49 (0) 6151 / 16 26 60

Fax: +49 (0) 6151 / 16 33 55

EMail: weiss@pmd.tu-darmstadt.de