INTERNATIONAL DESIGN CONFERENCE - DESIGN 2002
Dubrovnik, May 14 - 17, 2002.

APPROACHES OF KNOWLEDGE-BASED DESIGN

Sándor Vajna

Keywords: Design theories, knowledge-based design, knowledge taxonomy

1. Introduction

The most important success factors of a company are dedicated employees and the application of knowledge. This statement was made b*y Frank B. Gilbreth* already in 1907 in *Field System*, an instruction book for his contractor company of how to erect buildings effectively [1].

Besides the designer's creativity and ability, the provision and application of knowledge is especially of interest within product development, as he determines the relevant characteristics of a product (within given boundaries of requirements fulfilment, quality, time, and costs). Appropriate knowledge is needed and used everywhere in the processes of product development. But it isn't always visible and easily available-even within a group with common interests like e.g. designers.

- Knowledge isn't always easy to apply without preparation efforts, especially when using it from non-human (or "external") sources. Independent from its storage media, external knowledge seems always to be old and not covering all aspects of a problem description (with the exception of basic knowledge, i.e. a generally accepted set of knowledge within a domain with low change frequency). Obviously, non-human systems can't update knowledge automatically, can't build up analogies and interlinks, or generate new knowledge.
- There are only limited solutions of how to describe knowledge, to acquire it, to maintain, to administrate it, and to get rid of it when it becomes outdated or obsolete.
- It seems to be very difficult to model knowledge into the product model of a CAD system.

The success of product development is based on the creativity of the designers. Besides other descriptions, creativity can be regarded as a new combination or configuration of existing knowledge which is used to solve a task, which is not, structured sufficiently [2]. This knowledge seems to be always in the possession of an individual ("intellectual property"), who generates, acquires, explores, enhances, refines, gives away, buys, and sells knowledge (the later activities may also be allocated to a "Specialist" or to a "Consultant").

Knowledge can't be transferred simply neither from one person to another nor with an analogy to similar requirements. Examining how knowledge is handed over between designers, one may find out that the designer transmits but data, information (data with a context), rules (describing the handling of and the relations between each data and information), and meta-rules (describing the context between all these elements and the processing of rules). Knowledge and, based on that, experience are only created at the second person if the second one trusts the first one. The created knowledge isn't always exact and certain, mostly because of the possibility of analogous application. A designer needs experience and skills to decide by in-advance-interpretation whether the acquired knowledge fits to his field of work. He is "competent" if he can handle the acquired knowledge appropriately to a given task.

Is knowledge therefore the (human) ability to create and to apply rules (based on experience, supported by confidence) on a given set of data and information, i.e. distilled (respectively refined) or neutralised (respectively abstract) experience? The knowledge taxonomy (**figure 1**):

- tries to put some of the keywords and key activities describing the designer's work into a context, to show their possible interdependencies and their possible bases, especially the link between experience and knowledge,
- differentiates between "individual" (i.e. internal) and "portable" (i.e. external) knowledge,
- shows that, on a computer, knowledge can only be processed as a set of data, information, rules, and meta-rules.

If this taxonomy is applicable, it seems to be incorrect to speak of (computer-aided) knowledge-based design, but only of computer-based rules application in design. This seems to be true for most of the today commercially available knowledge-based design systems, e.g. ICAD [3] and Intent! [4] (nevertheless, as an established habit, an accumulation of data, information, rules and meta-rules on a computer will be still referred as "knowledge base" in this paper).

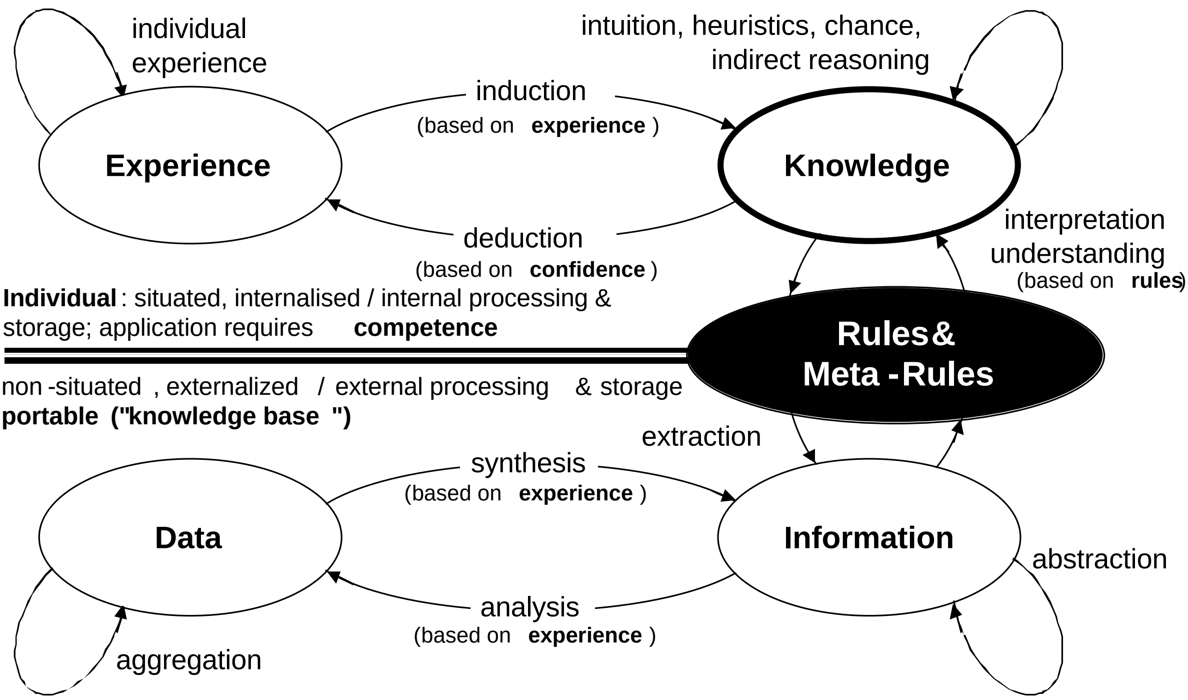

Figure 1. A knowledge taxonomy

2. Application and Use of Knowledge in Product Development

It is obvious that the development of a product has always been knowledge based, otherwise there would hardly be successful products. In order to work successfully and to make the right decisions at the earliest time possible especially when applying computer support with its multiple simulation possibilities the designer needs appropriate and sufficiently linked information. This information, e.g. functions, partial functions, product behaviour, costs, available manufacturing technologies, influences and constraints from other departments into his work (leading among others to Design for X), company strategies and philosophies, etc., should be provided on time, **figure 2**.

It is this information which mostly consists of technical data and of explicitly available knowledge about products and processes. However, this information normally is stored and processed in the brain of the designer and not on external media (e.g. in the model of the product he is creating in a CAD system). His knowledge "fuels" the design process, thus transforming the mental (i.e. virtual) model of the part to be designed into an external description of this part (product model, drawings, BOMs, working plans, etc). If he needs additional information he has to ask a colleague or has to look it up in external media, and has to transform it (and to refine it) in order to make it fit to his existing knowledge.

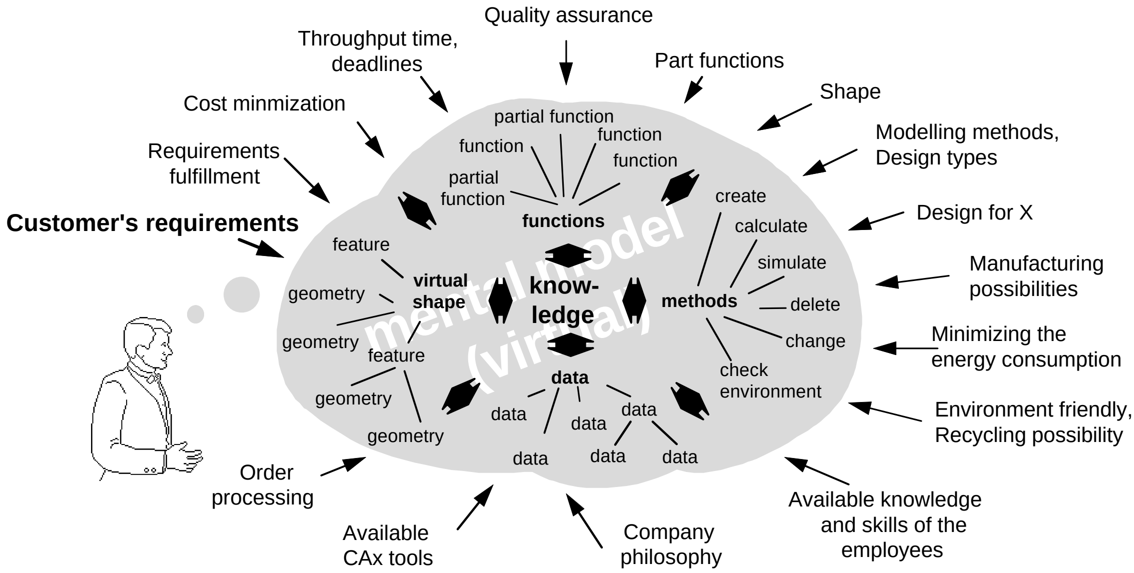

Figure 2. Influences on the designer's work

Looking onto the design process from the order processing point of view, **figure 3** shows first the well-known graphs of fixed costs and of incurring costs. Costs of the later product are fixed by early decisions within product development (e.g. for a shape, a material, a manufacturing technology, and an assembly order), of which the quality can be improved e.g. by using optimisation strategies [5].
Although the knowledge about a product is continuously growing along the development process, the necessary knowledge to take the right decisions isn't at all or at least not always completely available at the time required (shown in the third graph in figure 3). This highlights not only the higher importance of design work compared to manufacturing, but also the higher risks designers have to take during their work by using incomplete information for decisions thus relying heavily on their experience when "predicting" the characteristics of a product. It is therefore the aim of knowledge-based design (as well of any computer support) to provide the right amount of knowledge (as data, information, rules, and as simulation of the behaviour of the later product) at the earliest possible moment. This allows the designer to take the necessary decisions in good time during product development (symbolised by the arrow in figure 3).

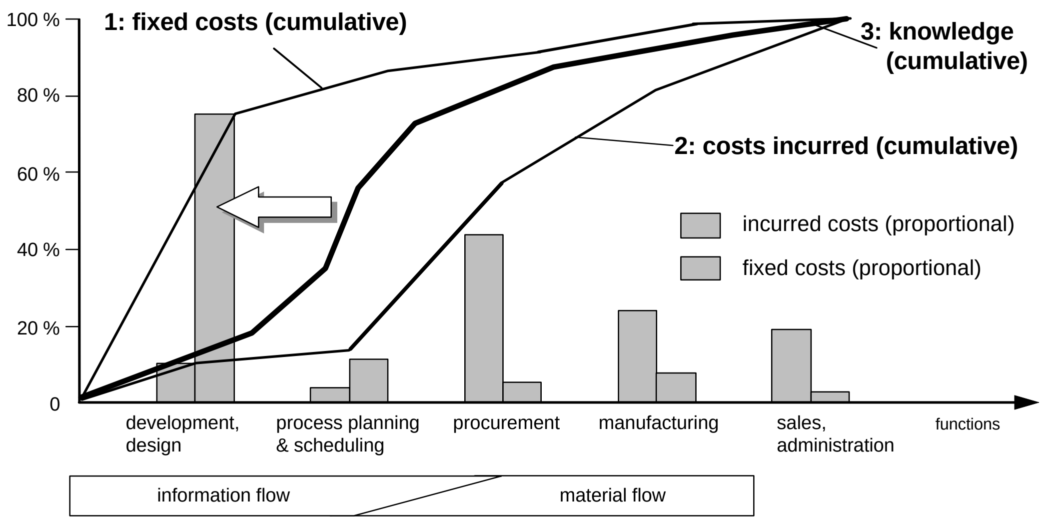

Figure 3. Availability of knowledge during product development

Knowledge can also be regarded as another resource for product development, which, for proper application, has to be developed and permanently adapted and updated. Therefore, there are a lot of analogies between the life cycle of a product and the life cycle of knowledge, **table 1**, which allow to handle knowledge along its life cycle with the same procedures and methods as products.

Table 1. Analogies between the life cycles of a product and of knowledge

Product life cycle	*Knowledge life cycle*
Product research	research for (new) knowledge
Product development	knowledge acquisition and development
Product shaping	knowledge limitation
Product design	knowledge description
• Original design	• create new knowledge
• Adaptation design	• (prepare) knowledge change / update
Process planning	Planning of knowledge creation and adaptation
Logistics	knowledge management
• Manufacturing	• perform knowledge creation and adaptation
• Distribution	• distribution and sharing of knowledge
• Search, find, provide	• search, find, provide
• Store	• storage of knowledge
• Service and support	• service and support of knowledge
Application and usage	knowledge application
Disassembly & recycling	revision and re-use of knowledge
Disposal / destruction	**???** of knowledge

Table 1 shows (besides the fact that knowledge is usually growing) that procedures must exist either to get rid of obsolete knowledge or to archive it in a way (e.g. linked to the product life cycle), that it doesn't disturb the actual development process. But it must be possible that knowledge can be re-activated when necessary, e.g. when liability questions arise where it has to be proved that the right decisions were taken along the development process.

In general, non-human systems may generate additional knowledge only in special cases (which have to be described very exactly in advance) by using combination rules, by deriving tendencies from huge data bases (e.g. when performing a patent research), or by specific use of contradictions (e.g. [6]). However, without an expensive interpretation, evaluation, and structuring of their results, such systems can't be applied yet always meaningfully.

When looking on possible computer support to knowledge-based design, a designer, first of all, needs a vision. This vision includes all possible information into a unique product model (which goes far beyond the representation of geometry), in order not to work with interfaces and data redundancy, thus fostering data consistency within the system, and providing a holistic support of the Engineering process. On the way to this vision, knowledge-based systems in product development will not replace established systems, but will exist additionally and parallel to CAx systems, Virtual Reality systems, and EDM/PDM systems [7]. In general, they may be classified into three categories:

- Product configuration systems (e.g. [3], [4]), which describe the links between the elements (products, components) of a given product structure tree either with fixed relations or with rules in order to model quickly complex family-of-parts. The description is mostly made using a special design language (which nowadays usually is an alphanumerical language, although graphical languages are under development). The language allows an ongoing adaptation of the configuration. Before the application of such systems it is necessary to prepare and to structure products with all possible alternatives and to acquire and to formalise the relations and rules between the elements. Most of original design today is made by configuration and combination [8]. Examples from automotive and aerospace industry applying these configurators show a dramatically decrease of product creation time [7], especially when, by using standard geometric modellers, not only the product structure but also the resulting product model is created and handed over to a CAD system.

- Process modellers and navigators, which administrate and support the processes within product development. These aids additionally shall provide in context the information and tools needed to perform a certain activity, thus allowing to navigate through the (chaotic) design process, to propose process parallelisation when reasonable, and to monitor the actual status within the processes. These approaches are mostly based on a generic process model of Engineering, as e.g. in [9], [10].
- Systems providing the appropriate knowledge in context to the actual stage of design (knowledge repositories), which, in future development stages, shall acquire a part of the knowledge being generated along the process (mostly by building internal knowledge patterns and knowledge topologies).

3. Characteristics of Knowledge-based Product Development

Computer support in product development resulted among others in a tendency of shifting activities from later phases in the product life cycle to the earlier phases like product development, thus making e.g. Design for X become more and more important. This tendency increases with the application of knowledge-based design. Especially the quality of decisions can be improved by better process knowledge and product know-how being provided more completely and earlier in time. In addition, applying simulation and animation routines already in early development phases can check decisions being taken easier. For example, virtual product development can't be applied completely without the appropriate knowledge provision.

One of the most important benefits is the increasing possibility of discovering more possible or hidden errors in the design of a product at very early stages, thus improving FMEA applications. **Figure 4** shows the well-known graphs of the (decreasing) probability of error detection in a standard environment in relation to the (increasing) costs of error correction. With the provision of appropriate knowledge in product development in conjunction with (mostly) simulation and animation tools the probability of error discovery is moved forward-in-time thus allowing detecting more errors already in development [7].

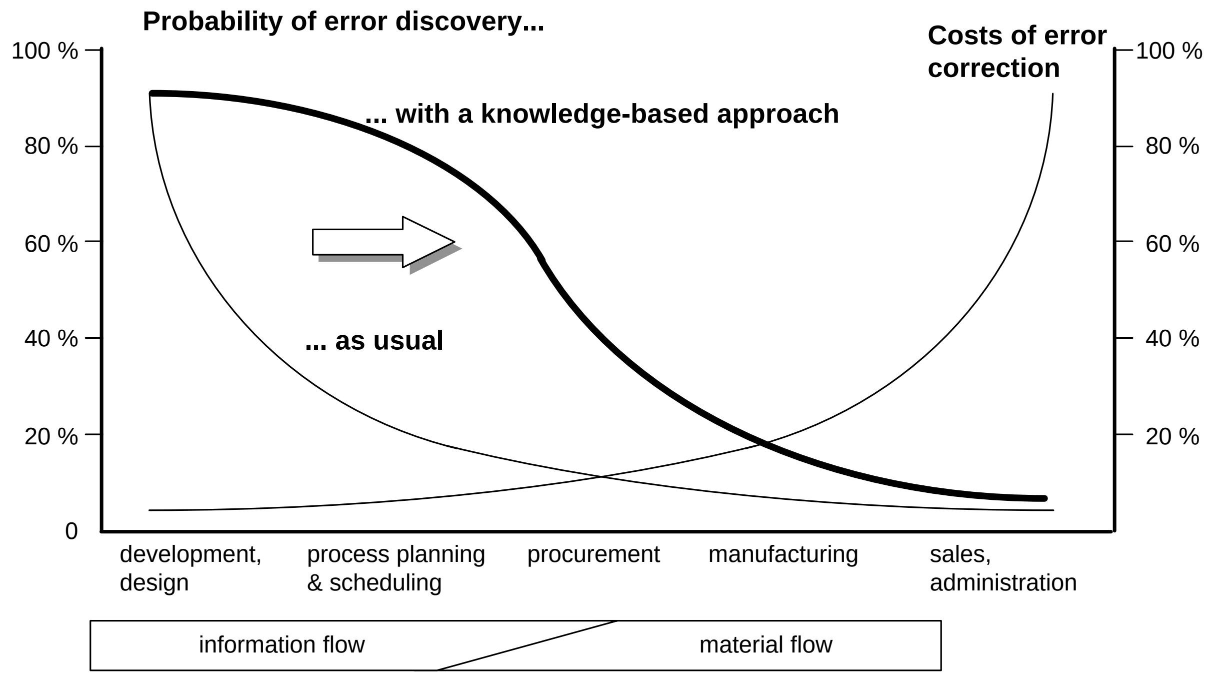

Figure 4. Improved error correction in knowledge-based product development

Besides product configuration systems, other tools and approaches for knowledge-based design are either already existing or will be provided in the near future, e.g.:

- Features to support synthesis and layout, which "decide" e.g. how two components shall be linked together, which calculate the stresses of the link, which select the link elements and place them appropriately into the product model, and which adapt the environment accordingly [11],
- Functions supporting a permanent quality assurance ("Design Spell Checker") by checking the given requirements against their respective fulfilment. These checks concern e.g. shapes against manufacturability (i.e. Design for Manufacturing), manufacturing processes against cost limits (i.e. Design to Cost), and the consistency of a feature (when inserted, edited, deleted) with its environment, thus supporting the earliest possible detection of errors and mistakes (figure 4), assuring design consistency and fostering design improvement,
- "Intelligent" component catalogues on the Internet which not only support the selection of a component based on a given problem description but which as well can download the CAD model of a selected component into the system of the user. They can support its embedding into the existing product model, can prepare proposals for this selected component, and, in case of an order placement, can handle the order processing automatically [13],
- External knowledge sources which may serve as pools of ideas providing results not only from research and development but also standards and best practices from industry (**figure 5**) [14]. A good example is the competence network of the "Berliner Kreis" (bkkn), which provides research and development results from its member institutions on the Internet [12].

As any kind of computer support, knowledge-based design will need high expenses of provision and maintenance. Among others, open issues concern knowledge representation, acquisition, management, and service:

- How shall knowledge be represented – with a (not necessarily formal) language, as symbols, or in a notation format? This representation should be almost universal but easy to adapt to specific needs, nevertheless independent from a user society. They should be transportable between systems, should be easy to be handled even by the end user (although the basics would have been created by specialists), easy to be maintained, and, finally should have only a small resource consumption.

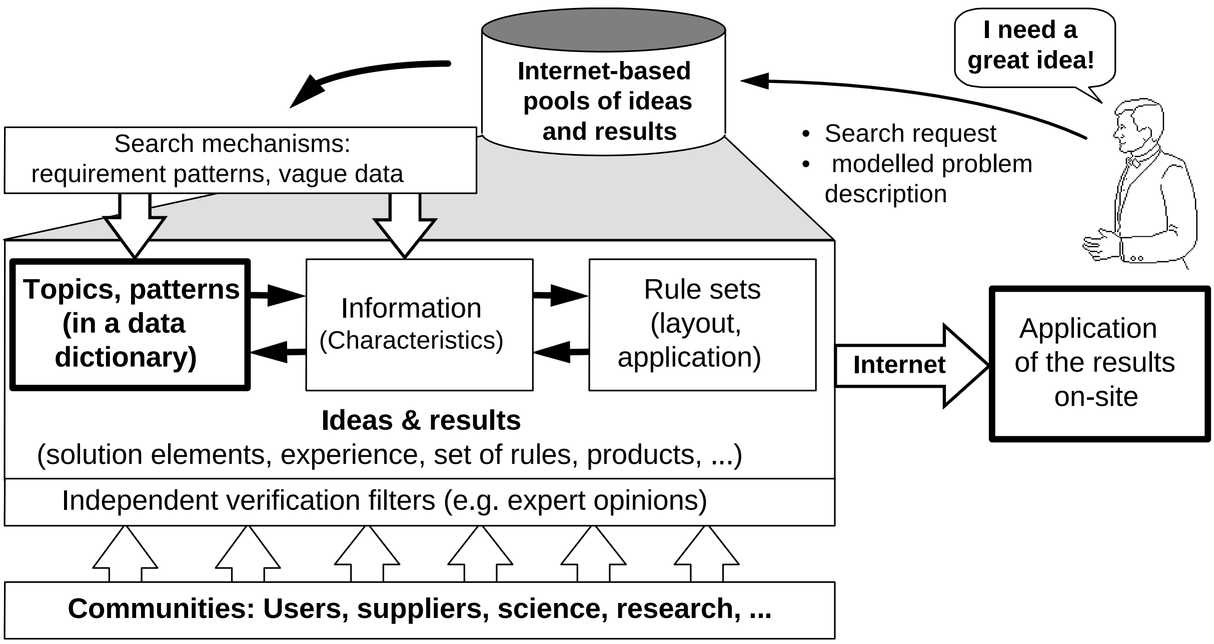

Figure 5. Pools of ideas on the Internet

- How shall knowledge be acquired – by manual interpretation and input into a system, by pattern and topology recognition from existing sources, or (mostly automatically) during the product development process without bothering the designer but with leaving him the responsibility of which knowledge is worth to be acquired?
- As it is of the same value to learn from success and from mistakes, is it possible for a system to "learn" automatically from development dead-ends and errors (e.g. in conjunction with the Design Spell Checker)?
- If geometry can be parameterised in a CAD system, is it accordingly possible to parameterise knowledge in order to provide a broader application base (by analogous transfer to other problem areas) and to minimise resources?
- How shall rules, methods, information, and data be provided in context to the designer?
- Based on the experience with product development data management in EDM/PDM systems, is it possible to transfer these approaches to knowledge management accordingly?

4. Conclusion and Outlook

Design can be described as "the art of making a plan/sketch/model to work from" and "the arrangements of parts, details, forms, etc. especially so as to produce a complete artefact" [15]. There is no statement on the kind and limits of methods, procedures, and tools to support design as long as these improve the design and its results. Therefore knowledge-based design is the logical evolution of computer-aided design as the provision of knowledge at the earliest time possible along the product development process allows not only sound decisions but fosters as well the complete digital representation of the life cycle of a product.

- In CAD systems, new developments in Parametrics will include the application of rules in the relations between the parameters. Thus the product configuration approach is re-used on a lower level as a parameter configuration approach.
- Because features are based on Parametrics ("features" in the sense of [11]), the parameter approach can be used as the basis for a rule-based configuration of features. As features in the sense of [11] may describe any kind of elements (i.e. products and components, not only partial/detailed shapes) it will be possible to create a complete product by rule-based configuration of existing features.

Research has been done for appropriate theories, methods, approaches, frameworks, and scenarios for knowledge-based design. Discrete tools (e.g. knowledge-based process modellers, features, product catalogues, idea pools on the Internet, etc.) have been developed or are already (at least partly) available. However, appropriate working techniques and modelling methods, which would support an "every-day-work" with computer-provided knowledge, are still missing. Additionally it is necessary in the future to elaborate benefits and advantages of knowledge-based design in regards to industrial applicability more clearly.

Especially at applications in such a sensible area as product development there are additional questions to be solved, e.g.:

- Who will provide knowledge on a commercial base ("knowledge broker"), what is their origin, where is the source of their knowledge? What is an appropriate price for knowledge? Who is liable for damages caused by the application of such provided knowledge?
- How can knowledge be provided, e.g. as knowledge "atoms", as elements of a knowledge building block system, or as agents (small turnkey programs which are downloaded from the Internet providing support on-site, deleting themselves after application)?
- Who will be responsible for knowledge actualisation, service and support? Shall this be done by the users themselves or by (independent) user groups? What are the appropriate quality assurance means and criteria? How often shall an update take place?

Within knowledge-supported product development, decisions can be taken closer to the earliest possible phase, with higher confidence and security. The designer's area of freedom will surely increase (although the danger is already visible that the increased area of freedom will be minimised by increasing demands of shorter throughput time and higher efficiency as it has happened with all applications of any support means). Therefore it is necessary that designers very clearly describe their proper needs (knowledge presentation, applicability, security and confidence, etc.) to those providing the knowledge, and that they feel responsible when using the various possibilities of knowledge application along the product development process.

References

[1] Gilbreth, F. B.: Field System, Private Publishing New York 1907 and Management History Series No. 30

[2] Váncza, J.: Artificial Intelligence in Design - A Survey. In: Kals, H. und van Houten, F. (editors): Integration of Process Knowledge into Design Support Systems. CIRP International Design Seminar 1999

[3] Breitling, F.: Wissensbasierte Konstruktion - Das Berufsbild des Ingenieurs im Wandel. Presentation within the Colloquium of Integrated Product Development, Magdeburg University 1999 (additional information may be found at http://www.ktiworld.com)

[4] Gittinger, A.: Intent! bringt Konstruktions- und Fertigungswissen zum Kunden. Presentation within the Colloquium of Integrated Product Development, Magdeburg University 1997 (additional information may be found at http://www.heidecorp.com)

[5] Vajna, S., Wegner, B.: Autogenetische Konstruktionstheorie - Evolutionärer Ansatz für eine erweiterte Konstruktionstheorie. Magdeburger Wissenschaftsjournal 2/99

[6] Linde, H.: How to Generate World Class Solutions - WOIS's Approach. In: Proceedings of ICED 97 Tampere/Finnland, edited by A. Riitahuhta

[7] Cooper, S., Fan, I., Li, G.: Achieving Competitive Advantage through Knowledge-Based Engineering (A Best Practice Guide). Knowledge Technologies International, Leamington (GB) 1999

[8] Birkhofer, H.: Konstruieren mit Maschinenelementen – in 30 Sekunden zum kalkulierten 3D-Grob-entwurf. 15. MeKoME-Workshop, Rigi-Kaltbad (Schweiz) 1998.

[9] Blessing, L.: A Process-Based Approach to Computer-Supported Engineering Design. Private Publishing Cambridge 1994

[10] Vajna, S., Freisleben, D., Scheibler, M.: Knowledge Based Engineering Process Model. In: Proceedings of ICED 97 Tampere/Finnland, edited by A. Riitahuhta

[11] Vajna, S.: Feature-based Design. In: Proceedings of the 1997 Unigraphics User's Group Spring Conference Anaheim/USA

[12] http.//www.bkkn.de

[13] Vajna, S., Schabacker, M.: Interpret - A Communication Tool with Customers via Public Network. In: Proceedings of 17th ASME Computers in Engineering Conference Sacramento/USA 1997

[14] Vajna, S.: Stand und Tendenzen der rechnerintegrierten Anlagenplanung. In: 19. Konstruktionssymposium der DECHEMA e.V., Anlagenentwicklung - Trends in der Gestaltung von Systemkomponenten zur Prozeßoptimierung. DECHEMA Frankfurt 1999

[15] Webster's New Universal Unabridged Dictionary (2nd edition), Simon & Schuster New York (USA) 1983

Prof. Dr.-Ing. Sándor Vajna
Lehrstuhl für Maschinenbauinformatik, Institut für Maschinenkonstruktion
Otto-von-Guericke-Universität Magdeburg
Post Box 4120
D-39016 Magdeburg
Tel. +49-391-67-18794
Fax. +49-391-67-11167
E-Mail: vajna@mb.uni-magdeburg.de