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

PRESENTATION OF ENGINEERING DESIGN RELATED KNOWLEDGE

Jörg Feldhusen[1], Johannes Lemburg[1]
[1]RWTH Aachen University

ABSTRACT
The presentation of engineering design related knowledge has a long time history by technical documentation particularly technical drawing and technical illustration. The complexity of the tasks to solve and the sharing of knowledge within collaborative engineering increase the need to present the output of the early steps of the design process in a human and computer compatible way. The assistance of computers within the concept phase of product development has until now been limited to mere data management. This paper depicts a user centred approach to visualize design engineering related knowledge within the conceptual design phase by computer aided sketching of principle solutions. An approach is made to store the content of prescriptive sketches of principle solutions with a language of appropriate symbols connected by a well defined but not rigid semantic. By this the outcome of the conceptual design phase can be represented by an object-oriented kind of product-data within product management systems.

Keywords: implicit knowledge, collaborative engineering, mental model, computer aided sketching, prescriptive sketches.

1 INTRODUCTION
The presentation of engineering design related knowledge aims to push the frontier from implicit towards explicit knowledge to achieve a higher quality of documentation and set the basis for more complex engineering tasks challenging companies and educational organizations. Several pushing and pulling factors cause the need to develop a more standardized, machine and human compatible way to deal with highly abstract expert knowledge as it is crucial to develop innovative products [1], [2].
Pulling factors are the rising complexity of the tasks to solve and the specialization of the participating disciplines. The complexity of the tasks is on the one hand founded in the amount of constraints associated with the products to be developed and on the other hand the networking of collaborative engineering. Specialization is one answer to the exploding amount of human knowledge. This development can also be seen within the discipline of engineering design and enhances the need to share generated knowledge.
Pushing factors are the development of increasingly powerful hard- and software and newest discoveries of disciplines as knowledge management and scientific visualization. The ongoing price decline of hardware, which is able to run complex programs, is the basis for introducing specialized expert knowledge in small and medium-sized enterprises. These programs have to be of high usability to be capable to transfer the knowledge to the user. The research of scientific visualization and human-computer interaction shows the potential to handle complex and abstract knowledge of engineering design in a user friendly way.

2 PROPERIES OF DESIGNS AND DOCUMENTATION
How does a design engineer explore a given machine? Is it possible to understand a design just by its documentation? In fact the documentation lacks some properties that the generated design in reality features. These properties, like rattling, squeaking or friction, are of a kind that reveals only under operating conditions. That's why an experienced design engineer will have a look at the machine itself, fiddle about it and maybe disassemble to conceive it. Some functions of parts can only be understood with the functional interplay of parts in the assembly in mind but others can only been seen

in combination of the real design with the documentation. For example missing or unwanted features can only be detected this way because the technical documentation describes an intended ideal state of the design that never will occur in reality.

This interplay of documentation and the real word design is not possible when a new design is to be developed. In this situation the experience of the design engineer and his ability to imagine the interplay of the parts fills the gap of documentation. Sketching a machine the designer "sees" it working. Unfortunately this may not be possible when he is just working on a sub area of a problem without knowing all needed data or if he himself lacks the required experience. This situation may occur in large-sized and complex engineering projects in a collaborative working environment. The lack of knowing enough data often roots in the problem that this data has not been precisely enough defined up to this point of the project and therefore can impossibly be delivered. Models of a higher abstraction level may give the right environment for generating a given sub solution without being dependent on precise data. But how to document these models and their environment?

Neither the finished design nor its technical documentation contains all of the intentions of the design engineer. Of course the design engineer is free to document his thoughts and intentions, but because this can mainly been done with words in written form or with hand made sketches this will not been done in real life because of the lack of time or fleetingness. The ideal case is that the finished design embodies all the intentions of the design engineer and a subsequent observer can reengineer from the designs properties what the designer did intend and what he did not. Unfortunately this correlation of feature to intention can bee ambiguous. Therefore the technical documentation, in particular the manufacturing documentation, and the finished product are the final result of the design process but are not the complete data to understand all of it. To describe the whole design process of a product there is the need to explicit document the design engineers intentions and ideas.

Furthermore the ambiguity of the correlation from product property or feature to the underlying designers intention eliminates the possibility of a overarching quality management within the design process. How to detect a forgotten function, if the final design does not contain a feature that embodies this function? How to detect the correlation from a feature to a function if the designer failed to implement this function properly? Of course the existence of functions or features described in the requirements list is easy to check, but not all requirements are documented. There are also procedures to capture requirements of the customer that he is not aware of at the beginning of the design process. But there are a lot of implicit requirements towards the product life cycle e.g. the manufacturing process that the design has to fulfil. Often these requirements contradict each other and therefore a clear definition of the chosen balance of them has to be done in an explicit and well documented way. There are methods within the systematic engineering design to describe abstract concepts of the design as e.g. the function structure or the solution principle but these concepts are, if at all, just deposited in the technical documentation. The creation of these concepts nowadays is not aided by computers and can only be checked for plausibility by humans.

3 MENTAL MODELS

The main challenge of displaying design related ideas and concepts are their diverse characteristics. The results of the early steps of the design process (function structure, solution principle, and preliminary embodiment design) are coherent. By contrast the mental model of a design engineer is a conglomerate of sub-solutions. These sub-solutions can differ in grade of abstraction and do not have to be consistent nor complete. This is expressed in the way sketches display solutions, where symbolic representations coexist and are connected with partial finished embodiment designed features. These solutions can not exhibit real word analogues and are not able to be expressed by today's CAD-models. Even incompatible solutions may coexist in preliminary sketches. Sketches feature the ability of focusing on single partial problems by suppressing unwanted details and illustrating the required quantity of degrees of freedom.

The inconsistency and incompleteness of human mental models is the key, why computer aided tools did not make their way into the early steps of the design process. Computer based product data has, at least on the level of data representation, to be coercively consistent. For example the idea of an open volume is impossible for a computer to grasp. The computerised representation of a volume is a space bounded by surfaces. As long as these surfaces do not stick together the composed structure is hollow and can not have a volume-property. For a human this inexactness is no trouble. Imagine a cube, set on its vertex as shown in Figure 1 on the left side.

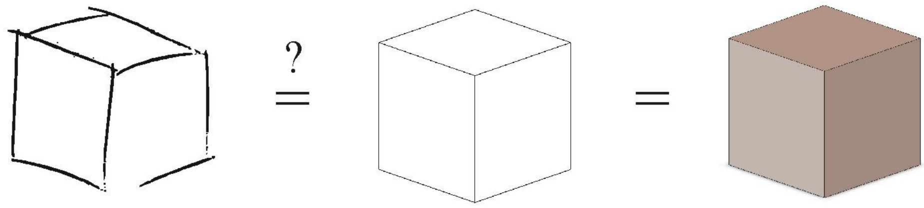

Figure 1. sketches of a cube

How does the rear view look like? You do not know yet. Although you even may not want to define it, you know that you are looking on a cube with a volume.
This inexactness is of major importance for finding a solution within engineering design, because it allows a kind of adaptability by just defining some of the relevant features. It focuses but leaves possibilities for variation. Furthermore the creation of such an inexact model is of less cognitive load.
To display the cube as in Figure 1. on a computer, you at least have to:

- Create a volume. Therefore you do have to define the rear view. You do have to create a volume, which features six connected surfaces.
- Create three surfaces. These surfaces have to stick together precisely otherwise you will see two lines for each edge. Furthermore you have to create the surfaces one by one.
- Draw single lines. Therefore you do have to draw every line one by one. The lines do not have to be connected.

The idea of a cube, in particular the volume property, is in last two described ways just implicit existent in the mind of the human creator. In fact in nowadays kind of computer aided drawing software it would be utterly impossible to add a volume property to a bunch of surfaces or lines. To display and document the intermediate results of early steps of the design process on a computer based systems the difficulty to represent incomplete and inconsistent mental models of humans by precise data models has to be managed.

4 CONCEPT OF COMPUTER AIDED SKETCHING

The main advantage in using computers for displaying or documenting data is its ability to let the user interact with the data. Text and graphics on static media can not change. Even most data processed by computers is of this static kind, like text, pictures or even film and sound files. The latter ones are just animated but not interactive in that way that the content shows behaviour towards the user.
Computer Aided Drawing data is on its way to be an interactive kind of data representation. The parts and assemblies can be moved, exploded and explored by analyzing properties as weight, centre of gravity or stiffness. With an underlying skeleton or collision detection enabled a quite intuitively way of examining the kinematics of an assembly can be done by dragging the parts around and seeing what happens. This interact-and-see-what-happens is exactly the way a design engineer identifies the properties of a given machine that are not stored within the technical documentation.
The design process according to Pahl/Beitz [3] features the phases planning and clarifying the task, conceptual design, embodiment design and detail design as shown in Figure 2. The outputs of the different steps of the design process are the requirements list, the function structure and principle solution, preliminary embodiment design and detail design with fully defined production documentation, e.g. a 3D CAD model as the basis for generating drawings, parts lists and operating instructions.

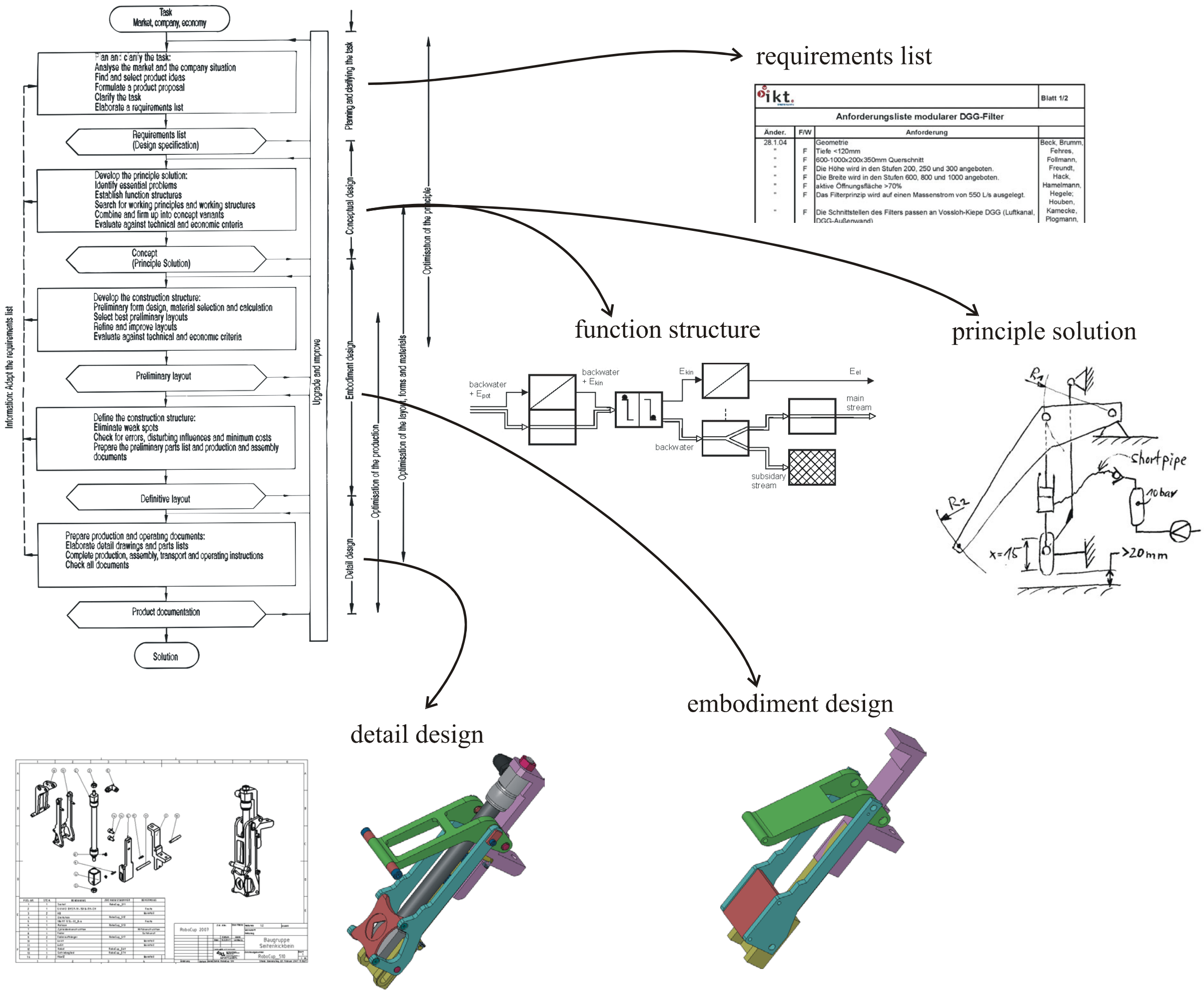

Figure 2. outputs of the design process phases

Today the only way to display a function structure is to draw symbols manually, connect them to a structure and append additional textual metadata. The quickest and most intuitive way to create a function structure within the conceptual design phase is to sketch it by hand. The use of computer in this case is limited to a more accurate but also more time consuming drawing. To implement interactivity to the creation of such structures, the symbols of the single functions have to have a logic built in that allows only possible connections of functions. This way a basic plausibility check can be implemented. Furthermore the symbols may be linked to certain parts of the principle solution or further on to the CAD-model to document the fact that all the functions are assigned to a physical effect and an effect carrier and later on to working surfaces.

The next step within the conceptual design is the effect synthesis where every function is realized by a physical effect in combination with an effect carrier and the qualitative embodiment parameters of the working location. The output of this step is the solution principle. Figure 3 depicts different kinds of representations of principle solutions varying in grade of abstraction.

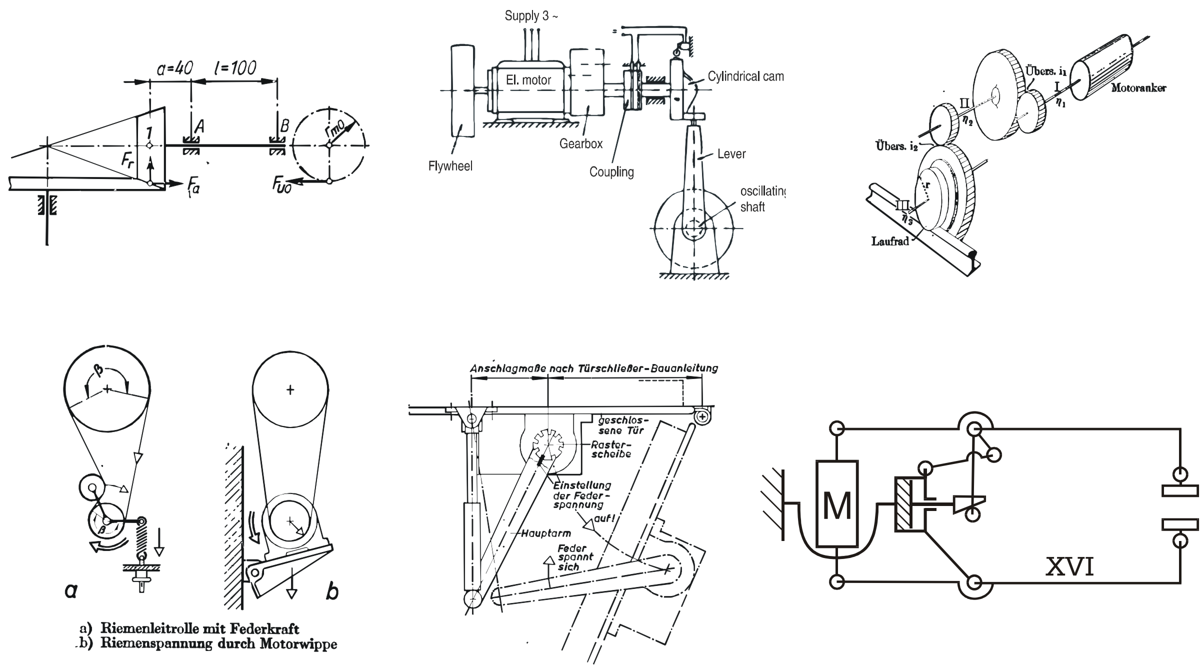

Figure 3. principle solutions

The important step of generating the fundamental properties of the product are up to this day done mainly by drawing non-standardized handmade sketches that often need auxiliary explanations by words or text. The use of computers is restricted to simple administration and storage of the documentation of the generated results. At this point the greatest benefit of interaction may be achieved.

Sketches of principle solutions are able to contain symbols, icons and semi-embodiment presentations [4] of parts all coexistent and working together. The less is known about a part the higher the level of abstraction of the depiction is (Figure 4).

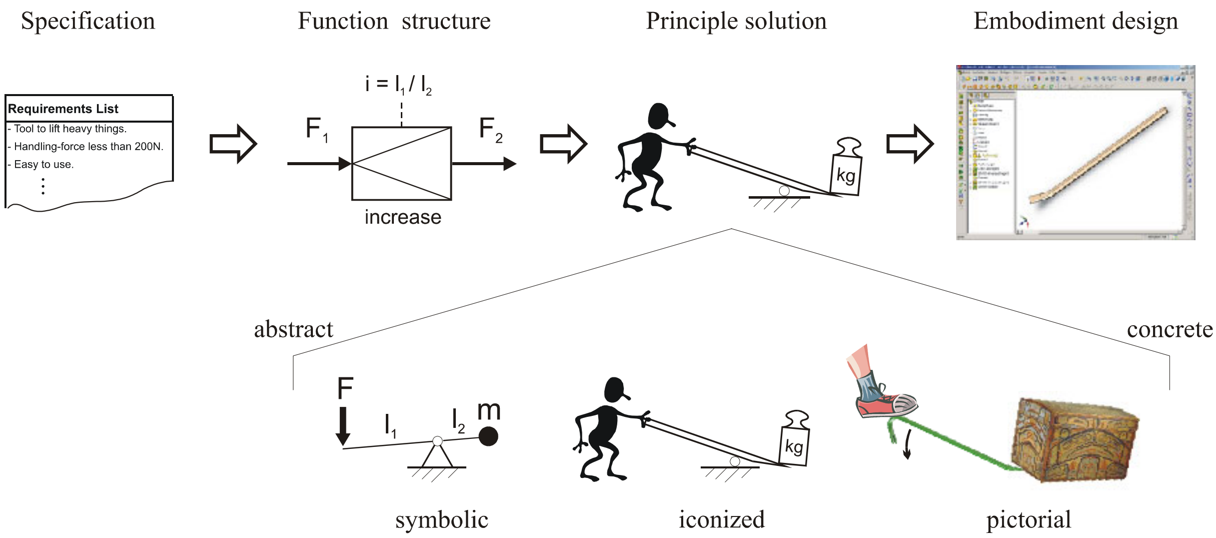

Figure 4. grades of abstraction

The possibility of ambiguity and inconsistency within sketches is the reason why there is up to nowadays no standardized language for creating sketches. The coexistence of different grades of abstraction within one picture is difficult to handle for accurate software. The key is, not to expect one single closed solution. The items of an imaginable computer aided sketch may vary their number of constraints to define according to the context they are used in. If a plausibility check is not possible because of an inconsistency in the solution or too less constraints defined the software still can be used for generating concept documentation more standardized and efficient. This interactive way to create a structure of abstract items playing together in a simplified and controllable environment have already been implemented in puzzle games as “the incredible machine” [5] or “widget workshop” [6] (Figure 5).

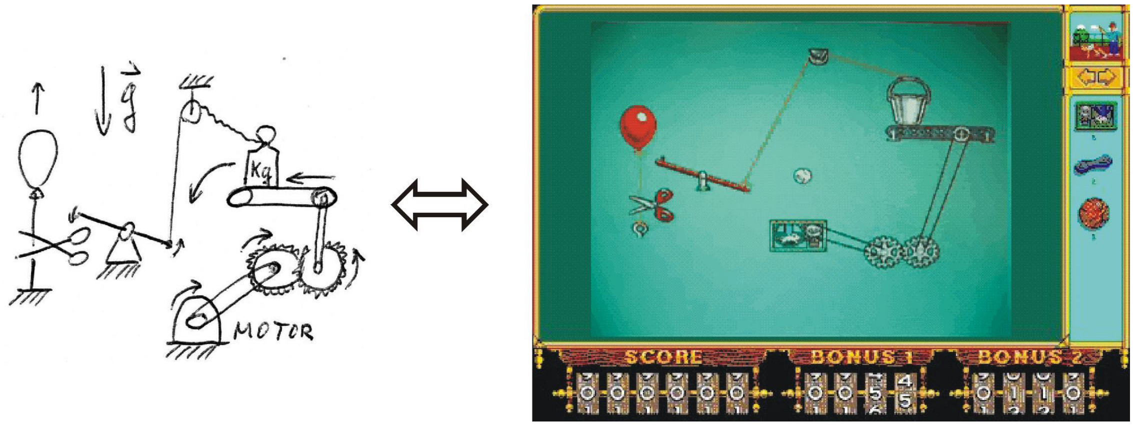

Figure 5. principle solution and puzzle game “the incredible machine”

In these games an action has to be performed by creating a chain reaction of given devices as generators, bowling balls, ropes etc. In these games, unlike in reality, the number of devices and the amount of their properties and constraints is limited to avoid a too large solution space.
Transferred to serious engineering design this kind of displaying principle solutions can lead the way to generate plausible documentation, because shafts do move and the action of levers and pistons can be explained without the use of words. Once a working principle solution is generated, it is imaginable to increase the complexity of the sketch and simulate a more complex environment by adding quantity to the used parameters or adding additional parameters as for example friction or stiffness.
The gap between the output of the conceptual design phase and the beginning of the phase of embodiment design may be bridged by a less rigid kind of product data. While the principle solution features symbolized and quite abstract items, the preliminary form design requires models that embody real parts. In today’s CAD software there is no in-between to parts with volume and symbolic items.

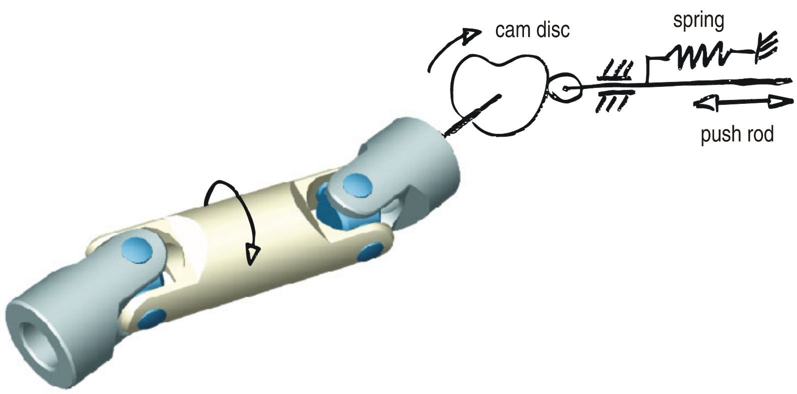

Figure 6. hybrid depiction

A kind of hybrid depiction as seen in Figure 4 may be the answer to the problem of combining CAD-parts and assemblies with not-yet embodied symbolic objects. While the cardan-shaft is an already embodiment designed assembly arranged within a spatial working space, it nevertheless transmits rotational movement to the two dimensional cam disc. Adding a mass property to the push rod and changing the stiffness of the spring the dynamic effect of fluttering can be visualized by increasing the rotation speed.
Another possibility to insert the abstract information of purpose into three-dimensional CAD-models is to reduce the products shape down to its underlying principle solution. For that purpose all boundary surfaces are rendered transparent or removed at all and the working surfaces and channel structures are highlighted by appropriate colours.

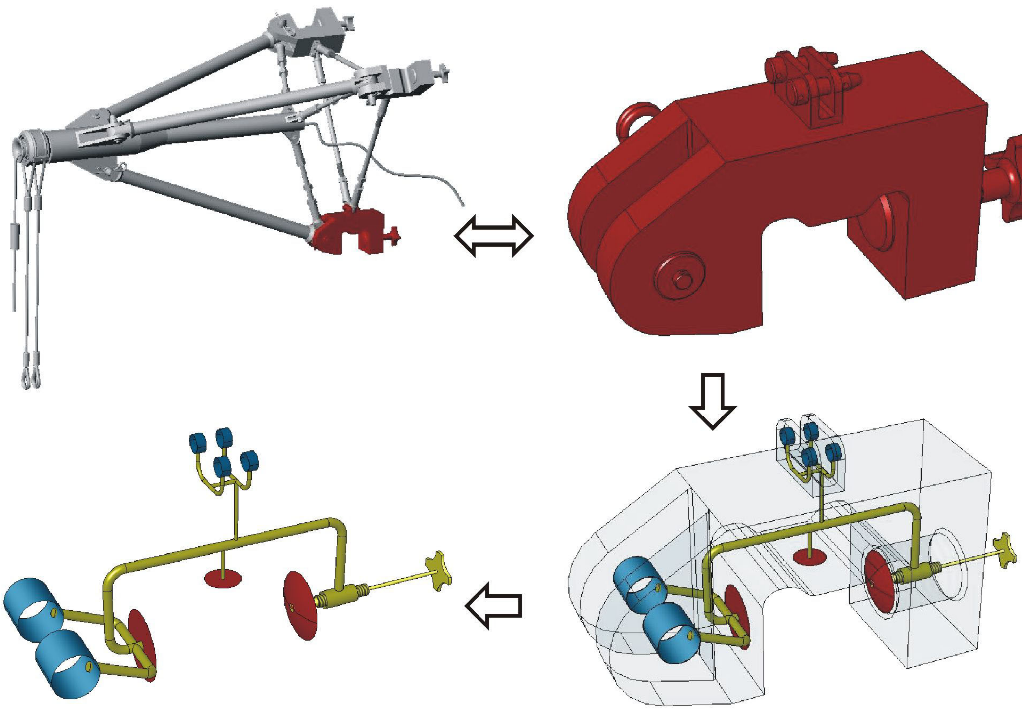

Figure 7. depiction of working surfaces & channel and support structures

Figure 5 depicts an example of a clamp supporting a mounting. The shape of the clamp has been shrunk down to its working surfaces & channel and support structures [7] to display a spatial and semi-embodiment kind of principle solution.

IMPLEMENTATION

As the basis for this type of documentation of design process results a product data representation has to be developed with the ability to store not only the single kinds of results but also the relations between them. This can be achieved by implementing an object-oriented information structure. Such information structures have already been introduced to represent CAD-data, where the single items of a model, like edges, surfaces up to parts and assemblies in combination with additional metadata are instances of well defined classes.

To transfer this object-oriented approach to the content of sketches, first of all the most common depictions of items of principle solutions have to be described. A standardised language of symbols and pictograms with different grades of abstraction for recurrent items within principle solution sketches will be the outcome. This language can not and must not be self-contained, because of the unlimited possibilities of real world design tasks. Similar languages of symbols are used for instance in the field of electronics, where schematic diagrams are the electrical analogue to mechanical principle solutions. The major difference between both is the amount of pre-definable modules. As it is obvious, that there can’t be an all-embracing canon of pre-defined symbols to depict every possible design solution, the language itself has to be adaptable to emerging challenges.

Together with the standardised symbols a semantic of the embodied objects will be developed. The object will have properties and methods, which link the objects among each other and reference them to data resulting from other steps of the design process.

Figure 8 depicts an example of an object named “weight” as it is used in the principle solution sketches of Figure 4. In the beginning of the design process weight might be just an abstract value, which is defined by the requirements list and not yet a resulting property of an embodied CAD-model. Therefore the depiction of weight can only be of an abstract kind. Already at this point the object weight can have methods as “accelerate”, which itself is dependent on the weight-value and has influence on other methods as for example “move”.

The use of such an approach to represent sketches on a computer based system will be restricted to distinctive type of sketches. Ferguson [8] differentiates between thinking, talking and prescriptive sketches. The first is used by the designer while the creative process of generating new ideas to catch fleeting, non-verbal thinking. The second is a pictorial way to transmit information from a sender to a receiver, often in combination with vocal explanations. Both kinds of sketches are situational and seldom follow predefined rules. The prescriptive sketch instead stores information for later use, namely for further developing and specifying. It contains more details, features qualitative embodiment parameters of the working location and can be seen as a preliminary stage to the technical drawing.

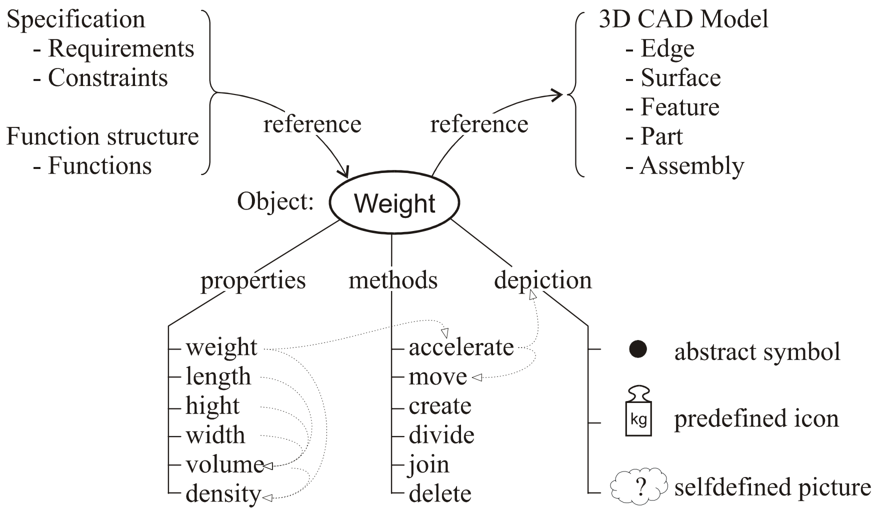

Figure 8. example of an object class

It has to be emphasized that the creative process of generating thinking sketches to develop new ideas or scribbling talking sketches to communicate shall not be replaced by some kind of restrictive sketch-drawing-software. In this paper an approach to access the knowledge within the pictorial outcome of the conceptual design phase is developed. With a language of appropriate symbols connected by a well defined but not rigid semantic it is possible to represent the content of prescriptive sketches of principle solutions by an object-oriented kind of data within product management systems.

REFERENCES

[1] Kroll E., Condoor S., Jansson D. *Innovative Conceptual Design*, 2001 (Cambridge University Press, Cambridge)

[2] Roth K. *Konstruieren mit Konstruktionskatalogen*, 1982 (Springer-Verlag, Berlin).

[3] Pahl G., Beitz W., Feldhusen J., Grote K. *Konstruktionslehre: Grundlagen erfolgreicher Produktentwicklung ; Methoden und Anwendung*, 2007 (Springer-Verlag, Berlin).

[4] Tjalve, E., *Engineering Graphic Modelling*, 1979 (Butterworth & Co Ltd., London).

[5] Ryan K.; Tunnell J. *The incredible machine*, 1993 (Sierra Entertainment, Los Angeles).

[6] Wright W., Braun J. *Widget Workshop: A Mad Scientist's Laboratory*; 1995 (Maxis Communications Berhad, Emeryville).

[7] Matthiesen S., *Ein Beitrag zur Basisdefinition des Elementemodells “Wirkflächenpaare & Leitstützstrukturen” zum Zusammenhang von Funktion und Gestalt technischer Systeme*, 2002 (Institut für Produktentwicklung Karlsruhe Universität Karlsruhe (TH), Karlsruhe)

[8] Ferguson E., *Engineering and the Mind’s Eye*, 1992 (MIT Press, Cambridge)

Contact: Johannes Lemburg
RWTH Aachen University
Chair and Institute for Engineering Design (IKT)
Steinbachstraße 54B
52074 Aachen
Germany
Tel: Int +49 241 8027351
Fax: Int +49 241 8022286
E-mail: Lemburg@ikt.rwth-aachen.de
URL: http://www.ikt.rwth-aachen.de/