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

# A CONCEPTUAL DESIGN SUPPORT SYSTEM USING PRINCIPLE SOLUTION ELEMENTS

Sören Wilhelms

## Abstract

The parametric modelling paradigm has become important in shape design and simulation. A prerequisite for product development with an integrated digital model is the availability of parametric models even for early design stages. This paper presents a parametric information model for conceptual design, consisting of requirements, hierarchical functions, function structure, principle solution elements and concepts, and a computer tool prototype based on the model.

*Keywords: concept generation, early phases of design, constraint-based design, functional modelling*

## 1 Introduction

Progress in parametric technology has led to powerful computer tools for later stages of a design process, e.g. parametric 3D volume modellers, parametric machine elements, manufacturing features or tools integrating shape design and calculations [1]. Since some time, research has been trying to transfer the shape feature thought to the earlier phases of conceptual design (principle features) [2].

According to literature, e.g. [3], the early stages of a systematic design process comprise specification of requirements, identification of functions and sub-functions, search for principle solutions and synthesis into concepts. In order to create significantly new or improved solutions, the need of abstraction (functions) and consideration of several solution alternatives (concepts) are stressed ([3], ch. 6.2.1 and 2.2.4, respectively).

In practice, however, the time to obtain a solution is an essential factor. Classical design methodology as described, mainly oriented towards providing good or best quality solutions, is time-consuming. To increase practical applicability of design methodology, methods for design in shorter time, with less effort and increased quality are desired. The idea is to achieve these goals by reusing and instantiating suitable principle solution elements, but, at the same time, permitting easy addition of own elements in order not to inhibit diversity. The long-term objective is an extensible conceptual design support system enabling modelling and reuse of existing principle solution elements (e.g. from design catalogues [4]), their instantiation with parameters and values (e.g. by an object oriented approach) and the modelling of their relations (e.g. in a semantic network of constraints [5], both inside and between principle solution elements).

Usually, the design task is initially poorly described and understood. Starting from a need, requirements, functions and possible principle solutions are gradually determined and concre-

tised by identifying and constraining parameters and assigning values, in this way even enabling quantitative investigations once the qualitative structure has been determined. Early calculations describing a concept with focus on only a few key characteristics relevant for the concept stage (cf. [6]) are desired as early as possible. In this way, the existing methods for abstraction (e.g. function structures and functional decomposition) and synthesis (e.g. function/means tree, morphological matrix) with their combinatorial and qualitative focus are extended by first quantitative calculations [7].

Existing tools such as TechOptimizer access solutions through a pragmatic classification or keyword search. The provided solution segments vary regarding the level of abstraction and require usually work outside the tool for obtaining concepts.

This paper therefore intends to contribute to a semantic information model with higher formalisation for reusable, parametric principle solution elements permitting instantiation of elements as well as limited support for synthesis and detailing. The model is based on theoretical considerations on a number of cases and has been tested on the example of a specific design problem from industry. In order to verify the principle appropriateness of a parameter constraint network approach, a prototype for a software tool has been implemented.

# 2 Background

## 2.1 Design methodology

Design methodology provides discursive, product-independent procedures for advancing from the clarified task (design specification) via abstraction (function structure) to principle solutions (working principles) and concepts [3]. Results are elaborated in the transformation domain (function structure based on systems theory), organ domain (means, working principles) and part domain (component geometry, part structure).

Requirements specify the necessary functions and properties of the product by verbal description, values (where possible) and type (required/desired/goal, min/max/interval/exact) [3]. Functions describe abstractly the tasks a product is to perform in a solution-neutral way by verbal data, transformation of operands (e.g. material, energy, information in a general function structure or more specific physical quantities in a special function structure [4]). By connecting functions, a horizontal structure is obtained which describes their interrelations. The causal relationships (which function a function carrier solves, which sub-functions it is decomposed into) are described in a vertical structure, visualised e.g. by a function/means-tree (F/M-tree) [6]. Means describe the principle solution for one single sub-function and benefit from inclusion of operands and relevant formulas that describe the physical effect [8]. Concepts collect a combination of means to form an overall principle solution.

## 2.2 Information models

Considerable effort has been put into developing information models for conceptual design, which has resulted in e.g. information models for function structures [9] [10], organs [11] and design units [12] as well as holistic models with requirements, functions, flows, principle solutions, working principles and embodiment representation [13].

Constraint networks (e.g. Delta Blue [5]) can be used for engineering purposes to assign allowed values, sets or intervals to parameters and to connect parameters with each other by relations [14]. This serves to answer which output parameter values a set of input parameter values will yield, which input values a desired set of output values requires or to determine if

a current state is free of contradictions. Truth maintenance systems can be used to justify why a parameter has a specific value, to handle assumptions (e.g. default values for parameters that are kept until a stronger justification for a specific value is obtained) and value assignments (e.g. due to requirements or user choices).

## 2.3 Existing systems

Among previously implemented prototypes are demonstrators for bidirectional parametric relations between shape design and calculations [15], constraint networks and truth maintenance for variant design [5], F/M-tree based synthesis [16] and object-oriented design objects with data and methods in a semantic network [17].

# 3 Information model for conceptual design support

Support for conceptual design involves requirement capturing, functional analysis and concept elaboration as shown in Figure 1.

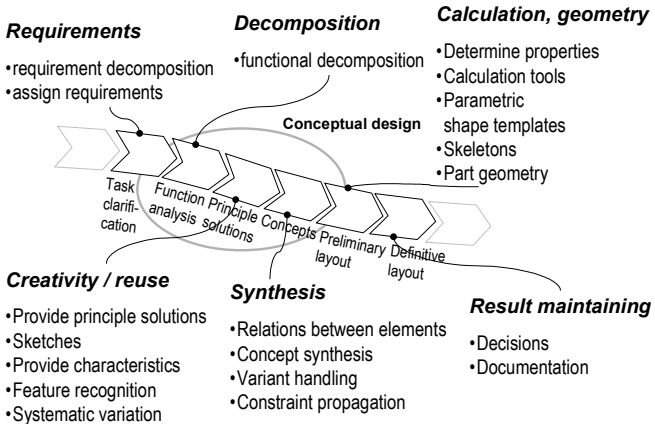


Figure 1. Tasks during conceptual design.

## 3.1 Task, requirements and goal

The aim is a flexible and generic information model for conceptual design suitable for a variety of design tasks, not a tool tailored for a special purpose or product. The qualitative description known from F/M-trees is extended by first quantitative reasoning and explicit capturing of design knowledge by parameters and their interdependencies (algebraic constraints). Assigning values to concept parameters allows instantiation. The underlying hypothesis is that such a framework for modelling and reuse of existing solution elements facilitates reducing cost and time consumption and increasing quality.

The goal is an interactive support with partly automated steps rather than a completely automated design process. The function structure is most important for the computer tool because it provides order and supports solution search and storage, cf. [9], as well as containers for later geometry and calculation models.

The model must enable parallel work with several concept alternatives to allow variation and comparison of different principle solutions.

Conceptual design is characterised by incomplete information, therefore support of a successive, interactive concretisation and work with yet incomplete solutions is desired.

Choosing parametric technology for the conceptual design model and describing the results with parametric solution elements and instantiated values facilitates:

- Traceability and quick assessment of consequences of changes based on a more formal description of parameter relations
- Checking requirement fulfilment
- Modelling of interrelations by a constraint network, cf. [5]
- Developing based on predefined principle solution elements, cf. [12]
- Extension by own solution elements, defined interactively at runtime
- Database storage for later reuse, which requires strong modularisation to enable the detachment from a prior context and the insertion into a new context
- Handling more complex solution fragments (e.g. complex drive chain consisting of energy supply, motor, torque converter etc.) in a hierarchy

## 3.2 Principle architecture and contents of the information model

The model is based on a succession of the phenomenon models requirements, functions, means and concepts (see Table 1). These objects reflect the gain in information and an increasing concretisation when carrying out a conceptual design task. By adding parameters and relations on parameter level between needs (requirements) and realisations (principle solution elements, form feature templates), determined numerical values can be stored and a constraint network connecting the parameters of means and functions is built.

Table 1. Elements of the information model, compared to VDI 2221 [18].

<table>
<tr><td rowspan="2">VDI 2221</td><td>Task clarification</td><td colspan="3">Conceptual Design</td><td colspan="2">Embodiment Design</td></tr>
<tr><td>List of requirements</td><td>Function structure</td><td colspan="2">Solution concept based on working principles</td><td>Structure</td><td>Rough 3D-shape</td></tr>
<tr><td rowspan="3">Partial models</td><td>Requirements</td><td>Functions</td><td>Means</td><td>Concepts</td><td>Assembly</td><td>Parts</td></tr>
<tr><td>Parameters</td><td>Parameters</td><td>Parameters</td><td>Choices, Values</td><td>Parametric parts</td><td>Parame-ters</td></tr>
<tr><td colspan="6">Constraint Network</td></tr>
</table>

In contrast to usual calculation tools expecting defined models, even underdetermined states can be described by modelling only some parameters and their relations in momentary solutions with partly defined requirements and partly defined/configured solution elements.

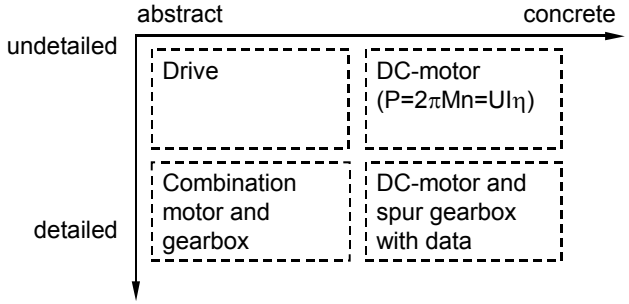


Figure 2. Degree of detailing and abstraction.

Progress from abstract to concrete descriptions of elements is done by successive addition of parameters and constraints (horizontal axis in Figure 2). The hierarchical decomposition is modelled by functional decomposition in a tree (vertical axis in Figure 2).

To enable execution of constraints, a formula parser and coupling to a symbolic calculator (Maple was chosen for the prototype) are added. In future, geometry handling is planned in a similar way. The resulting model maintains and documents semantic relations and values in concepts and the complete solution space, not only one final principle solution or sketch. The information model is shown in Figure 3 and further described below.

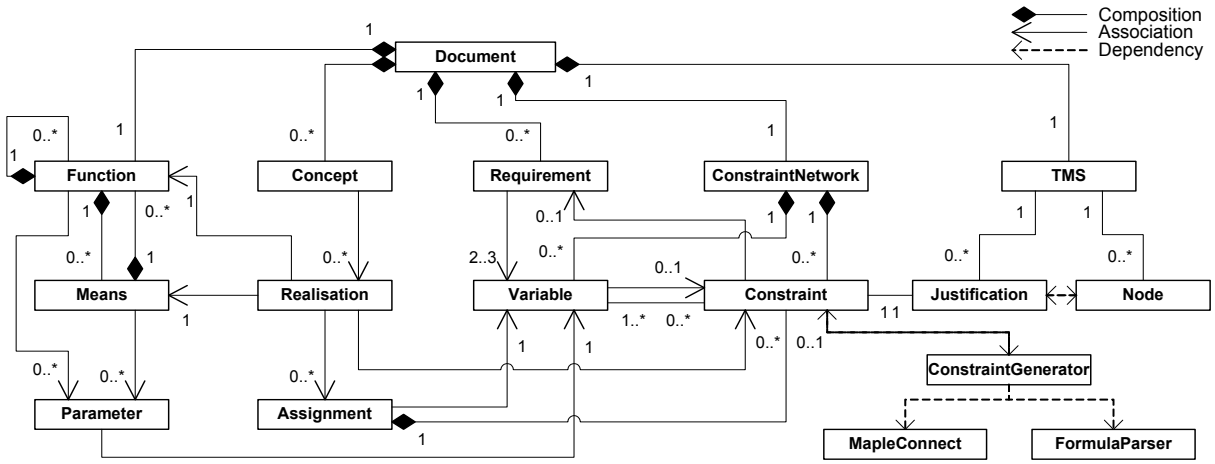


Figure 3. UML class diagram of concept model.

**Requirements** can be of goal-setting or restricting nature. Goal-setting requirements (functional requirements, e.g. output torque $M_t \geq 100$ Nm) are an input to the design task and are assigned to the parameter of a function responsible for their fulfilment (e.g. create torque). Restricting requirements limit parameters (e.g. limited diameter, $d \leq 50$ mm) and can even result from means (e.g. the initially acceptable diameter range being reduced to an exact value by a certain solution, $d \leftarrow 40$ mm). Requirements are connected to relevant functions and parameters in order to provide traceability, which addresses the problem of unclear requirements being a reason for negative outcome of conceptual design [19].

**Functions** are crucial for reuse, as they represent the context-independent part and are connectors for inputs and outputs across boundaries of principle solution elements. They even provide connectors for the transfer characteristics provided by solution elements.

A principle solution element consists of a **means** (verbal, qualitative description of the working principle, e.g. "cogwheels"), a set of parameters (e.g. $n_1$, $M_1$, …) and internal constraints (e.g. $i=n_1/n_2$). If present, internal constraints of a solution element determine the transfer characteristic from input to output parameters (cf. [10], p. 111) and correspond to a quantitative description of the working principle [3]. In connection with a database, internal constraints can be added and removed to represent the current degree of abstraction, cf. Figure 2.

**Parameters** are used to connect functions to a function structure and to assign input and output parameters to functions. The relations are modelled by external constraints connecting the parameters of functions or picking up function parameters of sub-functions inside a means.

**Constraints** of non-directed nature restrict one or several parameters, governing no or one parameter value at a time. Four types of constraints are used: requirement/function parameter coupling, value assignment, internal constraints between parameters inside a solution element and external constraints between parameters across the border of a solution element (cf. [9] generally for methods). Constraints are formulated as algebraic equality constraints, which

covers many properties (e.g. mass), but may require manual simplification of complex properties as e.g. manufacturing cost. Requirement constraints have desired and actual value and can be inequalities (exact, min, max, interval). Internal constraints (e.g. $i=n_1/n_2$ for a pair of cogwheels) are context independent and thus directly reusable. External constraints (e.g. $n_{motor}=n_1$ for the gear drive) are specific to the context and task and not directly reusable.

The **constraint network** [20] consists of a set of constraints $\mathbb{C}$ defining n-ary relations in a set of variables $\mathbb{V}$. Each constraint generally has a set of methods $\mathbb{M}$, with one $m_i \in \mathbb{M}$ being applicable to compute each desired output variable $v_j \in \mathbb{V}(c) \mid c \in \mathbb{C}$. For equality constraints, these methods are the solutions with respect to each variable, e.g. three methods $a \leftarrow c-b$, $b \leftarrow c-a$ and $c \leftarrow a+b$ for the constraint $c=a+b$. Inequality constraints can be converted to equality constraints prior to assignment, e.g. $a \leftarrow b$ for $a \leq b$. When not converted, inequalities only permit determination whether they are fulfilled or violated. The constraints defined in this way correspond to an equation system $A\underline{v}=\underline{b}$, which normally will be underdetermined and is therefore not solved explicitly. Instead, the current solution is taken as the basis for incrementally updating the adjacent parts of the constraint network to be consistent with the changed values.

Value **assignments** set parameters to a certain value and possess different strengths (default value, desired value, required value). Constraint propagation can cause other parameter values to become dependent, which is traceable through the respective constraint.

**Concepts** collect a set of chosen solution elements, modelled as **realisations**, i.e. associations between a function and the selected means. Each instantiation of a solution element can own concept-specific parameter values, allowing different values in different concepts.

**Geometry** can be modelled in a similar way by parametric volume model templates. Conceptual design is characterised by a pre-geometry state with lacking geometry. Therefore, a working geometry skeleton could be used for describing connections between principle solution elements and a shape template can provide parametric geometry for successive detailing.

Additional classes are used for truth maintenance (TMS, Justification, Node), to create constraints of all types (ConstraintGenerator), to solve algebraic constraints for one variable (MapleConnect) and to calculate values executing algebraic constraints (FormulaParser).

## 3.3 Modelling procedure

One aim of the outlined tool is to assist in doing the routine work while not inhibiting the designer's creativity. Therefore, the possibility to quickly insert own solution elements and relations is important. Easy definition and change of relations is even important as tasks allow decomposition into different relations. The process is an interactive sequence of modelling, choosing and calculating steps with mainly functional but even geometric quantitative information. The model even permits switching between different levels of abstraction, e.g. by working with a verbal description of a function on higher level after having detailed parameters of a different low level function.

Modelling starts by recording relevant requirements. Initially it is undetermined how functional requirements are solved. By assigning a parameter of a responsible function to the requirement, a check for fulfilment can be done later in the design process when solution elements are considered. This allows variation even for principles, not only parameters inside a solution element. Restricting requirements can similarly be assigned to function parameters, e.g. available space or hydraulic energy provided by an existing external technical system.

The functional description always starts with the main function. Sub-functions, some of them coupled to requirements, are added successively and decomposed into their sub-functions.

Before further decomposition, solution elements have to be added and assigned to functions. Concepts are formed by selecting means into them.

Parameters can be added to functions and solution elements and are initially set to default values. Internal constraints are added to describe the working principle by connecting parameters inside a means. External constraints connect functions to form the function structure. The choice of the right parameters and constraints remains the engineering problem and is not necessarily straightforward, but is supported by providing the possibility of quickly modelling relations and their consequences. Obviously, predictions can only be done on modelled parameters and constraints, and unexpected interrelations can still exist. If the designer of a high-speed train for instance only models the desired tilting used to decrease lateral forces on passengers, but does not consider harmful tilting, then the model will not predict derailing due to susceptibility to side wind. However, it can easily do so once the corresponding constraints for side wind are added. On the other hand, only those functional parameters and constraints that are important for the concept development and evaluation should be modelled, thus choosing a limited number of central parameters (cf. [6]). The basic idea is not building up a complete equation system (which due to the early stages can change rapidly and considerably), but instead model momentary solutions and important concept properties.

Values can be set and propagated through constraints, which successively will determine all relevant parameter values. This is done until all goal-setting requirements are fulfilled by suitable solution elements and restricting requirements are not violated. The undirected constraint network allows treating design problems, which do not start with a clear causality or design process scheme. Constraint satisfaction leads to equations automatically being solved for a needed variable, which then will be set to fulfil the constraint.

The transition to embodiment design can be prepared by a similar approach of parametric geometry patterns assigned to solution elements and coupled to the constraint network, but modelled inside a geometry modeller.

## 3.4 Reusability and extensibility

In the model, reusability is facilitated by a weak and partly automatic connection between functions and solution elements. Internal constraints operate only inside the solution element and are directly reusable. External constraints of the respective function parameters are automatically coupled to the means' parameters when the means is selected into a concept and separated if it is deselected (external constraints thus will operate correctly even if the means is changed as long as they provide identical parameters). In this way, the context of new applications of a solution element can be applied.

As libraries containing all possible solution elements for a broad range of problems are not feasible, the tool needs to be easily extensible. In [12], off-line or on-line capturing or a combined approach for adding new elements is identified. To overcome the disadvantage of high preparation time and low flexibility of design catalogues and similar off-line prepared solution repositories, here on-line capturing is chosen. Solution elements are mainly described by parameters and their internal constraints of arithmetic formula type, which instantly gain effect. A database of solution elements can successively be built by adding suitable segments from finished concept models, this way creating an applicable tool. Storing hierarchically detailed means with sub-functions and their constraint network facilitates reuse of whole segments. In this way, scalability is facilitated. By introducing a type hierarchy to the database entries, constraints could even automatically be added and removed, e.g. adding the hydraulic constraints when concretising a generic motor to become a hydraulic motor. This supports the initial abstraction to functions, search for solutions and the final concretisation [10].

Other possibilities at choice for describing new elements could comprise procedural interpretative languages, procedural compiling languages, interactive parsing languages, selection of predefined elementary description blocks etc.

# 4 Example

The model has been applied on a real industrial application involving concept generation for an original design problem. The task was to design a tube cleaner; the excerpt in Figure 4 shows a hydraulic motor with constraints, fulfilling the function “rotate cleaning head”, connected to the functional requirement to provide the necessary torque and restricted by the requirement given of the available space.

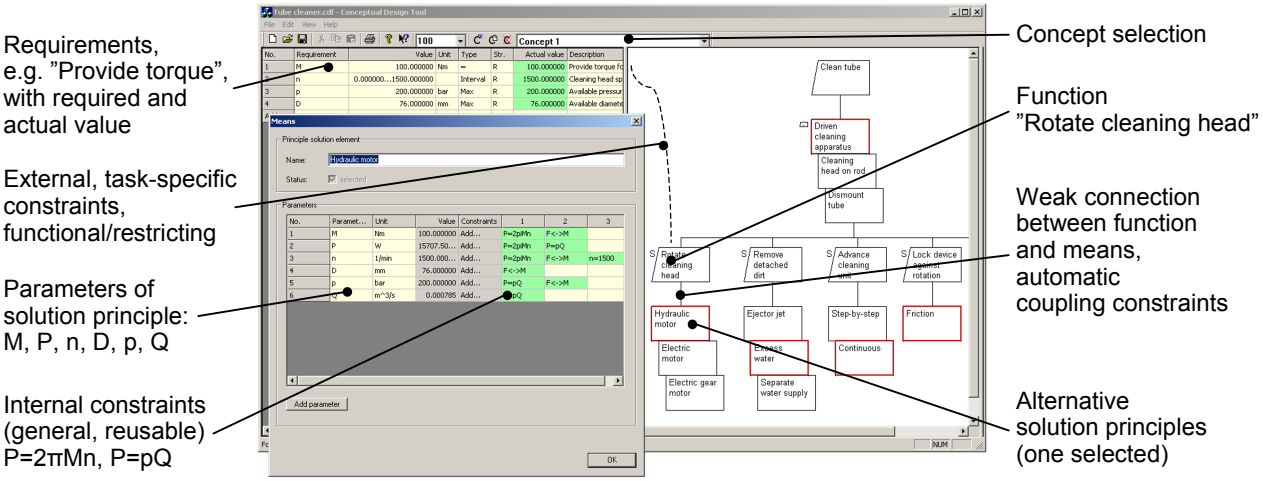


Figure 4. Example of the principle solution element “hydraulic motor” with relations.

# 5 Prototype

An object-oriented prototype implementing the information model has been built as C++ application. MFC was chosen to provide good interactivity and context-specific handling, at the expense of operating system dependency, tight coupling between user interface and product model and a rather high coding effort. The user interface is based on an automatically rendered F/M-tree and is well suited to visualise the causal structure of solution elements and the functions they provide in a hierarchy. Functions as well as solution elements can possess specific branches below them, activated by choosing a means. The constraint network is based on the DeltaBlue algorithm [20], numerical calculations are performed internally and formulas are solved through a Maple-interface.

Displaying means as nodes in the F/M-tree supports the limited human short term memory. Human perception is even taken into consideration by the staggered display of alternative solution elements underneath the function symbol, which increases clarity by visually grouping the set of alternative elements at choice for fulfilling one function. During elaboration of the F/M-tree, the need for rearrangements is likely to occur. Therefore, moving of functions and means is important and can be easily done by drag and drop. As F/M-trees grow quickly, adaptability is important. Branches can be collapsed and a zoom function is provided. Intuitive handling is increased by colour-coded constraints (green=fulfilled, red=not fulfilled) and parameters (grey=dependent parameter, white=independent parameter). In order

to maintain consistency, dependent parameters cannot be modified; the determining constraint is displayed instead. Symbolic pictures and sketches have been identified as crucial elements of concept models [21]. Support for assigning hand-made sketches to means in an F/M-tree can therefore be a further implementation issue in the prototype.

# 6 Conclusions

Presently, computer support for conceptual design is sparse. In early stages, sketches and qualitative reasoning are doubtlessly important, but quantitative reasoning on a limited number of parameters (describing the major properties of a concept) can improve the evaluation of concepts.

Constraint networks have previously been used for modelling configuration design problems. In this article, the main contribution is to address support and reuse even for original design problems [3] by applying functional decomposition and quantitative, parametric solution elements. In this way, even the finding of principle solutions ([3], ch. 6.3, 6.4) can be supported by a constraint approach. By using relations on functions and general parameters instead of subcomponents or spatial relations, the tool is suitable for more universally supporting a broad range of conceptual design problems. The concept object permits multiple alternative principle solution variants, not only parametric variants of the same principle solution. Reuse is facilitated by the division into internal, reusable constraints and external, inter-element constraints specific to each new task.

The outlined information model connects requirements, functions, reusable principle solution elements and concepts. The support is suitable for those stages of original designs where the solution is still being conceived, is underdetermined or with unclear causality and therefore cannot be treated by usual simulation techniques. Successively added constraints of formula type are more generally applicable as opposed to product-specific dimensioning tools. By assigning parameters and values for gradually developing solutions, the lacking possibilities for instantiation of some methods are improved and the solution space is well documented.

The possibility to quickly model own elements and their relations and reducing type classifications to solution element types and parameter types increases the general applicability.

Further work can encompass scalability and the verification on realistic-size problems, the treatment of time-dependent constraints and the inclusion of geometry in a similar constraint-based way, namely in the form of simplified parametric geometry templates coupled to the concept parameters. Further automation steps, e.g. automatic combination of available solution elements inside the tree, and collaborative work can be more future issues.

**References**


[1] Anderl, R. and Gräb, R., “Parametrics: A Key Technology for Efficient Development of Virtual Products”, ProSTEP Science Days 2000, Stuttgart, 2000, pp. 3-13.

[2] Vajna, S., “Durchgängige Produktmodellierung mit Features”, CAD/CAM-Report, 1998, pp. 48-65.

[3] Pahl, G.; Beitz, W.; Feldhusen, J. and Grote, K. H., “Konstruktionslehre”, Springer, Berlin, 2003.

[4] Roth, K., “Konstruieren mit Konstruktionskatalogen”, Springer, Berlin, 1994.

[5] Arndt, H.; Feldkamp, F.; Heinrich, M. and Meyer-Gramann, K.-D., “System Design for Reusability”, Workshop ECAI 2000, 2000, pp. 13-18.

[6] Hansen, C. T. and Andreasen, M. M., "The Content and Nature of a Design Concept", Proceedings NordDesign 2002, Trondheim, 2002, pp. 101-110.

[7] Weber, C.; Werner, H. and Deubel, T., "A Different View on PDM and its future Potentials", Proceedings DESIGN2002, Vol. 1, Dubrovnik, 2002, pp. 101-112.

[8] Wilhelms, S., "Modelling of Means in Conceptual Design", Proceedings DESIGN2002, Vol. 1, Dubrovnik, 2002, pp. 579-584.

[9] Kuttig, D., "Die Funktionsstruktur als integraler Bestandteil des rechnerunterstützten Konstruktionsprozesses", Konstruktion, Vol. 44, 1992, pp. 183-192.

[10] Langlotz, G., "Ein Beitrag zur Funktionsstrukturentwicklung innovativer Produkte", University of Karlsruhe, 1999.

[11] Mortensen, N. H. and Hansen, C. T., "Towards semantic modelling of organs in a computer based design support system", Proceedings Datenverarbeitung in der Konstruktion 1994, Düsseldorf, 1994, pp. 485-500.

[12] Jensen, T., "Functional Modelling in a Design Support System", Technical University of Denmark, Lyngby, 1999.

[13] Golob, B.; Jezernik, A. and Hren, G., "A Feature Based Approach For Conceptual Design", Proceedings DESIGN2002, Dubrovnik, 2002, pp. 483-488.

[14] Lin, L. and Chen, L.-C., "Constraints modelling in product design", Journal of Engineering Design, Vol. 13, 2002, pp. 205-214.

[15] Lindemann, U. and Amft, M., "Rechnergestützte Integration von Gestaltung und Berechnung im Konstruktionsprozeß", Konstruktion, Vol. 52, 2000, pp. 78-82.

[16] Bracewell, R. H. and Wallace, K. M., "Designing a Representation to Support Function-Means Based Synthesis of Mechanical Design Solutions", Proceedings ICED 2001, Glasgow, 2001, pp. 275-282.

[17] Werner, H. and Weber, C., "Ligo – an Object-Oriented Modelling Tool for Integrated Product Development", Proceedings IPD 98, 1998.

[18] VDI 2221, "Systematic approach to the development and design of technical systems and products", 1993.

[19] Wallmeier, S.; Badke-Schaub, P. and Birkhofer, H., "Training for designers: empirical results of trained designers in selected design processes", Proceedings ICED 1999, Vol. 1, München, 1999, pp. 211-216.

[20] Freeman-Benson, B.; Maloney, J. and Borning, A., "An Incremental Constraint Solver", Communications of the ACM, Vol. 33, 1990, pp. 54-61.

[21] Almefelt, L.; Sutinen, K. and Malmqvist, J., "Computer support for systematic design applied in a cross-functional commercial concept development process", Proceedings TCME 2002, Wuhan, 2002.

Sören Wilhelms
Linköpings universitet, Department of Mechanical Engineering, Division of Machine Design
SE – 581 83 Linköping, Sweden
Tel.: +46 13 28 27 30, Fax: +46 13 28 56 70
Email: sorwi@ikp.liu.se