NordDesign 2004 – Product Design in Changing Environment
18-20 August 2004, Tampere, Finland

EXTENDED SUPPORT OF CONCEPTUAL DESIGN USING OPERATING STATES AND DIMENSIONING CASES

Sören Wilhelms

Division of Machine Design
Department of Mechanical Engineering
Linköpings universitet
SE-581 83 Linköping
SWEDEN
E-mail: sorwi@ikp.liu.se

Per Johansson

Division of Machine Design
Department of Mechanical Engineering
Linköpings universitet
SE-581 83 Linköping
SWEDEN
E-mail: perjo@ikp.liu.se

Conceptual design, design support, design concept, operating states, dimensioning

Abstract

This article focuses on conceptual design. In this phase, systematic design provides models for function analysis and concept synthesis. Little support exists though for early consideration of different operating states and dimensioning cases while concepts are developed. This is important in order to comprehend the relation between product requirements and design characteristics in different or changing environments or use cases. This article suggests modelling each operating state and each dimensioning case using a dedicated object and describing conceptual design using a constraint network in order to explicitly capture the inter-element relations. The purpose is to facilitate early quantitative reasoning (e.g. to check if a requirement is fulfilled in a specific dimensioning case with specific chosen means and values), to simplify the change of views (e.g. analysing another operating state) and to flexibly support change operations on the designed product (e.g. changing a solution principle or a value choice in a concept).

1 Introduction

Systematic design, e.g. [Pahl et al 2003], provides models, methods and tools for engineering design and product development. Developing complex products in changing environments renders it unfeasible to design products anew each time without considering and reusing prior design solutions. Therefore, it is important to be able to access prior solutions and design history. Solutions, however, are used under different conditions, which necessitates the possibility to rapidly adapt prior solutions and stored solution elements to new, changed requirements and to model the changing requirements and dimensioning cases. This is especially important when several distinct operating states exist. In order for companies to maintain their competitiveness, the faster development of better-verified concepts is desired, especially in high-salary countries as in Scandinavia, and easier adaptability of prior solutions to new dimensioning cases can contribute to this goal.

Prior research in design methodology has resulted in synthesis models, e.g. the function/means tree (F/M tree) used for functional decomposition and concept synthesis [Hansen1995]. Concepts are ideas of overall principle solutions connecting needs and required functionalities to a solution and its structure [Hansen&Andreasen2002]. Due to the nature of early design, the models used by today's practitioners are mainly qualitative and static, i.e. do not consider the quantitative dimensioning or the existence of several different operating states. To some extent, this is meaningful, as some authors claim that including states too early will produce models, in which the conceptual decisions are no longer manageable for complexity reasons ([Oliver et al 1997], ch. 3.5). The separate modelling of operating states and conceptual decisions regarding chosen solution principles and essential characteristics, however, as suggested here, can provide a way of taking into account states, but not make the model unmanageably large.

Other researchers have also pointed out the need of describing concepts' essential differences by modelling a selected set of characteristics [Hansen1995] and the need of considering state transitions (cf. [Buur1990], pp. 75f; [Andersson2003]). There has also been work towards information models supporting partially quantitative conceptual design, e.g. [Wilhelms2003], where concepts are described as a number of means choices and value choices. The intention is to extend qualitative models to handle some quantitative, dynamic aspects rather than to replace them.

The applied method is theoretical reasoning, based on the assumption that there is a need of additional modelling entities for capturing dynamic aspects in conceptual design. A summary of applicable existing approaches is given in section 2, the new modelling entities are then presented in section 3. The examples in section 4 are to further illustrate the use of operating states and dimensioning cases.

2 Representing states in conceptual design

In literature, several different approaches to take into account states are known, cf. Figure 1. State transition modelling, e.g. in Petri networks or state charts, is common in detail design and in other domains such as software development, control technology or mechatronics. For conceptual design of mechanical products, the relevant objects that can be used for state modelling are the functions and the flows interconnecting them in a function structure. In the following, the known methods are summarised with regard to the special needs of conceptual design of mainly mechanical engineering products.

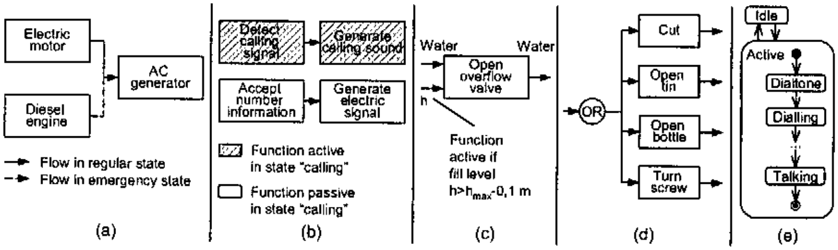

Figure 1. States represented through (a) flows, (b) functions, (c) flow condition, (d) alternative path and (e) state diagram

2.1 Representation using transient flows

A first example of how to represent states is to assign state information to the flows that are connecting different functions. In [Kuttig1993], it is shown how the assignment of working phases to each flow can be used to denote that flows are active only under certain time intervals. In the example of an uninterruptible power supply treated there, three operating conditions are identified: regular, change and emergency (see Figure 1 (a), where two of these are shown). In these conditions, different flow connections are activated, e.g. a flow connection from "diesel engine" to "AC generator", which is active only in the emergency condition. It is though not explicitly stated that for example the function performed by the "diesel engine" is active only during some states.

2.2 Representation using transient functions

In [Buur1990], states are modelled by declaring functions as active or passive in the different states. In his example of a telephone, the function "generate calling sound" is for instance only active in the state "calling", cf. Figure 1 (b).

The declaration of functions as active/inactive in various states can form a superset of representation using only flows. When states are combined with a functional breakdown, the applicable flows for the actual sub-functions and states can automatically become activated or inactivated (flows are active iff they connect sub-functions that are present in the current concept and active in the current state). This representation harmonises best with concept synthesis using several alternative functions, e.g. for mechatronics as described by Buur.

2.3 Representation using flow conditions

The model in [Langlotz1999], pp. 103, 113ff and 156ff, describes how functions can be active or inactive depending on individual start, stop or length conditions. The conditions are assigned to functions and are formulated using the functions' input and output flows or as relations between functions (e.g. function 1 must finish before function 2 starts). Possible conditions are manual start/stop, start/stop when a flow is present or start/stop when a flow parameter is above/below a threshold value (e.g. an overflow valve opening when the fill level of a tank reaches a maximum value, see Figure 1 (c)).

Flow conditions are suitable to represent the course of events during an operation sequence or to schedule functions over time ([Langlotz1999], fig. 4-27). They can also constitute information on when a transition between states occurs. For concept synthesis and administration of different conceptual decisions, the approach is though less advisable, as it easily may get too detailed and as states and functions are not modelled separately.

2.4 Representation using alternative paths

Logical conditions can be used to select different paths through a function structure. In [Oliver et al 1997], p. 56 and p. 60, an "OR"-operator is used to describe several functions, one of which is chosen depending on the presence of certain conditions or flows, or on user choice. These authors exemplify the "OR"-operator using a pocketknife with several functions involving the use of the different tools, see Figure 1 (d), but do apart from fairly simple examples not provide any realistic-size problems.

Modelling of this kind is suitable for estimating the temporal behaviour of several paths through the function structure. As the focus here is not on real-time systems where state modelling might be used to assess critical response times, this approach is less recommendable. Here, rather the handling of characteristics and conceptual decisions in different use cases is the central concern.

2.5 Representation using state transitions

State transition diagrams are common for modelling different states of a product and the state transitions between them. Two different ways of modelling functions and states can be distinguished ([Oliver et al 1997], ch. 3.5):

- Functions are active *while* the state transitions occur.
- Functions are active while the product is in one state, i.e. functions occur *before* or *after* state transitions (this corresponds to the representation described in section 2.2).

An example of the first way is given in [Andersson2003], p. C-8, where state transition diagrams are connected to function structures so as to provide a way of describing by which functions a state transition is achieved. His example treats the remote control for opening an automobile door lock. State charts are an example of the second way and use transitions without duration, with functions taking place between transitions. In [Harel1987], examples of state charts for a digital wristwatch are given.

Mainly for software development, various models are common such as structured analysis/design (SA/SD), which use data flow diagrams (i.e. function structures with information as operands) and state transition diagrams ([Yourdon1989], ch. 13). The similar, but more recent UML standard [OMG2003] contains a graphical language for state chart diagrams, see Figure 1 (c). An example of mechanical engineering use of state diagrams is e.g. given in [Gausemeier et al 2001], ch. 4.3.3.6 and 4.3.3.7.

State transition models have in common their focus on merely describing the structure, albeit some operations such as automatic code generation from the structure are possible in some tools. The models focus on maintaining the necessary functions the modelled objects should have (i.e. artefact functions such as "emergency stop quick turn-off"), not on information processing capability for the modelled elements (i.e. designing functions such as "estimate the time needed for quick turn-off" or "estimate power consumption in different states with different chosen principles"). The models do not either take into account different principles to achieve a required functionality (e.g. that a transition from standby to activation can be achieved by different principles such as button, photo sensor, voice control, timer etc.).

3 Extended concept model including states

In order to help to consider relevant operating states and dimensioning cases already in conceptual design, an extended parametric F/M tree is suggested (see Figure 2).

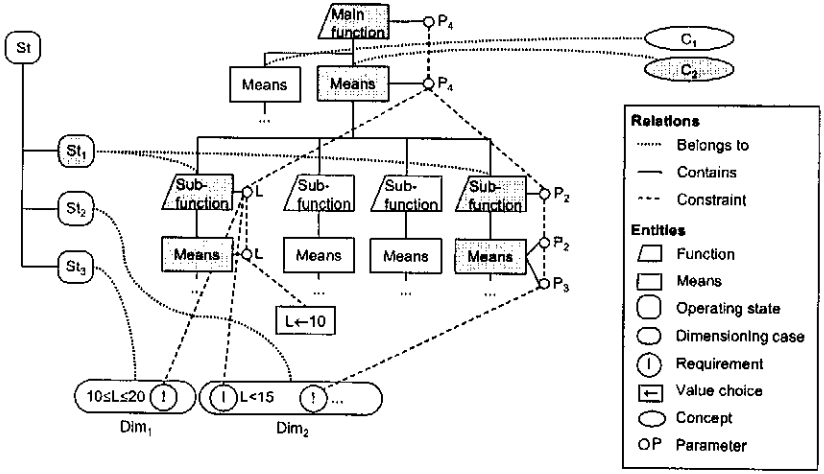

Figure 2. Extended F/M tree with operating states (St) and dimensioning cases (Dim), active entities shown in grey

The entities *operating state* and *dimensioning case* are added in order to embrace the dynamic aspects of concept development. By using explicit objects for dimensioning cases, it is easier to obtain better-verified concepts. Operating states facilitate keeping track of influences between different operating conditions and life cycle phases.

3.1 Operating states

A product is often used in several modes or different working conditions, e.g. an aeroplane with distinct states as takeoff, climbing, cruise etc. In these different conditions, various functions can be active or inactive. The entity *operating state* is therefore used to indicate which functions are active in a certain use phase by providing a list of references to active functions in that state.

Moreover, operating states allow expressing the design freedom of choosing between single or multiple functions. A given design problem could either be solved using one single function that is widely adaptable to meet the requirements (i.e. one function active in all operating states, e.g. for a continuously variable transmission) or several specific functions, one for each requirement (i.e. one function specially adapted to each operating state, e.g. a special function for "fast traverse").

Operating states and functions/means constitute hierarchies which are interrelated, but of equal importance. As neither of which is superordinate to the other, the designer is free to choose to start with operating states or functions at will, and is also free to change focus between them at any time. For some products, starting with operating states is most intuitive (e.g. the aforementioned aeroplane), and the states will then be extended by adding relations to functions they include. For other products, starting with functions is preferable, and the functions can then be detailed by adding operating states to them indicating when a function is active (e.g. a chainsaw, for which it was discovered that outdoor use in winter requires an additional state with special functions to increase the inlet air temperature).

Analogous to the other model elements, operating states are arranged in a modularised architecture. Operating states are thus objects merely referencing functions, but not containing any functional description themselves. This architecture is chosen to prevent redundancy and inconsistencies which otherwise could occur when functional descriptions would be located both in states and functions and modified at one place, but not the other.

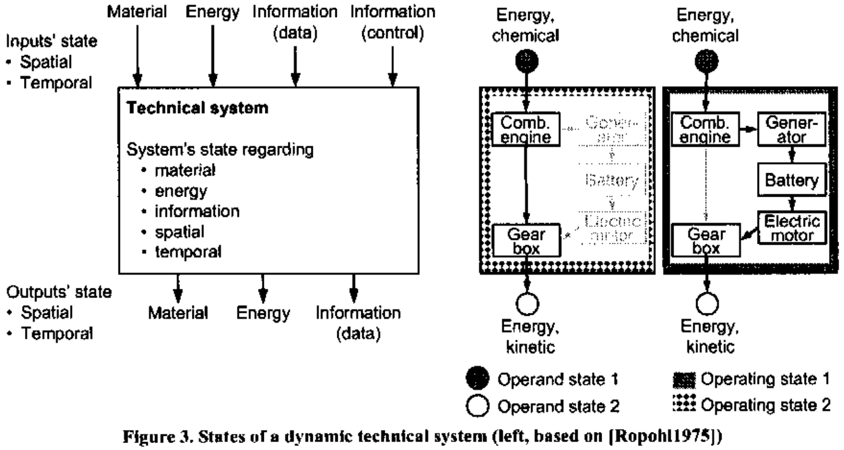

Figure 3. States of a dynamic technical system (left, based on [Ropohl1975]) and example of two operating states of a hybrid car (right)

Operating states and state transitions have a certain similarity to operands and transformations known from technical systems, cf. Figure 3. However, states in the technical system refer to the properties of the input and output operands, and state transitions to the transformations that are (continuously) performed by the "black boxes" in a technical system. Operating states as described here, indicate the one of the possible internal states of the technical system in focus that is currently active, i.e. with which functions it achieves the transformation from operand input state to operand output state it is to perform. It is therefore advisable to distinguish between operand states (i.e. the properties of inputs and outputs) and operating states (i.e. the properties of the system "between" input and output).

3.2 Dimensioning cases

The activity of making decisions in order to lay out, adjust or design a solution to meet a certain performance is in this section called *dimensioning*. Dimensioning relies on unambiguously defined loading cases defined with respect to quantity, type, frequency of occurrence, duration etc. (cf. [Pahl et al 2003], p. 288). Traditionally, dimensioning is relevant in early embodiment design, as it involves finding the right shape, geometric dimensions, materials etc. to meet the required lifetime, deformation, stability etc. within the given loading cases. However, in order to be able to evaluate concepts, some of these dimensioning decisions should already be made in conceptual design ([Pahl et al 2003], p. 267-271), e.g. the choice of a transmission ratio or the size of a flywheel. If functions and means are described quantitatively, as suggested in the parametric F/M tree, earlier dimensioning is facilitated.

For this purpose, *dimensioning cases* are introduced as entities containing a number of references to requirements. Dimensioning cases indicate which technical requirements (and their respective desired values) are relevant to consider in a specific use case of the product. They also provide a way of handling cases where multiple requirements are restricting the same parameter. Figure 2 shows such an example of two dimensioning cases Dim_1 and Dim_2 both restricting a parameter "L", for which an acceptable value "10" for both states was found.

The requirement object permits checking if a certain concept fulfils a single requirement. By using dimensioning cases, checking for requirement fulfilment in different loading cases is supported, i.e. checking if a concept fulfils a group of requirements.

3.3 Information model

Figure 4 shows the information model used to describe operating states and dimensioning cases, and their relations to the information model for parametric F/M tree based conceptual design [Wilhelms2003]. The basic approach is to use a constraint network to couple parameters owned by functions, means and requirements. In this way, the concepts can be built up incrementally and interactively. All elements in the model have information processing capabilities, which is a major improvement compared to the classical F/M tree or state diagrams. Changes made at any element are propagated to other elements through the constraint network.

Change operations are provided to easily be able to change a chosen solution principle or parameter value in a concept. Likewise, operating states and dimensioning cases can be changed in order to analyse a different phase or use case. In this way, through a semi-automatic update of the model, the analysis of various operating states, dimensioning cases or concept alternatives is supported by a simple change of selection operation.

Rather than a formal, descriptive modelling language that aims at producing an exact model for a single solution, but with limited area of applicability, a flexible and extendable concept model is achieved, which allows working with under-determined solutions and multiple solution variants and states and therefore presumably harmonises better with the needs of early phases.

By introducing operating states and dimensioning cases as new entities, it is furthermore assumed that the mental workload can be reduced. The short-term memory with its limited size of approximately 7±2 chunks is relieved by externalising states and dimensioning cases.
Each operating state either references a number of functions or a number of sub-states, but not both (cf. Figure 4). The aim of this restriction is to prevent contradictions between a state on one level and one of its sub-states.
Dimensioning cases point to the requirements that are relevant for this case. They even reference assignments, i.e. value choices that are only valid for this dimensioning case. Value assignments can thus either be valid for one specific dimensioning case (e.g. that a specific gear of a gear box is active for that dimensioning case) or several dimensioning cases (e.g. for all gears except the reverse gear) or globally valid for all of them (e.g. a chosen geometric length or the number of teeth).

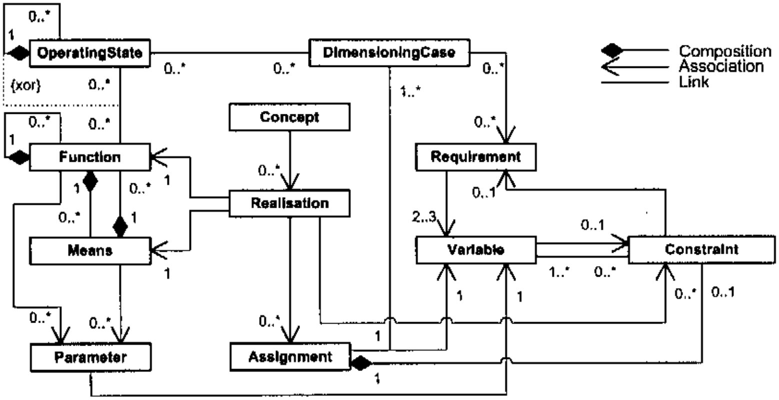

Figure 4. UML information model (based on [Wilhelms2003])

Shown in Figure 4 is the hierarchy relation between operating states (i.e. which sub-states a state contains). In order to also describe the conditions upon which transitions between operating states occur, further objects and relations are necessary (not shown in Figure 4).

3.4 Using the model

It is envisaged that the model is used in an assisting system, of which a prototype already has been implemented. The tool is intended to serve as an aid for creativity, a help to manage requirements and operating states in several concept alternatives and a help to adapt reused solutions to new operating conditions. By its iterative and interactive nature, the support approach is feasible throughout the concept phase, i.e. even for not yet completely determined solutions. The primary target group is designers having to conceive principle solutions and to select the best of them. The presented model is appropriate for original design situations involving the management and assessment of different solution principles that can easily be described by a few distinct parameters. The suitability may vary for other tasks and design problems. Generic elements allow the support to be used for any kind of product that can be described by parameters and relations.

4 Examples

In the following, two examples are given. The first example, a hybrid car, illustrates the use of different operating states and how properties are calculated. The second example, a hand drilling machine, indicates how hierarchies of states are modelled.

4.1 Hybrid car

In Figure 5, the function structure for a hybrid car is given. The five operating conditions are shown on the upper left.

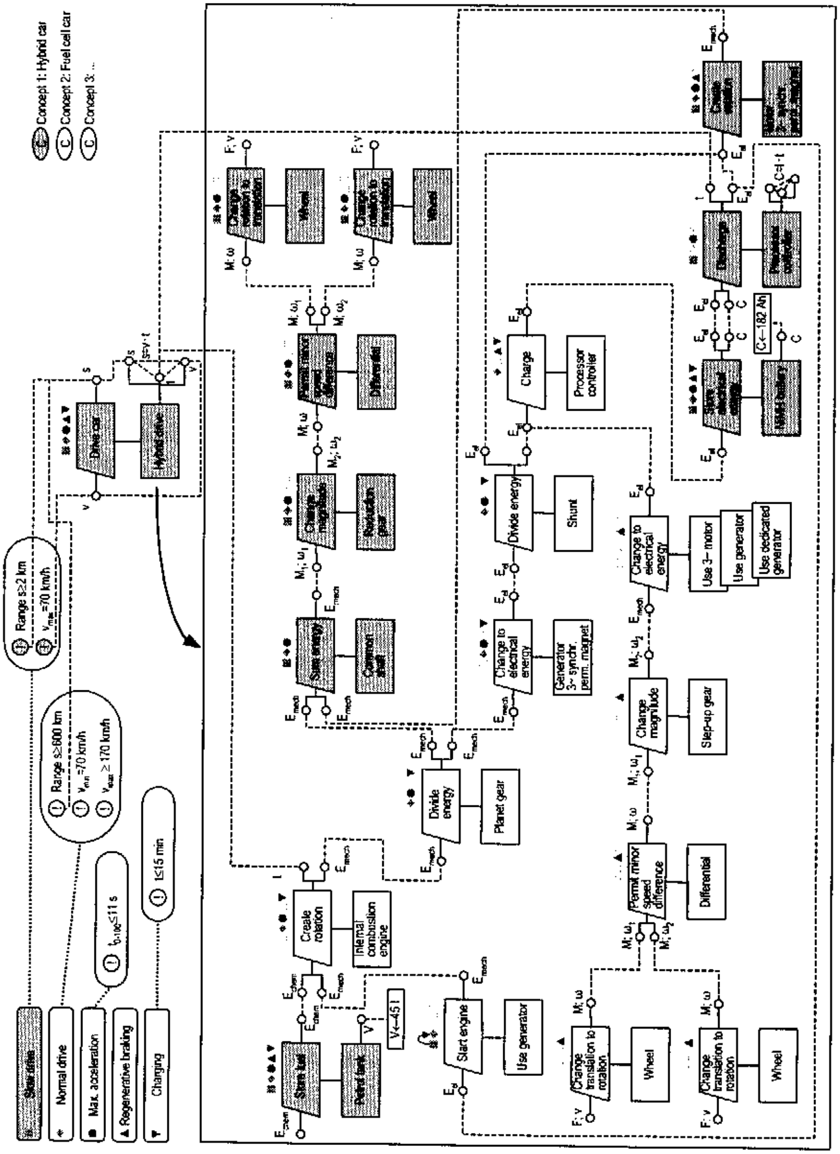

Figure 5. Hybrid car and its operating states (active objects shown in grey)

In the figure as given, the condition "slow drive" is active, the functions in grey thus being active. Constraints are shown as dashed lines, with the text next to the lines indicating the parameters that are constrained (E_{mech} abbreviates torque M and rotational speed ω, E_{el} likewise voltage U and current I). Above each function, symbols indicate the corresponding operating states, during which the function is active. The function "start engine" provides an example of a function that is active *between* two operating states. In the shown example, where the combustion engine is not used for slow speeds, "start engine" provides a means to achieve the transition from "slow drive" to "normal drive" (indicated by the arrow in Figure 5).
The calculation of the achieved range, related to the required value "Range $s \geq 2$ km", exemplifies how properties are determined by constraints using the chosen means and operating states. For the active operating state "slow drive", this calculation will, as shown, rely on the available battery capacity. In a different state, e.g. "normal drive", the calculation terminates at the function assigned to "internal combustion engine", which then would be active and calculate the range from the engine's fuel consumption and the tank volume.
Calculations such as the mentioned fuel consumption require that the relations can be expressed in a simple formula between the parameters, which is often the case, as simplified models or approximations from empirical data exist for many engineering applications.

4.2 Drilling machine

The example of a drilling machine in Figure 6 exemplifies hierarchies of operating states. The operating state "Use" contains three sub-states "Normal", "Slow/screwdriver" and "Percussion drilling". The state "Normal" in turn, shown as active in Figure 6, references the functions "Accelerate chuck", "Retard chuck", "Guide bit-tip", "Select rotational speed", "Guide drill" and "Remove bit from chuck".

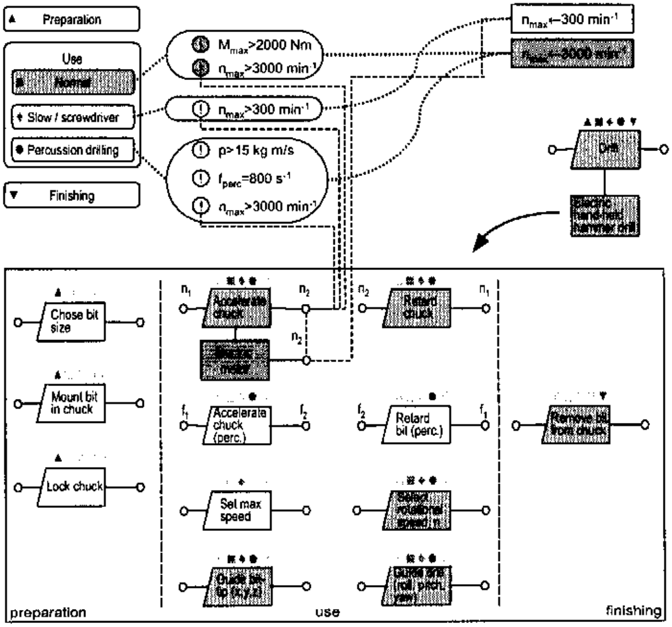

Figure 6. Drilling machine and its hierarchy of operating states

An example of different value choices in different operating states is given in the upper right corner. The chosen value "$n_{max} \leftarrow 3000$ min^{-1}" for state "Normal" constrains the corresponding parameter n_2 in the means "electric motor". Changing to state "Slow/screwdriver" would deactivate this constraint and instead activate the constraint "$n_{max} \leftarrow 300$ min^{-1}" associated with that state.

5 Conclusions

Many conceptual descriptions provide no natural support for easy reasoning on different operating states, as they do not include information on when functions are active, or which calculations that are relevant in different cases. In this article, an extended model based on a parametric F/M tree and constraint networks is presented. In this way, taking into account the dimensioning and different operating states already during conceptual design is made possible.

As the model is capable of describing evolving, incomplete solutions, it is more suitable for the mental model in the conceptual design phase. The easy exchangeability of means and value choices and the ability to quickly change the operating state or dimensioning case under consideration allows exploring the solution space in a more efficient way.

However, a limitation of the model in its current form is that it requires the relations to be expressible in plain algebraic formulas. This is often the case when approximate formulas are used, but may be impossible for products involving tightly coupled interacting components.

In general, the time effort for creating models is not negligible, and it is not self-evident that modelling, which is suitable for detail design, even is a suitable activity in conceptual design. As there are simulation tools for detail design, the presented model is not to replace the powerful modelling capabilities of these tools, but to enhance them regarding reasoning on early solutions, their different states and cases and on multiple principle variants.

References

Andersson, F., "The Dynamics of Requirements and Product Concept Management", Göteborg : Chalmers University of Technology, dissertation, 2003.

Buur, J., "A Theoretical Approach to Mechatronics Design", Lyngby : Technical University of Denmark, dissertation, 1990.

Gausemeier, J.; Ebbesmeyer, P.; Kallmeyer, F., "Produktinnovation", München : Hanser, 2001.

Hansen, C. T., "An approach to simultaneous synthesis and optimisation of composite mechanical systems", Journal of Engineering Design, Vol. 6, No. 3, 1995, pp 249-266.

Hansen, C. T.; Andreasen, M. M., "The content and nature of a design concept", Proceedings NordDesign 2002, Trondheim : NTNU, 2002, pp 101-110.

Harel, D., "Statecharts: a visual formalism for complex systems", Science of Computer Programming, Vol. 8, No. 3, 1987, pp 231-274.

Kuttig, D., "Potential and limits of functional modelling in the CAD process", Research in Engineering Design, Vol. 5, No. 1, 1993, pp 40-48.

Langlotz, G., "Ein Beitrag zur Funktionsstrukturentwicklung innovativer Produkte", Karlsruhe : Karlsruhe University, dissertation, 1999.

Object Management Group (ed.), "OMG Unified Modelling Language specification", 2003, http://www.omg.org/cgi-bin/apps/doc?formal/03-03-01.pdf.

Oliver, D. W.; Kelliher, T. P.; Keegan, J. G., "Engineering complex systems with models and objects", New York : McGraw-Hill, 1997.

Pahl, G.; Beitz, W.; Feldhusen, J.; Grote, K. H., "Konstruktionslehre", Berlin : Springer, 2003.

Ropohl, G., "Systemtechnik – Grundlagen und Anwendung", Munich : Hanser, 1975.

Wilhelms, S., "Reuse of principle solution elements in conceptual design : an information model for concepts in systematic design", Linköping : Linköpings universitet, licentiate thesis, 2003.

Yourdon, E., "Modern structured analysis", London : Prentice-Hall, 1989.