PERFORMANCE EVALUATION METHODS IN DESIGN: DISCUSSION

HANS GRABOWSKI
University of Karlsruhe, Germany

Product development has a close connection with the theory of decision making. That theory deals with the human behaviour of decision making in its social context. It is an interdisciplinary branch of economics. The interdisciplinarity is caused by the influences of mathematics, psychology and sociology.

Von Neumann and Morgenstern published their classic work on game theory called *Theory of Games and Economic Behaviour* in 1944. In their work, the authors prove the similarity between the behaviour of an economics system and the mathematical representation of certain technical games. While classical decision theory deals with the behaviour of individuals in deciding, game theory developed as an independent discipline which studies the conflict of interest between at least two decision makers.

In contrast to game theory, stands the area of vector optimization, which is also known under the terms multiobjective optimization or multicriteria optimization. It deals with the problem of multiple goals of one single individual with the help of multicriteria decision models. Multicriteria optimization also originated from economics and has been used by engineering disciplines, especially by civil engineering.

Decision theory distinguishes between normative, descriptive and prescriptive approach. Normative decision theory assumes the concept of an optimal decision. Its behaviour is described with a formal logic of decision making. Descriptive decision theory assumes, in contrast to the closed model of normative decision theory, an open model of decision making. This open model takes the creation of the decision criteria and the insufficient capacity of human information processing into account. This model explains why in complex decision making situations, the individual does not always decide in an optimal way. Prescriptive decision theory does not aim to develop its own models of decision making but supports the decision maker through the conservation of consistency.

There is a growing unification of decision theory and the theory of cognitive problem solving processes is developing. While decision theory focuses on the problem of the solution process as preparation for the decision, cognitive theory focuses on the decision as the end of the problem solving process. It is implicitly assumed that the problem definition also contains a request for optimization so that a requirement for a decision is explicit. The almost identity of the decision and problem solving processes is clarified by a schema of the phases of the decision process which distinguishes six steps:

- Identification of the problem,
- Acquisition of necessary information,
- Development of possible solutions,
- Evaluation of the solution,
- Selection of a strategy for the action,
- Action followed by learning and revision.

This phase schema of the decision process is very similar to the model of the systematic procedure for the problem solving process which was integrated into the phase oriented design process in design methodology. The iterative character of the problem solving process is explicitly shown in the last step. As a consequence, the phase schema develops a cyclic process which is called “cybernetic control cycle” or “cognitive program” by psychologists. Figure 1 shows this cybernetic control cycle in a so-called “Test-Operate-Test-Exit” (TOTE) unit.

Figure 1. TOTE unit.

The program of a decision and optimization process is caused by an incongruity between the requirements which are valid for a decision and the current situation. With respect to the design process, the situation is described

by the description of the design problem or by a first solution of this problem. In both cases an analysis (test) shows the necessity to correct detected errors. This cycle of analysis and synthesis is repeated until a sufficient (congruous) solution was found (exit).

With the help of low-level and high-level TOTE units, which represent strategic or tactical units respectively of a problem solving process plan, a hierarchical structure of TOTE units, can be designed where specific TOTE units can contain other TOTE units. In addition to the iterative character of the decision making and problem solving process, a recursive element occurs. This recursive element realizes a higher level goal by satisfying several low-level sub-goals. This produces problem structuring.

The efficiency of the product development process would be improved if realistic goals of the design intent could be derived from the requirement specification at an early stage of design. In most cases of product planning some idealistic concepts about the future product very often cannot be satisfied. The designer now has to answer the question which of these goals are important and must be reached by the design and which of them are not that important and can be neglected. A clarification of the task in advance would augment the efficiency of the product development process.

A number of approaches have been developed as part of design methodology to deal with these multiple and often conflicting goals. These approaches fall into two clearly distinguishable groups. The first group consists of methods which use weights attached to goals to indicate their relative importance. This approach has obvious difficulties associated with how these weights are arrived at and how they should change with a change in the available information. Often weights are implied but never explicitly stated. The second group consists of methods which explicitly do not use weights—the Pareto methods. These methods delay the final decisions to a later phase.

In many cases the weights only represent the importance of the respective design goals whereas a relation to the actual instances of the weighted parameters is not given. Design evaluation methods must consider both facts, the relative importance of a specific goal and the possible range of its values.

So, in the decision making process *evaluation methods* are necessary between every design stage because if a design solution cannot be evaluated no progress in the design process is possible. Three papers dealing with evaluation methods from their respective points of view were presented and discussed in this session. The paradigm form follows function or function follows form and the formalization of that paradigm is no longer the point at issue as it was in the past. Most of the authors introduced terms like *context* explicitly to express that the concept of causality between form and function does not lead to practical and “good” design solutions. So each author had

to incorporate and to formalize more semantics in the data models which have to be evaluated. The requirements specification has to be formalized so that formal methods can evaluate a design solution against the requirements.

Requirements can be specified quantitatively or qualitatively. Every evaluation method presented supports quantitative and qualitative evaluation explicitly. Differences could be seen in the support of the design process itself and the design stages in which the evaluation methods are embedded. Kalay et al. and Rodgers et al. embedded their evaluation concept in a design process while Rudolf's evaluation method supports evaluation at a unique design stage.

The paper *A Performance Based Paradigm of Design* by Yehuda Kalay and Gianfranco Carrara proposes an approach which is called performance based design. The main idea is that the relationship between form and function is not causal but contextual which means that finding a design solution does not only depend on the relationship between form and function but on additional information such as social, physical or cultural context information. We can say that contextual information is all the information which is not form and function. The computational model explicitly represents function and form, as well as the context. The main elements of the model are a structured set of goals, representing the functional requirements, a set of solutions, a context and a set of evaluators. In the paper *On a Symbolic CAD Front End for Design Evaluation Based on the Pi-Theorem* by Stephan Rudolph, an evaluation method which is based on the Pi-theorem of physics is presented. The basic idea to the evaluation problem is to transform a given functional relationship of design parameters into a dimensionless evaluation form. The design parameters are described in an object description space x , which is a homogeneous and complete description f of n physical quantities x_i : $f(x_1, \dots, x_n)$. The dimensionless representation is described by an equation F of m dimensionless quantities π_j . The mapping is performed by a mapping function ϕ . Using the theory of

Pi-theorems it can be proved, that the dimensionless graph of $F(\pi_1, \dots, \pi_m)$ is complete and consistent. It can be shown, that this evaluation method has theoretical properties such as minimality, consistency, granularity, hierarchy, sensitivity and modularity and provides a sensitivity analysis of the design parameters. In addition to that the Pi-theorem shows, that the evaluation problem can be reduced to the object description space x ,: $f(x_1, \dots, x_n)$. Rudolph demonstrated the practical feasibility of this method with a prototype of a symbolic CAD front end that has been integrated with a CAD system. The practical benefit is based on supporting the designer by automatic generation and display of the dimensionless groups. In order to obtain this, the designer only has to select the appropriate design variables and to construct the topological coupling conditions. For the graphical

output the symbolic information of the dimensionless groups is converted into a graph of the coupling of the design variables on the screen.

In the paper *A Formal Method for Assessing Product Performance at the conceptual Stage of the Design Process* by Paul Rodgers et al., the interrelation between form of an intended product and the context of its use is the driving factor of design, where context represents product behaviour from users' point of view. In the paper, context is understood to determine the design problem.

There are two different classes of design processes, the unselfconscious and selfconscious process. The first is classified by the designer operating directly on an actual form, determining misfits and correcting them.

One characteristic of the latter is that the designer is not able to directly observe the ensemble. The designer investigates, explores and researches the actual context and constructs a mental picture of it. With this mental picture of the context and with a representation of the product, the designer makes a mental picture of the ensemble in order to try to determine potential misfits.

In the approach to product performance assessment presented, attribute prediction plays an important role as well as the extraction (recognition) of product characteristics. Characteristics are inherent properties of any product, independent of the product use. They can be determined purely from their representation.

The methodology was implemented in an expert system where the following procedure for product performance assessment is used:

1. Define the attributes in users' terms,
2. Determine the model of each attribute (design of the related rule base),
3. Define the combination function.

The implemented system evaluates the proposed design by evaluating the attribute model with the help of the defined combination function. It answers the main criterion and returns an objective value determined for it.

All presentations explained that it is a unique problem to extract the criteria against which a design is evaluated. The discussion on this topic led to the conclusion that the product requirements represent these criteria and that this is the reason why the product requirements, which must contain information of all stages of the product life cycle and especially different views on the product (design view, users' view), have to be formalized in a computable model. However there still remains the problem of weighting different requirements if weights are used. An absolute weighting of the design parameters (requirements) is not useful and it does only makes sense to weight the requirements relative to each other. The question how to represent "trade-offs" leads to the result that trade-offs too can and should be modeled by requirements and that an evaluation system should not only be

used in order to justify a design but also to aid the designer in the decision making process.

In which way requirements can change throughout the design process was another important topic of the discussion and was discussed vigorously. For example, a designer is given the task of designing a bridge over a river and because of certain problems and restrictions, the design ends up as a tunnel. There was consensus that new requirements can appear during the design process but the point at issue was the nature of requirements, the abstraction level of the represented information and, of course, the than open problem of formalizing requirements. The inadequacy in this example was that the problem definition was not abstracted enough, which means that during the definition of the design task the product (bridge/tunnel) has been specified and not the problem represented by the requirements. The designer has to start at a fixed point. That point is the main function representing the intended design. If the main function of the design, to transport objects from one side of a river to the other, is given then the intended design will not be preordained and it is possible that requirements, especially costs etc., can change over time without necessarily producing a totally different product.