Proceedings of ICAD2000
First International Conference on Axiomatic Design
Cambdridge, MA - June 21-23, 2000
ICAD005

THE AXIOMATIC APPROACH IN THE UNIVERSAL DESIGN THEORY

Dr.-Ing. Ralf Lossack
lossack@rpk.mach.uni-karlsruhe.de

o. Prof. Dr.-Ing. Dr. h.c. H. Grabowski
gr@rpk.mach.uni-karlsruhe.de

University of Karlsruhe
Institute for Applied Computer Science in Design (RPK)
D-76131 Karlsruhe
Germany

1 ABSTRACT

UDT stands for Universal Design Theory which is a design theory with the aim of integrating a broad variety of engineering domains, such as mechanical engineering, material science, information science, chemistry, chemical engineering or pharmaceutics. All engineering domains have in common that their overall goal is to create something new in the world, a machine in mechanical engineering or a specific drug in pharmaceutics.

The UDT consists of several parts. One very important part of the theory is the kernel of the UDT which is based on an axiomatic approach. In this paper we want to describe theoretical fundamentals and practical requirements of UDT's axiomatic approach. One important requirement we are dealing with is to enlarge, expand or extend a theory. In the traditional view of axiomatic approaches this seems to be almost impossible. UDT wants to give an answer to this problem.

In our paper we describe the basic elements, an abstraction model of different abstraction domains and the axiomatic framework of the UDT.

Keywords: Universal Design Theory, Design Process, Solution Patterns (Case Based Reasoning), Conceptual Design

2 INTRODUCTION

The concept of *theory* is a *system of statements* which is empirical substantial and contentful, informative, general and rich in explanation. This concept is found mainly in the area of analytical theory of science and in natural science itself and has to be distinguished from the operative or instrumental theories representing structural concepts, e.g. mathematical theories. But nevertheless "Theory should never forget that it is nothing but applied practice" (Gabriel Laub) [8].

Theories of design e.g. [10], [11] or design methodologies in general [7] are different from the concept of theory in science in the sense that scientific theories view nature analytically and in an unbiased manner to find general explanations for its behavior, whereas design theories use phenomena to synthesize artefacts. The process of synthesizing and its result, the artefact, is driven by needs, estimations and evaluations of designers and customers.

A *Universal Design Theory (UDT)* is a design theory containing findings and knowledge about design from *different* scientific and engineering disciplines in a consistent, coherent and compact form. It serves as a scientific basis for rationalizing interdisciplinary product development. A Universal Design Theory takes all the common features of different scientific and engineering domains into account in order to find a *system of statements* of general validity with regard to the explanation and prediction of artefacts and the way of designing them.

In this paper we want to use the term “theory” and “design theory” in the sense it is used in science. And here we want to refer to any system of knowledge that is concerned with the physical world and its phenomena and that entails unbiased observations and systematic experimentation. In general, it involves a pursuit of knowledge covering general truths and the operation of fundamental laws.

To achieve this goal we first look at a simple mapping process which always takes place in design processes. This mapping process maps a problem (or a design task) onto a solution and we call this process a solution finding process or just knowledge about design solutions.

Succesful design requires a lot of experience and knowledge in a certain domain to find proper solutions for given design tasks. UDT which is able to support solution finding processes effectively requires the utilisation of domain independent design knowledge to solve given design tasks. At this we assume the following:

1. In different domains there is knowledge about design solutions which can be assigned to specific tasks (or problems).

2. Complex tasks can be decomposed into partial tasks.

3. The access to domain independent knowledge results from similar design tasks.

4. Knowledge of a certain domain can be applied to and utilised by another domain.

E.g. if there is a partial problem in the technology domain for wich there exists no solution for the time being then it should be possible to find similar problems or tasks in other domains (e.g. biotechnology) for which a solution exists. Then it should be possible to adapt and utilize this solution in the technology domain.

The aim is to establish a theory of design which explains and predicts design processes. This paper is concerned with the conceptual elements of the UDT. These basic conceptions are the *elements* the Universal Design Theory is based on and the *abstraction levels* for the navigation to the design process and the basic assumptions which form the *axiomatic framework* of the UDT.

2.1 ELEMENTS OF THE UNIVERSAL DESIGN THEORY

The explanation and prediction of design and its implementation in a design theory we define the principle elements the UDT is based on.

At this we assume that knowledge about artefacts and knowledge about designing artefacts are interconnected inseparably and can be described by the theoretical elements *object*, *process* and *solution pattern*.

Here an object pattern describes knowledge about design objects and process patterns describe knowledge about design processes. With these elements the mapping process between a design task and a solution can be implemented.

> **Object pattern**
> *An object pattern is an application independent description of a solution which is adaptable to a certain problem. A solution is concerned with knowledge about real objects. Object patterns support the solution finding process by the description of the solution itself in connection with a solution context and a solution supposition. Object patterns represent design knowledge in elementary units explicitly and declarative.*

The principle idea this definition is based on is that the intrinsic design solution is always related with the given design task (problem) and embedded in a design context. Object patterns can further be structured so that mechanisms for decomposition and abstraction etc. processes can be supported.

Artefacts and the process of designing them are interrelated with each other inseparably. Designing them, the design process itself, is controlled by so called process patterns.

> **Process pattern**
> *A process pattern is a set of related object patterns. A process pattern is application independent and adaptable. It includes a description of design steps. A design step is a process which leads to a defined target state with respect to a defined start state.*

A design solution in general which meets the requirements now can be an object, such as a peace of software or a mechanical machine, or a process, such as a workflow. This circumstances are described by a solution pattern.

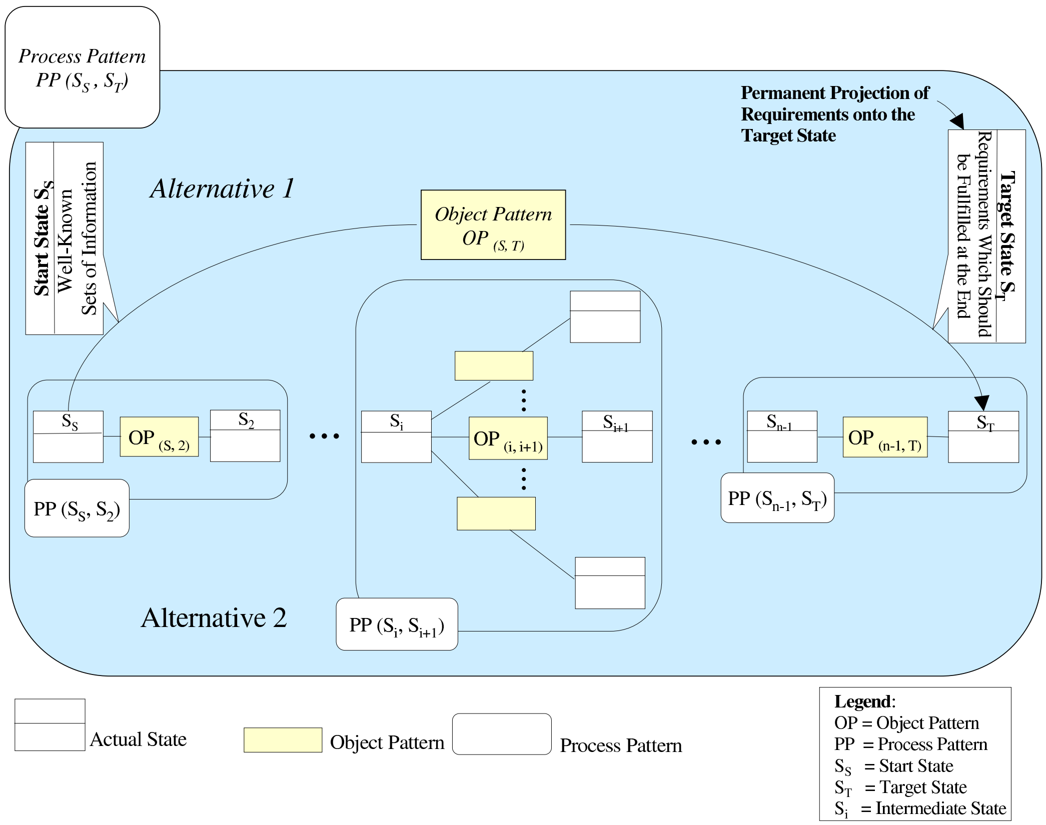

Figure 1: Strategy for implementing the UDT with the concepts of object and process patterns as basic elements

Solution Pattern
A solution pattern is a generalisation of an object and a process pattern. It describes application idependent and parameterised the interconnection between an artefact, its elements and its develompment.

In contrast to process patterns an object pattern describes a statical solution. A process pattern describes a procedure in time. Fig. 1 shows how these two elements work together.

Design processes can be interpreted as information processes which describe (partial) solutions in different states. Beginning with the start state S_S (fig. 1) which includes a set of all information of a design task a target state S_T is achieved via intermediate states S_i by object and process patterns.

If there is a known set of requirements, for which there has been designed a solution, then a known process pattern $PP_{(S_S,S_T)}$ together with an object pattern $OP_{(S,T)}$ leads directly to the target state S_T (Alternative 1 in fig. 1).

If there is no object/process pattern then the strategy is that a set of elementary object/process patterns is used to proceed successively to target state (Alternative 2 in fig. 1). In this case process patterns with a different complexity will be used. At this we distinguish between micro process patterns, such as $PP_{(S_S,S_2)}$ or $PP_{(S_{n-1},S_T)}$, and complex process patterns, such as $PP_{(S_i,S_{i+1})}$, which allows to use different alternative ways when there are different OPs.

Object and process patterns are the conceptual elements to describe a stepwise solution finding process in design. Design in general can be interpreted as a refinement process, leading from the abstract to the concrete, from the general to the specific. This refinement process is sub-divided in different design stages or we want to refer to the abstraction model of design.

3 ABSTRACTION MODEL OF DESIGN

Besides the elements of the UDT the abstraction model builds a basis for the UDT. The abstraction model consists of several abstraction levels each level describing the necessary *information sets* used in design. We concentrate mainly on mechanical and process engineering but assume that these information sets are also true for other disciplines, such as chemical engineering, chemistry, pharmaceutics etc. although the usage of single information sets might differ.

We sub-divide the design process in the abstraction-layers *specification*, *function*-, *physical*- and *embodiment layer*.

3.1 DESIGN SPECIFICATION

The design process begins with a *design specification* where requirements of a design object are specified. At this stage the needs are formulated as complete as possible indicating the intent of a design as precise as possible. Unfortunately in practice the specification doesn't contain a complete definition nor all relevant facts a designer respectively a computer (computational model) would need to come up with a proper solution. Therefore as result one gets a *conceptual formulation* of needs which has to be developed and evolved like the whole design has to be.

The goal of a design specification is to *analyse*, to *describe* and to expose the *aim* of a design so that the *purpose* and the *intention* of the design and not a possible solution is formulated. The result is a design specification or we will refer to a *requirement model* .

A requirement model contains requirements a product or an artefact has to meet. Requirements contain e.g. functional, behavioral, structural information sets and in this sense we talk about required functions, required geometry or the like. Requirements generally represent constraints of design objects and therefore they control the whole design process. Requirements can be classified as follows:

- Depending on their origin, we can differentiate between *external* and *internal* requirements. External requirements are formulated needs by a customer, internal requirements are enterprise-specific and depend on the knowledge of manufacturing processes or the like.
- We can further distinguish *explicit* and *implicit* requirements. Explicit ones have an elementary character. Implicit requirements can be further derived to explicit ones.

3.2 FUNCTIONAL DESIGN layer

At the abstraction level of the functional specification the behavior of an artefact is developed by decomposing the required functions and establishing a network of functions. The function structure and its sub-functions serve as a supposition for the subsequent stages, especially for the physical principle layer. In function modelling there are a lot of suggestions to formulize functions although they can be classified basically into two approaches.

One approach can be called the input-output (I/O) approach, the other one the verb-noun approach. Both approaches have in common that they are decomposed into sub-functions. In the latter case the result is a hierarchy of functions resprectively sub-functions, the result of the I/O approach is a network. The sub-functions establish the basis for the solution finding process for physical principles. Important for the solution finding process is the fact that all functions are described by a set of independent and complete function verbs and quantities. To require independence and completeness of function verbs is important because by this a definite and unambiguous assignment can be guaranteed and all problem areas can be described.

> **Product function**
> *A product function describes the product behaviour by terms of inputs and outputs as well as by a normative set of independent and complete function verbs. The function verbs determine how the input is transformed qualitatively into the output.*

The most abstract structure of I/O functions is the *general function structure*. A general function structure has the three in/output quantities *material*, *energy* and *information* and e.g. according to ROTH [9] the relations *store, channel, transform, change* and *connect.*

> **General Function**
> *A general function is a formal description of the product behavior. The function quantities are*

matter, energy and information and whose function verbs are store, channel, transform, change and connect.

With this concept it is possible to describe *all* technical systems [9]. A function which is builded up by relations composed of physical quantities like force, torque etc. is called a special function. With a special function the energy flow of a technical system can be described very well. Such functions are important e.g. in mechanical, liquid, electrical and thermodynamical systems.

Special Function
A special function is a formal description of the product behaviour. The product is identified by its system boundary and described by a normative set of function verbs and function quantities. The set of function quantities is the set of physical quantities. The combination of special functions by relating them by function quantities is called a function structure.

After completing the function structure the designer determines how to realize each function. First the designer determines physical principles realizing a function and then he/she determines the physical structure and geometry where these physical principles occurs.

3.3 PHYSICAL PRINCIPLE LAYER

In the layer of physical principle modeling each function is mapped on a physical effect. Effects (in the sense law of natures) are the building blocks made available by nature *to cause* a certain effect.

There are physical, chemical, biological etc. effects which establish a relationship between physical quantities. The effects themselves are bounded to material structures the so called *effect carrier*. The result is an effective structure (chain of effects).

Effect
An effect describes a physical phenomenon. An effect is determined by a mathematical formula which defines the quantitative formulated relationship between the physical quantities.

There can be systematically selected those physical laws which match the physical quantities in the function structure. But if there does not exist a direct relationship between the input/output quantities a chain of effects has to be established.

Then an *effective geometry* is established. Here, all information about the structural physical solution is modelled. Each physical effect is assigned an effective geometry which is assembled from the elements *effective line, surface* and *space.*

3.4 Embodiment LAYER

The final shape is determined in the embodiment layer. In this stage features, parts and assemblies of the product are modeled.

4 AXIOMATICAL FRAMEWORK OF UDT

In order to work out a Universal Design Theory, some general assumptions on the process of design can be made. At this as a starting point we have the following axiom:

Axiom 1 (*Axiom of finite physical effects*)
This axiom is concerned with the components or basic elements defined on the abstraction level of physical principles. We know that we cannot design anything which is incompatible with natural principles and laws. And we know that these well-known physical principles are finite.

The finiteness of physical effects is true for a certain point of time, the state of the art in natural science; finding new effects is concerned to be research in natural science.

From this point of view and research being done in design methodology, theory and especially in the field of computer aided design we argue that all the elements used in any abstraction level are finite.

Hypothesis 1 (*Hypothesis of finite basic elements*)
On every abstraction level there is only a finite number of basic elements.

E.g. basic elements on the functional level are basic inputs and outputs, such as *information, energy* and *matter* and operators, such as *change, channel* etc. There are several proposed ontologies for function modelling, such as the verb-noun approach which models functions in another way than the input-output approach, but nevertheless there must exist a limited set of basic elements, as there is a limited set of words we use to express our ideas. Other ontologies are expressed at different abstraction levels.

Hypothesis 2 (*Hypothesis of finite abstraction levels*)
There is a finite number of abstraction levels one can use to model an artefact or to describe design processes.

With regard to the domain of mechanical engineering, these are e.g. the above-mentioned levels of the functional, physical principle layer etc. Different ontologies of functional descriptions are different abstraction levels. The mapping between the elements on each abstraction level can be called a transition between abstraction levels.

Hypothesis 3 (*Hypothesis of finite transitions*)
The number of possible transitions between different abstraction levels is finite. The mapping between the abstraction levels is a mapping between the basic elements defined in each abstraction level.

The mapping between abstraction levels and in abstraction levels is performed by the elements object and process patterns, e.g. from a sub-function to an effect or from functions to its sub-functions.

Based on these, we can state the following:

Hypothesis 4 (*Hypothesis of invention*)
New artefacts are always made and created from a new combination of known basic elements. That applies to the basic elements of the abstraction levels, e.g. a function, a physical principle, an effective surface etc., which are to be combined.

With the hypothesis of invention we have concluded that a new design solution can be generated systematically by mechanisms of combination. And these mechanisms are controlled by a set of requirements given at the outset.

Hypothesis 5 (*Hypothesis of solution finding*)
Each product requirement points at least to one solution area. From this follows that a solution is determined unambiguously if the set of requirements is complete, consistent and valued.

This hypothesis is based on the assumption that there must exist a complete set of requirements after a design has been finished and the same must be true in reverse order. In addition to that the set of requirements must be consistent and valued because the process of requirement modeling is subjective.

4.1 DESIGN EXAMPLE

To illustrate these rather abstract concepts we introduce a small design example. This example shows the design of a robot gripper with the design system DIICAD-Entwurf[1]. This example is not meant to prove or validate any of the statements made but serves as an example.

The design with DIICAD-Entwurf starts with modeling the specification, in a broader sense also called requirements. The specifications are won by clarification of the task and contain the *preconditions* of the design, the *to-be properties* of the future product, the requirements, and the description of the product's immanent *task structure*. In the following the task structure is mapped onto the function structure. This can be done manually or automatically. The reasoning process is done in a case-based manner by object patterns.

On the functional modeling level the product function and its network, the function structure, is established. In this case a product function describes the product behavior in terms of inputs and outputs as well as by a normative set of independent and complete function verbs. The function verbs determine how the input is transformed qualitatively into the output. Functions respectively function verbs are selected from a limited set of function verbs and connected with each other by physical quantities.

Figure 2 (1) shows the established special function structure of a robot gripper in terms of the input-output model. The product behavior, the required transformation process, is described by the energy type *pressure* (input) which is *changed* in the energy type *force*. Then the force is *channeled*, *distributed* and *amplified* (output).

After that or in most cases during function modeling the solution principle is elaborated by combining physical effects according to the required function structure. Here an effect is determined by a law which defines the quantitative formulated relationship between the physical quantities.

An example is shown in Figure 2 (2). Each single function (1) is assigned a physical effect (2). The reasoning process is performed in a case based manner by means of solution patterns.

The solution gained covers all information of a product's *physical* solution, such as the *physical effect* that is described

[1] **DIICAD** in "DIICAD-Entwurf" stands for **D**ialog-oriented **I**ntegrated and **I**ntelligent **CAD** system and **Entwurf** means literally translated design, blueprint, sketch or rough draft with which the *early stages* of a design are intended to support. DIICAD has been developed since 1978. And the module DIICAD-Entwurf supports the early design stages and the embodiment stage and is based on so called design working spaces [6], [5], [2], [3], [4].

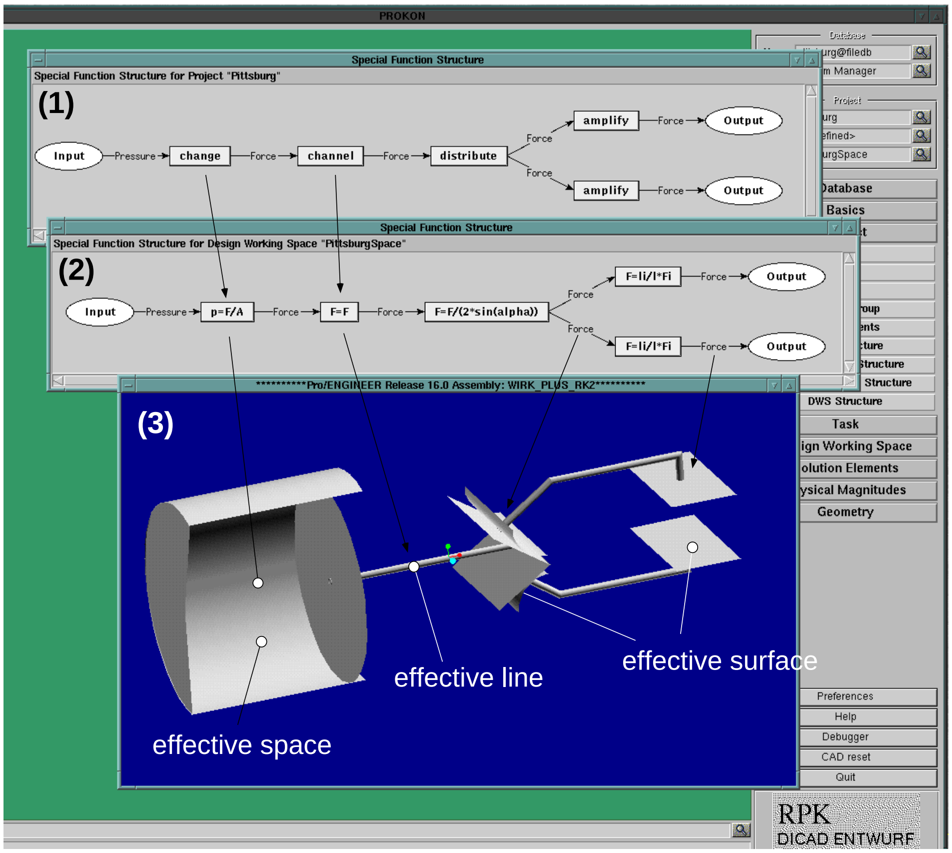

Figure 2: Modeling Structures of the Robot gripper

by a *mathematical equation*, geometrical and structural information such as *effective lines*, *effective surfaces* and *effective spaces* which can be seen on layer (3) (Figure 2).

5 Summary and future work

In this paper we dissussed a proposal for a universal design theory. The research is strongly motivated by the research being conducted so far by the design methodology community with a strong practical background but a more or less weak formal fundament. In this paper we presented the conceptual framework. One aim in the future is to formulate this conceptual and practical model on a strong formal fundament and being able to validate it by observations and experimentation. Another goal is to make this design theory universal, which means to bring together as many different domains as possible, such as chemistry, chemical engineering, material science, technical biology etc. [1].

With the result we want to achieve the following advantages:

- A theory of design which is applicable in practice.
- Experience-based knowledge will be made available for future design processes.
- The design process becomes easier to control since it does not longer depend on the interpretation of a single designer.
- The design process becomes more efficient and reliable.

References

[1] Grabowski et. al. *Universal Design Theory*. Aachen: Shaker, 1998.

[2] Grabowski H., Lossack R.-S. *Cooperation between the Preliminary Design and Manufacturing Stages*. 5TH IFIP WG5.2 WORKSHOP ON GEOMETRIC MODELING IN CAD, May 19-23, 1996, Pittsburg, 1995.

[3] Grabowski H., Lossack R.-S. *Knowledge Based Design of Complex Products by the Concept of Design Working Spaces*. Knowledge Intensive CAD-2, Tomiyama T., Mäntylä M., Finger S., 1996.

[4] Grabowski H., Lossack R.-S. *Formal Aspects and Realisation of Cooperative Design with Design Working Spaces*. IFIP WG 5.2 Workshop on Formal Design Methods, 1997.

[5] Grabowski H., Lossack R.-S., Weis C. *A Design Process Model based on Design Working Spaces*. Knowledge Intensive CAD-1, Tomiyama T., Mäntylä M., Finger S., 26.-29. September 1995, 1995.

[6] Grabowski H., Lossack R.-S., Weis C. *Supporting the design process by an integrated knowledge based design system.* IFIP WG 5.2 Workshop on Formal Design Methods for CAD, Mexico City, 1995, 1995.

[7] Hubka V. *Theorie technischer Systeme.* Berlin; Springer, 1984.

[8] H. Lenk. Epistemological Remarks Concerning the Concepts "Theory" and "Theoretical Concepts". In *Universal Design Theory*. Aachen: Shaker, 1998, 1995.

[9] Roth K. *Konstruieren mit Konstruktionskatalogen, Band 1.* Springer-Verlag, Berlin Heidelberg New York; Bd 1: Konstruktionslehre. -2. Aufl. - 1994, 1994.

[10] N.P. Suh. *The Principles of Design*. Oxford Press, New York, 1990.

[11] T. Tomiyama and H. Yoshikawa. Extended General Design Theory. In *Design Theory for CAD*. Elsevier Science Publishers B.V. (North-Holland), 1987.