A theoretical approach to mechatronics design

Buur, Jacob; Andreasen, Mogens Myrup

Publication date:
1990

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Buur, J., & Andreasen, M. M. (1990). A theoretical approach to mechatronics design. Kgs. Lyngby, Denmark: Technical University of Denmark (DTU). (IK publication ; 90.74 A).

Institute for Engineering Design

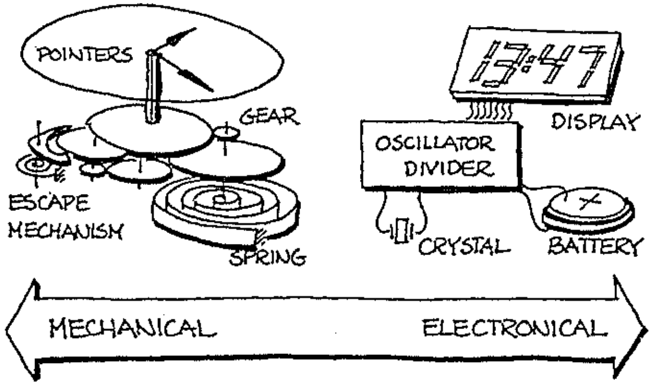
Technical University of Denmark
Building 421
DK-2800 Lyngby   Tel. +45 4593 4316

A THEORETICAL APPROACH TO MECHATRONICS DESIGN   Jacob Buur

KONSTRUKTIONSTEKNIK/DTH

A Theoretical Approach to
MECHATRONICS DESIGN

Jacob Buur



A Theoretical Approach to
MECHATRONICS DESIGN

A THEORETICAL APPROACH TO
MECHATRONICS DESIGN

Jacob Buur

Abstract

This thesis defines mechatronics as a functional and spatial integration of mechanical, electronics and information technology, and it analyses the characteristics and typical properties of mechatronic systems.

Based on interviews with designers in both Danish and Japanese companies, the activity of developing such systems is scrutinized, and it is found that successful product development requires special strategies, interdisciplinary education, suitable project organization and design methods.

The thesis claims that only by treating mechatronics engineering as an independent discipline, is it possible to exploit the full potential of this new technology.

The main concern here is design methodology and, in particular, the methodical basis for the hypothetical activity of 'cutting the cake': that is, of creating the optimal mixture of mechanics, electronics and software in mechatronics concept design.

Literature on mechatronics and related design fields is reviewed for contributions to such a design methodology, and a theory of mechatronic systems is presented in a set of axiomatic statements and theorems. The theory treats such aspects as transformation functions, state transitions, technology allocation and interface organs.

In particular contrasting design principles, i.e. rules of thumb and their contrasts, are found advantageous for creating alternative design concepts by 'moving the boundaries' between mechanics, electronics and software.

Institute for Engineering Design, Technical University of Denmark, 2800 Lyngby

Jacob Buur:
A Theoretical Approach to Mechatronics Design

IK publication 90.74 A

Technical University of Denmark
Building 421
DK-2800 Lyngby
Tel. +45 4593 4316

Preface

This dissertation was born out of the inadequacy of machine design theory to cope with the integration of mechanics with electronics and computer control. It investigates and defines the area of mechatronics in order to establish a theory of mechatronic systems to support the development of new design methods.

But design methodology is not the one and only solution to successful mechatronic product development. The experience from my 20 months study in Japan shows that a much broader view is necessary. Therefore part of this work is devoted to pointing out the requirements that mechatronics puts on business strategies and on interdisciplinarity in engineering training and the development organization.

The dissertation is the result of my PhD study, which was initiated i 1985 at the Institute for Engineering Design, the Technical University of Denmark. It concentrates on the design methodology aspects of the study, since the results of my work in Japan have already been published in the first part of this thesis.

It is primarily intended for researchers in the field of mechatronics, and it is published in the hope that it may add some structure to discussions and future work on the design of mechatronic systems.

I want to thank dr Mogens Myrup Andreasen for his encouragement and patient supervision throughout the process. I am grateful also to all my colleagues at the Institute for contributing to a stimulating and supportive atmosphere for design research.

During the project, a small group of industrialists has followed my progress and willingly discussed proposals and results. I am indebted to Knud V. Valbjørn of Danfoss A/S, Niels E. Modvig of GN Telematic A/S and Knut Meyer of the Institute for Product Development for continuously pressing for sharper formulations and industrial relevancy. Likewise I want to acknowledge the willingness of professor Joachim Heinzl, Technische Universität München, for taking on the task of refereeing my work.

Thanks are due to Kirsten Roikjer and Vibeke Nørly for wordprocessing of this thesis, to Johnny Jensen for reprography, and to Robin Sharp for improving my English. Also my wife Helle deserves mentioning for her loving support and patience.

Lyngby, 25 November 1990
Jacob Buur

Resumé (in Danish)

Denne afhandling definerer mekatronik (apparatteknik) som en funktionel og rumlig integration af mekanisk, elektronisk og informationsteknologi, og den analyserer karakteristika og typiske egenskaber for mekatroniksystemer.

På grundlag af interviews med konstruktører i både danske og japanske apparatvirksomheder granskes udviklingsaktiviteten for sådanne systemer, og det konkluderes, at den succesrige produktudvikling indenfor mekatronikområdet kræver specielle strategier, tværfaglig uddannelse, tilpasset projektorganisation og konstruktionsmetoder.

Afhandlingen hævder, at kun hvis mekatronik behandles som et selvstændigt fagområde, er det muligt at udnytte denne nye teknologis muligheder fuldt ud.

Afhandlingens hovedtema er konstruktionsmetodik og fortrinsvis det metodiske grundlag for det at ’skære kagen’, dvs. at skabe den bedste blanding af mekanik, elektronik og software i apparatudviklingens konceptfase.

Litteraturen for mekatronikkonstruktion og tilstødende områder gennemgås for bidrag til en sådan konstruktionsmetodik, og en teori for mekatroniksystemer opstilles i et sæt af axiomer og hypoteser. Teorien behandler aspekter som proces og funktion, logiske tilstande, teknologiallokering og interfaceorganer.

Især synes anvendelsen af modsatrettede konstruktionsprincipper, dvs. konstruktionstekniske tommelfingerregler og deres modsætninger, at være fordelagtig til at skabe konceptalternativer ved at ’flytte på grænserne’ mellem mekanik, elektronik og software.

Contents

Page

APPENDIX

1 INTRODUCTION

The subject of this research is mechatronics design: The development of technical systems which integrate mechanical solutions with electronics technology and software.

The introduction of microprocessors only 15 years ago has spurred an explosive development in new products of combined technologies. It has also disclosed an enormous business potential, which Japanese industry in particular has been fast in exploiting.

The design of mechatronic systems requires not only the integration of technological know-how, but also the coordination of design practice and design tools from the three fields of mechanical, electronics and software engineering.

The task chosen for this research is concerned with the second problem. It is to examine the characteristics of mechatronics and of mechatronics design in order to establish a theoretical basis for the development of appropriate design methodology. No design theory from any of the separate engineering fields are capable of fully supporting mechatronics design.

The research will focus on the conceptual design phase, since this is the stage where mechatronics is most clearly distinguishable from the traditional technologies which it is composed of.

1.1 The research project

This research was initiated in anticipation of the fact that machines as purely mechanical artifacts are quickly becoming a thing of the past, and that the design methodology taught for machine design is insufficient for describing the design of mixed electromechanical systems.

It was also an opportunity to exploit my personal background in electronic engineering and combine it with my interests for product design and machine design methodology.

The purpose and goals

The project was established on the hypothesis that it would be possible, based on earlier work on theory for the design of machines, to:

- Create a synthesis theory for mechatronic systems.
- Develop terminology and descriptive models.
- Derive methods and working procedures.
- Describe those in a pedagogic and operational methodology for mechatronics design.

It was also assumed that for the research to achieve results of any practical value, it would be crucial to:

- Achieve an understanding of the problems and needs of mechatronics industries both nationally and internationally.
- Contact international discussion partners.
- Include experience of the working practice of successful Japanese mechatronics companies.

My research in Japan made it apparent that a strict view of design as a technological synthesis activity would be insufficient to explain the Japanese success. Mechatronics has a much wider impact on product development in industry.

When taking this into account, the main goals of the present research can be formulated as follows:

1. To define the concept of mechatronics.
2. To determine the required preconditions for successful mechatronic product development in industry, and hereby identify the need for special design methodology for mechatronics.
3. To establish a theoretical basis for mechatronics design, including
   - terminology and descriptive models
   - axiomatic statements on the characteristics of mechatronic systems
   - general theorems describing fundamental relations of importance for the synthesis of mechatronic systems.
4. To formulate design models, methods and principles in agreement with the theory basis, which are applicable to mechatronics design, and to describe them in a pedagogic and operational form.

This research is limited to the design of *mechatronic* systems, like telefaxes, video recorders, robots. The basis of reasoning is an understanding of mechatronic systems and their characteristics, not for instance the designer's way of thinking or the organizational pattern of companies.

Since the focus is on conceptual design, I will not study Computer Aided Design systems, because CAD (in particular for mechanical and software engineering) is simply not yet sufficiently developed to handle conceptual design reasoning.

Scientific approach and verification

This research belongs to the field of applied science, i.e. research aiming directly at the practical applicability of results. According to ROPOHL 1969, such research has always two types of aims:

"Angewandte Forschung hat also stets ein praktisches und ein teoretisches Ziel, wobei ersteres nur über lezteres erreichbar scheint. Da theoretische Erkenntnis bei angewandter Forschung nicht um ihrer selbst willen angestrebt wird, müssen sich die theoretischen Erkenntnisziele aus praktischen Gestaltungszielen ableiten lassen."

(Applied research always has a practical and a theoretical goal, where the first apparently can only be reached through the last. Since the theoretical results of applied research are not pursued for their own sake, then the theoretical goals must be derivable from the practical goals.)

This is indeed the case for the goals of this project: The explanation of preconditions for mechatronics product development (2) requires a sharp definition of the object (1), and applicable design tools (4) require a theoretical basis (3).

HUBKA 1990 has described the structure of design science. He states that all contributions can be distinguished according to their prescriptive (instrumental) or descriptive (explanatory) nature and to their focus on technical systems or on the design process. This is illustrated in Figure 1.

This work belongs mainly to the lower left corner of the Figure. It is concerned with the nature of mechatronic systems, not with the design process as such, and it is descriptive, since it tries to base design methods on an understanding of the object of design. Naturally, the end result will incorporate prescriptive elements, because methods are instructions for how to work.

A major obstacle in design science - and also the reason why design research still has difficulties in being accepted as a field of science - is that it is almost impossible to verify theoretical results empirically.

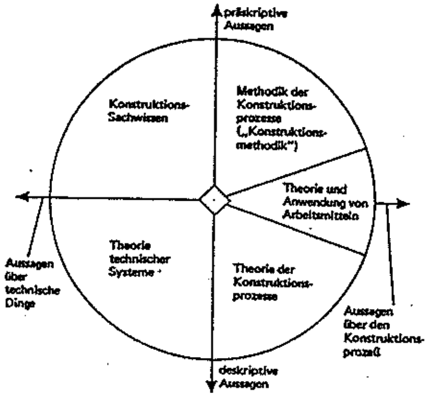


Figure 1 The structure of contributions to design science, HUBKA 1990

The classical verification of design methods demands that their application to the practical design of artifacts is successful.

Two factors make such design experiments unrealistic: The first is that the design process is stochastic in nature, i.e. a new design method may raise the probability of success, but does not guarantee it. The second is that the number of influencing factors is extremely large, which makes a precise repetition of an experiment virtually impossible.

So for this research, other means of verifying the validity of design theory must be found, for instance:

Logical verification

- Consistency: there is no internal conflicts between individual elements (e.g. axioms) of the theory
- Completeness: all relevant phenomena observed previously can be explained or rejected by the theory (i.e. observations from literature, industrial experience etc.)
- Well established and successful methods are in agreement with the theory
- Cases (i.e particular design projects) and specific design problems can be explained by means of the theory.

Verification by acceptance

- Statements of the theory (axioms, theorems) are acceptable to experienced designers
- Models and methods derived from the theory are acceptable to experienced designers.

Logical verification has the draw-back that design theory is basically confirmed by *analysis* of cases and observations. This does not automatically provide a guarantee for the validity in *synthesis* activities.

Verification by acceptance implies a pedagogic problem: The willingness of a designer to accept a statement or method depends on both his own situation (his need, knowledge, experience), the complexity of the information (how much training is required?), and the pedagogic presentation.

Both types of verification method will be applied in this research. It must be noted also that verification of design tools is not so much a question of wether they 'work' or not, it is a comparison relative to the qualities of existing tools and working practice.

Research method

I will base this study on the design theory advocated by the Workshop Design Konstruktion (WDK), in particular the Theory of Technical Systems suggested by HUBKA 1984 and the Theory of Domains of ANDREASEN 1980.

Also, for an overall understanding of the role of design methodology in product development, I will apply the three-level approach presented by ANDREASEN & HEIN 1987: problem solving, product synthesis and product development.

This research project has been a constant alternation between observing real phenomena in industry or in literature, formulating hypothetical statements on the nature of mechatronics, experimenting with these hypotheses on product examples and design cases, and confronting designers and researchers with preliminary findings.

I will briefly comment on the sources of information exploited in this research:

Literature Mechatronics is still a very new field without major publications. Material on design aspects is found almost exclusively in articles of journals and in conference proceedings. The number of contributions is accelerating, but they present solitary ideas with very little synergy (cross-referencing) between authors. Theoretically well-founded discussions of methodological aspects of mechatronics design do not exist. A number of Japanese contributions on design are available, but only in the original language. The neighbouring fields of Feinwerktechnik (BRD), Feingerätetechnik (DDR), and Instrumentation Engineering (UK) share some similar design problems and are much better documented.

Teaching material Unpublished lecture notes from mechatronics engineering courses in Finland, Germany, Britain and Japan have provided useful information on university teachers' ideas on design.

Interviews in industry Designers in industry will seldom voice a need for design methods, because that would be like criticizing their own way of working (and even their way of thinking). With some practice, however, it is possible through an interviewing technique to detect problems of methodological nature in a designer's working practice. One important experience is that only designers with some systematic method (acquired by experience or through education) are able to explain in general how they work: "We usually do like...". This research comprised 24 interviews in Danish mechatronics industry and 38 in Japanese industry, including case-studies. The interviews have mainly provided background information for this thesis.

Case-studies in industry To analyse completed or still running development projects in companies is a means of obtaining information on practical design problems and problem solving patterns. There have been no attempts to directly influence projects. Industrial projects differ very much as to initial conditions, goals, ambitions, resources etc. Therefore it is almost impossible to derive any quantitative results from such studies, and even general conclusions are difficult to draw from a limited sample of projects. This research included 14 case-studies in Denmark and 9 in Japan. The experience achieved through case-studies will mostly be included in the thesis as examples.

Discussions with university professors Engineering education in mechatronics design aspects is often provided by university teachers with little personal design experience and varying understanding of and attitude towards the design activity. But the fact that they are forced to abstract and generalize knowledge of design in order to teach students, makes discussions worthwhile. This project included visits to universities with mechatronics activities in Denmark, Finland, Germany, Britain and Japan and discussions with approximately 20 professors.

Educational projects Some experience was also achieved from students' design projects and MSc-theses at my home institute. In students' projects it is possible to experiment quite freely with new concepts and methods, but one must accept that most engineering students contribute with very little design experience or practical feeling for technical solutions.

All interviews in industry and universities and the case-studies have been documented previously in a series of 6 progress reports. Please see page 111 for particulars.

Research in Japan

As a part of this project, 20 months research was carried out at the Department of Mechanical Engineering, University of Tokyo, during 1987-89.

The study focused on activities related to mechatronics design research and education in universities, and on mechatronics product development in industry. A period of intensive language training was included to achieve sufficient abilities in Japanese to allow visits to companies and discussions with designers in their native language.

The main conclusions of the investigations in Japan were that:

1: Mechatronics companies in Japan compete chiefly in growth markets, and they apply a set of product development strategies quite different from their western competitors. They seem to react fast to changes in competition, to shorten the product cycle to spur demand, to emphasize competitive product properties, and to plan carefully for new market opportunities.

2: Japanese companies create the interdisciplinary basis and cooperative attitude necessary for mechatronics product development by generalistic training of their employees, favorable employment conditions and suitable project environments.

3: Japanese designers apply design methods mostly for general problem solving tasks, for project management, for quality assurance and for industrial design, - but not for the synthesis of mechatronics.

4: Japanese universities play a minor role in design methodology research and in mechatronics engineering education.

In this thesis, the Japanese experience will contribute to Chapter 2 and 3, i.e. to the investigation of mechatronics technology, and the discussion of product development characteristics.

Results from the research in Japan have already been published in 1989 in the first part of this thesis, intended mainly for designers and executives in industry. It describes strategies, methods and working practice in Japanese mechatronics companies, and it includes an appendix of Japanese design methods and an extensive bibliography.

To support the message of the report in a more comprehensible form, but also to add to authenticity, I have completed a video movie on the subject of Japanese product development in mechatronics industry. Through interviews with industrialists it illustrates attitudes to competition and employee training, and it points them up with scenes from Japanese culture.

Experience on design methods and industrial design, which exceeds the scope of this thesis have been published in separate articles. For details on previous publications, please refer to page 111.

1.2 The structure of this thesis

The dissertation basically consists of three parts: a study of the mechatronics concept, a review of design methodology literature, and the development of theory and tools for mechatronics design.

Each part will be treated in two separate chapters as briefly described below:

Chapter 2 focuses on mechatronics as an integrating technology and claims that mechatronics must be treated as an independent engineering discipline oriented toward product development. I conclude the chapter by suggesting a rigorous definition of mechatronics.

Chapter 3 states that the development of competitive mechatronic products requires company strategies, interdisciplinarity and dedicated design methodology. The level of design methodology is identified.

Chapter 4 gives an extensive review of available literature on methodology for mechatronics design. No consistent design theory is yet to be found, but when observed thorough the framework of the Theory of Domains, the fragments add up to a total understanding.

Chapter 5 discusses design theory and methodology from the fields of mechanics, electronics and software for contributions to a design theory for mechatronics.

Chapter 6 examines all those aspects which set mechatronic systems apart from machines, electronic devices and software. A new theoretical basis is formulated in a set of axioms and theorems.

Chapter 7 suggests models, methods and principles derived from the theoretical basis, which are advantageous for the conceptual design of mechatronic systems.

The thesis is concluded with a discussion of results and their relevancy in the final chapter. The appendix contains two previously published articles on the characteristics of mechatronics design and on mechatronics design modelling.

2 WHAT IS MECHATRONICS?

In the mid 1970's the word *mechatronics* started appearing in Japan, first in a company advertisement, later it was used by a ministerial committee to describe the rapidly increasing tendency to combine *mecha*nical technology with ele*ctronics* and computer control to enhance performance and flexibility of products and manufacturing equipment.

Mechatronics has become a catch-word, and the penetration of this word in Europe and the USA has indicated not only a growing awareness of the competitive potential of combining mechanics and electronics, but also a growing concern about Japan's strong leader position in this field.

Mostly robots and CNC (computer numeric controlled) machine tools have been in focus as mechatronic equipment, but in this thesis I will argue that a much wider, product-oriented view is necessary, and that in fact mechatronics must be regarded as an independent engineering discipline, and not just as a combination of traditional engineering fields, in order to exploit the full potential of the symbiosis of mechanics, electronics and software.

In this chapter I will study the characteristics of mechatronics in order to define the subject of research. A definition of the technology itself is a necessary precondition for treating the development of mechatronic systems.

2.1 The concept of mechatronics

Besides robots and computer controlled machinery, the group of typical mechatronic systems includes video recorders, photocopiers, electronic cameras, electro-hydraulic actuators, sensors and so on. There is not yet any generally accepted definition which clearly distinguishes mechatronic systems, but some typical patterns may be found.

When using the term *mechatronic system* in the following, I will understand a subset of all technical systems, as illustrated in Figure 2. When we distinguish according to applied technologies, mechatronic systems are hierarchically at the same level as mechanical, electrical, software systems etc. I will also use the word *product*, to emphasize the system as a commercial artefact.

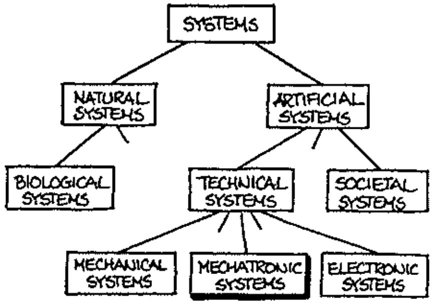


Figure 2 Mechatronic systems as a subset of technical systems, after HUBKA *1973*

Typical characteristics

A number of authors classify mechatronic systems according to the impact of electronics on the systems concept, FRÖLICH 1982, FUJIWARA 1984, GREGORY 1986, KISAKA 1986:

1. Electronic control is added to traditional machine systems (e.g. CNC machine tools, electronic controls for engines).
2. Some of the mechanical functions are replaced by electronic devices (e.g. in sewing machines, the gear mechanism has been replaces by stepping motors with electronic control, and in telex transceivers, the mechanical coding mechanism was replaced by digital logic).
3. The mechanical function is totally replaced by an electronic concept (e.g. electronic calculators have totally replaced mechanical types).

The mechanical functions, which are replaced by electronics, are often functions with a control purpose, as shown in the example in Figure 3, where the rigid, mechanical coupling between three shafts is replaced by three independent drives under computer control. Mechatronic systems are *flexible* in the sense that software control algorithms are easier to alter and diversify than machine structures, ISHII 1989.

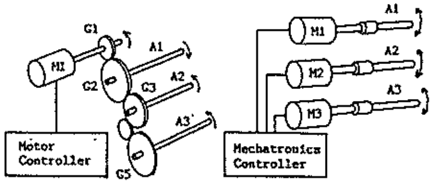


Figure 3 Example of a mechatronic concept: a complex mechanism is simplified by multiplying the electric drives and adding computer control, YAMAZAKI 1987

A mere replacement of mechanics with electronics or addition of electronic control to machines however, does not fully explain the magnitude of mechatronic systems. Some systems, like telefax and video recorders, are only possible because of the integration of mechanics and electronics, and some systems obtain completely new functions, FURCHERT ET AL 1979, TAKEUCHI 1986, like the auto-focus function of electronic cameras and the memory function of electric typewriters. Mechatronic systems seem to take over *intelligent* functions from the user, where machines have traditionally replaced the muscular, energy functions of man.

When introducing microprocessors into machines, the system seem to leap into a kind of *multifunctionality*, where a large number of features become available under software control at low cost. The trade-in is an increase in *complexity*, which is likely to cause difficulties in the interface between the system and its user, ISHII 1989.

Further, a number of general properties will be improved by using the mechatronic concept, KAJITANI 1986, RIETDIJK 1989: reductions in *weight* and *size* due to the miniaturization of electronics, increase in *reliability* by reducing the number of moving parts, savings in *energy consumption* because of the shift from mechanics to electronics and due to more efficient control, and saving in *cost* by applying electronic standard components rather than specially manufactured machine parts.

The mechatronics evolution

According to Japanese authors mechatronics has developed from mechanical technology. The fusion of mechanics with electrical technology led to electro-mechanics, which then integrated with microelectronics and software technology to form mechatronics, as illustrated in Figure 4, KAJITANI 1986, 1989 and KUROSAWA 1983.

There are in particular three circumstances related to microelectronics, which have pushed the development of mechatronics forward:

1. Cheap, mass-produced integrated circuits have made it attractive to replace mechanical functions by electronics.
2. The appearance of the microprocessor has made it possible to control mechanical processes simply, precisely, and at low cost.
3. The reliability of electronic components and circuits has become high enough to withstand vibration, heat and other influences caused by physical integration with mechanics.

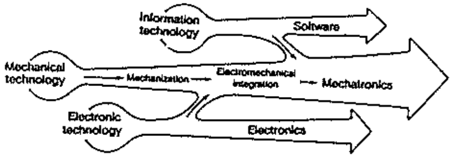


Figure 4 The evolution of mechatronics from mechanical technology, KAJITANI 1989

VORBACH 1982, KUROSAWA 1983, ITAO 1986 and YAMAZAKI 1987 have traced the tendency of the mechanisms part giving way to an increasing amount of electronics and software in different product types. Figure 5 shows roughly the ratio of contents of mechanisms, electronics and software changing through six successive models of teleprinter equipment.

The 'softening evolution of machines' is not limited to the systems themselves, but it has affected also the means for their development and production: CAD/CAM systems, flexible manufacturing systems, CIM etc., FUJIWARA 1986.

The new technologies which will most strongly influence mechatronics in the near future, are IC technologies (ASICs and Smart Power chips), optical technology, micro-mechanics and distributed intelligence.

ASIC technology (Application Specific Integrated Circuit) spurs a movement away from the general purpose, off-the-shelf components towards ICs designed specially to suit every company purpose. YAMAZAKI 1987 envisions complete one-chip microcontrollers including microprocessor, peripherals and algorithms for intelligent mechatronics control, to be designed by engineers without special knowledge of semiconductor fabrication through the use of silicon compilers. One-chip solutions have the advantages of small size and high reliability, AMSTUTZ 1987.

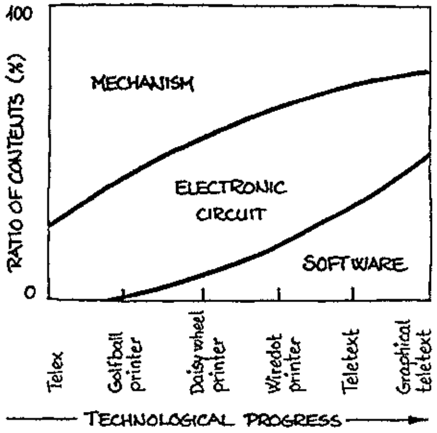


Figure 5 Content of mechanisms, electronics and software in teleprinter models, ITAO 1986

ASIC technology is also a movement opposed to the tendency towards an ever increasing part of 'flexible software' in mechatronic systems. Some authors have labeled this new prospect 'super-mechatronics', DINSDALE 1988.

Smart Power technology means integrating digital control electronics with power electronics on the same chip, YAMAZAKI 1987. The advantages are fast control of high power, simple wiring, small size and high reliability compared to conventional solutions with separate switching transistors. Also it becomes possible to monitor continually the power components for rises in temperature and malfunctions. A potential area of application is automotive electronics.

Optical technology provides devices for input (image sensors, hologram scanners, etc.), output (semiconductor lasers, light emitting diodes), transmission (fibers, waveguides), storage (optical disks) and eventually for signal processing. Examples of current 'opto-mechatronics' applications are bar-code readers, laser printers, CD-players, optical disk storage systems, high precision measuring systems, ITAO & NISHIDA 1989.

Micro-mechanics technology is the technology of designing mechanical structures on semiconductor material, like in available IC pressure sensors. The perspective of micromachines and microactuators integrated with sensor and micro processor on a single chip opens a new field of ’micro-mechatronics’, BAJOREK ET AL 1989, ISHII 1989.

Distributed intelligence will make every component of a system autonomous, so that they can react independently to changing working conditions. Intelligent sensors and actuators will not just be connected by signals and power, but communicate by data. This will add redundancy, reliability and flexibility to future systems. We may see an increasing tendency for mechanical structures to have a ’live state’ and ’dead state’, like the Control Configured Vehicle, an aircraft which is only aerodynamically stable under computer control, ISHII 1989.

Besides these technological innovations, an increasing user consciousness is bound to change mechatronics. The days when mechatronic products were selling solely because of their functions have passed - now, with competing models in plenty, buyers tend to pay more attention to the man-machine interface and to whether the product is expressing ’life-style’. This is also a consequence of the fact that functions of mechatronic systems are becoming less visible with increasing content of electronics and software.

The forerunners

The evolution towards integration of mechanics with electronics and software in industry products has taken place in all industrialized countries, though maybe not with as much focus as in Japan. In Europe, every country seem to have adopted its own term for areas corresponding more or less to mechatronics: Feinwerktechnik, Gerätetechnik, Instrumentation Engineering, Apparatteknik and so on. Like mechatronics they are not sharply defined and do not allow a clear distinction between systems belonging to this and to other industrial fields but one will notice differences which mirror the industrial tradition of each nation. This section will briefly review some of these terms and their origin.

Feinwerktechnik (BRD) appeared as a new word in West Germany just before the second world war, but not until 1968 was there an agreement to apply this name cosistently in industry and in engineering universities, ZWICK 1971. ’Feinwerktechnik’ is strongly related to ’Feinmechanik’, i.e. the science of designing and manufacturing small-scale mechanisms and parts, so Feinwerktechnik education is mostly located in the mechanical engineering faculties of universities. There is, however, also a strong tradition of electromechanical engineering education in the electronics universities of West Germany, mainly aimed at the electrocommunications industries, KEILWERTH ET AL 1980.

In the design of ’Feinwerktechnik’, the Germans have emphasized an abstract, function oriented design methodology, based on the observation that ’Feinwerktechnik’ systems process signals, as opposed to transforming energy or materials, ROTH 1968.

The terms *Feintechnik* and *Microtechnique* are occasionally used for the same field in Switzerland (in the German and French speaking areas respectively).

Gerätetechnik (DDR) or *Feingerätetechnik* is the name used in the German Democratic Republic for the area identical to ’Feinwerktechnik’, HILDEBRAND 1969, FURCHERT 1979. It is derived from a distinction in the German language between the words ’Maschine’, ’Apparat’ and ’Geräte’, which may be defined to describe systems with the main purpose of transforming energy (e.g. engines), material (e.g. compressors, lathes) and information (e.g. measuring instruments) respectively, KOLLER 1976.

Courses in ’Feingerätetechnik’ are mostly based in Departments of Electrical Engineering in the DDR.

Instrumentation Engineering (UK) is the term used in Britain for the discipline of designing and establishing measuring systems for evaluating physical variables, e.g. BARNEY 1985. This field is clearly electronically biased, i.e. it is implied that the processing of measurement information is processed by electronic circuits. Compared to the German ’Feinwerktechnik’, the field of instrumentation engineering is more sharply defined, but also much narrower in scope.

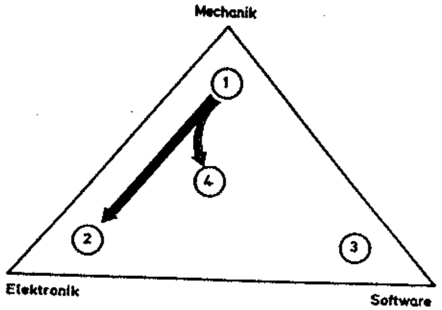


Figure 6 The characteristics of mechatronic products (4) in relation to mechanics, electronics and software, FRÖLICH & SCHLOTTMANN 1982

Apparatteknik (DK) was suggested in the early 1970’s in Denmark to cover the design of electro-mechanical systems. It includes mechanical aspects of electronic systems, like cabinet design and electronics packaging, NIELSEN 1976, MEYER 1979. Quite in contrast to the German ’Apparat’, the word in Danish means ’small-sized electric equipment’. There is no great tradition of precision mechanical work in Denmark like the one in Switzerland or Germany, so ’Apparatteknik’ implies an electronic foundation, and the existing engineering courses in this field are mostly based in departments of electronics.

Since 1985, or at about the time, when the Commission of The European Communities established a working group on the subject, many organizations and national bodies in Europe have adopted the term mechatronics for activities already in progress, but often without formulating any clear definition. As a result, the meaning of the word varies from country to country, depending on traditions in industry and university systems.

Distinguishing mechatronics and electronics

It seems difficult to draw the line between mechatronic and electronic systems: a telefax device is surely mechatronics, but how about a push-button telephone or a computer? The electronic circuit itself is not mechatronics, but the design and manufacture of the components and the circuit both have strong mechatronic aspects, DETTER 1984.

As illustrated in Figure 6, FRÖLICH & SCHLOTTMANN 1982 characterize systems with respectively mechanical, electronic and software orientation in order to point out the features of ’mixed’ systems. In electronic systems (e.g. pocket-calculators, communication equipment) and software oriented systems (e.g. programmable machine controllers, microcomputers) the mechanical functions are limited to housing and parts assembly, but in ’mixed’ systems (e.g. electronic typewriters), the main functions are assigned to both mechanics, electronics and software during the concept design phase.

Even in systems which are considered ’purely’ electronic, there are two aspects of mechanical importance:

1 When the electronic system exchanges information with the user, and when it exerts control on mechanical processes, then interface elements such as sensors, actuators, control elements, displays and connectors become indispensable. Such interface elements are always mechanical or electro-mechanical in nature, ROTH 1985.

2 The manufacturing of electronic circuits is a mechanical process, so the detailed design phase of electronic systems (’electronics packaging design’) is in fact mechanical design, but with a number of additional constraints: heat dissipation, electromagnetic noise suppression, nonignorable resistance of conductors, voltage insulation etc. are determined in this phase, FRÖLICH & SCHLOTTMANN 1982.

The best examples of intricate integration of mechanics and electronics on the detailed design level can be found in transducer and actuator designs, where performance is determined by not only mechanical and electrical properties, but also by physics, acoustics, chemistry and so on.

The definition of mechatronics must determine, how mechatronic systems should be distinguished from electronics. If it includes electro-mechanical integration in the design of interface components and in the detailed design of electronics, then also very ’electronic’ systems must be considered as mechatronics.

2.2 A new engineering discipline

Should mechatronics be regarded as anything but a clever combination of existing engineering fields? It seems that the full potential of mechatronics cannot be described by focusing solely on traditional technologies. PRESTON 1989 states that "mechatronics is also a mental attitude - a way of looking at problems", and KAJITANI 1986 points out the need for a ’philosophy of mechatronics’.

Not just a technology combination

In spite of what the word may suggest, mechatronics is not just a replacement of mechanics with electronics or a simple combination of mechanics, electronics and computing technologies. It is more than that, TAKEUCHI 1986, KAJITANI 1986, DINSDALE 1988, JSME 1989, PRESTON 1989.

The extra ’something’, which can be achieved using the mechatronic concept, is likely to include one or more of the following five features:

1 Realization of functions not seen before (because they were either not technologically possible or economically realistic), e.g. recording video signals on magnetic tape.

2 Extension of the range of parameters used for machine control into the ’intelligent functions’ of man, e.g. numerical control of machine tools, adaptive heating control.

3 Increase of flexibility both during design and use and multifunctional abilities, e.g. videorecorders.

4 Compensation for weaknesses in mechanism designs or mechanical structure design, using electronic control to increase quality of performance, e.g. tangential record player, stepping motor drive in electric sewing machines.

5 Physical integration of mechanics and electronics in one body to reduce size, manufacturing costs etc. e.g. integrated, intelligent sensors, electro-hydraulic components.

The symbiosis of mechanics and electronics means that previously unknown types of solutions evolve, so that the designer has to work without design examples in mind.

Mechatronics is not a new specialization, but an *integrating discipline*, DINSDALE 1988, RIETDIJK 1989:

> "The whole concept of mechatronics is different: instead of being a new specialised subject area developing out of an existing discipline, it is drawing together elements of existing subject areas, (...) a movement which opposes the traditional style of academic development."

> "Mechatronics is not a new branch of engineering, but a newly developed concept that underlines the necessity for powerful interaction between different fields of engineering."

Some authors regard mechatronics as the future discipline of mechanical engineering, ENGLER 1990, KAJITANI 1989. They offer the opinion that ’purely mechanical’ machines will not exist in the future, so all mechanical engineers need to be able to handle mechatronic type solutions. RIETDIJK 1989 suggests that mechatronics will have a rejuvenating effect on mechanical engineering.

It is not only its interdisciplinary nature, which sets mechatronics apart. A major difference in relation to mechanical and electronic engineering is that mechatronics is designed based on *knowledge of available manufacturing technologies* rather than of what is realizable according to physical laws and engineering fundamentals. Also this is a tendency in conflict with the academic approach to engineering.

A total approach

A mechatronic solution to a problem involves principles from a wide range of theory areas like control theory, acoustics, optics, physics, electromagnetics, fluid mechanics. In particular this is true for transducer and actuator design.

This combination of specialized subject areas suggests that the set of alternative mechatronic solutions to a problem is very large. To handle mechatronic concepts, one needs both a total view of available technologies and a systematic attitude towards technology combination.

In contrast to most other engineering fields, mechatronics engineering is characterized by a *generalistic approach* rather than a specialized attitude.

Roles in industry are changing

It follows that the role of the engineering designer is changing in mechatronics industries. Besides sufficient knowledge of specialized areas, the designer now also needs to have the ability to suggest and evaluate alternative combinations of technologies. He must be able to see how principles from different areas may complement each other and enhance the overall performance.

The impact of mechatronics, however, is not limited to the design department. Also the roles of marketing, manufacturing and service experience a change:

Marketing finds an increased need for application knowledge and knowledge of buyer and user characteristics. This is because mechatronics increases the content of application specific information in products by taking over intelligent functions from man, and because the flexibility of mechatronics encourages the proposal of product variants.

Manufacturing will as a result feel the urge to produce smaller lots of product variants but at costs equal to those of mass-production. Manufacturing of mechatronic systems is multiskilled, since it includes production and test of both mechanical (and sometimes optical) parts, electronic circuits and computer software.

Service demands multiple skills, since defects in mechatronic equipment may locate in mechanical, electronic or software systems. Service organizations geared for repair of mechanical parts are forced to change towards replacement of electronic modules and subsystems, with resulting changes in spare part policy etc.

In conclusion, mechatronics engineering cannot be seen as a simple pooling of skills from the traditional fields of mechanical and electronic engineering. Mechatronics is a generalistic, integrating discipline based on wide technology knowledge and systems thinking. The impact of mechatronics is not limited to the design department, but is altering the roles of marketing, manufacturing and service too.

2.3 A product oriented activity

The term mechatronics applies to components (e.g. sensors, actuators), products (e.g. video recorder) and to larger systems (e.g. flexible manufacturing systems, computer integrated manufacturing). In this section I will argue that a common point for all applications is that mechatronics design is a *product oriented* discipline.

Here, ’product-oriented’ should be understood in the sense that mechatronics is not a strictly technological discipline, since its very existence is justified only by business aspects. The main purpose of applying mechatronics is to create competitive products that respond to needs, not merely to perform technological functions at a high level of sophistication.

The business viewpoint

The business aspect is an intrinsic part of mechatronics, HEIN 1989:

> "Mastering the individual mechatronics disciplines, without a strong perception of the rules of disposition, does not make for good mechatronic products."

The main assertion here is that the team of engineering designers pre-decides the busi-

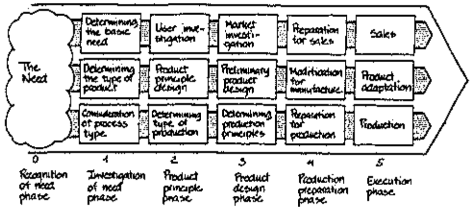


Figure 7 The strategy of 'Integrated Product Development', ANDREASEN & HEIN 1987

ness potential of a product already during the design phase. This is because the dispositions of the designer severely limit the gap between sales price and production cost to be achieved by marketing and manufacturing departments. ANDREASEN & HEIN 1987 have formulated a strategy for 'Integrated Product Development' to clarify the interactions between the activities of marketing, design and manufacturing, Figure 7.

The idea that the designer pre-determines the business potential to be created by the product, is true not only for mechatronics, but for all industrial products. In mechatronics however, the cooperation of specialists of very different technologies seems to make it even more difficult to let design be governed by business aspects rather than technical performance.

The mechatronic product must be understood both as a response to needs among customers and as the business foundation of the company. The design requirements must therefore contain not just functional specifications, but all determinative aspects of design, manufacturing, marketing, use, and destruction, as in TJALVE 1979.

In mechatronics it becomes increasingly important to distinguish between *duty bound* and *success parameters* that is, between the properties, which are just necessary preconditions for selling the product ('state-of-the-art'), and those which actually increase its competitive edge.

As an example, let us discuss the design of video tape recorders for home use. At present potential buyers expect a VTR to record and playback TV programmes at a certain quality level and to allow programming, picture search, and remote control. Recording duration on standard tapes, the number of TV channels, and the number of events to be pre-programmed for automatic recording have all reached a level where they hardly count as sales points. The properties that actually sell the product - the success parameters - are quite different: industrial design, ease of operation, price level, and marginal features like teletext and editing functions. So when a major Japanese electronic company introduced a bar-code reader as alternative to the tiresome button programming of VTRs, this was not merely an additional feature, but a deliberate attempt to exploit 'ease of operation' as a success parameter.

It seems that mechatronics has developed to a stage where mere functional performance does not determine the success of the product in a competitive environment - at least for a great many applications . So the understanding of success parameters is indeed important, but it is not included in the traditional engineering disciplines of mechanics and electronics; they are strictly limited to functional considerations.

Linking product- and production technology

One characteristic of mechatronics is that design solutions are closely interlinked with available manufacturing technologies, and that they often have to be developed simultaneously. Most clearly this is seen in the development of sensors and actuators (e.g. optical scanners in telefaxes, recording heads in video equipment, LCD displays, computer printing heads), but it is also the case in products like the Walkman, where miniaturization is the main issue.

This link between product and production technologies differs in strength in different areas:

- The components in electronics are mostly standard parts to be connected by a limited choice of well-described packaging technologies (cabling, printed circuit board, thickfilm etc.). Manufacturing technologies are mostly developed quite independently of applications.
- Mechanical products consist mainly of unique parts designed with a large choice of alternative materials and processing technologies. Manufacturing technologies are often critical.
- Micromechanical structures depend totally on the IC manufacturing technologies. The technology is still very much at the basic research level.

2.4 Defining mechatronics

A universally accepted definition of mechatronics does not yet exist. Many individual researchers, national boards and engineering associations have given the field their attention, so a number of proposals for definitions are available.

They range from very narrow descriptions of electronic control of mechanisms to open, all-embracing definitions, for instance those of TAKEUCHI 1986 and ITAO 1986:

> **"Mechatronics is characterized by the use of electronics in controlling mechanical systems to enhance the controlling performance."**

> **"Mechatronics is a system for transmitting, processing, transforming and preserving energy and information."**

The key-words found in most definitions are the *combination* (of mechanics and electronics) to form *intelligent* and *flexible* equipment, like in the formulation used by the Finnish Federation of Metal and Engineering Industries, SALMINEN & VERHO 1989:

> **"Mechatronics is a combination of electronic information technology and mechanics, intended to raise the intelligence level of machines and devices and at the same time increase their flexibility, versatility, efficiency and reliability."**

The definition proposed by a working group under the Commission of the European Communities describes the activity of mechatronics design rather than the substance of mechatronics, IRDAC 1986:

> **"Mechatronics is a synergistic combination of precision mechanical engineering, electronic control and systems thinking in the design of products and manufacturing processes."**

That there is indeed confusion as to the nature of mechatronics, can be seen in the following definiton proposed at a recent conference in the UK, HURST & JAMES 1990:

> **"An attitude or philosophy which instills an integrated multi-disciplined approach to problem solving. It is by its nature a team activity involving mechanical, electronic and software engineers, along with personnel from other departments in the company. It aims to produce optimum solutions to many different classifications of problems. Mechatronics is not a science but a technology."**

Is mechatronics an attitude, an activity or a technology? No consensus on a definition is yet evolving. For the time being, the meaning of mechatronics depends on the viewpoint and interests of the people using the word, and the word seems to inherit a different bias in every country, depending on traditions in industry and education system. There is a need for a concise, generally acceptable definition.

The purpose of a definition

What is the purpose of defining mechatronics? In this work, a definition is necessary as a basis for design methodology research. It is required for determining the scope of systems, for which the design theories and methods should be valid. But there are other reasons for formulating a definition:

- The establishment of education for specialized personnel (engineering designers, technicians, equipment-operators). Mechatronics needs personnel with wider qualifications than are taught in traditional mechanical or electronics courses.
- The coordination of special initiatives and support to promote mechatronics industries. For many countries, mechatronics industry is attractive, because it is an expanding, knowledge intensive business.

It seems indispensable to distinguish between a definition of *mechatronics technology* (i.e. the substance, of which systems are composed) and a characterization of *mechatronics engineering* (i.e. the activities and skills, which are needed to design and produce such systems).

Which criteria define mechatronics?

The purpose of this paragraph is to discuss the general characteristics of mechatronic systems described in section 2.1, in order to single out criteria for distinguishing mechatronics from mechanical and electronics technology.

Main function The defining criteria of the related areas Feinwerktechnik (BRD), Gerätetechnik (DDR), Instrumentation Engineering (UK), and Apparatteknik (DK) are all identical, namely that the main purpose of systems is to *transform information* (processing signals), as opposed to the energy and materials transformation of machines, ROTH 1968, RÖHRS 1980, KRAUSE 1986, MEYER 1980, see Figure 8.

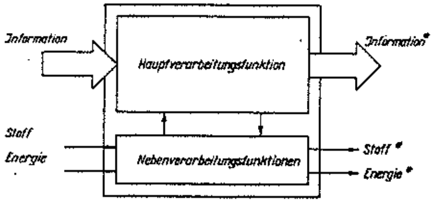


Figure 8 General model of the transformation function of products in the 'Gerätetechnik', RÖHRS 1980

Mechatronic systems also transform information, but there are some exceptions: the main purpose of an assembly robot is to move objects (i.e. transform material), a sewing machine assembles cloth (i.e. transforms material), and an actuator turns electric power into mechanical rotation (i.e. transforms energy). In the case of material or energy transformation however, the effort in controlling the process is predominant in mechatronic systems.

Intelligence The intelligence level of a system is determined by its control functions. Intelligence is a word related to human brain activity, and different authors have tried to measure advanced electronic control function against human thinking. Instead, I will propose a 'technical' definition of intelligence as a basis for this research:

> *Definition*
>
> **Intelligence** of a mechatronic system is the ability to utilize a range of parameters for the control of functions.

A room temperature control system for instance, when designed using only mechanical technology, is likely to adjust the heater (or cooler) according to only one parameter: room temperature. When thinking in terms of mechatronic technology and microcomputer applications, one can easily imagine several levels of more advanced control based on the measurement of different parameters, like out-door temperature, time of the day, air humidity, air velocity, number of persons present.

Note that the definition insists on an assessment of the 'range' of parameters, rather than on counting the simple number of parameters.

If 'parameter' is interpreted to include combinations of parameter values at previous points in time, *adaptability* too can be covered by this definition of intelligence.

It is possibly to realize intelligence in mechanical technology, but an increase in intelligence level is a direct consequence of applying computer technology.

RZEVSKI 1990, 1991 bases his definition of mechatronics on the property of intelligence. He states that machines can be classified as intelligent, if they exhibit some or all of the following behaviours: Programmability, communication, self-regulation, self-diagnosing, self-repair, negotiation, learning and self-organization.

Flexibility Systems are seen as flexible, if they can be changed easily to fit altered requirements or situations. This term too needs a more precise definition:

> *Definition*
>
> **Flexibility** of a mechatronic system is the ease with which a product can be adjusted to a new environment during its lifecycle.

Such alterations or product variants may be desirable both during design (adapting to new markets, standards etc.), during manufacturing (modifying production method, raw materials etc.), and when in use (adapting to new applications or situations).

Flexibility can be increased drastically with the introduction of software control, since computer programmes have practically no manufacturing phase, when compared to mechanics and electronics hardware.

Flexibility and intelligence are predominant features of mechatronic systems, but they are both properties of relative values (also a mechanical system has some degree of flexibility and intelligence), so they cannot be applied for a scientific definition.

Size and weight In Feinwerktechnik (BRD) as in Apparatteknik (DK) there is a general feeling that products are usually small, when compared with energy and material handling machines. ROTH 1968 examined whether 'size' is a valid criteria for distinguishing Feinwerktechnik from mechanical engineering as in Figure 9, but he concluded that this was not the case.

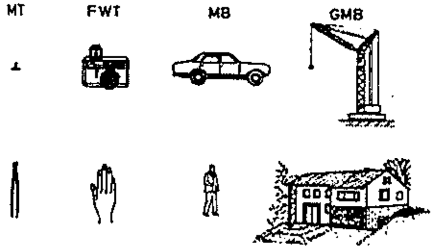


Figure 9 Comparing product dimensions of Mikrotechnik (MT), Feinwerktechnik (FWT), Maschinenbautechnik (MB), and Großmaschinenbautechnik (GMB), ROTH 1968.

For mechatronics it is true that microelectronics technology offers the possibility of shrinking size and weight due to the miniaturization of components, but product size seldom depends on electronic circuits alone. In the mechatronics programme of the Finish Federation of Metal and Engineering Industries, there is a project of developing a 2 meter long hydraulic cylinder as an electronically controlled actuator with position feedback. Is that too big to be mechatronics? Likewise, the Technical University of Lancaster include the design of vehicles and excavators in their mechatronics course.

Small size and light weight of systems are not satisfactory criteria for defining mechatronics technology.

Precision A separate field of mechanical engineering has been devoted to studying the design and production of precise mechanisms: Precision Mechanics or Fine Mechanics. Precision has been used as a criteria for defining Feinwerktechnik (BRD), ROTH 1968 and ZWICK 1971, and was suggested as a distinct feature of mechatronics by for instance KAJITANI 1989.

Mechatronic products may be precise in two respects. Firstly, their performance may be precise, secondly, their internal parts may require precise manufacturing processes. Precise performance is indeed a common feature of mechatronic products, though not necessarily a direct consequence of utilizing mechatronic technology. Precise manufacturing processes are required for some product types (e.g. sensors and actuators), but one of the advantages of bringing in electronics is that high precision requirements for the mechanical system can be compensated using electronic control.

Precision is not a sufficient criterion for defining mechatronics.

High reliability, low energy consumption and *mass-produced parts* have also been suggested as typical characteristics of mechatronic systems, but since none of them are direct consequences of applying mechatronics, they can hardly be used as defining criteria.

A new definition

The only tenable criterion for defining mechatronics seems to be the *combination* of technologies, which I have elaborated in the following proposal:

Definition

Mechatronics is a technology which combines mechanics with electronics and information technology to form both functional interaction and spatial integration in components, modules, products and systems.

This definition is based on the following observations:

1 Mechatronics is regarded as a group of function realizing principles based on similar physical phenomena.

2 Mechatronics combines principles from the existing technologies of mechanics, electronics and informatics.

3 There are two types of combination which must occur in mechatronics:

functional interaction (i.e. the functions of the device are split between the three technologies) and

spatial integration (i.e. the subsystems of each technology are physically realized in one unit).

4 Mechatronics technology is applicable to both component, module, product and system level of a design, and recursivity exists.

This definition rules out electronic products in simple mechanical housing (there is no functional interaction) and control systems using a separate computer (there is no spatial integration).

One must acknowledge that mechatronics can be recognized by certain typical properties like intelligence and flexibility. Therefore the definition should be supplemented with the following general characteristics of systems realized in mechatronics technology:

Properties typical of mechatronic products

The transformation of information (signal processing) is predominant in a mechatronic system, either as the primary function or as the control function of material or energy transformations.

The intelligence level and flexibility of mechatronic systems are likely to be improved, when compared with the performance of pure mechanical or electromechanical products.

Recursivity in mechatronics?

The aspect of *recursivity* needs a closer study. What is the lowest level of subsystem which may still be termed mechatronics? Should a large system be called mechatronic, even if only one of its subsystems is covered by the definition?

First, a hierarchy of subsystems must be defined. I have chosen to classify mechatronic systems by *degree of complexity* in agreement with HUBKA 1973:

0 A *part* is an artefact produced without assembly operations. Mechanical parts belong to this class.

1 A *component* is a simple subsystem assembled from parts and other components. Machine elements and electronic components belong to this class, and so do sensors, actuators, controls and so on.

2 A *module* or *sub-assembly* is a subsystem assembled of components and parts, which fulfills an independent function in a product. E.g. separately testable mechanisms, electronic boards, optical unit.

3 A *product* is an assembly of modules, components and parts, which performs a closed function. Robots and photocopiers belong to the class of products.

4 A *(product) system* is an assembly of products, which interact to perform a number of functions. FMS systems and Office Automation systems are examples.

It is not possible to distinguish clearly between all levels. In particular, the multiplicity of interpretations of the terms 'system' and 'product' in common use makes it difficult to formulate any scientific definition. In this research, the term *system* will be applied both (1) to designate technical artifacts in general, (2) for a class of artifacts of high complexity, and (3) in a systems theoretical sense.

In the hierarchy described above, a *part* cannot be a mechatronic system, since parts are produced without assembly operations. Electronic components are not considered parts, since they are generally assembled from several mechanical parts. Also a 'naked' IC chip will be included in the class of components.

Does the term mechatronics apply to the *component* level? Yes, devices like intelligent sensors and actuators with built-in control electronics must be considered mechatronic components.

A *module*, which includes mechatronic components is also to be understood as mechatronics and likewise on higher levels.

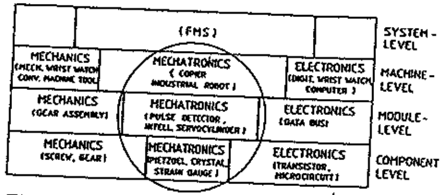


Figure 10 Definition of levels of subsystems in mechatronics, SALMINEN ET AL 1990-C

To conclude: *recursivity* applies to this definition of mechatronics in the direction o increasing complexity, but not necessarily i the opposite direction.

The levels of subsystems defined here ar in close agreement with those of SALMINE ET AL 1990-C, as shown in Figure 10.

The discipline of mechatronics engineering

Mechatronics engineering is the engineering discipline of synthesising mechatronic components, modules, products and systems. This discipline mainly comprises the following skills:

Characteristics of mechatronics engineering

Interdisciplinarity - a general knowledge of product and manufacturing principles from a wide range of technologies, including mechanics, electronics and informatics.

Systems thinking - a systematic attitude towards combining principles of unlike technologies to form an optimal concept.

Creativity - the courage to suggest and experiment with previously unknown combinations of technologies.

Business viewpoint - a business oriented attitude towards evaluating design concepts in a competitive context.

3 PRECONDITIONS FOR PRODUCT DEVELOPMENT

After the discussion of mechatronics technology in the previous chapter let us turn our attention to the activity of developing or designing mechatronic systems. How does this activity differ from the design of 'pure' mechanical or electronic systems?

In the following I will argue that competitive mechatronic product development needs particular company strategies, that it needs measures to ensure interdisciplinarity during design, and that it needs special design methodology.

3.1 Competitive development strategies

Because of the characteristics of mechatronic systems, the markets for such products respond in particular ways and demand particular strategies for product development from the competitive manufacturer. Not that such strategies should be the same for all types of mechatronic products - naturally video recorders and assembly robots are developed and marketed under very different conditions - but still one will see similar patterns caused by the nature of mechatronics.

It is profitable to study the success of Japanese corporations since they exhibit a remarkable understanding of mechatronic markets and competitive strategies in product development. For this reason my experiences from Japan will form the basis for the statements of this section.

Industry characteristics

A common feature of mechatronics companies is that they are technology oriented, salary intensive, and that they have a high level of product application knowledge, VALBJØRN ET AL 1989.

They are technology oriented in the sense that management mostly have a technical background and that development, marketing, manufacturing and service are dominated by engineers. The companies have a tradition of creating technical devices and selling them, rather than hunting up peoples' needs and solving problems.

Mechatronics industry is rather independent of raw materials, since the applied components are freely available in international trade, and the demand for special raw materials is small. It is salary intensive and the level of raw material refinement is very high. NÜCHEL 1984:

> "Und erstmals in der Geschichte der Technik werden die Fähigkeiten eines MC-gesteuerten Produkts weniger durch den Preis und die Eigenschaften einzelner Komponenten bestimmt, als vorrangig durch die Vorstellungskraft und Kreativität des Entwicklungsteams geprägt."

> (For the first time in the history of technology the abilities of a microprocessor controlled product are determined less by the price and properties of single components than by the imaginative power and creativity of the design team.)

The amount of application knowledge that goes into designing mechatronic products is very large, especially for the professional type products. Since microprocessors take over 'intelligent' functions previously performed by the product user, the development of control software demands a thorough insight into the user's situation and needs.

Conditions for product planning

In order to understand the strategies of successful mechatronics companies, we must examine the circumstances which influence industrial product development most.

The technologies are changing very fast and this encourages a constant stream of innovations in new products. It also seems to demand a shorter and shorter market life-time for products. There is pressure especially from Japanese competitors to introduce new models in a cycle much faster than previously.

In general, markets for mechatronic products are moving very fast, following the technological developments in electronics. Especially in the consumer area, the very existence of the product itself may create the need, STONE 1984:

> "Mechatronics markets are often supply-driven, with very strong force-feeding of or discounting into distribution channels to achieve widespread product awareness, coupled with strong advertizing. In other words, the forces operating on the buyer change substantially as the market starts to move."

It seems that intelligent functioning and automatization is in great demand among customers. Product users are no longer satisfied with simple functions, which substitute physical labour only. Many mechatronics areas have turned into buyers' markets: with ample supply of competing models, which fulfill roughly the same needs, customers start to evaluate secondary qualities of the products, and competition gets harder.

A consequence of being independent of raw materials and expensive processing is that starting new business in mechatronic products needs rather small investments. This makes for severe competition with low-priced copy products.

One other condition to which should be paid attention is that the internal functions of a mechatronic product are mostly invisible to humans (e.g. electronic control functions). This causes the buyer to regard the product as potentially unreliable, STONE 1984. It means that customers purchase chiefly on trust in the manufacturer or agent.

Typical strategy patterns

The mechatronics business is a field of fast and constant changes: changes in technologies, in markets and buyers' attitudes, in competition. The mechatronics company will only survive if it is constantly on the watch for new developments, and if its organization is geared for frequent upheavals in both marketing, research, and manufacturing. It seems that in the strategies of successful companies one will find the following patterns:

Far-sighted technology management to determine the future key technologies for the company and methods to acquire them (by licensing, research collaboration, in-house research etc.). To maintain a sufficient stock of both product and production technologies throughout the field of mechatronics. To decide on measures for protecting technological innovations (patenting etc.).

As an example, many machine tool manufactures in the first encounter with mechatronics preferred to sub-contract the design of control electronics and software to specialists. They have since realised that electronics and software development must be an in-house activity, because control electronics carries a major part of the added product value, and because much of the manufacturer's application knowledge has to be implemented in computer software, ISERMANN 1985.

Careful product planning to recognize opportunities in new markets, new customers' demands etc. To track competitors' actions and marketing results. To formulate new product concepts in accordance with an overall company strategy.

FRÖLICH 1982 claims that the area of product planning needs development of systematic methodology to increase the accuracy in product definition and market segmentation. HEIN 1989:

> "While mechatronics certainly represents a potential for creating new products and businesses, it also represents a threat in a competitive environment: The ones to make the most intelligent use of the new possibilities will persist over the others."

Market segmentation in markets with hard competition to exploit the flexibility of electronics for creating product variants for different customer groups. To exploit the opportunities in flexible manufacturing for producing smaller lots of product variants at the low cost of mass production.

Japanese companies in the mechatronics consumer industry put much effort into perceiving new *life-style* trends among consumers not only in Japan but also in USA and in Europe. The purpose is to design product variants that exactly fit the expectations and the style of customers in narrow market segments. Sharp for instance has established a 'Life Soft Center' with 50 employees in Japan and an additional 10 in New Jersey and 10 in Hamburg to provide information about changes in life-style.

Fast product development to increase manoeuvrability and speed of reacting in the fast moving competitive environment.

Of course development projects cannot be shortened ad libitum. The development time is linked to the innovative change between two subsequent models. The Japanese strategy is to shorten the product cycle and introduce new models faster than Western makers, at the cost of smaller technical improvements, a pattern referred to as 'incremental design', ABEGGLEN & STALK 1987. In this way they tend to both outrun Western competitors and spur demand among customers. They encourage a profitable buy-and-throw-away mentality.

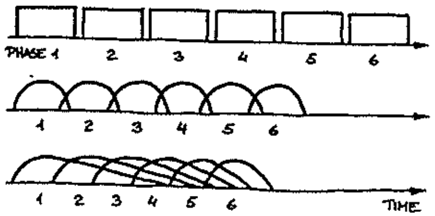


Figure 11 Sequential versus overlapping phases in the product development cycle, TAKEUCHI & NONAKA 1988.

TAKEUCHI & NONAKA 1988 describe how progressive companies change to a holistic strategy in product development. By increasing the autonomy of the project team, the development time can be reduced drastically, as compared to a traditional phase plan project with checkpoints at phase transitions, Figure 11.

Emphasis on secondary product properties like man/machine interface, industrial design, quality etc. to increase competitivety in markets with ample supple of competing products.

We are in the 'Human-Ware Age' where products need not only function, but must be agreeable to the user, EVANS 1985. As a consequence of the missing visibility of mechatronics functions, the communication between product and user becomes a key issue. Industrial design has less restrictions on the shape of mechatronic products when compared with machines, but there is an increased responsibility to create a functional symbolism in the outer form.

Emphasis on brandname to foster trust among customers in the products of the company. This may be by advertising the company name and slogan rather than individual products, and by putting emphasis on corporate identity design.

Mechatronic product development demands strategies which are different from the ones applied for machines, electronic devices and software packages, because the nature of mechatronics is different, and because the conditions for design differ.

The patterns discussed in this section naturally do not constitute a complete strategy for the development of competitive mechatronic products, but they seem to be those elements, which are independent of product type (e.g. consumer or industry products) and markets. They will all be recognized to some extent in the strategies of successful manufacturers.

3.2 Interdisciplinarity

It is a basic principle in problem solving techniques that design alternatives in quantity increase the probability of finding an optimal solution to the problem. Likewise in mechatronics: To form a good design concept, the designer must be able to suggest many alternative combinations of technological principles. Therefore a wide knowledge of different technology areas must be available to each designer or to all of the design team.

There are in principle two ways of achieving such an interdisciplinary basis for product development. One is to educate all designers with a generalists' knowledge of many technologies, the other is to put together technology specialists and encourage communication and cooperation within the team, HERZOG 1974.

Industrial product development is mostly a combination of the two, for instance a design team of specialists headed by a project leader with a generalists' education. SALMINEN ET AL 1990:

> "Because it is impossible to store the entire mass of knowledge needed in a product design and development project in the brains of all project participants, we must in the future train a new kind of chief designers to control the multi-dimensional mechatronic design process. The chief designers keep an eye on the whole: they draft the products and act as interpreters between the various field experts. We must bear in mind that experts on narrow fields will always be needed and they will more and more important as their capacity for cooperation increases. The chief designers must be familiar with the properties and uses of the methods in order to be able to manage a multitechnical expert group efficiently."

Educating designers for mechatronics

Here, I will treat education in a wide sense, meaning not only university courses but also the training and experience gained within the company.

In universities the traditional distinction between departments of mechanical and electrical engineering has created organizational obstacles to interdisciplinary education. More than often an attempt to create a mechatronics course results in a quarrel on teaching quotas between subject areas or individual professors, since interdisciplinarity threatens the principle of specialization, which is fundamental to all engineering educations today.

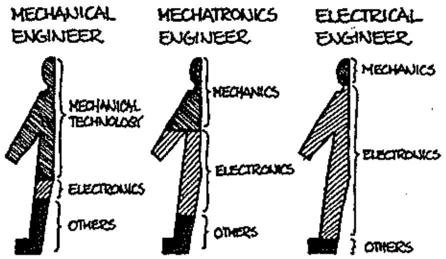


Figure 12 The technology competence of the mechanical, electrical, and mechatronics engineer respectively, MEYER ET AL 1980

It is really a conflict between depth (specialization) and width (general knowledge) in engineering education. If one wants to increase the electronics competence of a mechanical engineer within the same four years course, something else has to go out. WASSELL 1980 categorises the technological knowledge required of design engineers into four depths:

"Depth A	The awareness depth
Depth B	The buy and use depth
Depth C	The theory depth
Depth D	The design/make depth"

He states that most subject courses aim at covering the theory depth (C), even though effective designers need a balanced knowledge at all depths ranging from a wide awareness of many types of product technologies to complete knowledge in a specific product area.

> "The depth and mix of the products technological knowledge required by the designer is dictated by the kind of products to be designed and not by any vaguely generalised traditions."

Further, he recommends that at least one area of technology should be studied in full D depth, so that the students learn to appreciate the full technical professionalism of the designer's role.

The Danish Mechatronics Association has suggested a detailed course curriculum for the establishment of mechatronics education, MEYER ET AL 1980. It is based on a similar 4-level scale of knowledge depth. This report states that besides the depth/width dilemma, also the teaching attitude is a major problem in creating mechatronics courses in universities. Most subjects are taught analytically, leaving very little room for synthesis or design activities.

A third obstacle in creating interdisciplinary education is the timing problem, WASSELL 1980. New technologies arise continually, and the efficient designer needs a wide awareness of potential principles. In university environments there is the risk that education is either too slow in reacting to trends in industry, or puts too much emphasis on esoteric front-end research areas, which may never find application in industry.

4 year mechatronics courses have been established at a number of universities in Europe, either as new initiatives or as developments of previous courses on electromechanics, Feinwerktechnik or precision mechanics, FASS 1981, MEYER 1988.

In such courses, it is common to find electronics taught by the mechanical engineering department and vice versa, simply because the barriers between departments are too large and the acceptance of the need for interdisciplinary education too little for cooperation. Thus at a Danish engineering college for instance, there are two mechatronics courses with very little collaboration: One in the mechanical and one in the electrical engineering department.

Surprisingly, the number of engineering universities in Japan with a full mechatronics course is very small (KATO & TSUTSUMI 1986). It seems the Japanese try instead to increase the contents of electronics/microprocessor education in the curriculum of *all* mechanical engineering students and leave the rest to in-house training within the mechatronics companies.

The number of universities with laboratories researching in areas of mechatronics technology (robotics, FMS, computer control, micro-mechanics etc.) is naturally much larger, both in Europe and in Japan. Especially laboratories with hands-on micro processor control exercises are becoming increasingly popular, YAMAZAKI 1985, 1989; MILNE & FRASER 1989. The students graduating from such laboratories will also achieve a kind of interdisciplinary foundation.

In companies, design engineers will achieve a general knowledge of a wide range of technologies after working on several dissimilar projects or by following post-graduate courses on technologies (company internal or external).

There is a noteworthy difference in attitude between European and Japanese companies in this respect. In Europe design engineers will mostly be employed in a specialist position (even when hired directly from university), and they will seldom move to another specialist area within the company, unless they are promoted.

Figure 13 The Japanese priority is for generalists: designers with not just one specialization

In Japan, because of the life-time employment practice, all engineers are employed right after university graduation and in no particular position . The company will automatically assume all freshmen in need of additional training, before they can be posted in their first designer's position.

Through job-rotation between different departments the company will ensure that its designers acquire a generalist's competence within a few years, Figure 13.

In this way the Japanese company achieves a flexibility in its work force to respond to changing market situations. In Europe a similar flexibility can only be exercised by laying off abundant specialists and hiring new employees with the required competence.

Communication and cooperation

The second way of providing the interdisciplinary basis for mechatronics development, is to encourage cooperation between specialists from each of the necessary technology fields (mechanics, electronics, software etc.).

Besides being an attitude problem ("do I *need* to cooperate?"), there are first of all the aspects of personal cooperative skills, language of communication, physical project environments and documentation techniques to be considered.

The designer's attitude towards cooperation depends very much on his understanding of the product development activity, and the role he plays in it: The rules of disposition, success parameters, business aspects etc. In Denmark, industry courses in 'Integrated Product Development' have had great success in changing the attitude of designers, ANDREASEN & HEIN 1987.

Another factor influencing the attitude of the design engineers is the organization of the design department. Many companies in Europe have established a matrix organization in order to provide the best possible technology back-up for all development projects. This means however that the specialists on a project are responsible not only to the project leader, but also to their department managers, who may not agree on the priority of the project.

The cooperative attitude also depends on personal relations. To establish cooperation with people whom you already know, is much easier than if both the personal and working relationship have to be established at the same time. Here is one of the advantages of the job-rotation system of Japanese companies: not only do the Japanese engineers obtain knowledge about a number of product areas and technologies, they also build a vast network of personal contacts throughout the company.

Personal skills in communication and cooperation are not traditionally seen as the responsibility of the engineering universities, though occasionally departments of engineering design include such training in their exercises: cooperation in teams, oral and visual presentation of ideas, cooperation with industry, report writing etc.

At the Technical University of Darmstadt for instance, students at the Electromechanical Engineering Department have to complete four one-semester design projects in sequence during their 3rd and 4th year. Each project is accomplished in teams of five members, and the teams are rearranged for every new project, to train the students in cooperating with different human types, BUSCHMANN & WEISSMANTEL 1984.

A design course in Denmark assembles students from engineering, industrial design and business administration universities to complete a one-semester design project in mixed teams, GJERLØV-KNUDSEN ET AL 1989.

At Chiba University in Japan, a class of 20 industrial design students are requested to cooperate on completing the design specifications and concept design for a complex system. Then they branch out in smaller groups to design a subsystem each.

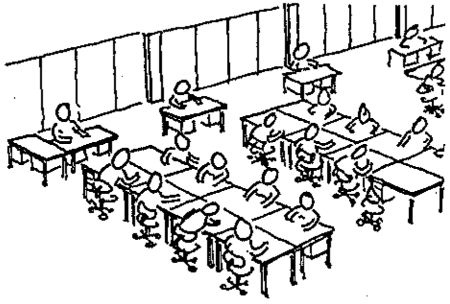


Figure 14 The open office style in Japan encourages communication

Common language is a necessary precondition for specialists of unlike technological fields to cooperate. Ways of expressing functions and principles, for instance, are very different in mechanical, electronics and software engineering.

There is an urge to develop such a common language in mechatronics design, HEINZL 1984 and ISHII 1989. We need a common language on three different levels, claim SALMINEN & VERHO 1989:

- "language for determining the functions that the customer needs, 'symbolic language for required functions'
- language for the task setting and conceptual design phases, 'conceptual phase language'
- language for embodiment- and detailed design phases, 'expert phase language'."

Since the topic of a common language for designers is closely related to design modelling in mechatronics, this aspect will be treated in detail in a later chapter.

Project environments The physical surroundings of the development team (the lay-out of the design office, available facilities etc.) influence the communication between team members.

The Japanese open office style, in which designers work in the same room or even around the same table, encourages informal communication, since all the groups will overhear telephone calls, overlook each others drawings etc. see Figure 14. A System of individual offices, as found in many companies in Europe, may increase concentration and creativity, but it prevents efficient communication. The individual office requires both a conscious *decision* to communicate and *action* from the engineer (pick up the telephone, walk to the next room). Another dilemma is whether design engineers should be placed close to colleagues in the same technological field, to ensure specialist backing, or whether they should be placed with the development team to emphasise cooperation in the project. This is in particular a problem with electronics engineers, for whom the knowledge of new electronic components is crucial. In the specialist group, there is an informal sharing of knowledge.

Documentation The handling of written information on the project results (ideas, specifications etc.) is important for the communication between team members.

Some companies utilise common document files for all engineers on one project. Instead of individual bookshelves, team - members share the same files for their calculations, notes and drawings. The result is that activities are documented more carefully and communication within the group gets easier, since all documents become openly accessible. It is however a system which is difficult to keep up in the West, because engineers here tend to regard information as private property or as an object of trade.

In conclusion, the interdisciplinary basis necessary for mechatronics product development is not just a question of creating mechatronics courses at university. There are a range of possibilities for organizing the project team and design department to achieve a suitable width and depth in competence and cooperative attitude.

3.3 Special design methodology

When interviewing designers in industry, there are indications that methodology for mechatronics design is missing:

- Designers find it difficult to describe and discuss the way of working of a total mechatronic system.
- To choose the right design concept in mechatronics is regarded as very important, but the decision is often made early in the design process with very few designers involved.
- There are difficulties in dividing the design activities in a mechanical, electronics, and software part, and in managing the interfaces between the three areas.
- The function of the total concept will mostly not be verified until a very late stage of the project ("software design has to be finished first"). At that time alternative concepts are not available for comparison, and only details may be altered.

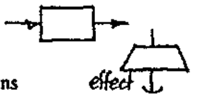
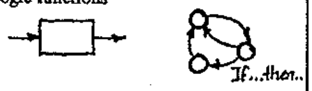
	MECHANICS	ELECTRONICS	SOFTWARE
FUNCTIONS	Transformations of material energy information Purpose functions  	Transformation of information attatched to energy (electric signals) 	Transformation of information Logic functions 
CONCEPT DESIGN	**Function principles, organic structure** 1. Problems appear to be 'new' 2. Many alternative solutions to be found	**Module structure, circuit design** 1. Standard circuit solutions available 2. Choice of standard components	**Program structure, algorithms** 1. A set of basic operations exists 2. Recycling of algorithms uncommon
CONCEPT REALIZATION	**Embodiment design and element form** 1. Many production technologies available 2. All elements must be specified (form, dimensions,material, surface)	**Electronics packaging design** 1. Limited number of circuit technologies: PCB, thick-film, IC etc. 2. Mostly subsupplier components	**Program coding, assembly of modules** 1. There is no true production phase, once program is coded
DESIGN MODELLING	1. Specially manufactured elements required for hardware models 2. Models are easily understandable for non-specialists (sketches, mock-ups)	1. Standardized diagram symbols 2. Easy access to functional modelling with off-the-shelf components 3. Reliable computer simulations	1. Until coding phase only graphical models (diagrams) 2. Easy access to functional modelling (trial-runs on standard computer) 3. Function is difficult to explain to non-specialists
DESIGN METHODS	1. Design methods are available 2. Methods are not widely accepted in industry	1. Few methods for 'new' circuitdesign 2. Few methods for packaging design	1. A number of methods are available 2. Increasing interest from industry
COMPUTER AIDS	1. 2D and 3D systems only applicable in the final design phases 2. Parametric design available for well-described design procedures	1. Computerized circuit synthesis for logical designs 2. PCB and IC lay-out and simulation 3. Silicon compilers for IC lay-out 4. Prototype simulation	1. Only matemathical tools for early design phases (for specification) 2. High-level programming languages and compilers 3. Graphical documentation systems

Figure 15 Comparison of some methodical characteristics of mechanical, electrical, and software design

The purpose of this section is to examine, how mechatronics design differs from the design of mechanics, electronics and software in a methodical sense. Also I will identify the type of design methodology needed for mechatronics.

Design methodology on three levels

The design activity is too complex to be described in a one dimensional process, starting with a need situation and resulting in a complete technical system. This research is based on the framework for design methodology as taught at the Institute for Engineering Design of the Technical University of Denmark. It states that design must be regarded from three viewpoints or on three *levels*, ANDREASEN & HEIN 1987:

1 The designer's way of solving problems in general (specifying, creating ideas, evaluating).

2 The synthesis of technical systems (functions, technologies, working principles, form of components).

3 The total activity of product development as performed by the company (relations between design, manufacturing and marketing).

The activities on each level may be divided into phases, and design methods can be attached to each phase.

Problem solving is the elementary activity carried our repeatedly in every design project. Methods on this level may be applied to any type of 'open' problems, i.e. problems with a multitude of solutions. They include methods for specifying problems, for generating ideas (e.g. brainstorming, synectics) and for evaluating and presenting solutions. These methods are based on the human way of thinking and are designed to enhance the designer's capabilities, to prevent mental blocks arising from traditional thinking etc.

Synthesis of technical systems covers the engineering development of a machine or technical system from abstract functions (e.g. black-box models) to concrete and detailed form (parts drawings). Methods on this level are based on the characteristics of technical systems and therefore specific for classes of systems. For machine design, methods, which handle transformations (material, energy, information), physical effects, function carriers, form variation etc. belong to this level.

Product development includes all the activities necessary for the company to establish business based on a new (or revised) product. As describes in the model of Integrated Product Development in Figure 7, this involves the cooperation of marketing, development and production. Methods on this level are based on the structure of companies and on the complex nature of product development. They include scenario-techniques, competition analysis, design reviews, planning etc.

The three level approach has been adopted here, because it shows a consistency and completeness in describing the design process not found in any other 'school' of design research. It also allows an identification of the level, which is of particular interest for mechatronics.

On the first and third level, design methods are rather technology independent, and therefore mechatronics does not show any differences compared with 'pure' machine design. However, difficulties must be expected, if we try to apply machine design theory to mechatronics design on the level of synthesis. This is the level, I will examine in the following.

Machine design, electronics and software

There are significant differences between designing mechanics, electronics and software. Not only are different technical skills required, but the very nature of design problems differ in terms of functions to be realized, types of solutions available, and realization of the intended functions. Because of this, also the options for design modelling and the existence of design methodology and computer aids differ greatly. Figure 15 sums up the characteristics of the three fields. For details, please refer to Appendix 1. Major differences are found on the following two points:

- The term *function* is understood very differently, ranging from physical effects and transformations of material/energy/information to logical relations between data operations.

In machine design theories, it is common to describe the main purpose of a machine in terms of transformations of material, energy and information. However, designers do not only think of transformations, when using the word 'functions'. They also think of 'effects needed in a machine' or purpose functions, HUBKA 1973 and ANDREASEN 1980, and they mostly ignore logical functions. In electronics, designers handle signal transformations that is, transformations of information with the energy aspects (e.g. heat dissipation) mostly ignored. Software is manipulation of data (i.e. transformations of information only), but with the logical relations between transformations (order of execution, conditions) explicitly described (e.g. in Petri-nets).

- The conditions of *design modelling* are different, both in the ease of making functional models and in the tools available for the designer to illustrate his ideas for colleagues. In particular, abstract software principles are found very difficult to 'visualize'.

Design models describing function are on very different levels of abstraction. Mechanical models are rather concrete (3D sketches, principle diagrams), where models of electronic circuits and computer programs are more abstract, and thus more problematic as a basis for discussions in a mechatronic design group. An electronic circuit diagram can hardly depict functional states of the device (e.g. switches open or closed), while a program flow-chart mainly describes functional states and conditions for progressing from one state to another.

The first point means that it is difficult to come to a total, abstract understanding of a mechatronic system, i.e. to describe a functional structure. The second point means that suggesting new design concept ideas in mechatronics is difficult, because the designer cannot describe a total concept in just one language (e.g. symbols, drawing, diagram), he needs a different language for each of the mechanical, electronic and software parts.

I will discuss the differences in design methodology for mechanics, electronics and software in detail in Chapter 5.

The need for new methodology

Can we describe more precisely the kind of methodology necessary for mechatronics design? At this point it may be concluded that:

1 Only design methodology belonging to the level of synthesis of technical systems is specific for mechatronics.

Activities relating to problem solving and product development - though important also in mechatronics - are not performed differently from other fields of design.

2 Design methodology which handles total aspects of the mechanics/electronics/software combination is missing.

Methods and models for each of the individual engineering disciplines exist (to varying degrees), but cannot easily be combined.

3 In particular the early stages of mechatronics are in need of methodology: for functional description, technology assignment, conceptual design etc.

After the concept has been decided, the design process tends to separate into specialized activities of mechanics, electronics, and software, where the need is rather for technology specific methodology.

Why develop special methodology at all, would it not be possible to adapt the theories and methods of existing design disciplines? To some extent it is certainly possible to apply elements from the fields of machine design and software design, where such theories have been developed, but they are not fully sufficient for mechatronics, because:

- Machine design methodology has no means of abstractly describing the logical relations between functions (i.e. when, in which sequence, and under which conditions the functions must be performed), since these relations are in a complex way built into the physical structure of the machine.
- Electronics design methodology is mainly based on analysis of 2-dimensional structures. There are neither tools nor traditions for formulating alternative concept ideas.
- Software design methodology is not capable of bridging the gap between abstract functional descriptions and physical effects and spatial relations, since such effects and relations do not exist in the software domain.

The comparison in the previous paragraph of the design characteristics of mechanics, electronics, and software, indicate that the most important areas for research are a theoretical basis for the functional understanding of mechatronic systems, and design models ('common language') to describe functional structures and design concepts for such systems. These will be the main topics of this thesis.

There are two areas which I will not include in this research, even though in particular the Finnish project (SALMINEN ET AL 1990(2)) has called attention to their importance for mechatronics design:

Design specification methods are in principle independent of technologies and therefore not specific for mechatronics, i.e. it should be possible to specify the desired system without indicating a technical realization. Such methods belong to the levels of problem solving and product development. The same argument holds for design evaluation methods.

Design procedures, i.e. recommended step-by-step plans for activities and results to be achieved in a design project, are closely related to the company structure and cannot be based solely on an understanding of mechatronic systems. They belong primarily to the level of product development.

4 MECHATRONICS DESIGN THEORY IN LITERATURE

The field of mechatronics design is very new, so to talk about the existence of any *design theory* in literature is hardly appropriate. No single author has yet presented a general theory of mechatronic systems or a theory for the design of such systems.

A number of articles offer elements of design methodology: systems models, design procedures etc, but only if studied through a theoretical framework of design, do those elements give some contribution to a total understanding. In the first section of this chapter I will discuss such a framework.

Articles, which directly discuss mechatronics design, have been published mostly by Japanese, Finnish, and British authors. This study will also include literature on Feinwerktechnik (BRD), Gerätetechnik (DDR), and Instrumentation Engineering (UK), primarily where aspects of the *interaction* of mechanics and electronics are treated.

In Chapter 5, I will review literature on design methodology for the traditional fields of mechanics, electronics and software for contributions to mechatronics design theory.

4.1 A theoretical starting point

After defining four important terms related to design methodology, I will describe a theory for the synthesis of technical systems as a basis for the literature review.

Defining design terms

In this research I will distinguish the four terms design *method*, design *model*, design *principle*, and design *procedure*, which all describe tools to aid the designer during the development of technical systems.

> *Definition*
>
> A **design method** is a set of instructions for how to perform activities to proceed one or more steps in a design process.

A design method describes a transformation process in which information about a problem is transformed into information about one or more solutions.

According to ANDREASEN 1980, methods are characterized by their

1 scope of validity (e.g. technology specific or general),

2 level of operation (e.g. the 'size' of the design step: a full design process or an elementary operation), and

3 basis of origin (e.g. theory of human reasoning, theory of machine systems, theory of company organization).

A few examples: 'Brainstorming' is a method of general validity for the elementary design operation of generating ideas for problems of limited complexity. It is based on human association forming mechanisms. 'Establishing functional structure' is a method with validity for mechanical systems for the design operation of deriving an abstract description of the functions of a product. It is based on the theory of machines transforming material, energy and information.

In spite of many attempts it has not been possible to develop design methods or design *algorithms*, which automatically lead to design solutions. Only for problems of limited complexity, or where the solutions are slight variants of well-known concepts, have such design algorithms been implemented successfully in CAD systems. In general, design methods can only recommend ways of thinking and operation sequences for the human designer to follow.

> *Definition*
>
> A **design model** is an artefact, which reproduces a subset of the properties of an object.

Here, *property* means any attribute or characteristic of the object. The *object* is the product or rather the designer's idea of the product to be designed. This understanding of design models may be illustrated as in Figure 16. The model has a set of properties, the *modelled properties*, in common with the product, but it carries also properties, which do not belong to the product, and which are irrelevant for the modelling activity. A circuit diagram for instance has the property of structure in common with the electronic circuit of the product, but it is drawn on paper and does not work.

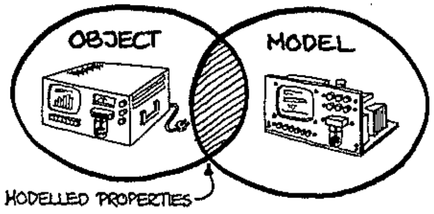


Figure 16 The relation between a design model and the product

The activity of design modelling can be characterized by

1. the object (e.g. the need situation, the total form, the working principles),
2. the modelled properties (e.g. design costs, design appeal, function),
3. the purpose of modelling (e.g. to specify, to evaluate), and
4. the model user (e.g. design colleagues, managers).

The design model itself can be regarded as part of a communication between the designer and a model user (which may be himself), and can therefore be specified in terms of model medium (talking, paper, etc.) and modelling code (language, mathematical symbols, etc.). Please refer to Appendix B for a more detailed account of this understanding of design models.

Design is a propagation through a large number of models, each of them modelling different aspects of the not yet finished product, e.g., mathematical formulae, verbal descriptions, sketches and prototypes.

Design models and design methods are closely related, because the designer works with nothing but product models until the product itself exists. Methods are often described by the kind of models they create or act on, e.g. 'specification methods' prescribe how to establish a design specification.

One can think of design methods as instructions for how to establish a product model or how to propagate from one model to another.

> *Definition*
>
> A **design principle** is a general rule for the design activity, which will frequently favour good solutions.

Design principles are recommendations for the designer to follow, usually in the form 'Best designs are achieved by doing ...'. They are established through experience from development projects and will have a frequent, but not general validity. An example from the machine design field:

- Low assembly cost is achieved by minimizing the number of parts in the design.

This principle is obviously true for as long as a small number of assembly operations is cheaper than a large number. However, if the small number of assemblies each becomes very complex or requires great precision, then the principle will lose its validity.

SCHILLING 1984 defines two related terms: 'Richtlinien', which are recommendations or directions, and 'Prinzipien,' which are basic options for structuring the technical artefact and its parts. The first term involves an evaluation, whereas the second is an objective description of a design possibility or strategy ('minimizing the number of parts' in the above example). I will not distinguish between the two meanings in this research.

Design principles can be looked upon as short-cuts during design. They can be applied for rejecting solution alternatives without too much investigation.

> *Definition*
>
> A **design procedure** is a recommended sequence of activities to be performed during one or more phases of a development project.

The procedure will prescribe results to be presented at the end of each phase (keypoint plan), and may suggest design methods to be applied. It is an organizational type of plan and will mostly depend on the company structure, ANDREASEN & HEIN 1987.

A design procedure does not - in contrast to a design method - describe *how* activities should be performed, it merely lists the activities and results.

The design procedure is based on a *model of the design process*, but it tries to recommend how it should be, rather than to describe how it is. Design procedures are very common in design literature, for instance VDI 2222 1977 and PAHL & BEITZ 1977.

One other frequently used term needs commenting: I will apply *design methodology* to denote a collection of all four elements: methods, models, principles and procedures.

A systems approach to mechatronics

Systems Theory is a very important basis for understanding and generalizing the nature of mechatronics. Only by regarding the mechatronic artefact as a number of abstract systems, each amplifying a different characteristic, is it possible to derive design tools of some general application value.

Systems Theory provides a concept and a set of rules for the abstract modelling of technical artifacts and for the decomposition of such artifacts into subsystems on hierarchical levels.

Any object, which is composed of *elements* and their *relations*, and which can be regarded separately from its surroundings, may be considered a system. It will be characterized by the two systems properties *function* and *structure*. Please refer to Figure 17 for an illustration of the systems concept.

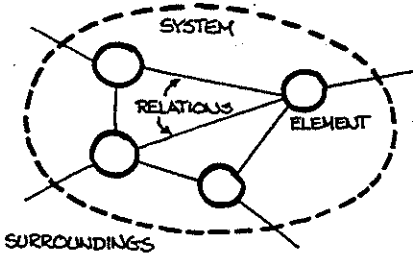


Figure 17 Basic terms in systems theory

Systems Engineering is a field that originated in the USA, and it is closely related to systems theory. It has been mostly concerned with the procedures of designing large, complex systems like aircraft, defence systems, and the NASA space programmes. Systems engineering divides the design process into sizeable phases, and it has developed methods for predicting design costs, for computer modelling, for specifying and evaluating solutions etc, HALL 1962.

The present research, however, is not so much concerned with the design procedure, as with an understanding of mechatronic systems (which may then eventually form the basis of a procedure oriented synthesis methodology).

ANDREASEN 1980 in his efforts to formulate a scientific basis for machine design, has developed a 'Theory of Domains' based on a systems perception of machines.

The Theory of Domains states that the synthesis of machines consists in successively establishing four systems, each corresponding to a (mental) working domain for the designer, Figure 18. These systems represent four different aspects of the machine:

1 The *process system* - a structure of processes, which transform material, energy, and information.

2 The *functional* system - a structure of purpose functions or effects needed in the machine to create the specified transformations.

3 The *organic* system - a structure of organs, each of which realises one or more functions through physical effects.

4 The *parts* system - a structure of single machine parts that make up the embodiment of the machine.

The process of machine synthesis cannot be described as a simple sequence of activities belonging to each domain. The designer is likely to jump back and forth between the four different perceptions of the machine. The main advantage of the Theory of Domains is that it allows a precise positioning of design models and design methods either within a particular domain or on a transition from one domain to another.

The domains should not be confused with the *level of abstraction* or the *number of details* in the designer's momentary activities. In fact each domain should be regarded as a two-dimensional plane spanned by those two parameters, as in Figure 19.

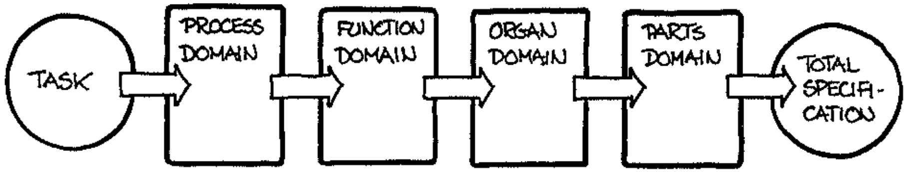


Figure 18 The Theory of Domains: a general strategy for machine design, ANDREASEN 1980

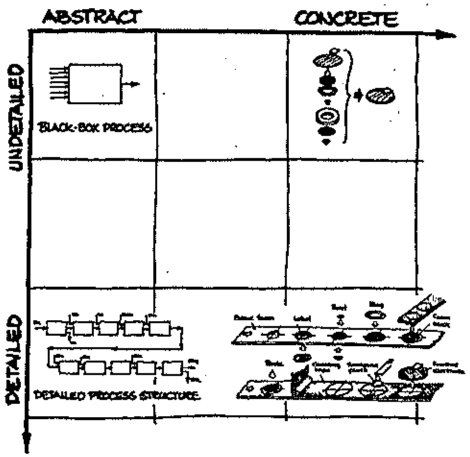


Figure 19 The Process Domain as a two-dimensional plane of design models

BOSMAN 1978 has published a systems approach to the design of instrumentation systems, which shows some similarities to the Theory of Domains. He recommends that the designer prepares five structures ('goal-related aspect systems') during instrument design:

functional structure
organic structure
information structure
activity structure
sociotechnical structure.

Of those, the first two can be directly equated to the process structure and organic structure of ANDREASEN 1980. Bosman's information structure is also a process structure, only it is restricted to transformations of information. The last two structures represent aspects which are not treated in the Theory of Domains.

The activity structure describes control procedures for the 'operation' of organs, i.e. the sequence and timing of effects. With the sociotechnical structure, Bosman expands his systems understanding to include not only the technical system itself, but also its human operator. This structure stipulates the interrelations between the activities of the operator and those of the equipment.

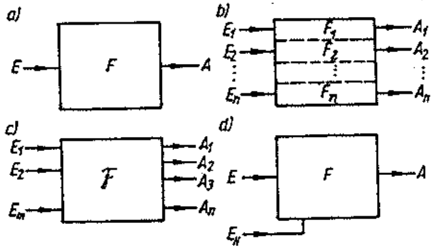


Figure 20 Symbolic representation of transformation functions: (a) single, (b) multiple, (c) integrated, (d) dependent on external energy supply, BÖHME 1978

The Theory of Domains is developed on a *machine* systems understanding and is not fully adequate for mechatronics, but being currently the most consistent theory for the design of technical systems, it forms a good starting point for this research.

In the review of mechatronics design literature I will concentrate on publications which treat the nature of mechatronic systems, assuming that the topic can be dealt with on two levels:

- A functional level: the technology independent systems of transformation and purpose functions.
- An organic level: the systems of technical principles chosen to fulfil the required functions.

At the end of the chapter, I will briefly discuss the few suggestions for mechatronics design procedures found in literature.

4.2 The functional level

Though a definition of the term *function* is rarely formulated in literature on mechatronics design, it is mostly understood as a *transformation of input into output*, for instance BOSMAN 1978:

"*Function* is defined as the characteristic action by which the instrumentation fulfills its purpose. For a measuring system the action is that specified input variables are transformed into desired output variables. The input variables can be matter, energy and information."

The total function of a mechatronic system can be divided into subfunctions according to various criteria, KRAUSE 1986:

"1 nach der Bedeutung für die Erfüllung des Zwecks: Haubt- und Nebenfunktionen

2 nach der Art der Veränderungen von Funktionsgrößen innerhalb des Funktionsflusses in einem Gerät: Grundfunktionen (Wandeln, Leiten, Speichern usw.) oder Elementarfunktionen

3 nach dem physichalischen Charakter der Funktionsgrößen: Teilfunktionen der Stoff-, Energie- und Informationsverarbeitung."

(1 According to their significance in fulfilling the purpose: primary and secondary functions.
2 According to their types of transformation of variables in the functional flow of the system: elementary functions (transform, conduct, store, etc.)
3 According to the physical character of the variables to be transformed: subfunctions for material, energy and information transformation.)

In his method of 'Fewest Necessary Subfunctions' for the design of electro-mechanical systems, TAYEFEH-EMAMVERDI 1981 states that the subfunctions of a functional structure are indispensable and sufficient, if their solution principles make a complete synthesis of the total function of the system possible.

Logical functions are frequently mentioned in literature on mechatronics design. They are necessary to create causal relations of the type *cause and effect* between actions in a system, ROTH 1963, WEBER 1987. The means for realising logical functions has changed significantly with the introduction of microelectronics, as can be clearly seen in the case of the teletypewriter, ROTH 1963 and VORBACH 1982, see Figure 21.

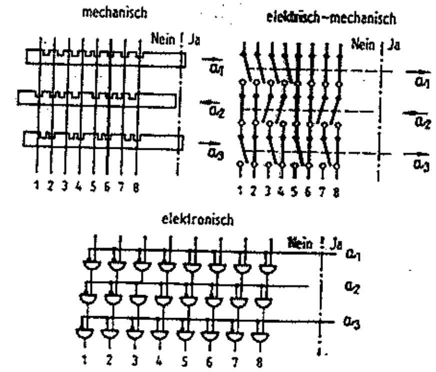


Figure 21 The logical function of decoding signals in teletypewriters applying different technologies, ROTH 1963

An explanation of the relationship between transformation functions and logical functions in mechatronic systems was not found in the literature.

Understanding functional structures

A number of authors have suggested general functional structures, which apply to mechatronic systems. They claim that besides the primary function (main function, 'Verarbeitungsfunktion'), a set of auxiliary functions must exist in any system to support the main function. RÖHRS 1980/KRAUSE 1986 name three such support functions, Figure 22:

1 *Communication function* ('Kommunikationsfunktion') for the exchange of (control) information between the system and the user or other systems,

2 *Protection function* ('Sicherungsfunktion') for protecting the main function against undesirable inputs (disturbances) and for protecting the environment from undesirable outputs,

3 *Control function* ('Interne Steuerung') for controlling the main function and for adjusting the interface between main function and communication function.

From the functional structure of KAJITANI 1989 in Figure 23, two additional support functions may be added to the list:

4 *Power function* to supply the main function with the required energy,

5 *Structure function* to fix the components of the system in a spatial relationship.

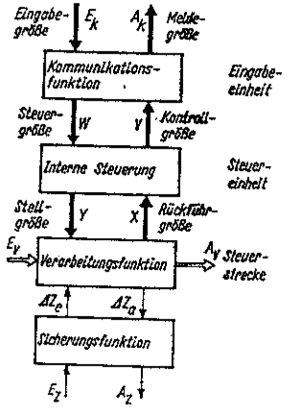


Figure 22 General functional structure and control structure of mechatronic systems according to RÖHRS 1980/KRAUSE 1986.

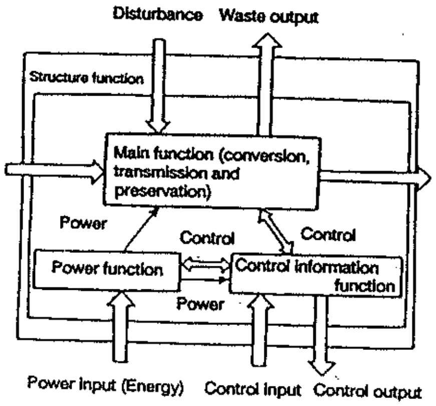


Figure 23 Functional structure of mechatronic systems including a 'structure function' KAJITANI 1989

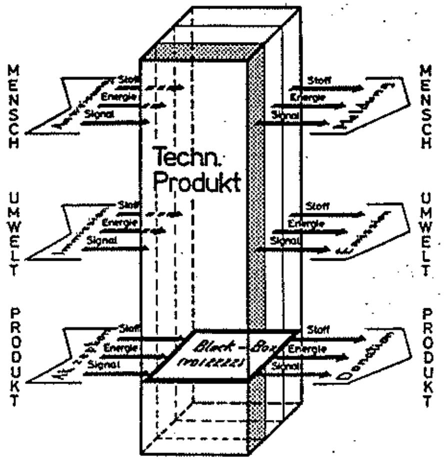


Figure 24 Characterizing the relations between the technical system and its surroundings, GERHARD & LENART 1982

Concerning the relations between the system and its surroundings, most authors agree, that three types of *functional relations* must be considered: input/outputs for the main function, input/outputs for control purposes (communication with man and other systems), and input/outputs which are not functionally relevant (disturbances).

GERHARD & LENART 1982 in their model of systems relations in Figure 24 establish the connection between the functional purpose of input/output flows ('Anweisung, Immision, Akzeption') and the corresponding systems of the surroundings (man, environment, other products).

As to the purpose of their model, GERHARD & LENART 1982:

"Das ganzheitliche Produktmodell-Denken führt während des Konstruktionsprozesses schon in der Konzeptphase quasi systematisch zu einem vollständigen Definieren aller Kenngrößen an den Schnittstellen des zu realisirenden Produktes mit seinen Umfeldsystemen (Mensch, Nachbarprodukt, Umwelt)"

(The thinking in the total product model concept leads already in the concept phase of the design process almost systematically to a total definition of all flows in the interfaces between the product and its surrounding systems (humans, neighbor-products, environments).)

Usually the authors of such functional structure models attempt to explain general characteristics of mechatronic systems as a basis for improved design behaviour.

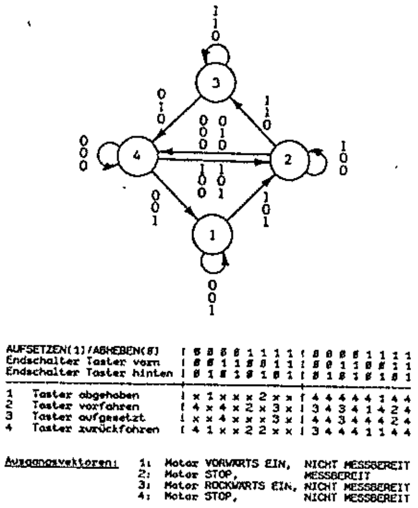
States and transitions

As demonstrated by HEINZL 1984 and CORDES 1984, 1986, the function of a mechatronic system not only depends on the combination of input variables, but also on the previous activities of the system (i.e. the 'history' of the system). The same combination of input may cause different functions at different points in time.

The functional structure of a mechatronic system seems to be variable and change with the momentary *state* of the system, BÖHME 1978, FURCHERT ET AL 1979.

The mechatronic system can be regarded as a finite state machine, and it can be described in terms of *states* and *transitions* from one state to another. CORDES 1986 suggests the application of a state transition diagram ('Zustandsgraph') and a state transition matrix ('Zustands-Ausgangs-Matrix') for modelling the system functions, as illustrated in Figure 25. These are tools borrowed from electronics design.

The state transition diagram and matrix complement each other in that the first is a graphical tool for easily understandable



```
AUFSETZEN(1)/ABHEBEN(0)    | 0 0 0 0 1 1 1 1 | 0 0 0 0 1 1 1 1
Endschalter Taster vorn    | 0 0 1 1 0 0 1 1 | 0 0 1 1 0 0 1 1
Endschalter Taster hinten  | 0 1 0 1 0 1 0 1 | 0 1 0 1 0 1 0 1
---------------------------------------------------------------
1   Taster abgehoben       | x 1 x x x 2 x x | 4 4 4 4 4 1 4 4
2   Taster vorfahren       | 4 x 4 x 2 x 3 x | 3 4 3 4 1 4 2 4
3   Taster aufgesetzt      | x x 4 x x x 3 x | 4 4 3 4 4 4 2 4
4   Taster zurückfahren    | 4 1 x x 2 2 x x | 3 4 4 4 1 1 4 4

Ausgangsvektoren:    1:  Motor VORWARTS EIN,  NICHT MESSBEREIT
                     2:  Motor STOP,          MESSBEREIT
                     3:  Motor ROCKWARTS EIN, NICHT MESSBEREIT
                     4:  Motor STOP,          NICHT MESSBEREIT
```

Figure 25 State transition diagram and matrix describing the 4 states of a surface tester drive system, CORDES 1986 modelling, and the second allows a systematic analysis of the design. A practical example of application to robot gripper design is described by SALMINEN ET AL 1990-A.

It will be recognized that the *logical functions* discussed in the previous paragraph are closely related to the state transition understanding of the mechatronic system. The transition from one state to another will only occur if certain conditions are fulfilled, i.e. the states of the system are related in a causal way.

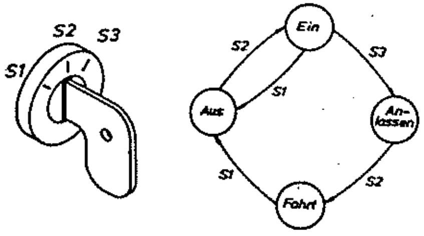


Figure 26 The function of an ignition key for automobiles described with a state transition diagram, HEINZL 1985

From the user's point of view the mechatronic system as a whole will also react as a finite state machine, HEINZL 1984. Depending on the previous entries performed by the operator, the system will be at rest in a state where some control elements may be activated, others may not. A new combination of inputs will make the system change to another state, Figure 26.

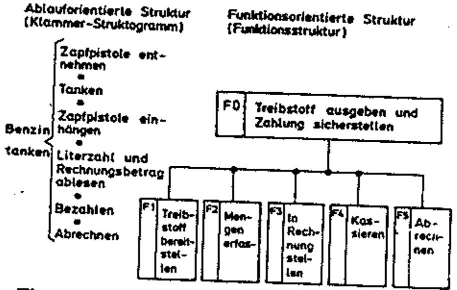


Figure 27 A sequence oriented and function oriented structure describing a self-service petrol filling system, FRÖLICH & SCHLOTTMANN 1982

The conflict between transformations and state transitions

There seems to be a conflict between the two ways of modelling the function of a mechatronic system: The transformations approach, where function is described as a continuous flow of material, energy, and information, and the logical approach, where the system changes mode of functioning from one state to another.

When describing a complete structure of transformation functions, the different states of the system cannot be expressed explicitly, and when modelling the state transition behaviour, the flow of information is not clear, - information is required for a change of state.

This conflict is a primary obstacle to mechatronics design, since mechanical and electronics engineers have a tradition of functionally oriented thinking, whereas software designers are preoccupied with sequences of operation and causal relations, FRÖLICH & SCHLOTTMANN 1982 and HEINZL 1984. Figure 27 illustrates the structure of a self-service petrol filling system seen from both a sequence oriented and function oriented viewpoint. It should be noted that the functional structure of this example does not exhibit subfunctions in a transformation fashion, but rather in the form of purpose functions (effects needed in the system).

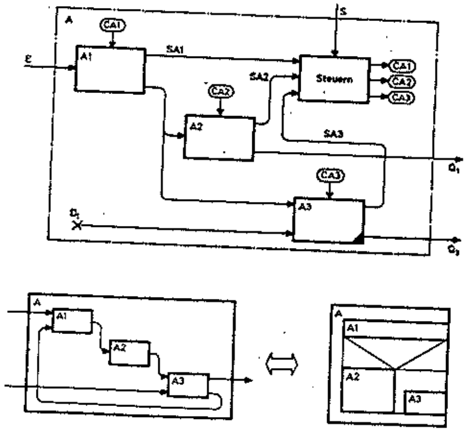


Figure 28 Expanding the functional description by adding (a) a control function to the black-box diagram or (b) a separate structogram, NÜCHEL & PILLER 1986

NÜCHEL & PILLER 1986 state that a transformation structure model (i.e. black-box diagram) does not give any information as to the sequence, the pace, or under which conditions subfunctions must be activated. To give a more complete description of the system function, they recommend that a control function should be added to the transformation diagram, or that the diagram should be complemented by a flow chart-model of the logical structure, Figure 28.

Reports from the Finnish mechatronics research programme also show attempts to handle both transformation and state transition aspects in the same design models, SALMINEN ET AL 1989, 1990-B. I will discuss design models for mechatronics in a later chapter.

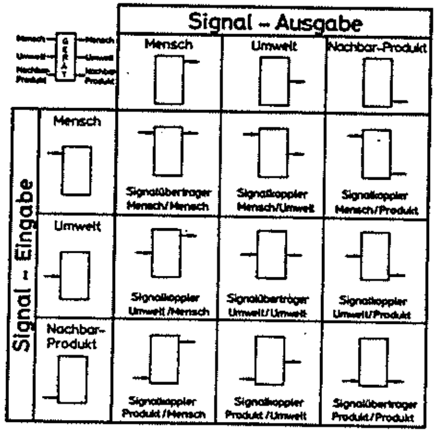


Figure 29 Classifying mechatronic systems according to their main flow of information, GERHARD & LENART 1982

Information in mechatronic systems

It was established in Chapter 2 that the transformation of information is a primary feature of mechatronics. ROTH 1968 and KAJITANI 1989 point out that two categories of equipment must be distinguished according to their handling of information:

1. Equipment with the main purpose of transforming information (e.g. equipment for signal processing).
2. Equipment which mainly applies information for the control of energy or material transforming processes.

Along the same lines, KRAUSE 1986 suggests that two kinds of information must exist: the first kind is *processed* by the system ('Arbeitssignal') and the second kind is *applied* by the system for control purposes ('Steuer- und Kontrollsignale').

The term information is hardly defined in mechatronics litterature except for a loose explanation given by BARNEY 1985:

> "*Information* is the data or details relating to an object or event."

Other authors point out the strong relationship between information and human beings, KAJITANI 1989 and KRAUSE 1986:

> "Information is produced by the mental activities of human beings. The resources of information are human beings."

> "Die Information stellt eine letzlich stets auf den Menschen bezogene erkenntnistheoretische und kommunikationswissenschaftliche Kategorie dar, die für den Sendenden und Empfangenden mit einem bestimmten Bedeutungsinhalt, einer Semantik, verbunden ist."
>
> (Information forms a knowledge theoretical and communications scientific category which always relates to human beings, and which the sender and receiver will associate with a specific meaning, a semantic.)

The concept of information is very important in mechatronics, and it will be taken up again in Chapter 6.

It is possible to systematize different categories of mechatronic products according to the main flow of information, as shown by GERHARD & LENART 1982 (Figure 29) and HEINZL 1984. The information to be processed by the main function of the system derive from the operator, from other technical systems or from the environment, and the information outputs will be directed to the same three recipients. A measuring instrument for instance, obtains information from the environment and presents it in processed form to a human being.

Such a categorization is useful for studying the types of components which take care of the *interface* between the system and its surroundings: sensors, actuators etc.

According to HEINZL 1984, the analysis of the type and amount of information entering and leaving the system through interface components is extremely important in mechatronics design. Especially the flow of information between the system and its user is increasing in quantity with increasing intelligence of the system, - one only has to think of the complicated operation procedure of a videotape recorder, ISHII 1989.

4.3 The organic level

On the organic level, the mechatronic system is described by the technologies and functional principles which realize the functions of the system. Ideally, the organic structure is a result of *allocating* technologies to the functions of the system.

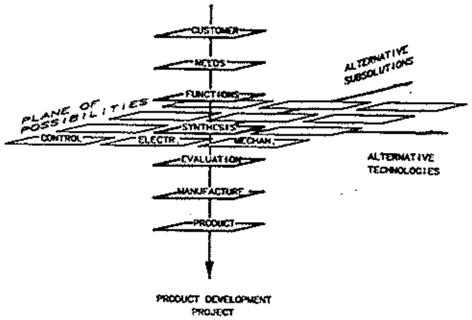


Figure 30 Technology allocation in the synthesis of mechatronics: a plane of possibilities, SALMINEN & VERHO 1989

A question of technology allocation

The technology allocation is a choice between a large number of alternatives, since many functions can be realized using either mechanical, electronical or software technologies, and within each field a range of alternative principles are available. ISHII 1983:

> "Therefore, the strategy for the development of mechatronics technology will have a very wide range of alternatives. Naturally, not all possible combinations are practical, which will make the dependence of decision-making on strategy even more important in the future."

FRÖLICH & SCHLOTTMANN 1982 express the dilemma of technology allocation in terms of *displacing interface lines*:

> "Erst die Verschiebung der Schnittstellen zwischen Mechanik, Elektronik und Software ermöglicht eine Optimierung des Gesamtsystems gemäß technischer und wirtschaftlicher Anforderungen."
>
> (Only the displacement of interfaces between mechanics, electronics and software makes it possible to optimize the total system according to technical and business oriented specifications.)

The two expressions technology allocation and interface displacement reflect idealised, but slightly different attitudes towards design: The first is the perception of 'revolutionary design,' where the product functions can be formulated theoretically, independently of technological realizations. The second is the perception of 're-design,' in which a technical solution already exists and must be modified by changing principles.

Neither of these two views give full credit to the iterative nature of the design synthesis activity, but in this work the term 'technology allocation' will chiefly be used to explain the transition from the functional structure to the organic structure. This is a key issue in mechatronics design.

As illustrated by SALMINEN & VERHO 1989 in Figure 30, there are two dimensions in technology allocation. One is to choose an appropriate technology (mechanics, electronics or software) for each function, the other is to choose a suitable working principle within one technological field. In some projects, the circumstances will justify that the choice of technology is made before the choice of actual principles. For instance a demand for design flexibility may favour a software solution, confidentiality may require ASIC technology, and available manufacturing facilities may dictate a particular mechanical technology.

An activity which is closely related to technology allocation is the *man/machine allocation*, i.e. the division of functions or activities to be performed by the human operator and by the technical system. This allocation cannot be completed in the functional domain, but needs decisions on principles in the organic structure, BOSMAN 1978:

> "The sociotechnical structure is the network of interrelations between human activities and equipment activities. It can be derived only when the activity allocation and the choice of the organic structure have been made."

The organic structure

The organic structure of a mechatronic system is a collection of those principal entities chosen to fulfill each subfunction and the relations necessary to establish the total function of the system. BOSMAN 1978:

> "... the organic structure shows the distribution of equipment according to constructional criteria. The familiar diagram of a computer is depicted with organs instead of functions. It shows the central processing unit, core memory, disc units, a tape reader, etc."

For this research I will propose the following definition of organs ('Funktionsträger'), adopted from ANDREASEN 1980:

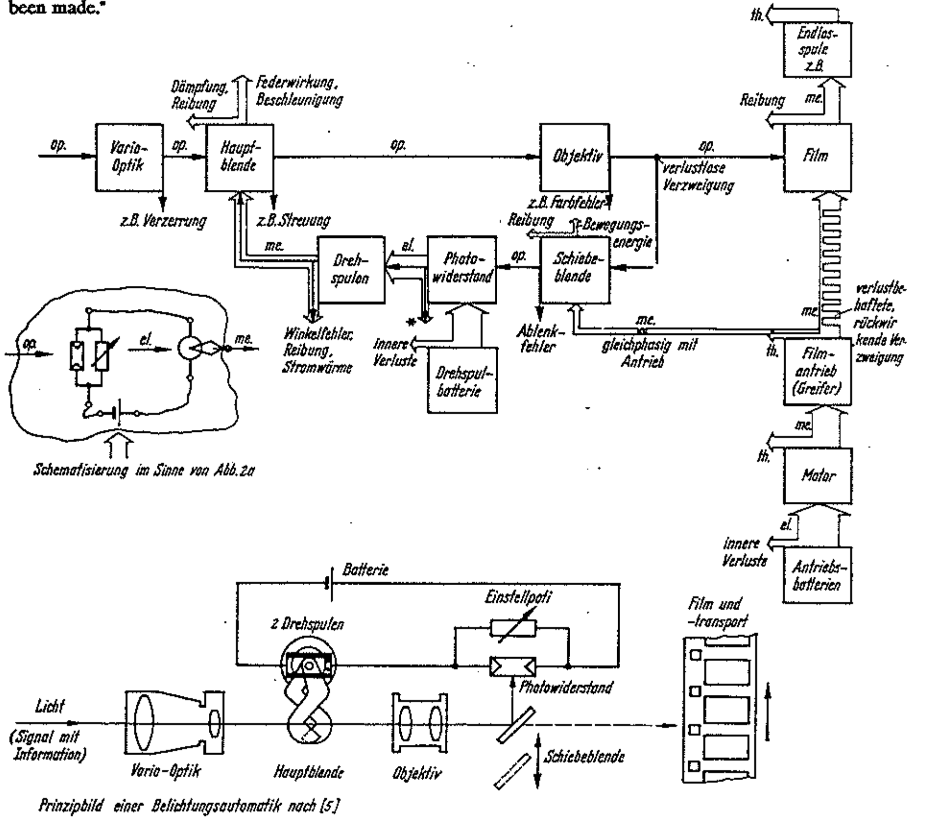


Figure 31 The organic structure of a movie camera modelled in two diagrams, ZWICK 1971

Definition

An organ is a class of entities consisting of parts, which exploit physical, chemical or biological phenomena to create a particular, required effect (function).

According to KRAUSE 1986, two types of relations between elements are interesting for the design process, the geometrical and the functional:

> "*Anordnungen* sind Relationen zwischen Systemelementen, die die geometrischen Relativlagen der Elemente beschreiben."
>
> "*Kopplungen* sind Relationen zwischen Systemelementen, die der Übertragung von Stoff, Energie oder Information zwischen den Elementen dienen."
>
> ('Spatial relations' are the relations between system elements which describe the geometrical relative positions of elements.
> 'Couplings' are the relations between system elements which provide the transfer of material, energy or information between elements.)

As an example, Figure 31 shows two diagrams of the organic structure of an electro-mechanical movie camera, ZWICK 1971. In the first diagram, each organ is described in verbal form and the flows of energy and information between organs are illustrated. The second diagram depicts the system elements in symbolic form, and the relations are electrical connections for the electrical system and the optical alignment for the mechanical system.

Several authors have described organic structures which they claim have general validity for mechatronic systems. Most of them show a computer in a central position. A typical example of such a structure composed of actuators, sensors, electronic interface circuits, man-machine communication etc, is shown in Figure 32.

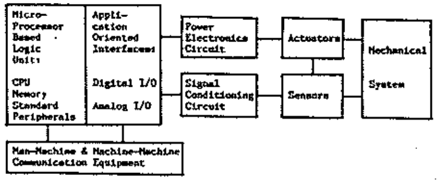


Figure 32 General organic structure of mechatronic systems, YAMAZAKI 1987

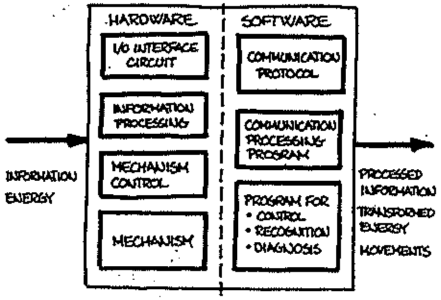


Figure 34 General organic structure including hardware and software elements, ITAO 1986

A more detailed model in Figure 33 describes the functional relations between elements. It was suggested by BÖHME 1978. ITAO 1986 adds software elements to his model of a general organic structure, but he does not specify relations between the elements, Figure 34.

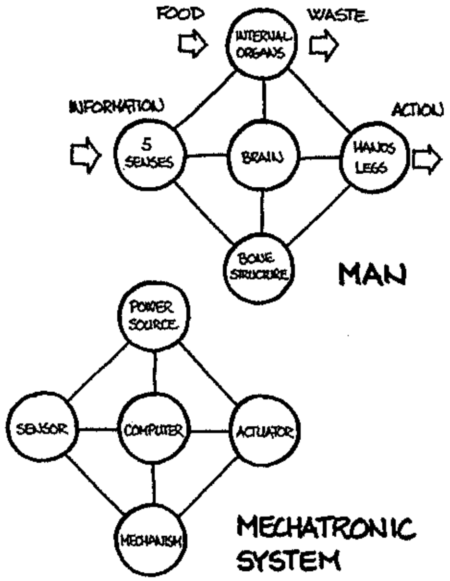


Figure 35 The five mechatronic elements resemble the organs of man, KAJITANI 1986

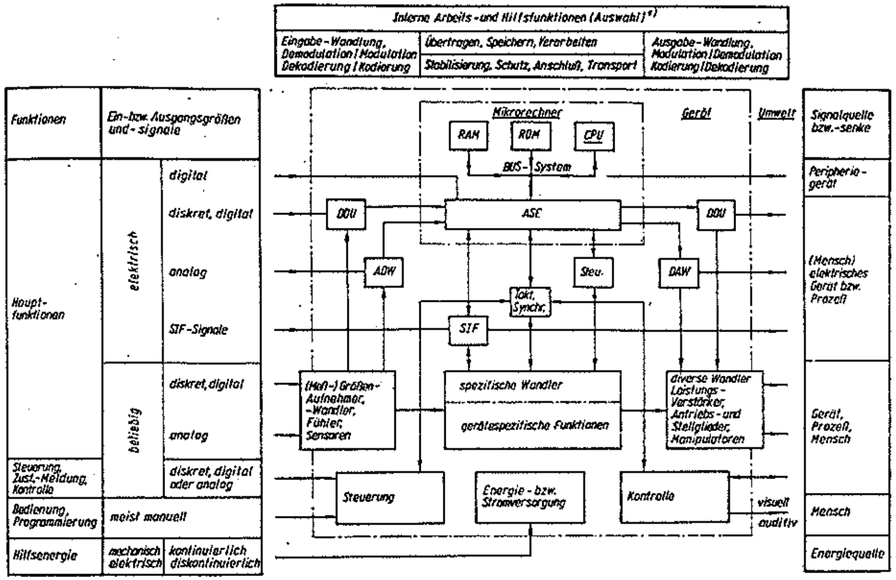


Figure 33 The organic structure of a typical mechatronic system with microprocessor, BÖHME 1978

Typical mechatronic organs

Many Japanese authors count five types of organ, which generally make up all mechatronic systems: sensors, computers, actuators, energy sources, and mechanism/structure. One may think of this as evolved from traditional machine systems, which were made up of only mechanism/structure, actuators, and energy sources, ITAO & NISHIDA 1989.

FUJIWARA 1984 and KAJITANI 1986-B find that these five elements of mechatronic systems resemble the organs of man: the senses, the brain, the hands and legs, the internal organs, and the bone structure, Figure 35. They find that man himself is an ideal mechatronic system to imitate. This understanding is clearly robot-biased: Humans are not able to replay a video recording or to copy a document at high speed, for instance.

ISHII 1989 argues that future mechatronic designs will more and more acquire the characteristics of biological systems. They will have a pronounced 'live state,' when the control system is working and a 'dead state,' when it is not. An example is the Control Configured Vehicle, an aircraft which is only aerodynamically stable under computer control.

Let us take a closer look at some of the most common types of mechatronic organ:

Sensors A sensor is a device, which in a reproducible way transforms a non-electrical property into a processable electrical output signal, SCHAUDEL 1982. The signal transformation properties of a sensor depend strongly on the mechanical design, and this makes the design process very complex, BUUR 1987. The mechanical housing of the sensor, besides contributing to the main function, has a protective function and a 'window function,' SCHAUDEL 1982.

The development of sensors points towards increased integration of the electronic signal conditioning and signal processing circuits, see Figure 36, KOHLER 1986. Another trend often referred to as 'intelligent sensors' involves the clustering of several

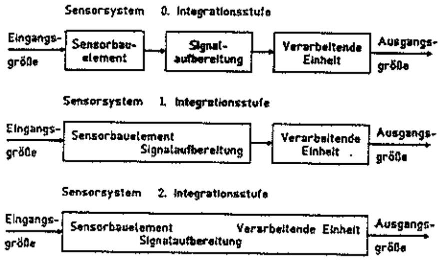


Figure 36 Levels of integration of a sensor system, KOHLER 1986

unlike sensor elements into one unit. Under microprocessor control it becomes possible to measure the momentary condition of the main sensor (e.g. temperature, calibration) and to calibrate or correct the output signal accordingly.

ISME 1989 suggests that sensors should be categorized in internal and external sensors in a functional sense. The internal sensors measure the internal state of the system (e.g. the position of a robot arm), while the external sensors measure features of the environment of the system (e.g. ambient temperature).

Actuators A (mechatronic) actuator is a device which transforms an electrical signal into a non-electrical physical property in a reproducible way. It seems that mechatronic actuators are limited to electrically powered devices and do not include actuators, which are powered by for instance hydraulics or pneumatics.

The design of actuators is similar to sensor design in that the physical realization is critical for achieving the functional properties. The main trend is to increase power and efficiency while at the same time minimizing size and weight, STUTE 1981. A development towards integrating electronics and microprocessor control in the actuator housing can also be seen. Like in intelligent sensors, additional sensor elements are built in to provide data on internal working conditions for more ideal control, e.g position feedback sensors.

Computers The micro computer allows fast and precise software dependent control of actions in the system. It also allows complex digital processing of signals, for instance pattern recognition.

A major problem in microprocessor application has been that conventional computers work sequentially while events in the real world mostly appear in parallel. This has motivated the development of processors with semi-parallel features (interrupt functions) and fully parallel working computers.

Mechanical design has hardly any influence on the functional properties of the IC computer, though a mechanical housing is needed for protection of the chip.

Microprocessors are increasingly designed specifically for particular application areas, like for instance motor control and signal processing. Such dedicated processor chips allow the integration of circuits for signal conditioning, analog/digital conversion, power control etc, into one-chip controllers.

For mechatronics design, it is important to note that the CPU (Central Processing Unit) of the microcomputer is flexible in a functional sense: it is programmed to perform a set of functions in a desired sequence, BOSMAN 1978. So it is not possible to read the function from a model of an organic structure which includes a microprocessor, unless it is supplemented with information about the program structure.

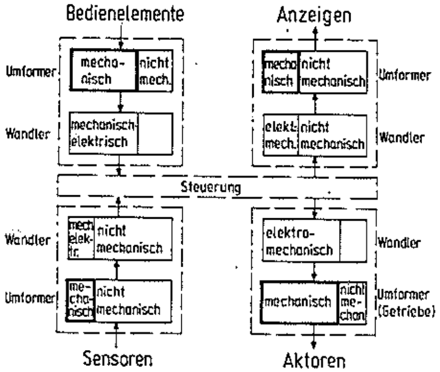


Figure 37 The organs of electronic control systems require mechanical and electro-mechanical energy transformers, ROTH 1985

Mechanisms The term mechanism is here used to denote all kinds of mechanical transformers of energy and signals, like gears, levers, link mechanisms etc. Sensors and actuators in mechatronic systems will seldom be able to treat physical properties in exactly the form in which they are available/required, so mechanical-mechanical transformers ('Umformer') are required, ROTH 1985. The same goes for those organs, which communicate with man: control elements and displays, see Figure 37.

External and internal interfaces

The concept of *interfaces* is important in mechatronics, because organs based on different technological principles are coupled together to achieve the total function of the system.

KAJITANI 1986-A finds it advantageous to regard the mechatronic system as a chain of interface elements, as in Figure 38. Designers must consider the quality of input/output for each element and think in matching functions to convert output from one element into input of the next.

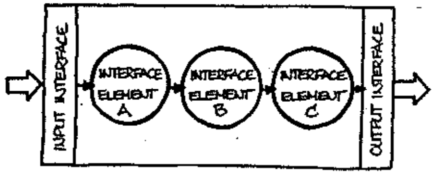


Figure 38 A mechatronic system regarded as a chain of interface elements, KAJITANI 1986-A

Two types of interfaces must be distinguished: The interface between the system and its surroundings (*systems* or *external interface*) and the interface between organs within the system (*internal interface*).

Systems interfaces According to GERHARD & LENART 1982 we may define 18 different types of 'Grenzeelemente' that facilitate the relations between the system and its surroundings, see Figure 24. They classify these organs according to the direction of exchange (input or output), to the type of corresponding system (man, environment or other products), and to the type of flow object (material, energy, or information).

HEINZL 1984 concentrates on the interface organs of electronic control systems for machines and proves that only four types of element are relevant when describing the flow of information, Figure 39.

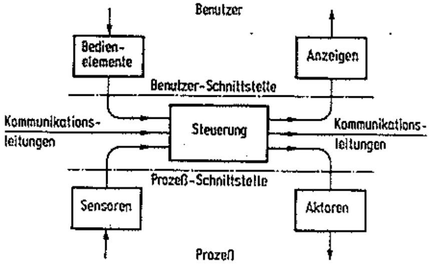


Figure 39 Model of the information flow in microelectronic control systems for mechanical processes, HEINZL 1984

These are sensors and actuators to measure and influence the state of a mechanical process, and control elements and displays to communicate with the user. This model has been adopted as the basis for the West German VDI/VDE Richtlinie 2422 for the design of machines with microelectronic control systems.

In his model, Heinzl does not explicitly prescribe any interface organs for the exchange of information with other products (except for 'Leitung'), and the model does not allow for aspects of environmental interaction (disturbances, etc.).

Internal interfaces The topic of interfaces between subsystems of mechatronic systems for process control has been studied by CORDES 1986. He suggests that the central block in HEINZL 1984's model should be divided into two communicating elements, 'Prozeßsteuerung' (process control) and 'Benutzerführung' (operator guidance), as in Figure 40.

He then shows, how functionally corresponding sensors and actuators can be grouped with dedicated control electronics to form a structure of autonomous subsys-

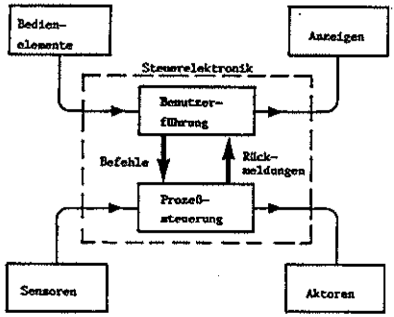


Figure 40 Dividing the microelectronic control system into two subsystems, CORDES 1986

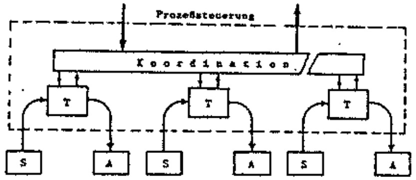


Figure 41 Process control by autonomous subsystems consisting of sensor (S), actuator (A), and dedicated control (T), CORDES 1986

tems, by means of a systematic synthesis process, see Figure 41. Such subsystems must then be coordinated by a higher level control organ.

A detailed account of interface levels in microprocessor based control systems can be found in SAUPE & LÄMMEL 1986. They describe 10 interface levels between both hardware and software organs, Figure 42.

The mechanical transformers ('Umformer') and the electro-mechanical transducers ('Wandler') described by ROTH 1985 in Figure 37 must also be considered internal interface organs.

Interfaces can be classified according to the type of conversion needed to match the output signal of one organ with the input requirements of the next, BÖHME 1978:

- Change of physical property ('physikalische Größe'), e.g. mechanical into electrical signal.
- Change of signal coding ('Darstellungsform'), e.g. analog to digital.
- Change of signal timing ('Zeitverlauf'), e.g. parallel to serial, unsynchronized to synchronized.

KAJITANI 1989 suggests a classification according to the effort necessary for matching output to input:

- Zero interface, meaning no conversion necessary.
- Passive interface, without energy supply.
- Active interface, with additional energy for conversion.
- Intelligent interface, applying a microprocessor for programable signal conversion.

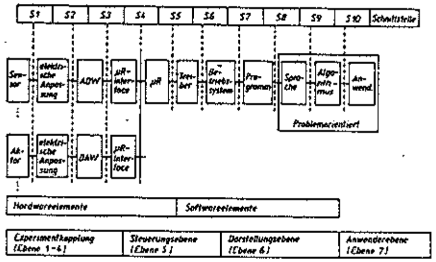


Figure 42 Internal interfaces on 10 levels in a microprocessor based control system, SAUPE & LÄMMEL 1986

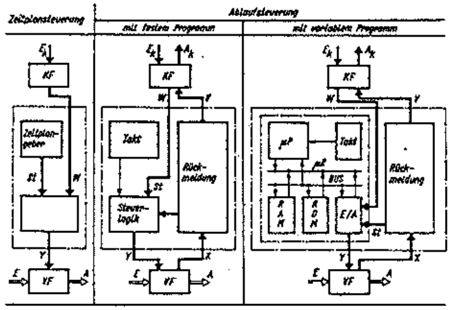


Figure 43 Organic structure of three types of controller hardware. Communication function (KF) and main function (VF), RÖHRS 1980

Control functions in the organic structure

It does not seem possible to illustrate a control strategy in an organic structure diagram. Organs carrying control functions are likely to be depicted as black-boxes with all appropriate input and output, like in Figure 39 and Figure 41. This leaves the reader the only choice of guessing about internal structure and working principles of the control organ.

In many cases the control algorithm is realised in computer software, meaning that the hardware structure of the control organ is a standardized microprocessor configuration not directly reflecting the actual control procedure. Figure 43 shows three such standardized hardware structures for different levels of control sophistication, RÖHRS 1980.

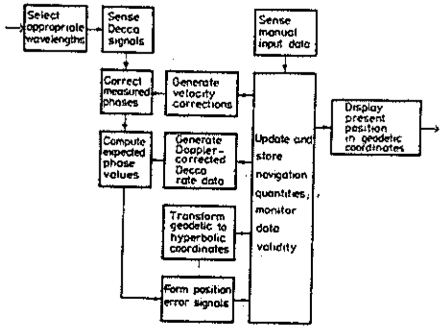


Figure 44 Activity structure (Functional flow diagram) for a radio position navigation system, BOSMAN 1978

BOSMAN 1978 claims that the organic structure must be supplemented with a description of the 'activity structure' of the system, i.e. a description of the activity sequence and execution rules, which are laid down in computer programs and in operation procedures for the human operator. He finds that the activity structure must be a flow system, because of its sequential nature and gives the example shown in Figure 44.

Another means for explaining the control principle in conjunction with the organic structure, is the timing diagram, which illustrates the states of different organs or signals of the system in relation to an axis of time. A timing diagram was for instance applied by D'ANGELO & UFER 1988 to model the logical functions of an electronic single lense reflex camera.

The application of models from the area of control theory for describing mechatronic systems is illustrated in VDI/VDE 2422, Figure 45. The purpose of such diagrams is to model the dynamic behaviour of the systems, and for the professional control engineer, there is direct correspondence between the model and a mathematical description of the transfer function.

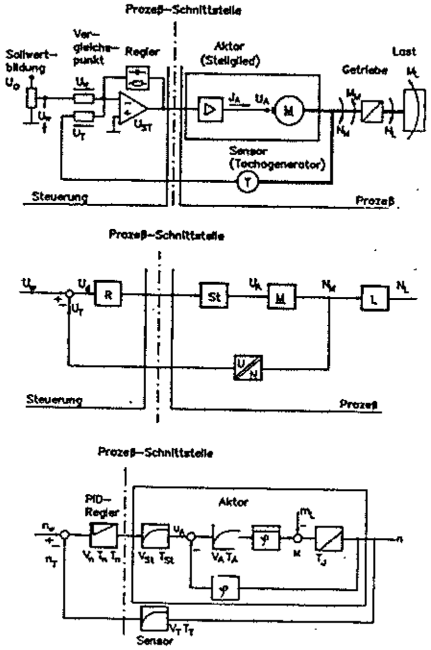


Figure 45 Three types of control theoretical models for describing a DC-motor control system. (a) 'Übersichtsplan', (b) 'Blockschaltplan' and (c) 'Signalflußplan', VDI/VDE 1986

Signals in mechatronic systems

To allow manipulation in a technical system information must be tied to either energy or material. So in discussions of the organic structure, information can only exist in the form of signals (e.g. electrical signals) or coded in material objects (e.g. a blood sample, text printed on paper). A definition after KOLLER 1976 and KRAUSE 1986:

> *Definition*
>
> A **signal** is the physical carrier of information as the time variance of en energy form.

A signal can be characterized according to the signal carrier, the information parameter, and the signal form, see Figure 46. It should be noted that Krause includes the coding of material in his understanding of signal ('geometrical' and 'chemical signals').

4.4 The synthesis activity

In Sections 4.2 and 4.3 I considered phenomena related to the nature of mechatronic systems: their purpose, functionality, structure etc. In this section, I will review literature related to the synthesis of such systems in general.

No *descriptive* analysis of the synthesis activity were found in the literature, but a few authors offer *prescriptive* statements, i.e. recommendations as to how the design of mechatronic systems should be carried out. These statements take the form of either design procedures (i.e. recommended sequences of activities) or design principles (general rules for designing). I will discuss each kind in a separate paragraph.

General theory of systems or synthesis strategy to support the suggested design procedures and design principles is not found in the literature, so the reader has to guess their basis.

2. Signalträger	3. Informationsparameter P	4. Signalform
2.1. Mechanisches Signal (Geschwindigkeit, Beschleunigung, Kraft, Masse, Druck, Arbeit usw.) 2.2. Geometrisches Signal (Länge, Dicke, Winkel, Fläche, Volumen, Niveauhöhe, Schriftzeichen usw.) 2.3. Hydraulisches Signal (Druck, Druckdifferenz, Flüssigkeitsmenge usw.) 2.4. Pneumatisches Signal (Druck, Druckdifferenz, Gasdurchsatz usw.) 2.5. Akustisches Signal (Schallstärke, Tonhöhe usw.) 2.6. Thermisches Signal (Temperatur, Wärmemenge usw.) 2.7. Magnetisches Signal (Induktivität, Feldstärke, Magnetfluß usw.) 2.8. Elektrisches Signal (Strom, Spannung, Leistung usw.) 2.9. Optisches Signal (Leuchtdichte, Brechungsindex, Wellenlänge usw.) 2.10. Kernphysikalisches Signal (Neutronendichte usw.) 2.11. Chemisches Signal (pH-Wert, Gaskonzentration usw.)	3.1. Amplitude 3.2. Frequenz 3.3. Phase 3.4. Anzahl von Impulsen 3.5. Dauer von Impulsen 3.6. Folge von Impulsen 3.7. Lage von Impulsen 3.8. Anzahl von Punkten 3.9. Anordnung von Punkten 3.10. Abstand von Punkten zu Bezugspunkt bzw. von Winkeln zu Bezugswinkel	4.1. Analoges Signal P kann innerhalb eines bestimmten Bereichs beliebige Werte annehmen 4.2. Diskretes Signal P kann nur endlich viele Werte annehmen 4.2.1. Binäres Signal P kann nur genau zwei Werte annehmen 4.2.2. Digitales Signal Die Werte von P entsprechen Wörtern eines vereinbarten Alphabets 4.2.3. Mehrpunktsignal Diskretes Signal ohne vereinbartes Alphabet 4.3. Kontinuierliches Signal P kann sich zu jedem beliebigen Zeitpunkt ändern 4.4. Diskontinuierliches Signal P kann sich nur zu bestimmten Zeitpunkten ändern

Figure 46 The characteristic properties of a signal, table compiled by KRAUSE 1986

Design procedures

I will concentrate on the conceptual activities in mechatronics design with the central problem being how to split the design concept into mechanical, electronic and software subsystems.

The VDI/VDE Richtlinie 2422 suggests a design process of five phases: problem definition, concept design, embodiment design, detailed design and systems test, Figure 47, VDI/VDE 1985 and LEHMANN 1985. It builds on the model suggested by HEINZL 1984 (Figure 39) as a fundamental functional structure for microelectronic controlled machines, and it recommends the following steps for finalizing the design concept:

1 Functional structure:
  - A Describe the operational procedure on the user's level.
  - B Describe the system function on the process level.
  - C Describe the control interface.

2 Choose the means for central control (analog, digital, microcomputer)

3 Establish the subtasks and balance the subconcepts for the mechanical subsystem incl. industrial design, for the software, and for the electronic circuit.

As soon as the concept has been established, the procedure recommends to divide activities into the traditional electromechanical, electronics, and software design and proceeds to describe detailed steps of the embodiment and detailed design phases.

Comprehensive research on the conceptual design phase has been carried out in Finland. Based on the experience from a number of development projects in Finnish industry, SALMINEN ET AL 1990-A recommend methods to be used along with their mechatronics design procedure: VDI 2222, Quality Function Deployment (QFD), Structured Analysis and Structured Design (SA/SD), Fault Tree Analysis etc.

Of interest here is that Salminen distinguishes between a functional and a behavioral description of the mechatronic concept, as in Figure 48.

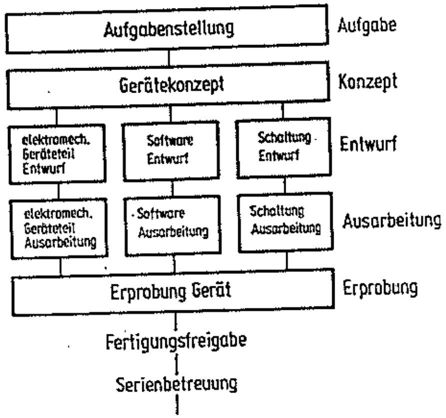


Figure 47 Coarse phase plan for designing machine systems with microelectronic control, VDI/VDE 1985

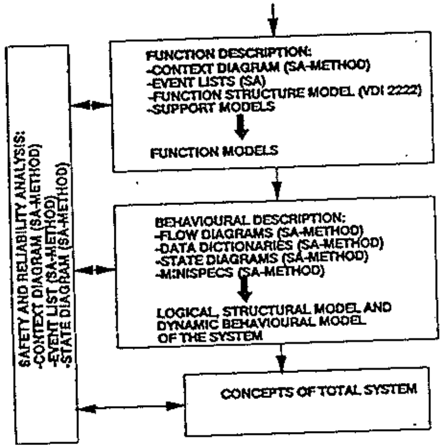


Figure 48 Conceptual activities in a Finnish procedure for designing mechatronic systems, SALMINEN ET AL 1990-A

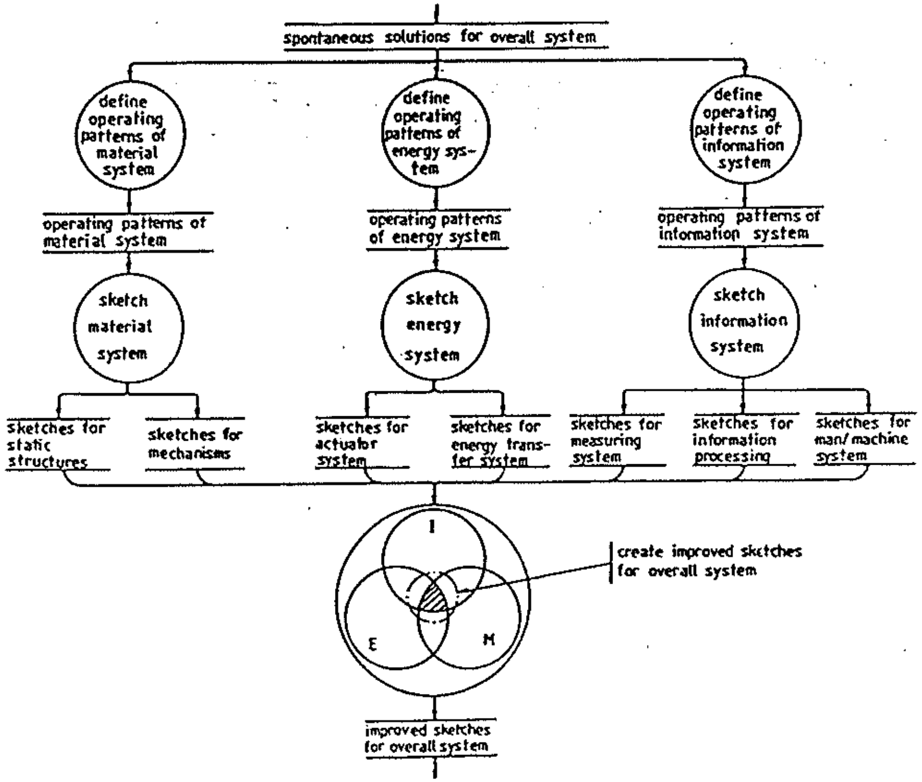


Figure 49 Splitting the conceptual design activity into material, energy, and information systems design, SALMINEN & VERHO 1989

For the initial proposal for a design concept, SALMINEN & VERHO 1989 suggest that tasks should be split into three groups: The sketching of the material, energy, and information subsystems respectively, rather than groups of mechanical, electrical and software design, see Figure 49. In this way they focus on the purpose of the systems, not so much on their technological realization.

One other suggestion on design procedures for mechatronics should be mentioned here, though it is not too explicit about how the split into mechanical, electronic and software subsystems should be performed. YAMAZAKI 1983 has published a procedure for the design of micro-computer control systems for machine tools, part of it is illustrated in Figure 50. The method includes detailed flow chart work sequences for the specification, hardware development, software development, and debugging phases. Yamazaki prescribes 6 types of design specification to be completed, and he advises the use of standardized working documents (like data sheets, I/O port maps, module structure charts) to improve communication between team members and minimize double work and errors. Yamazaki has integrated elements from the software methods 'Structured Analysis and Design Technique' (SADT) and 'Hierarchy plus Input-Process-Output' (HIPO) in his design procedure.

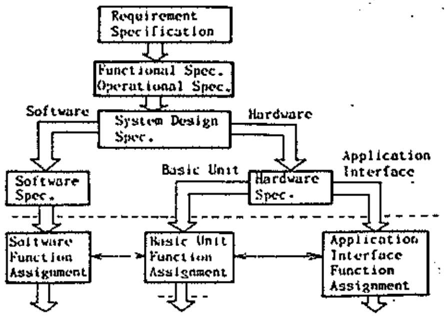


Figure 50 The conceptual activities of a design procedure suggested by YAMAZAKI 1983

Of further interest for the conceptual design of mechatronics are procedures for establishing the man/machine interface.

BOSMAN 1978 presents such a procedure, based on his concept of *activity structure* of the mechatronic system, as discussed on page 49. He describes how the allocation of activities between operator and machine determines the *sensory architecture* and the *functional architecture*, see Figure 51. The activity structure of Bosman bears a strong resemblance to the behavioral description of SALMINEN ET AL 1990-A.

Also for the design of user interfaces, HEINZL 1984 and CORDES 1986 recommend the following steps:

1. Estimate the flow of information in both directions.
2. Choose control elements and displays.
3. Describe briefly the main steps of the operation procedure and the main alternative choices.
4. Model roughly in a state diagram.
5. Complete a state-transition matrix.
6. Describe the operational dialogue in detail.

Here one will recognize a similar distinction between sensory architecture (control elements and displays) and functional architecture (operational procedures).

Design principles for mechatronics

Design principles are tied to the present state of technological progress. So for instance a principle which was popular in the 1950s: "Do not apply electronics for functions which can be realized with mechanics" is hardly applicable at present. In fact, the opposite statement is more likely to be true today.

Very systematic work on design principles has been carried out by SCHILLING 1984 for mechanical design, see Figure 52.

The design principles dedicated to mechatronics found in the literature mostly reflect experience derived from design projects. A few examples follow. ITAO 1986 (design of data storage systems):

- Minimize the role of mechanisms by using electronics and software.
- Make the mechanism simple, i.e., decrease the number of moving parts and minimize friction.
- When high performance is wanted, make specialized designs instead of general purpose modules.

VORBACH 1982 (teletypewriter design):

- Move the electro-mechanical transducers as close as possible to the mechanical functions, in order to minimize the accelerating masses and thus decrease noise and wear.

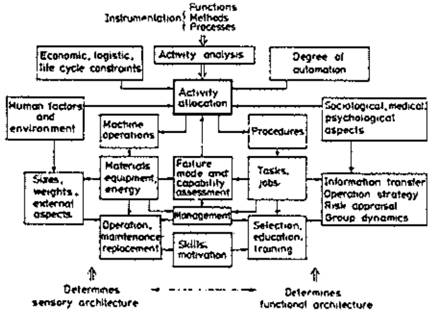


Figure 51 Design procedure for man/machine systems design, BOSMAN 1978

TAKEUCHI 1986 (machine tool design):

- Divide functions into modules to increase flexibility and the degrees of freedom during design.

From interviews with designers in industry:

- Accept any complication in software, if the electro-mechanical design becomes simpler.
- Digitize input signals as early as possible.

Konstruktionsprinzipien	Erläuterungen
1.1. Funktionentrennung	Eine von anderen unabhängige Teilstruktur übernimmt eine Teilfunktion
1.2. Funktionenintegration	Eine Teilstruktur übernimmt mehrere Teilfunktionen
2.1. Strukturtrennung	Eine Teilstruktur wird in mehrere Strukturbestandteile aufgelöst
2.2. Strukturintegration	Mehrere Teilstrukturen werden zusammengefaßt
3. Funktionswerkstoff an Funktionsstelle	Beschränken des funktionsnotwendigen Werkstoffes auf funktionsnotwendige Strukturbestandteile
4. Vermeiden von Überbestimmtheiten	Wahl von Elementenpaaren, deren Gesamtanzahl an Unfreiheiten die zulässige Anzahl nicht übersteigt
Prinzipien der fehlerarmen Anordnungen	Wahl einer Struktur mit minimierten Fehlern
5.1. Fehlerminimierung	Minimierung der Fehlereinflußfaktoren
5.2. Innozenz	Funktionsgröße hängt nur in 2. und/oder höherer Ordnung von der Störgröße ab
5.3. Invarianz	Funktionsgröße ist in weiten Grenzen unabhängig von der Störgröße
5.4. Kompensation	Störgröße beeinflußt gleichzeitig zwei sich gegenläufig verändernde Strukturparameter
Prinzipien der Selbstunterstützung	Hilfsfunktionen unterstützen Hauptfunktionen
6.1. Selbstverstärkung	Ausnutzung einer sich verändernden Größe zur Verstärkung der Hauptfunktion
6.2. Selbstausgleich	Hilfswirkung einer Nebengröße zur Erfüllung der Hauptfunktion
6.3. Selbstschutz	Einleiten eines zusätzlichen Kraftleitungsweges bei Überlast
Prinzipien des Kraftflusses	
7.1. direkte und kurze Kraftleitung	kürzester und direkter Weg für Weiterleitung von Kräften und Momenten
7.2. gleiche Gestaltfestigkeit	strebt überall gleich hohe Ausnutzung der Festigkeit an
7.3. abgestimmte Verformung	Verbindung von Teilen so gestalten, daß in der Paarungsstelle keine Relativverformungen entstehen
7.4. Kraftausgleich	Als Nebenwirkung entstehende Kräfte (Momente) auf kurzem Wege schließen
7.5. definierte Kraftverzweigung	Verzweigung der Kraft durch statisch bestimmtes Arbeitsprinzip

Figure 52 Summary of design principles for precision mechanical design, SCHILLING 1984

5 METHODOLOGY FOR MECHANICS, ELECTRONICS & SOFTWARE DESIGN

The purpose of this chapter is to describe the characteristics of mechanical, electronics and software design and to highlight important aspects for the design of mixed systems.

I will not attempt any complete review of original publications on design in these fields, since literature is far too extensive for that. This chapter is mostly based on contributions which summarize the state of the art, to achieve an overview of design characteristics, basic theories and methods.

Broadly speaking, the developments in each field may be summed up as follows:

Machine design was established as a science in the 1960's with two main approaches: a functional one in West Germany/DDR and a procedural one in Britain and the US. The main motivation was that systematics could help in obtaining the best solution out of many alternatives, and that a methodical approach could increase efficiency and controllability in the design process. Machine design methodology has not been well accepted in industry, though the recent quest for increased development speed and product quality seems to add to the interest.

For this research, it is mainly the functional approach which is of interest, because the synthesis activity must be based on knowledge of the technical system to be designed, whereas procedural aspects can largely be explained on the problem solving and product development levels.

Electronics design is dominated by theories of circuit analysis and logic design, but design methodology in the conceptual sense is very rare. The electronics field seems to have concentrated on finding and optimising the one solution which fulfills specifications, rather than searching for several alternatives to be evaluated systematically. The systems engineering approach of decomposing complex tasks into subsystems with separate specifications and interface considerations is well accepted in electronics industry.

There seems to be a growing awareness that the detailed design phase in electronics needs special design methods to cope with the problems arising from the physical realization of circuits: Electrical disturbances (EMC), heat dissipation etc. Such problems cannot be dealt with by traditional circuit design tools.

Software design has been forced to apply a structured approach since the early 1970's, when the complexity of systems became too high for intuitive programming methods, and the number of errors rose to an unacceptable level. Software engineering has become a scientific discipline with research concentrating on procedural and specification tools.

The need for structured design methods is well recognized in industry, and also here does the quality consciousness increase the demand for better transparency and documentation in the design process.

5.1 Machine design

The purpose of a machine system is to facilitate a *technical process*, i.e. to effectuate the transformation of a process object from a given to a desired state. An engine for instance turns petrol into rotary energy and a lathe (with the help of the operator) transforms raw material into a machine part.

HUBKA 1973 in his Theory of Technical Systems strains to prove that the machine is not in itself a technical process, it exerts the *effects* ('Arbeitswirkungen'), which are necessary to make the process happen. For most technical processes, the effects are created by the machine system in collaboration with a human operator, see Figure 53.

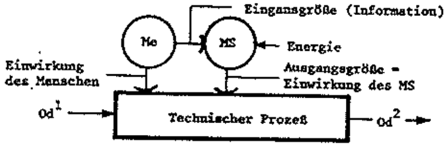


Figure 53 The technical process and the necessary effects realized by machine system (MS) and operator (Me), HUBKA 1973

The process objects, which are transformed by machine systems, can be classified as either material, energy or information. This is fundamental to the German schools of design theory, since it allows us to regard the machine as a system of processes (transformation functions) interacting through the flow of material, energy and information, Figure 54. In ANDREASEN 1980's Theory of Domains, this is the process domain.

The effects created by the machine can be understood as *purpose functions*, or more precisely, ANDREASEN 1980:

> *Definition*
>
> A **(purpose) function** is the ability of a machine to create an expedient effect

In the functional domain the machine is regarded as a structure of purpose functions with causal or logical relations. An example of a functional structure diagram for a tea brewing apparatus is shown in Figure 55.

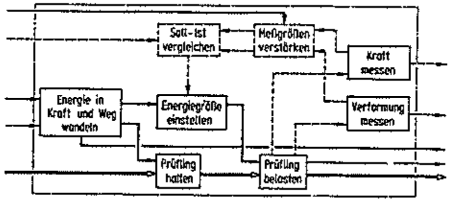


Figure 54 The machine system regarded as a structure of transformations: A mechanical test bench, PAHL & BEITZ 1977

According to ANDREASEN 1980, the link between process and function is the choice of technology, where the term *technology* is defined in a narrow sense:

"Technology must be understood as the kind and sequence of the subprocesses of a process and the interaction in space and time between the process object and the effects, which create the transformations."

This means that the designer has to decide on a general technology, before he can transform a process structure into a structure of purpose functions.

The concept of purpose functions is not widely accepted in design theory literature, but it is important, because it explains a step in between the description of transformations and the actual realization of functions through physical principles in the machine (organs). Also, the term comes close to the designer's practical understanding of machine functions, for instance the function of a ball bearing and of a static structure is much easier explained in terms of effects than transformations.

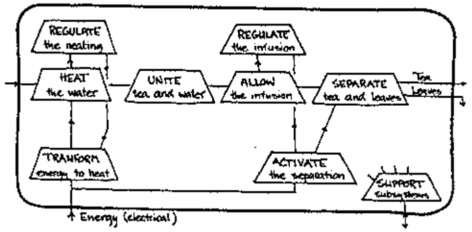


Figure 55 The machine system regarded as a structure of (purpose) functions: A tea brewing machine. Words in capitals denote activities, HUBKA ET AL 1988

Theory of properties

The benefit obtained by using a machine is measured by its *properties*, i.e. how fast, how heavy, how expensive, how durable, how beautiful it is etc. The main function can be regarded as the most important property of the machine system.

The machine is characterized by a wide range of properties, and it is the value of all its properties which determines the quality of the machine. Any machine will for instance have a size, an appearance and a life-time, no matter whether they were chosen consciously by the designer or not.

The product properties can be classified in a number of ways, HUBKA 1973: according to technology areas, their importance, quantifiability and so on. It is advantageous to classify according to the five stages of the product life-time cycle, TJALVE 1979-A: design, manufacturing, marketing, use, and destruction.

One may understand design as a process which establishes the values of all the properties of the product. The thing which makes design such a complex activity, is that each property has a different priority, and that properties are greatly interdependent. Therefore a small change of the design will affect not only one, but a whole range of properties.

The Theory of Basic Design Properties states, that the designer can only manipulate a small set of properties directly, and that all other product properties stem from those, HUBKA 1973. For machine design, the set of basic design properties have been listed by TJALVE 1979:

for the total product	*structure* (elements, relations)
for each part	*form* *dimensions* *material* *surface quality*

HUBKA 1973 has also suggested *tolerances* and *manufacturing method* as basic properties, but since they can be covered by dimensions and material respectively, they have been omitted here.

Vertical causality

The decomposition of a particular function into subfunctions is only possible, when a means has been chosen to realize the function. This is the essence of *The Law of Vertical Causality* as formulated by HUBKA 1976 and ANDREASEN 1980. Here *means* is a general designation for the solution to a problem, i.e. a technical principle or an organ, which can realize the required function.

There is causality in the sense that once a function is formulated, then it is possible to designate a number of alternative means, which may all carry out the desired function. Every means will however need the realization of a set of subfunctions on a lower level.

The Law of Vertical Causality is best illustrated in the *function/means tree*, TJALVE 1979-A. This is a hierarchical structure of alternating levels of required functions and of alternative means to realize the functions. In Figure 56 a function/means tree for a tea brewing apparatus is reproduced.

Please note how the symbolics in the figure underline alternative sets of means (either... or....) and obligatory sets of functions (both.... and...) on alternating levels.

This Law of Vertical Causality, as expressed in the function/means tree is an important tool for the functional synthesis of products, and it seems to have general validity for all technical systems, including mechatronics.

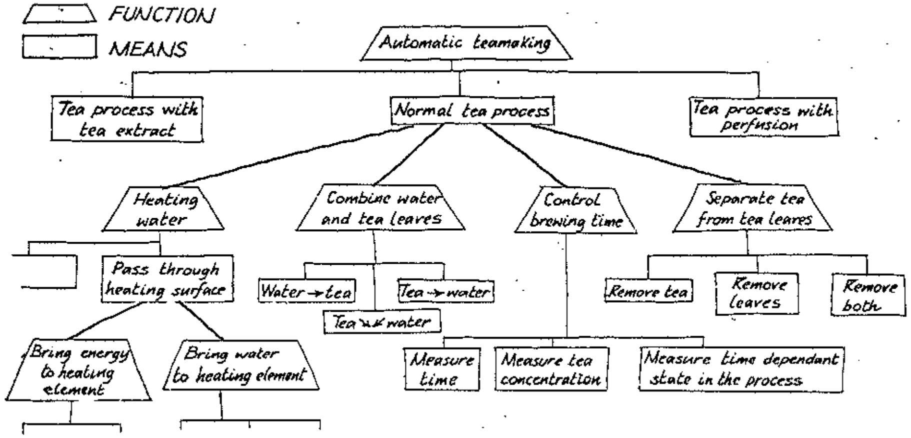


Figure 56 A function/means tree for an automatic teamaker: The normal tea process requires the realization of four purpose functions, TJALVE 1979-A

Complex of secondary functions

This principle proposed by HUBKA 1973 states that any function will need the simultaneous realization of secondary functions of the types:

power (or propulsion) function
control function
assisting (or auxiliary) function
support (or structural) function

For instance the halogen lamp of an overhead projector can only fulfill its function when sufficient electricity is available (power function), if power can be switched on and off (control function), if there is cooling (auxiliary function), and if the lamp is fixed in a suitable electrical socket (support function).

The principle states that in general, some (but not necessarily all) of the secondary functions of the complex will be required for any function of a machine system.

When this principle is combined with the Law of Vertical Causality described in the previous paragraph, then we will see that it is the means (principle) realizing the function, which determines what types of secondary functions are needed on the next level, see Figure 57.

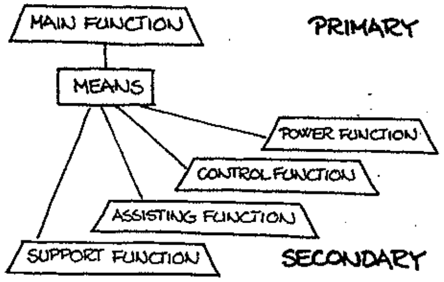


Figure 57 The choice of means determines the complex of secondary functions on lower level, ANDREASEN 1980

Logical functions

In machines, it is often necessary to assure that functions are performed in the proper sequence and with appropriate timing. This task requires the realization of *logical functions*, i.e. functions which basically deal with the semantic value of information in order to derive decisions.

A logical function is a discrete type of control function, so for instance the purpose of the speed controller of a steam engine should *not* be termed logical function, since it is to exert continuous control.

There is no tradition of the explicit handling of such logical functions in machine design, so literature on the topic is rare. I will treat logical functions in detail, because this is an important aspect of mechatronics design. ROTH 1982:

"Mechanisch-logischen Techniken wird vielerorts im Zeichen der rapiden Verbreitung von Mikroprozessoren weniger Bedeutung zugemessen. Das ist berechtigt, insbesondere für Prozesse der Nachrichtenverarbeitung; es ist jedoch nicht gerechtfertigt
- für zahlreiche Fälle der Nachrichtenein- und -ausgabe,
- für sehr einfache Verknüpfungen, bei denen der Übergang von der Mechanik zur Elektronik mindestens den gleichen Aufwand wie eine mechanische Logik erfordert,
- bei besonderen Sicherheitsanforderungen (z.B. mechanisches Sperren nicht erlaubter Lagekombinationen)."

(With the rapid spread of microprocessors, mechanical-logical techniques are mostly given little consideration. This is reasonable in particular for processes of information transformation, but it is not justified
- for many cases of information input and output,
- for very simple connections, where the change from mechanics to electronics requires at least the same effort as mechanical logic,
- for cases with exceptional safety requirements (e.g. mechanical blocking of non-permitted combinations of movements).)

ROTH 1982 has demonstrated that mechanisms with logical functions can be systematized in design catalogues. Examples of mechanisms which realize the *elementary logical functions* of storing, conducting, transforming, transducing, and connecting information (Speichern, Leiten, Umformen, Wandeln, Verknüpfen), are shown in Figure 58. In Roth's opinion, logical functions are *transformations of information*. He distinguishes mechanisms with combinatorial logic (AND, OR, NOT) and mechanisms with sequential logic (flipflops with 1-bit memory).

RODENACKER & CLAUSEN 1973 suggest the formulation of *logical relations between functions* ('logische Wirkzusammenhänge') as an important precondition for the decision on the technical/physical realization of functions. For example the tasks of turning on and operating a machine will include logical activities from the operator.

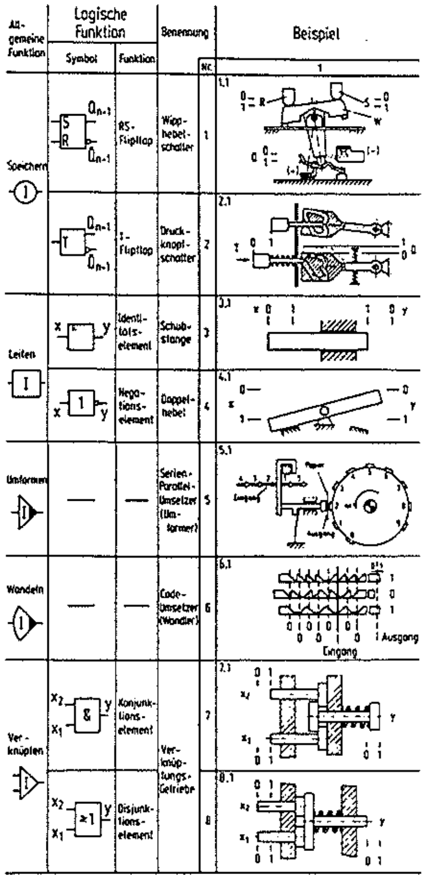


Figure 58 Mechanisms which realize logical functions on information, ROTH 1988

Such logical relations can be clarified by asking questions of "*if... then...*", for instance: "*If* somebody enters the elevator and *if* a button is pushed, *then* the doors of the elevator will close". An example of logical relations is illustrated in Figure 59.

"*If... then...*" sentences relate to characteristics of the functions: position, route, duration, condition, number (Ort, Weg, Zeit, Bedingung, Zahl), according to RODENACKER & CLAUSEN 1973.

RIESCHEBERG & WOITASCH 1976 have completed an analysis of information transforming logical mechanisms with the purpose of improving design procedures for such devices. They report the main conclusions as follows:

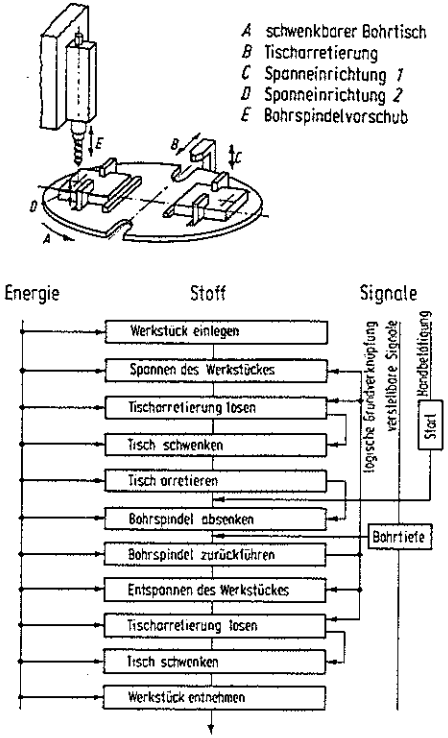


Figure 59 A drilling system consisting of five parts A-E, which all have two possible states. Logical relations are symbolized by arrows, RODENACKER & CLAUSEN 1973

"1 In informationsmechanischer Konstruktionen sind alle Grundfunktionen einschließlich deren Kombinationen, wie Leiten und Verknüpfen, Wandeln und Speichern, zu beobachten.

2 Die informationsmechanische Grundfunktion 'Verknüpfen' erweist sich von der Signalart abhängig, d.h. eine durchgängige Zuordnung zwischen Funktion und Funktionsgrößen giebt es in der Informationsmechanik nicht.

3 Diskrete mechanische Signale äußern sich als wechselhafte Bewegungsschritte von Körperflächen in einem allgemein krummlinigen Koordinatenraster.

4 Informationsparameter einer diskreten Signalbewegung kann deren Schrittweite, Schrittrichtung, Schrittzeit oder Schrittort sein. Sein Wertevorrat ist i.allg. unbestimmt."

(1 All the elementary logical functions of conducting, connecting, transforming, and storing, together with combinations of these, can be observed in information transforming mechanisms.

2 The elementary function of 'connecting' proves to be dependent on the type of signal, i.e. there is no consistent attachment of functional parameters to the functions of the information transforming mechanism.

3 Discrete mechanical signals appear in the form of steps of movement of object surfaces in a usually curved plane of coordinates.

4 The information parameter of a discrete signal movement is either step distance, step direction, step duration or step position. Its number of available values is generally undetermined.)

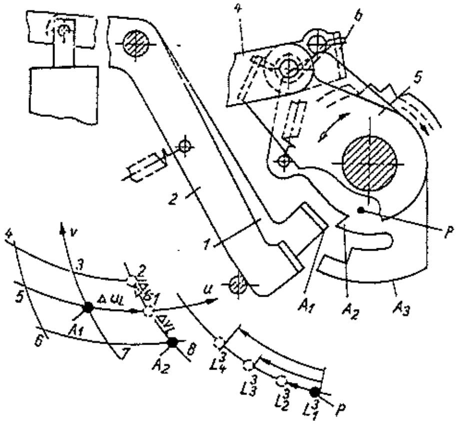


Figure 60 The logical mechanism of a line shifter device. The possible steps of movement are illustrated in the graphical model below, RIESCHEBERG & WOITASCH 1976

The last statement proves that an application of logical symbols from digital electronics is not appropriate for describing logical mechanisms, because such symbols are dedicated to systems with a single information parameter (i.e. electrical potential). The authors voice the need for graphical models of logical functions in mechanisms and suggest the application of a 'Schaltliniendarstellung' as in Figure 60.

The organic structure

The units which realize the required effects in a machine are denoted *organs* (HUBKA 1984 and ANDREASEN 1980) or *function carriers* ('Funktionsträger' - PAHL & BEITZ 1977, ROTH 1982).

Organs are units in a material-geometric sense, which independently perform one or more functions, just like the organs of the human body. They are for instance a bearing, a gear, a hydraulic cylinder, a mechanism or a clutch. The term organ is convenient, because for a functional purpose the designer can handle such types of solution without considering the detailed assembly structure of machine parts and their interactions.

The organic structure realizes the total function of the machine. As proposed by TJALVE 1979-A, we can distinguish between two types of organic structures according to the kind of relations between organs.

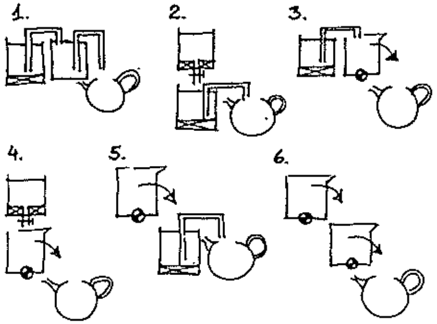


Figure 61 Basic organic structures of an automatic tea maker. Organs are illustrated by simple machine symbols, TJALVE 1979-A

The *basic structure* ('prinzipielle Struktur') is a structure of organs and their functional relations ('Kopplungen'), i.e. relations, which transfer material, energy or information, see Figure 61. This structure is mainly used to express the functional principle of the machine.

In the *quantified structure*, the relations between organs are of geometric-spatial kind ('Anordnungen'), Figure 62. This structure expresses the relative, spatial arrangement of the organs, and it gives an idea of the total form of the machine.

It is easy to see from the two figures that the use of simple symbols for the organs makes it very easy to alter the functional order of organs and the spatial arrangement. TJALVE 1979-A has suggested a number of variation methods for creating alternative solutions based on the organic structures.

For machine design it can be concluded that a theoretical basis exists for describing the functions of the machine from both transformation and effects viewpoints. Likewise the physical realization of the machine can be described in an organic sense emphasizing functional and spatial arrangements.

Fundamental theorems link together the different systems viewpoints or domains: The principles of Vertical Causality and the Complex of Secondary Functions. Knowledge of logical functions however, is not sufficiently incorporated in machine design theory.

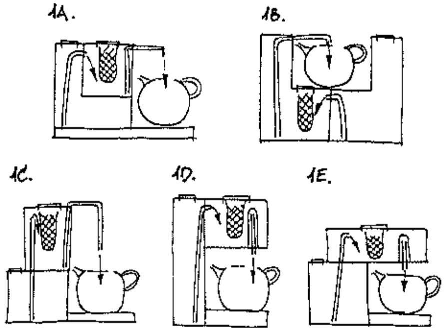


Figure 62 Quantified organic structures of an automatic tea maker. Geometric relations can be recognized, TJALVE 1979-A

5.2 Electronics design

The purpose of an electronic system is to transform information in the form of electrical signals: voltages and currents.

The *electrical signal* is characterized by the information parameter (e.g. amplitude, frequency, phase) and by the form of the signal (e.g. analog, digital), refer to Figure 46. According to KRAUSE 1986 it is possible to formulate an *information efficiency* ('Informationswirkungsgrad') by analogy with the energy efficiency of engines:

> "...maximale Erhaltung bzw. minimale Verfälschung der Information, d.h. Einhaltung einer vorgegebenen Informationverarbeitungsfunktion durch minimale lineare und nichtlineare Verzerrungen der Information und keinen Ausfall von Teilen bzw. der gesamten Information."
>
> (...respectively the maximum retention and the minimum falsification of Information, i.e. the fulfilling of the specified information transformation function with minimal linear and non linear distortion of the information and without omission of part or all of the information.)

The information efficiency is the most important property for evaluating systems like measuring instruments and equipment for speech, image and data processing.

The kinds of signals to be transformed by the electronic system are defined by its environment that is, by *interface components* such as sensors, actuators, control elements and displays, and by the communication with neighboring systems. This perception was illustrated in Figure 39.

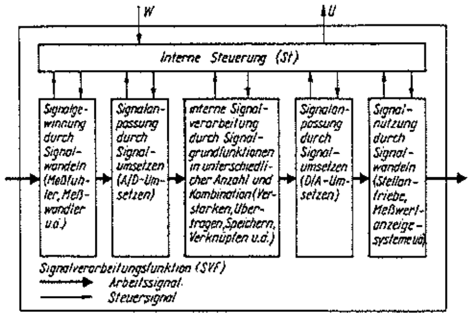


Figure 63 General structure of a mixed analog/digital signal processing electronic system, RÖHRS 1980

The electronic system can be regarded as a functional structure of information transformations or - since information in electronics exists only in the form of electrical signals - as a structure of signal processes.

No theory to support the decomposition of the total function into a process structure is found explicitly formulated in the literature. Some types of subfunctions will generally be found in all electronic systems: power function, user interface, communications interface, driver function for actuators etc. RÖHRS 1980 has suggested a general model of the structure of a signal processing electronic system, shown in Figure 63.

If it has been decided to apply a microprocessor in a design, then the kinds of secondary functions required (memory functions, clock-generator etc.) and their structure will also be specified, see Figure 33 and Figure 43.

Electronics design can be divided into three comparatively independent design activities: component design, circuit design and electronics packaging design. I will proceed to consider each activity separately.

Electronic component design

To design standard components for electronic systems is a process not unlike machine design.

For simple components like capacitors, resistors, potentiometers and transformers, the Theory of Basic Design Properties is applicable. A component is fully defined by a (mechanical) structure of single parts, each specified by its form, dimensions, material and surface quality, see Figure 65. Naturally the electrical and magnetic properties are essential for the choice of materials.

The relations between parts are functional (conducting electric current) and geometric-spatial (ensuring mechanical stability).

For components like semiconductors and ICs, which perform complex functions, the Theory of Basic Design Properties does not sufficiently explain the manipulable design properties. The central element in an IC component, the semiconductor chip, has in itself a complex material structure ('architecture') of infused areas and conductor networks. With the number of single transistors running above 1.000.000 for VLSI components, design can only be accomplished using a hierarchical, systems-oriented approach aided by computer design systems, RÜCHARDT 1985 and DARBY & ORTON 1986.

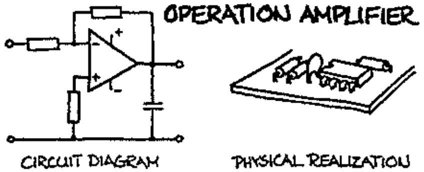


Figure 64 Circuit design and electronics packaging design: 2- and 3-dimensional thinking

Electronic components are primarily designed by specialized companies and they are to a very high degree standardized in component values and tolerances, functional specifications, mechanical packaging (e.g. DIL-packages, SMT-packages), temperature range etc.

Only in exceptional cases will the mechatronics designer need to design electronic components himself:

- When applying thick-film technology, the resistors, capacitors and inductors are produced directly in a printing technique.
- For production of custom specific chips (ASIC technology), IC semiconductor design is required.
- For sensor and actuator designs, special components may be required.

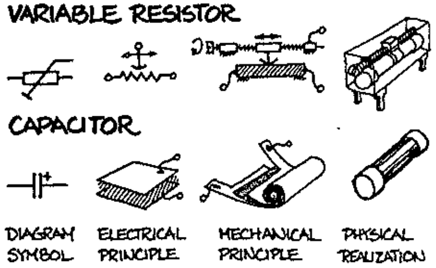


Figure 65 The structure of electronic components: electrical and mechanical functions

Circuit design

In electronics it is possible to design the function of a circuit almost independently of its physical realization, BECKER & NEUMANN 1983. The result has been a separation of the design activity into *circuit design* and *packaging design*, Figure 64. Some companies, especially in the USA, even prefer to separate the two activities in the organization, so that specially trained packaging engineers take over from the circuit designers and design the packaging system.

An electronic circuit is basically a 2-dimensional structure of functional elements connected by conducting relations. The elements correspond to discrete (single function) or integrated (complex or multiple function) standard components.

Circuits are seldom designed from scratch. Electronic engineers seem to share a large fund of knowledge on solution concepts for typical functions: amplifiers, oscillators, filters, A/D converters, microprocessor circuits etc. Such solution concepts are well-documented in teaching material, handbooks, technical journals and application notes of component manufactures.

The design of digital logic circuits differs from analogue circuit design, in that it is actually possible to derive the structure of logical gates (NAND, NOR etc.) from a mathematical description of the required function, VDI/VDE 1985.

Circuit design can be compared to the scheming of process plants and hydraulic systems: creating a new structure from existing elements. The most frequent recommendation in design literature is therefore to subdivide the task systematically into subproblems in a top-down manner, until the modules correspond to known solution types or to integrated components, SMITH 1983, COOKE 1984, HEATH 1986.

Electronic circuit design is mostly based on analysis and dimensioning methods. Once a diagram structure is suggested, then the performance can be calculated and often simulated in detail. The general approach is thus to suggest one solution proposal quickly for circuit analysis, and to modify that solution until specifications can be met, HEATH

1986. Methods for initially suggesting alternative concepts are not presented in literature.

There are a few factors which prevent the function of an electronic circuit from being completely independent from the physical realization (packaging) of the circuit diagram, for instance:

- The geometry of conductors limits power transfer and switching rates.
- Heat dissipation in a circuit depends totally on the mechanical structure.
- Electrical shielding is critical for proper microprocessor operation (EMC).
- Small dimensions cause feed-back and cross-talk between signal carriers.
- Manufacturing tolerances can add up to deviations from the stipulated function.

It should be noted that most of these aspects can be related to the energy content of signals, which is commonly ignored in electronic engineering education. They are problems in the interface between circuit designers on one side and packaging and mechanical designers on the other.

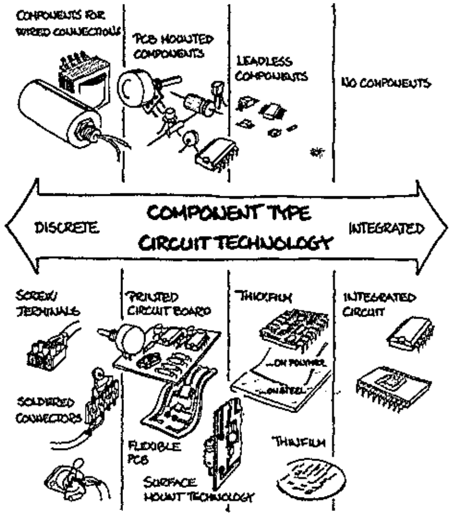


Figure 66 Electronics packaging design requires choice of component type and circuit technology (compiled by the author)

Electronics packaging design

The *packaging* of an electronic circuit is the manufacturable realization of the circuit diagram. It is a spatial structure of standard components and a specially designed 'connecting medium' like printed circuit boards (PCBs), thick film substrates or conventional wiring. The connecting medium contains all the interconnections between components (the functional relations) and will often mechanically fix the components in the spatial structure as well.

Typical parameters of electronics packaging design include the following, MATISOFF 1982:

- Mechanical design: Structure, shock and vibration control, manufacturability, repairability.
- Heat transfer design: internal and external heat transfer.
- Electrical design: circuit lay-out, interconnections, interference reduction.
- Industrial design: styling, human factors.

HARPER 1967 has described the design of electronics packaging in terms of two major decisions: a choice of *packaging technology* (component types and 'general packaging methods') and a choice of *circuit grouping* (functional and manufacturable modules).

The choice of packaging technology The number of available technologies is limited, at least when compared with mechanical manufacturing technologies. The choice of technology includes two aspects: A choice of component type and of the technology of the connecting media, see Figure 66. Component types (leaded or leadless etc.) are critical for the function of the circuit, and HARPER recommends the use of a checklist to ensure that all necessary information for making the choice is transferred from the circuit designer. The choice of circuit technology is based on the properties listed above rather than on functional aspects.

Choices of component type and circuit technology are naturally interrelated, as indicated in Figure 66, and in general solutions are of hybrid type, i.e. a combination of several technologies. A study of Japanese electronic products has indicated that Far Eastern companies are much more bold in choosing compound packaging than their European counterparts, whose engineers go for the beauty of a pure-bred, but often more expensive solution, JØRGENSEN 1986.

The choice of circuit grouping The modularisation of electronic circuits is very important for a range of activities: manufacturing, test, quality control, installation, repair, troubleshooting etc. On the other hand modularisation introduces less reliable, separable connections and higher manufacturing costs.

HARPER 1967 recommends a conscious choice of *plug-in level* and *throwaway level* for the design. The first refers to the levels of subassemblies which can be replaced without soldering. The second is the highest level of subunits which will not be repaired if service is needed. Figure 67 illustrates the two terms.

Electronics packaging design is similar to the detailed design phase of machine design. Both require spatial thinking and attention to manufacturing properties. In packaging design however, the uniqueness of a solution is caused by a clever combination of known techniques. In machine design, solutions often contain previously unseen principles.

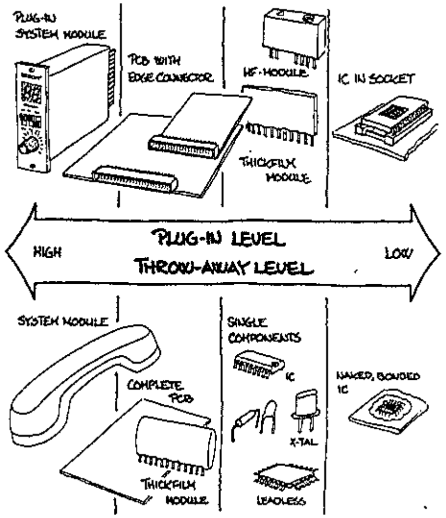


Figure 67 The 'plug-in level' and 'throwaway level' as parameters in circuit grouping (compiled by the author)

Some activities during the packaging design process are trivial and are for the most part already computerized, BALINT & HASLINGER 1980. The sophisticated lay-out of printed circuit boards or thick film substrates for instance, is hardly possible without computer aids.

Analogies of electro-mechanical systems

The well-developed set of theories for the behaviour of electronic circuits can be applied for the analysis of mechanical systems by establishing *dynamical analogies*, RASMUSSEN 1970. This is based on the fact that mechanical systems can be considered a network of components, which in a dynamical sense have functions analogous to the capacitor, inductor, and resistor etc. of an electronic circuit, see Figure 68.

An analogous network is a model which is established in a physical system different from the one to be examined. It is an analytical tool, which allows the establishment of:

- A mathematical model for calculating the dynamical functions of a system.
- A functional model using electronic standard components for measuring functions and response of the system.
- A computer model for simulating the dynamic functions of the system.

The technique is not limited to mechanical systems, but applies to acoustic, hydraulic, pneumatic, magnetic, biological and thermal systems as well. It is useful for the design of such systems, since the dynamic function can be optimised in a 'cheap' medium by calculation or electronic measurements, before the actual system is built.

From a mechatronics viewpoint, dynamic analogies are interesting for technology assignment in the class of designs in which dynamic properties are of great importance. By establishing an analogous network, it is possible to express the function of a mixed electromechanical system in one uniform

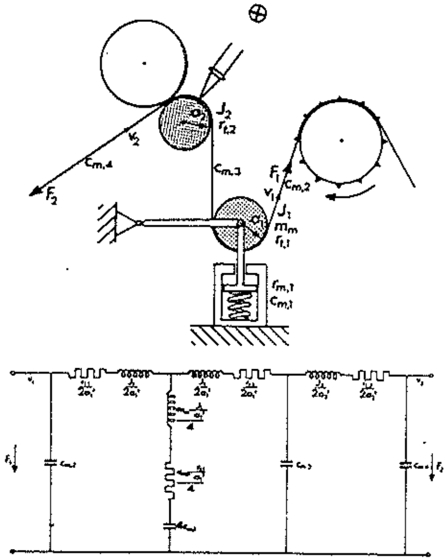


Figure 68 A film transportation mechanism and its analogous network, symbolizing the dynamic behaviour, RASMUSSEN 1970

language: The diagram symbols of electronic circuits, see Figure 69. It is possible for the designer to change the component values and add or remove components with the same effort, irrespective of the technological realization.

The technique of dynamic analogies is mainly useful for electrical engineers, who have learned the theories of electronic circuits, and who can imagine circuit modifications for improving the system.

To conclude this discussion of electronic systems design: The functional design and the physical realization (i.e. the basic and quantitative structure of organs) can be determined much more independently than in machine design.

Theory exists for describing the electronic functions in the sense of transformations, and for modelling the organic structure in diagram symbols, but theory for synthesis is very scarce. To some extent, machine design theory is applicable to electronic component design and to electronics packaging.

5.3 Software design

A computer program is a sequence of coded instructions for the processing unit of an electronic computer, specifying how to perform information transformations. It exists in an electronic memory medium either in a form which can be directly executed by the processor (machine code), or in the form of a higher level programming language, which can be automatically compiled to machine code by a standard program.

The term *software* is mostly used to cover more than just the computer program, BERSOFF 1984:

> "Software is information that is:
>
> - structured with logical and functional properties,
> - created and maintained in various form and representations during the life cycle;
> - tailored for machine processing in its fully developed state.
>
> So by our definition, software is not simply a set of computer programs, but includes the documentation required to define, develop, and maintain these programs."

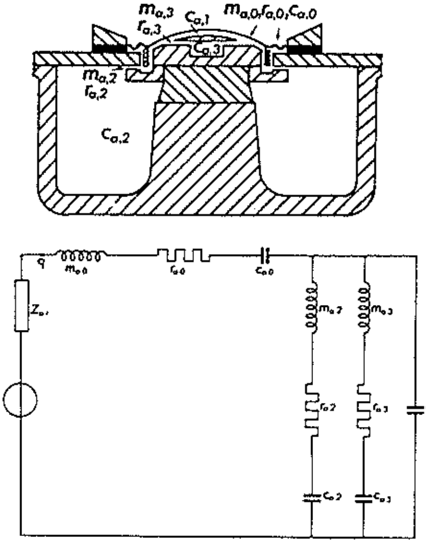


Figure 69 The analogous network for an electrodynamic microphone. Acoustics, mechanics and electrics expressed in a single model, RASMUSSEN 1970

Computer programs are immaterial. The magnetic disc and the IC memory are not programs in themselves, they are merely carriers of coded information. The program listing on paper does not have a function in itself, it is only a model of the program.

This is probably the reason why the term software has come to take on a wider meaning than just the computer program. To modify, test or transfer the program to a different hardware system is simply not possible, unless its structure and functions are thoroughly documented.

Software in itself is not capable of transforming information. The main purpose of software is to prescribe the types of transformations to be executed by the computer, and to control the sequence and timing of their execution.

The *types of transformation* are limited to a basic set of operations determined by the computer hardware (basic arithmetic and logic operations, data storage and retrieval). The processing units and the hardware architecture may be designed explicitly for special applications, for instance fast data handling for data management systems or fast multiplication for FFT signal processors.

Also the *sequence of operations* is limited to certain standardized patterns representing the possible logical conditions for execution (*if... then...; do... while...* etc.). Seven such elementary patterns are shown in Figure 70.

Those seven patterns are not dictated by electronic hardware, but based on conventions laid down in the *structured programming* strategy, which evolved following a 'software crisis' in the late 60's. The strategy suggested a number of principles for software design to improve testability and maintainability, and to reduce the sky-rocketing failure rates. The main principles (DIJKSTRA 1969) were to:

- Subdivide the program into small, testable modules.
- Design modules with only one entry and one exit.
- Avoid the use of *goto*-instructions.
- Permit a limited set of elementary logical sequence patterns only.

Data flow and sequence of operations

There are two concepts for describing the program function in software design, and each is supported by different design models or methods. One emphasizes the flow and transformation of data, the other emphasizes the sequential structure of operations.

In the first concept, software is regarded as a *system of transformation functions* related by the flow of data (information) to be transformed. The methods, which support this concept include the Structured Analysis and Design Technique (SADT), ROSS 1985, and Structured Analysis (SA) by Yourdon Inc, WARD & MELLOR 1985, see Figure 71 and compare with Figure 28. This perception of software comes very close to the transformation functional models of machines, as pointed out by BRUNTHALER 1985.

The transformation system has a dual representation, in which the data forms (stages) are regarded as system elements and the transformations are relations between elements. This representation is advantageous for the design of large data administration systems, but ROSS 1985 states that the application of *both* the data and function oriented models gives the most complete understanding of the system.

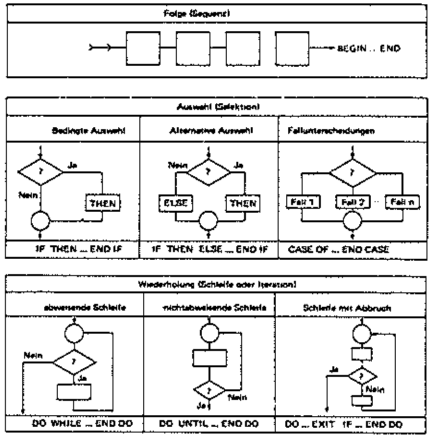


Figure 70 Seven elementary logical program sequences in flow-chart-symbols, HERING 1984

In the second concept, software is described as a *sequence of functions* to be executed by the computer in a prescribed order. The functions are operations on data (transformations and data storage/retrieval) or operations which control the sequence of execution. This perception of software is derived from the program listing and the computer's way of executing instructions. Typical models are program flow charts and the structograms of Nassi/Schneiderman, e.g. SNEED 1986.

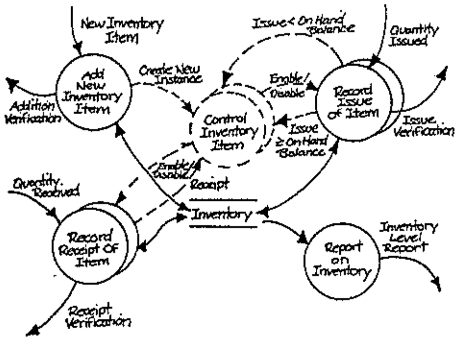


Figure 71 Software regarded as a system of transformations and data flow in the notation of WARD & MELLOR 1985 (SA-method). Note the control function

In both the flow and sequence oriented concepts, the computer program can be regarded as a *hierarchical structure* of modules and submodules. The methods which support this understanding include Jackson Structured Development (JSD, JSP) JACKSON 1983, the Structured Development (SD) of Yourdon, WARD & MELLOR 1985, and the Hierarchy plus Input Process Output method (HIPO), e.g. HERING 1984.

Any theory to support the hierarchical decomposition of the structure of transformations or the sequence of functions is not found in the literature.

Design properties of software

Many authors describe such properties as user friendliness, efficiency and reliability as important aspects of software quality, but in fact the computer program itself cannot possess such properties. They are a result of the combination of software and electronic hardware. So it does not make sense to compare software design alternatives on the basis of such qualities, unless the hardware has already been determined, for instance a standard personal computer with alphanumerical keyboard and CRT screen.

Mechatronic systems will usually be *embedded systems* i.e. systems where the environment and the functions can be rather precisely defined (this is not the case for a PC). In an embedded system, both hardware and software will often be unique, and programs cannot be transferred to different systems without modifications.

Some authors attempt to formulate hardware independent properties for computer programs, for instance *modularity*. When software is divided into independent modules, then testability, portability, ease of documentation etc. are improved. Yourdon's SD method recommends the evaluation of the *coupling* between modules and the *binding* within each module. A high degree of modularity is achieved by weak coupling and strong binding, WARD & MELLOR 1985.

Formalism in design specification

Due to the immaterial nature of computer programs, it is very difficult to specify complex software systems. This has lead to the development of formalized, mathematical methods for the formulation of program specifications. Such methods prevent the ambiguities and contradictions found in specifications based on natural language. The formalized methods require very abstract thinking and extensive education, and the issue has split software scientists into two different camps:

- The 'formalists' who believe formalism to be a necessary precondition for efficient software development, MEYER 1985.
- The 'anti-formalists' who find formalism harmful to design, because it makes the specification incomprehensible to both customers and designers, NAUR 1982.

Based on formal specification models, computer scientists work to develop design methods, like the Vienna Development Method (VDM), LØVENGREEN 1980, which allows the designer to prove that every development step is in agreement with the original specification. The designer still has to suggest and choose design alternatives at every step, but the verification can be computerized.

Fast prototyping

Another topic of discussion in software design circles is the application of early functional modelling in the design process. Fast prototyping offers the advantage of early identification of high-risk issues and provides a good basis for discussing design specifications with the users ('I'm not sure what I want, but I'll know when I see it').

Fast prototyping has been described as a method in opposition to the specification driven design approach. BOEHM ET AL 1984 have undertaken a multiteam design experiment to investigate the different characteristics of both approaches. They found that the prototyping approach yielded better human-machine interface designs with less effort than structured programming. Their conclusion was not unexpectedly that software design is not a question of either-or. Design specification activities and fast prototyping should complement each other to achieve the best designs.

The major issue here is really the missing visibility of computer programs. It requires a high level of abstract thinking to understand the function of software systems before they are actually implemented. This applies to program users, customers and design managers alike, and often to the designer himself too. Likewise it is difficult for outsiders to see any progress within the software design project until it is finished. BRYAN & SIEGEL 1984 strongly recommended *'making software visible'* through the use of standardized documents and design reviews, to motivate customers and management to participate in the design process.

Concurrent programming

In traditional computing systems, program instructions are executed in sequence, one by one. For some applications however - real-time control of technical systems being one -input does not arrive in sequence, and several inputs might require simultaneous action from the computer. This has forced the development of *parallel processing* systems and also methods for programming such systems, ANDREWS & SCHNEIDER 1981, RAVN ET AL 1986.

Physically, parallel processes are realized by multiple processing units, which operate in synchronization and share the same data files. For some simple applications, semi-parallel processing can be realized on microprocessors with an interrupt facility, i.e. with an organized way of breaking off a current program sequence, to execute more urgent instructions before turning back.

Methods for concurrent programming are becoming popular with companies designing embedded, real-time control for mechatronic systems.

It can be concluded that software design is based on theories describing transformation and sequence aspects of computer programs. The fact that software does handle logical relations, sequence and timing of functions, makes it very difficult to relate its design methods to machine design theory, which does not.

To conclude this review of design methodology for mechanics, electronics and software, there is a general tendency that the literature on design shows very little consensus on theories and methods, and it is mostly based on the personal design experiences of individual authors. There are no indications that such an agreement will be achieved in the near future.

The surge into CAD-systems development seems to unveil the lack of design theory in all three disciplines, and it motivates a stream of new contributions to the design theory field. Such contributions are however often entrapped in computer programs and not explicitly explained.

6 A NEW THEORETICAL BASIS

In this chapter I will present a theoretical foundation for the development of new methods for the synthesis of mechatronic systems. It has been my intention to achieve consistency in the theory by ensuring that all those aspects which set mechatronics apart from mechanical and electrical systems are discussed and related to the design activity.

In agreement with the findings in Section 3.3 and Appendix A, I will concentrate on issues related to the *functional interaction* of technologies in mechatronics. I will not deny that a study of spatial integration in mechatronics (particularly in transducer design and in electronics packaging in machines) could also lead to efficient design methodology, but this goes beyond the scope of the current research.

As in Chapter 4, the discussions here will be structured in two parts: The aspects that belong to a (rather) technology independent functional description of mechatronics in Section 6.1, and those aspects which deal with the choice of technological principles (technology allocation) in Section 6.2. In the final section, the discussions will be summarized in a set of axiomatic statements on the nature of mechatronic systems. Those principles which seem most important for the design activity will be formulated in seven theorems.

6.1 The functions of a mechatronic system

The most important issues here are the applicability of the purpose function concept in mechatronics, the role of state transitions in a functional framework, and the structure of information processing in the mechatronic system.

Transformation and purpose functions

A discussion of transformational versus purpose functional description of mechatronic systems really comes down to discussing whether the machine itself is transforming the process object involved, or whether it is merely providing the effects necessary to facilitate the main transformation, which is then somehow realised externally to the machine.

The concept of purpose functions was derived from observing material transforming machines, because there the distinction between the two concepts of function is evident. The effects created by the machine can be described almost independently of the transformation taking place.

One example is the fully automatic ECG electrode manufacturing machine illustrated in Figure 72. A basic layer of foam is cut, a label is attached, then a rivet, a ring and a

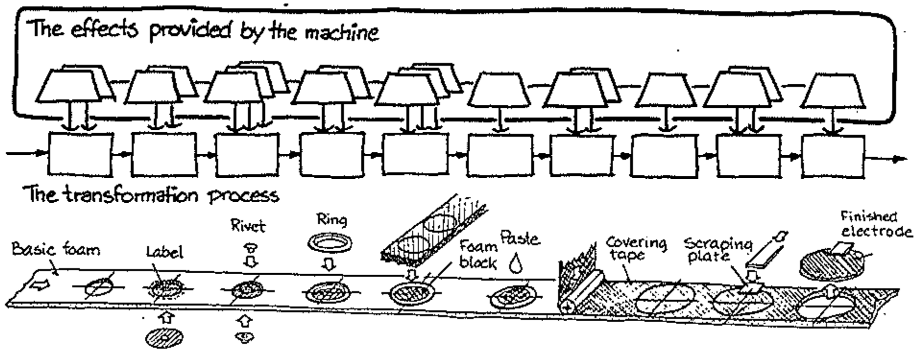


Figure 72 The manufacturing process of an ECG electrode, ANDREASEN & HEIN *1987. The purpose functions of the machine can be described independently of the transformation*

foam block are added and so on, all in a sequential pattern. The effects of the machine corresponding to those processes are to provide cutting effect and to generate a circular pattern, to establish position and provide attachment force for the label etc.

The relations shown in the figure between purpose functions are of causal type: All purpose functions are necessary to accomplish the required transformation.

For energy transforming machines the distinction between transformations and purpose functions is not so evident. A mechanical gear, for instance, transforms rotary energy from one speed of revolution to another. Here it becomes difficult to point out the effects which the machine exerts to facilitate the transformation.

Energy transforming processes are central to any mechatronic system, because information is mostly tied to energy. Electronic circuits for instance realize energy transformations (of electrical signals) only. The interpretation of purpose functions in energy transforming systems must therefore be examined more closely.

To do this, reasoning based on the analysis of well-known systems is necessary, i.e. one must ask which functions existing systems and components fulfill expressed in terms of transformations and exerted effects. From such reasoning, it may be possible to generalize findings which are applicable also to the synthesis of non-existing systems.

ANDREASEN 1980 favours distinguishing transformation functions (or processes) from purpose functions by strictly verbal formulation of verb/noun combinations.

Transformation functions are expressed in passive constructions of the form:

{object (noun)} is {transformed (verb)}

For instance for a turntable: *'record* is *rotated'*, the state of the object 'record' is changed from non-rotating (input) to rotating (output) in the process.

Purpose functions are expressed actively to denote the purpose of the machine:

to {activate (verb)} {effect (noun)}

For instance for rotating the record in the above example, the purpose function of the turn table is 'to *create rotation*', since 'rotation' is the effect exerted by the machine. The following table shows examples of components and the associated transformation and purpose functions:

Component	*Transformation*	*Purpose function*
Motor	electric energy is transf. into rotat.	create rotation
Gear	rot.energy changes revolution/torque	ensure suitable speed of revolution
Electronic amplifier	signal is amplified	ensure sufficient amplitude
Battery	energy is stored	provide power
Diode	AC signal is rectified	reject signals of neg. polarization

It is evident from the list that the same component may serve several different purpose functions, depending on the system it is part of. The purpose of the gear for instance could also be 'to ensure sufficient torque' or 'to ensure correct orientation of movement'.

When observing the purpose function and the transformation of individual components, we seem to formulate functions on different levels. The purpose relates to the effect, which the component provides to a system on a higher level, but the transformation relates to the object (material, energy), which is processed by the component itself. The purpose of the motor for instance is 'to create rotation' in order to facilitate a transformation of some object in a system, where in the motor is a component.

If we ask *'How* does the motor create rotation?', then the answer is 'By performing the transformation of electric energy into rotational energy'. This transformation process is only one out of a number of alternative ways to 'create rotation'. Another is 'By performing the transformation of potential energy into rotational energy' (a spring).

There is a causal hierarchical relation between purpose functions and transformations. When asking for the effects necessary for a transformation, we get purpose functions, and when asking, how the effects can be realized, we get transformation functions on a secondary level.

Also typical electronic (energy transforming) components can be described in purpose function terms. Such a description is not common for electrical engineers and will certainly appear alien to them, but the purpose function concept can help to clarify the hierarchical pattern of functions and alternative subsolutions in electronics design.

Is a purpose functional description of computer programs possible? In fact the definition of purpose functions (page 56) rules out this option. Software cannot in itself exert any effect on a technical process - only the *combination* of computer hardware and software may do that. The functional description of software is limited to the process domain, i.e. data transformation and state transition modelling.

State transitions and transformation functions

It was established in Chapter 4 that a full description of the functions of a mechatronic system cannot be accomplished when using transformation functions only. The mechatronic system works in different states (at minimum an *on-* and an *off*-state), and the function depends on the state of the system. Transitions between states are controlled by logical conditions (e.g. *if* switch is turned on by the operator, *then* the machine changes to its on-state).

There is general agreement in design literature on the definition of transformation functions:

> *Definition*
>
> The **transformation function** (or process) of a mechatronic system is the action which changes a process object (material, energy or information) from an input state to a desired output state.

There seem however to be two different interpretations of the transformation concept:

1. Transformations in which output has a 'continuous' quality, i.e. depends only on the state of input (e.g. conducting, amplifying, transducing).
2. Transformations in which output may take on several different states for the same input, depending on logical relations (e.g. coupling, storing).

When mixing the two interpretations in the same functional structure, the perception of states of the mechatronic system gets blurred. This is what happens, for instance, if one wants to read the function of an electronic circuit from the circuit diagram. For every switch, relay, zener diode etc. one has to imagine two states and deduce the corresponding changes in the overall circuit function.

I will argue here that transformation functional descriptions of mechatronic systems must be limited to interpretation (1), i.e. the continuous type of transformations. This description must then be supplemented by a model explicitly describing the states of the system and the transitions between states, for instance a Petri-net.

A simple example may illustrate this discussion, Figure 73. We want to design an intelligent system for room lighting. The system should only turn on the light, when persons are present in the room, and the lighting level should be adjusted according to the ambient daylight.

The system has one switching function and thus two states. If the function of the system is described in transformation terms only, then the output object of one or more blackboxes will exhibit those two states (light - no light). Instead, a 'continuous' transformation structure can be established for each state of the system, and the structure of states and transitions can be described separately.

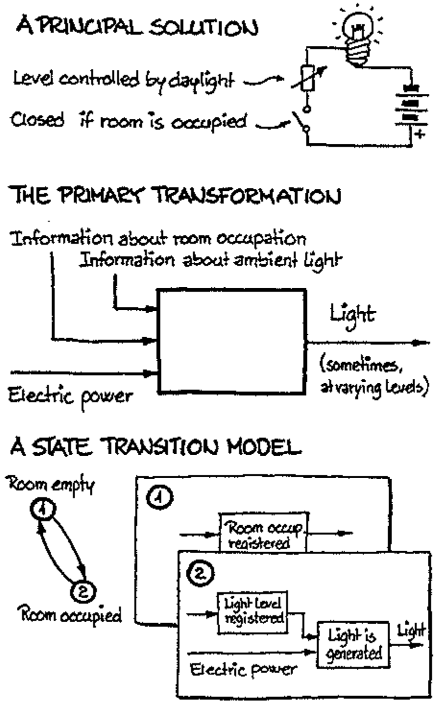


Figure 73 The functions of an intelligent lighting system. An example of the relations between a transformation functional model and a state transition model

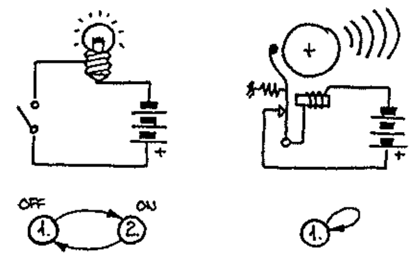


Figure 74 The function of an electric switch. A one- and a two-state system

To distinguish between continuous and multistate transformations is no easy task. The function of a switch has two states, but if the switch forms part of a pulsing relay, the function becomes continuous, Figure 74. It is rather a question of the scope of observation in a hierarchical structure of systems and subsystems. Since the multistate function is a direct cause of there being two states in the system, a system having multiple states may on a next higher level also be regarded as a stable one-state sub-system.

I will therefore define:

> *Definition*
>
> A transformation function belonging to the functional structure of a mechatronic system must be considered of **multistate type** if it causes the system output to change state momentarily due to an external logical input.

If a multistate function is present in a transformation functional structure, then it is possible to replace this function by a state transition structure plus a continuous transformation structure for each state. During the design of mechatronic systems, such an operation allows us to single out the logical functions of the systems.

State transitions and purpose functions

Another aspect that has to be clarified here is the relationship between the state transition point of view and the concept of purpose functions. A purpose function is the ability of the mechatronic system to create an effect needed to realize a desired transformation, and the sum of all necessary effects constitute the purpose of the system.

I will regard the functional structure as a *table of contents* of necessary effects (on the hierarchical level observed). If the system has multiple states, then a different subset of purpose functions will be required in each state. Figure 75 shows the purpose functional structure of the previous example of an intelligent lighting system. The two states of the system require each a set of purpose functions.

It should be noted that the function of the on/off switch, which causes the transition between the two states, will appear only as a *secondary control function* on a lower hierarchical level. It is dependent on the physical realization of the functions 'register room condition' and 'create light'.

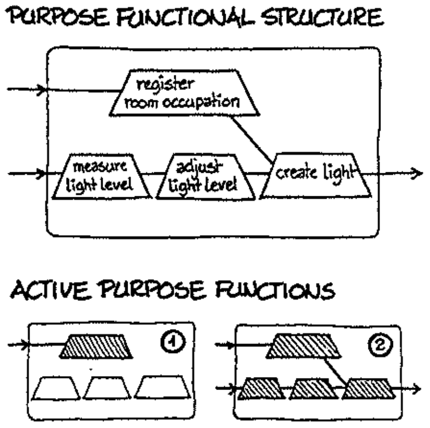


Figure 75 The purpose functional structure of an intelligent lighting system. Each state of the system requires a subset of 'active' purpose functions

It is only possible to a certain extent to illustrate the logical relations between functions by connecting lines. I believe that the main advantage of the purpose function structure is the total view of necessary effects in the system. Therefore it provides a good starting point for allocating technology, i.e. for suggesting solutions in the organ domain.

A second example of a functional description of a mechatronic system is shown in Figure 76. It is a telephone modelled in the process domain (transformation structures and state/transition diagram) and in the functional domain (purpose functional structure). The telephone has four states, of which one is idle.

The structure of information processes

The concept of information in mechatronics needs a more careful treatment than found in the mechatronics literature considered in Chapter 4. A good starting point is FRANKSEN 1984's discussion of the nature of *data:*

> **"In the sciences, measurements constitute the only link between reality and theory. Hence, since data are facts, they must originate in measurements"**

In his search for a theory of data, Franksen discusses two viewpoints: the *operational*, which ascribes meaning to measurements in accordance with the operations of measuring, and the *symbolic* viewpoint, in which measurement is the assignment of a numeral scale-value to an observed property.

> **"The crucial property of a datum in contradistinction to a measurement, is therefore that it captures this information (of the identity of the object or event) in addition to the numeral ascribed to the observable property."**

Like BARNEY 1985 (page 41), one could suggest that *information* is simply the same as data, i.e. that information is facts which *always* originate in measurements. This is clearly not the case, and it would contradict the statements of KRAUSE 1986 and KAJITANI 1989 that information is produced by and for man. Instead, I will consider data a subset of information and likewise accept that the message (*'Nachricht'*) exchanged between people is also a subset of information.

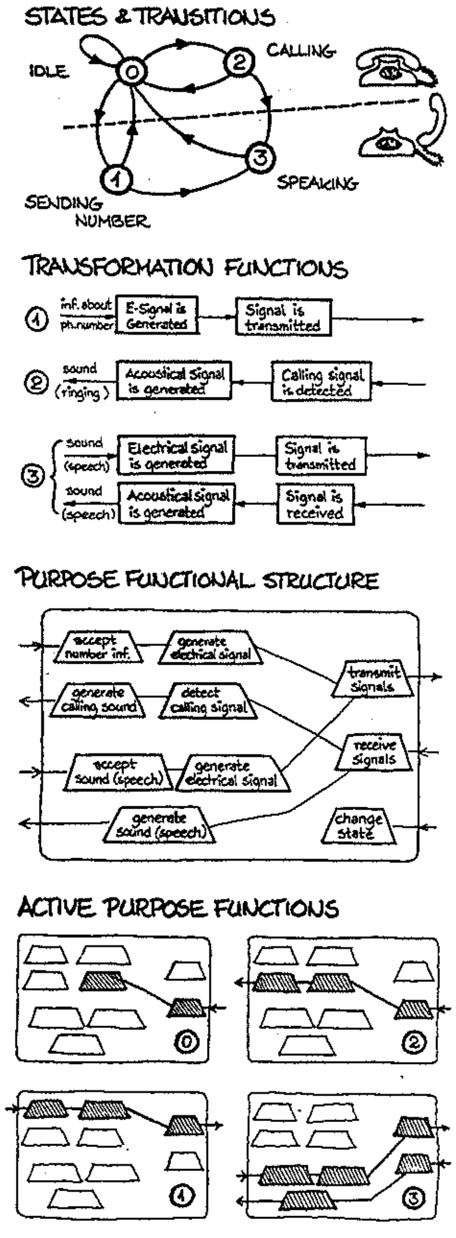


Figure 76 The functions of a telephone

To define information is hardly within the scope of this research. The definitions found in the literature (e.g. KLAUS 1968) mainly originate in probability theory (information is a measure of the uncertainty of the occurrence of an event) and information theory (information is a selection of characters transferred from a sender to a receiver).

The meaning adopted here is rather cybernetic (information is an independent category of transformation objects besides energy and material). Also of value here is the control theory interpretation, which distinguishes between numeric information (measured or reference values) and control information (instructions for necessary actions). Control theory is - as opposed to information theory - concerned with the semantics, the ’meaning’ of the treated information.

A mechatronic system basically handles two types of information:

1 *Process information*, which is transformed by the system regardless of its semantic value, and

2 *Control information*, which is applied by the system for controlling its function (i.e. which is ’understood’ by the system).

At any one level it is comparatively easy to distinguish between process and control information, but when observing the system at different hierarchical levels, confusion is bound to arise.

An electronic feed-back loop in a robot, for instance, clearly carries control information, since the purpose of the arrangement is to control the movements of the system. When observing the feed-back sensor and the signal conditioning circuits locally, however, the type of information treated has process character. As far as the sensor and pre-amplifiers are concerned, the semantic value of the processed information has no influence on their functions.

I will claim that in general, process and control information in a mechatronic system appear alternately in a hierarchical structure: Control information needs processing functions, and processing functions are likely to be governed by control information on the next lower level etc, see Figure 77.

6.2 Technology allocation

For the activity of establishing the organic structure of a mechatronic system, the most powerful principles are the Law of Vertical Causality and the Principle of the Secondary Function Complex, both borrowed from the field of machine systems theory (page 58). The first paragraphs of this section will verify the validity of those principles to mechatronics. After that, the basis for allocating technology will be examined closely.

Vertical causality in mechatronics

The Law of Vertical Causality states that there is a causal relationship between the (purpose) functions of the system, and the alternative means of realizing it. Once the means for a function on one level is found, it can be decomposed into subfunctions on a lower hierarchical level, each of which in turn requires its choice of means.

If the law applies to mechatronics, then we must expect means of purely mechanical, of purely electronical and of mixed principles on the same level of alternatives.

This is indeed the case. When analysing the principles utilized in mechatronic design, one will find several levels of functions, which could be realized in a very mechanical way, in a very electrical way, or as an integrated combination, Figure 78.

If the *purpose function* term of the vertical causal chain is interpreted more widely as a problem statement rather than an effect in the machine, then *means* may cover a transformation process, an organ or a parts structure. I will elaborate a little on this statement put forward by ANDREASEN 1980.

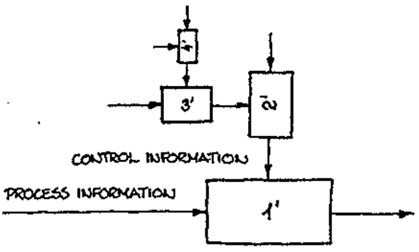


Figure 77 The process/control information hierarchy in mechatronic systems

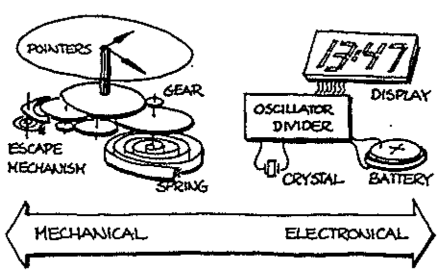


Figure 78 Different principles for realizing the function of a watch

A *technical process*, i.e. a sequence of transformations, which basically solves the problem, may appear as a means on the upper levels of the function/means hierarchy. An example of a process in a mechatronic system is: ’The line pattern on the paper original is read, and a similar pattern is printed on copy paper’ (electronic photocopier).

An *organ* can be understood as a category of physical entities, which exhibit similar working principles to realizes a required function. Examples of organs are: a motor, a mechanism, a battery, a display, a potentiometer. A microprocessor is in itself only an organ if it is supplied with a software program. Without it, the processor cannot realize any particular function. Software modules alone cannot be considered organs.

A *parts structure* is an assembly of those mechanical parts which constitute an organ, for instance for a potentiometer: The circular resistor path, the rotating contact pin, the axle bearing and the mounting base. Parts structures will appear as means close to the bottom of the function/means hierarchy, where the construction of the system is described in every detail.

When moving down through the causal chain, the function term will change from general problem statements at the top to specific descriptions of required effects and geometrical arrangements towards the bottom. At the same time, means cover technical processes, organs and parts structures successively. Figure 79 shows an example of the function/means tree for a telephone system.

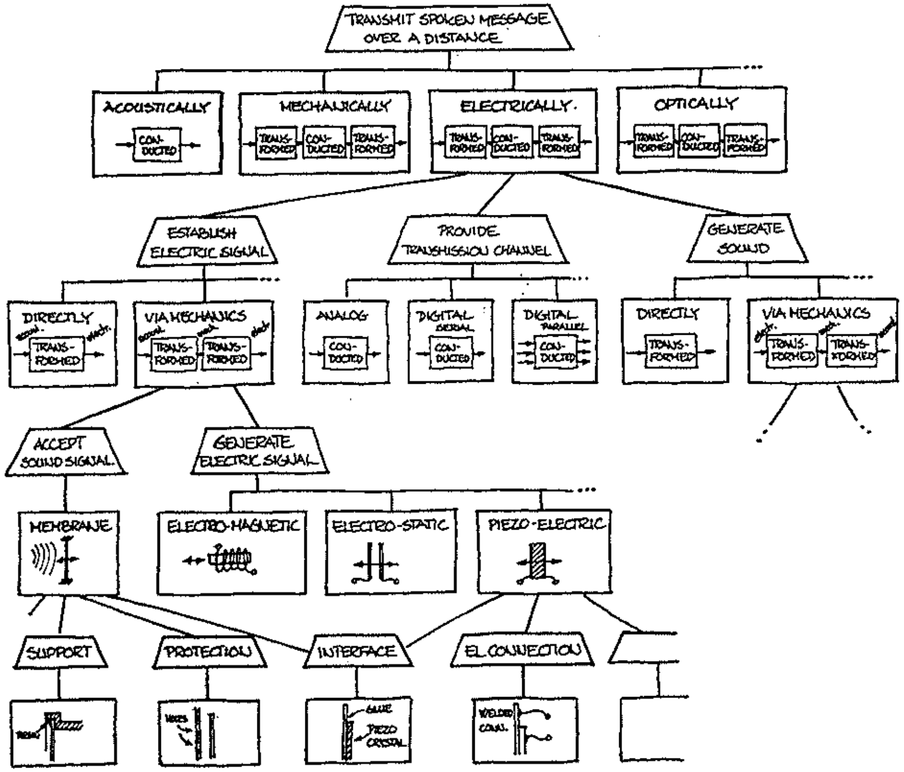


Figure 79 The function/means tree for a telephone system. The means change from processes to organs and part structures

The Law of Vertical Causality has one important consequence for the understanding of functional structures: When preparing a model of the functional structure of a system, only one level of partial functions from the function/means hierarchy can be included in the structure model. This observation is illustrated in Figure 80.

It is not possible to include lower level subfunctions, since they require that the choice of means to realize functions on the level above has already been made. If higher level functions are described in the functional structure, then the fact that their means have been already selected, will be disguised.

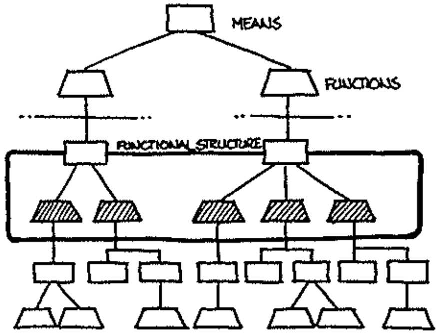


Figure 80 A functional structure model cannot contain more than one level of the function/means hierarchy

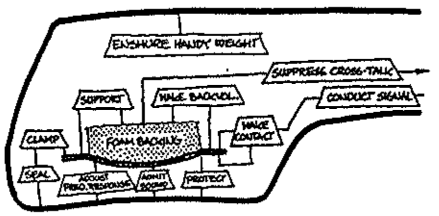


Figure 81 The functional structure for a telephone handset. The transducer principle (curved PVDF membrane) and the form of the casing have been chosen at this level

When applying functional structures for concept design, I think the preconditions for each structure diagram must be stated clearly, i.e. the decisions on means made up to this point. An example of such a structure model with clear-cut preconditions is shown in Figure 81, BUUR 1984.

The complex of secondary functions

The system models proposed by RÖHRS 1980 and KAJITANI 1986 (see page 38) both support the idea that there is a general set of secondary functions to be found in the vicinity of any main function.

The Complex of Secondary Functions as formulated by HUBKA 1973 in his machine systems theory, is basically sound also for mechatronic systems, but not fully adequate. As an appropriate set of secondary functions, I have chosen:

power function
control function
interface function
protection function
communication function
structural (or support) function

The rather vague term 'assisting function' from HUBKA 1973 has been omitted, since it is most likely to be identical to either the interface, the protection or the communication function. I will briefly discuss each function in the following.

The *power function* provides energy for the primary means, if energy supply is required.

The *control function* governs the state of the means. In the simplest case it may just be an on/off control. It also controls the functional performance in accordance with external inputs, e.g. a feed-back loop.

The *interface function* is needed, if the main inputs or outputs do not directly fit the environment of the primary means. If an electromotor was chosen to power a linear moving mechanism, some means of converting rotary motion into linear is necessary.

The *protection function* ensures that the functional parameters of the primary means are kept within a permissible range. The fan of an overhead projector for instance, protects the lamp from overheating. In addition, a second type of protection function prevents the means from exerting unacceptable impact on the system environment.

The *communication function* permits the means to exchange (status-) information with its surroundings. A laser system may for instance give a warning signal that it is working (output), and the system may function according to some reference settings (input). In general, the communication function will mostly provide information to a control loop on a higher level.

The *structural function* ensures that spatial conditions are satisfied to make the primary means work. The socket of an electric lamp, for instance, has a structural purpose, but it also provides an electrical interface.

The principle does not claim that *all* the secondary functions of the complex are required for the realization of any function of the system, only that some or all of the set *are likely*. The aspect of recursivity is important: the secondary functions of the complex can themselves be regarded as primary functions on the next lower level of the hierarchy, each requiring a new complex of secondary functions. So the power function of a higher level means may for instance need a control function, which again may require power.

By applying purpose functional terms for formulating the complex of secondary functions, one circumvents the problem that for instance the structural function cannot be described in transformation language (see Figure 23), and that the control and communication functions are intertwined in an input/output sense (see Figure 22).

The control function occupies a special position in mechatronic systems, since it is often realized by a (multifunctional) microprocessor with software. Two important statements can be derived from the Law of Vertical Causality and the Principle of the Secondary Function Complex:

1 Control functions are secondary functions, always depending on the choice of means to realize a primary function.

2 Control functions - though often realized centrally in the same computer - belong to different levels of the function/means hierarchy.

In other words: It does not make sense to discuss the control function until the means to realize the function to be controlled has been decided on. And the interrelations between different control tasks in the system quickly become complex, because they connect controls on different hierarchical levels.

The example in Figure 83 explains the control functions of the on-off switch of an overhead projector.

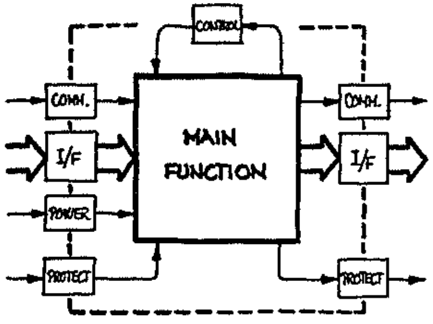


Figure 82 A system model for mechatronics, which complies with the principles of Vertical Causality and the Complex of Secondary Functions

A system model for mechatronics

Is it possible to suggest an elementary transformation functional structure which has general validity for mechatronic systems? If so, then it must comply with the two principles discussed previously:

1 It should include all the functions of the complex of secondary functions.

2 Full recursivity corresponding to the levels of the function/means causal chain should be maintainable.

A suggestion for such a system model is shown in Figure 82. I have placed all the secondary functions on the systems boundary to indicate that they may be shared with connecting systems. In a process chain of systems for instance, there will only be one interface function between each pair of systems.

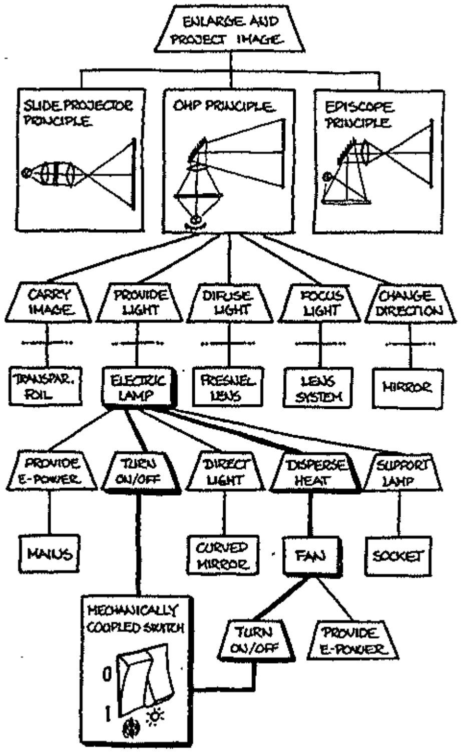


Figure 83 The function/means tree for an overhead projector: Two control functions on different hierarchical levels interact in a simple mechanical control organ: an electric switch

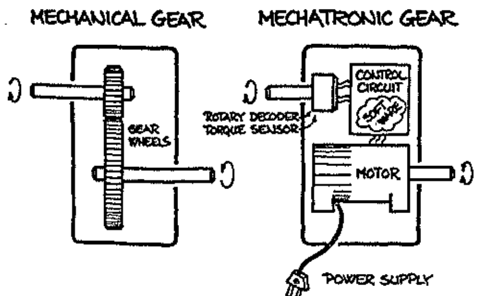


Figure 84 Comparing the principle of an imaginary mechatronic gear-reducer with a traditional mechanical gearbox

The structural function has not been included in this model, since it cannot readily be expressed in transformation terminology. Likewise, the formulation of the protection function as an input/output relation may limit the understanding of its full scope. For instance the function of a mechanical safety pin which prevents a mechanism from moving out of the safe working region, can hardly be described in transformational terms.

When comparing the system models suggested by RÖHRS 1980 and KAJITANI 1989 (page 38) with this one, it becomes clear that they each emphasize certain aspects and omit others.

The main advantage of such a system model is its value as a pedagogic tool. It allows us to explain the Complex of Secondary Functions and the structure of input/output relations between primary and secondary functions.

To apply the system model as a design tool, i.e. for abstractly describing the functional structure of a particular artefact, is hardly expedient. When several blocks are combined, the relations between them will quickly become too complex to allow easy sketching on paper.

Realizing control functions

First, let us conduct an experiment of thought: we will design a mechatronic system to replace a traditional, mechanical gear-reducer, Figure 84. The gear wheels are replaced by a sensor and an electric motor connected by electronic control. The movements of the input and output shafts are completely independent except for the control system. The energy transmission properties of the mechanical gear are lost in this design, so external power is required.

To achieve a shaft speed and torque conversion similar to the mechanical gear, we need computer software to control the motor in accordance with the signals from the rotary decoder and torque sensor. This software represents the logical relations which in the mechanical gear have been designed into the structure by its designer.

In the mechatronic system, the control function, which realizes these logical relations, can be totally separated from the geometry of mechanical parts and accomplished in a different media altogether. It is important to note that the logical relations do not suddenly appear because we introduce a computer into the system. Even in a purely mechanical system, we will find a kind of *mechanical software.*

The control function can be realized in mechanics, in electronic logic, or in computer programs. The main differences are the ease with which complex logical relations can be designed, and the flexibility of making changes in the design.

The mechatronic gear will be *programmable* in the sense that the speed and torque conversion ratios, the direction of rotation etc. can be altered independently by modifying only the computer program. Programmability is not a property which depends on the presence of an electronic computer. The street organs of older times, for instance, would claim programmability, since the music could be changed just by altering the hole pattern of a paper program. Electronic programmable logic devices (PLDs) contain no computer, but their logical structure of gates can be decided by the designer after purchase.

I will define programmability as follows:

> *Definition*
>
> A mechatronic system is **programmable**, if logical instructions for the execution of functions exist in a separate medium which can be altered without re-manufacturing the parts structure.

So in conclusion, the control functions of a mechatronic system can be realized in mechanics, electronics or software, and they can be programmable or not, but naturally different properties are achieved by different technologies in terms of complexity, design effort, costs, flexibility etc.

But also the operator of the mechatronic system is required to carry out control functions for the system to serve its purpose. Examples of a current shift in the allocation of control functions between operator and system are CNC machines, electronic sewing machines and auto-focus cameras, where the system takes over part of the control tasks previously performed by the operator.

The sequence and logic of operation, as described for instance in the operation manual of the system, must be considered *software of operation* from a design viewpoint.

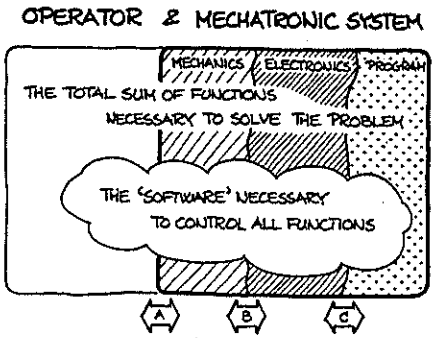


Figure 85 The allocation problem in mechatronic systems. Interfaces between man/machine (A), mechanics/electronics (B), and electronics/program (C)

The choice of technology and principle

As indicated in the discussion in Section 4.3, technology allocation in mechatronics design has two important aspects:

1 The general choice of an appropriate technology (mechanics, electronics or software) for each function of the system.

2 The choice of suitable working principles (organs) to realize each function.

For some designs the synthesis of principles can be made directly, which means that the first choice becomes obsolete. For others, an initial choice of technology is justified. In this paragraph, we will investigate whether there are any general rules for technology allocation to be derived from a systems understanding of mechatronics.

When we assume that both working functions and control functions can (at least theoretically) be realized in any of the three technologies mechanics, electronics and software, and that furthermore the operator of the mechatronic system performs both working functions and control functions, then we may draw a rough model of the technology allocation problem in mechatronics, see Figure 85.

The allocation of technology inside the system and the man/machine allocation is a complex activity, which depends on the particular application area and on the technological state-of-the-art. Nevertheless, we can derive useful knowledge by studying the characteristics of those interfaces indicated in the figure.

The man/machine interface mostly conveys information in both directions, but energy input may also be required for some systems. We know that information may be categorized in process and control information, and that it must be tied to either energy or material. This could provide the basis for categorizing the type of organs, which constitute the man/machine interface. A catalogue of displays and controls for electronic systems presented by VDI/VDE 1985 represents this line of thought. Another starting point for systematising the interface organs is the capability of man to perceive information by means of his five senses and to convey information/exert control by hands, voice etc.

The environmental interface between the system and its environment is more difficult to characterize, since both material, energy and information can be exchanged, see Figure 24.

The systems interface between two mechatronic systems is provided either electrically (i.e. via energy or signals) with appropriate connecting organs, or mechanically (material, energy or information) by means of coupling type organs. For electronic signal interfaces, a communication standard (protocol, common set of characters) is necessary.

The electro/mechanical interface internally in the system is provided by only two types of interface organs, as stated by HEINZL 1984: Sensors for information retrieval and actuators for controlling or powering the mechanical process.

The electronic hardware/software interface conveys information only. It is important to note that the computer programs are totally 'embedded' in hardware, i.e. there are physically no other interfaces to the program. An account of this interface was given by SAUPE & LÄMMEL 1986 in Figure 42.

The analog/digital interface is a special type of interface rendered necessary by the working principle of the digital computer and by the fact that most physical properties are analog in nature. One could suspect that this is a strictly internal electronic/electronic interface, but this is not the case. One will find examples of designs where the analog/digital conversion is realized mechanically, for instance in pressure sensors and even microphones, which due to their membrane design give a digital signal directly. Another example is rotary encoders.

The result of the allocation activity is the organic structure depicting those principles in mechanics, electronics and software, which will realize the functions of the system, *and* a description of the expected behaviour of the organs (software) and of the operator.

It is convenient to apply the term *design concept* for such a principal solution:

> *Definition*
>
> The **design concept** of a mechatronic system is a principal solution, characterized by:
>
> 1 The structure of those organs, which realize the most important functions.
>
> 2 The structure of interface organs, which define the system boundaries and the borders between mechanical, electronic and software subsystems.
>
> 3 The activity structure describing the expected operator behaviour and software for programmable organs.

In the organic structure, the relations between organs are of functional ('Kopplungen') and spatial ('Anordnungen') kind. Therefore we can establish two types of organic structures, which focus on functional and spatial properties respectively (the *basic* and *quantified structures* in the terms of TJALVE 1979).

The design concept should not be confused with the *product concept*, which is a term often applied in the early phases of a product development project. Besides the technical idea of the product (i.e. the design concept), the product concept must also include basic assumptions as to market needs, sales channels, production methods, financing etc.

6.3 Axioms of a mechatronic systems theory

I will sum up the considerations of the previous sections in a set of axiomatic statements describing general characteristics of mechatronic systems. The axioms will form the basis for discussing design models and methods for mechatronics in Chapter 7.

Axioms are statements which are accepted as true without proof, so there is no way of scientifically proving that the set is sufficient and complete. The main criteria for suggesting this set of axioms are that:

1. Each axiom expresses fundamental characteristics or relations between characteristics of mechatronic systems.
2. The axioms correspond to observations made by experienced designers.
3. The terms which are treated in the axioms have been previously defined or are generally understood.
4. The set of axioms is logically consistent and does not show ambiguities.

Axiom 1

A mechatronic system complies with the Theory of Domains, i.e. it can be regarded as a system of transformation functions, of purpose functions, of organs, and of parts.

Axiom 2

Transformation functions and purpose functions can be distinguished by strictly verbal formulation of verb/noun combinations.

Axiom 3

There is a causal relationship between transformation functions and purpose functions: A transformation requires different effects (purpose functions) from the system, and an effect can be realized by alternative transformation functions on a secondary level.

Axiom 4

The function of a mechatronic system depends on the state of the system. Every system has a minimum of two states (an *on*- and an *off*-state).

Axiom 5

The transition from one state to another is caused by logical inputs which are external to the mechatronic system.

Axiom 6

A subsystem with more than one state may be regarded as a one-state system on a next higher level, if the external input causing the change of state has become internal.

Axiom 7

Two types of transformation functions can be distinguished: continuous and multistate types. The multistate type is characterized by its external control input.

Axiom 8

A structure of transformation functions has multiple states, if at least one of its elements is of multistate type.

Axiom 9

Each state of a mechatronic system defines one particular structure of (continuous) transformation functions.

Axiom 10

Each state of the system requires a different set of effects in the total structure of purpose functions.

Axiom 11

A mechatronic system handles two kinds of information:

1. *process information*, which is treated regardless of its semantic value, and
2. *control information*, which is directly applied ('understood') by the system.

Axiom 12

Control functions constitute the logical relations between the purpose functions of the mechatronic system.

Axiom 13

Control functions are secondary functions, which always depend on the choice of means to realize a primary function.

Axiom 14

Both working functions and control functions in a mechatronic system can be realized in alternative combinations of mechanical, electronic, and information technologies.

Axiom 15

There is a causal relationship between purpose functions and organs: A function can be realized by alternative organs, and each organ will in turn require purpose functions on a secondary level.

Axiom 16

A mechatronic system is characterized by those organs, which provide the external interfaces between the system and its environments.

Theorems for mechatronics design

Based on the axioms, I will suggest a set of theorems which are directly applicable to mechatronic systems design. The two first are adopted from HUBKA 1973 and ANDREASEN 1980 and extended to mechatronics.

Theorem 1: Vertical causality

The structure of a mechatronic system is determined by a causal chain of alternating (purpose) functions and means, which realize the functions. Means denote transformation processes, organs and parts.

Every level of means shows alternative solutions, so from a design viewpoint, the causal chain is a hierarchical tree structure (the *function/means pattern*).

Theorem 2: Complex of secondary functions

For the realization of any function, some or all of the following set of secondary functions are required simultaneously:

power function
control function
interface function
protection function
communication function
structural (or support) function

Theorem 3: Control/process inform. hierarchy

In a mechatronic system, process and control information is transformed alternately in a hierarchical pattern of systems and subsystems.

Theorem 4: Substituting multistate functions

If a transformation functional structure representing a mechatronic system includes one or more multistate elements, then it is possible to substitute these functions by a state transition structure and a continuous transformation structure for each state of the system.

Theorem 5: Man/machine allocation

In a man/machine system, the total set of both working and control functions necessary to achieve the purpose must be divided between the mechatronic system and its operator.

Theorem 6: Technology allocation

Internally in the mechatronic system, the split between functions realized by mechanical, electronic and software means is specified by interface organs.

Theorem 7: Control organs

The state transition behaviour of a mechatronic system is determined by the structure of control organs and their programmable instructions (activity structure). Control organs realize the logical relations between purpose functions on the same and on different levels of the causal chain of functions and means.

7. DESIGN TOOLS: MODELS, METHODS AND PRINCIPLES

The purpose of this chapter is to discuss tools, which support the conceptual design of mechatronics. Some of these are tools which already exist in the fields of mechanics, electronics and software, but which have been extended or combined to fit mechatronics, and brought into the context of the theory of Chapter 6 and the Theory of Domains.

In addition I will indicate possible directions for the development of new methodology in response to the need for a more structured mechatronics design process.

The chapter will first present various types of mechatronic design models since a set of models is a precondition for devising new methods (a method is an instruction for how to propagate from one model to another). The second section describes design methods, and the final section explains how design principles can be utilized to generate concept alternatives in mechatronics.

7.1 Design models

All the examples of design models discussed in this section represent 'produktdarstellende Modelle' in the sense of ROTH 1982, i.e. they are models of the (imaginary) product or of product aspects (the functional structure, the organic structure etc.), not of the design process.

We are looking for design models for the synthesis activity, so in a model morphological sense (see Appendix B) the task of modelling can be described as follows:

- *The object* of the models is the process structure, the functional structure, the organic structure, or the parts structure of the product. We also want to take into account the behaviour of the system, i.e. the state transition structure.
- *The properties* we want to model are typically: function, ease of operation, appearance, design costs and development time requirements.
- *The purpose* of synthesis models is mostly to generate and describe new design concept ideas, and to evaluate roughly the consequences.
- *The user* of such synthesis models will be the designer himself, colleagues within the project team and sometimes 'outsiders' from management, manufacturing, marketing etc. It is important to note that, in mechatronics, even colleagues from the same team must be regarded as 'non-specialists' in fields other than their major one (mechanics, electronics, software).

I will base this study on the assumption put forward in the Theory of Domains that design models can be characterized according to their level of abstraction and the number of details they reproduce. In other words: All design models belonging to the same domain can be positioned in a two-dimensional plane, as in Figure 86. Here, since it will be impossible to list all design model types available, the four corners are of chief interest, because a discussion of the extremes in terms of detail and abstraction will clarify the understanding of the two axes.

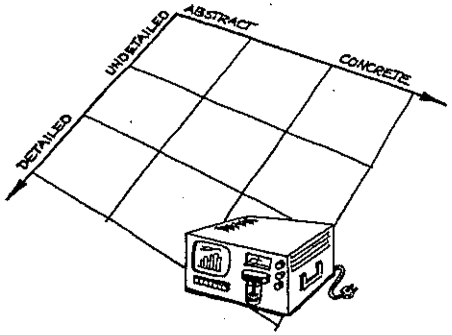


Figure 86 Design models can be systematized within a two-dimensional plane according to their concreteness and number of details

In every domain the near corner with the most concrete and detailed design model is occupied by the product itself. Depending on our viewpoint, we can regard the product as a structure of parts, of organs, of purpose functions, or of transformation processes.

Here, I will concentrate on the process, function and organ domains, and discuss the design models belonging to each domain in a separate paragraph.

The transformation structure and state transitions

In the previous chapter it was suggested that the process structure and the state transition structure of a mechatronic system complement each other and therefore should be described separately, as in Figure 76. The logical dependencies between subprocesses can be expressed in a number of different model types, according to which aspects are in focus. In Figure 87 some examples illustrate the following four aspects:

1 *States and transitions*, focusing on the states of the system and the conditions for changing from one state to another. Typical model types: State/transition diagram, Petri net.

2 *Sequential procedure*, where the execution of operations one by one is emphasized: Flow chart, structogram.

3 *Hierarchical pattern*, where the structure of subordinate levels of processes is important: Jackson diagram.

4 *Timing conditions*, where the timing of parallel transformations is critical: Timing diagram, event score.

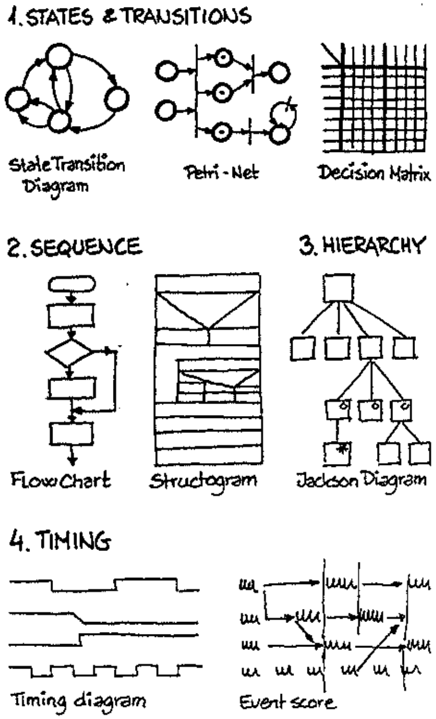


Figure 87 Different model types expressing the logical dependencies between transformation functions of a system

These model types are all roughly on the same level of abstraction, because they are all diagrammatic structures with verbally formulated operations and conditions. They may differ very much in their abilities to express details, however. The flow chart, for instance, is very suitable for expressing detailed, one-dimensional sequences, whereas the state transition diagram gets quite incomprehensible if the number of states is increased above a dozen. In the hierarchical diagrams, the number of details expressed is mostly a question of how many levels of the tree structure are included.

The philosophy of keeping the flow and sequence aspects separate in different model types is realized in several design methods for software development, for instance Yourdon's SA/SD method, WARD & MELLOR 1985, (data flow diagram, state transition diagram, hierarchical structure cart) and the HIPO method (hierarchical table of contents, input process output diagram).

A second philosophy is to integrate the flow and sequence aspects consciously into the same model, and thus emphasize the flow of control information. Here it is basically assumed that *all* transformation processes are to be controlled and thus in need of control information. Two examples will be mentioned here: the SADT method for software design, ROSS 1985, and a model developed specially for mechatronics design at my home institute, BERG ET AL 1989.

In the data flow diagram of the Structured Analysis and Design Technique (SADT) every subprocess may have separate control data inputs, which govern the transformation of the main data flow, Figure 88. The diagram emphasizes the structure of data flows, and the interaction of control and process information can be clearly observed. To obtain an overview of the primary transformations is much more difficult, since the levels of superior and inferior processes are not easily recognizable.

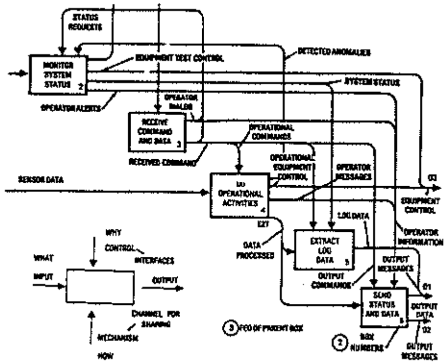


Figure 88 An example of a SADT diagram of data flow. Control inputs are on the upper edge of each black-box

In the SADT-model the influence of each control input on the execution of a subprocess is not explicitly shown. The model only shows that control data are present, and where they originate from.

The second model type is a little more detailed about the control aspects. It is shown in Figure 89. Each basic element in this model consists of a transformation and

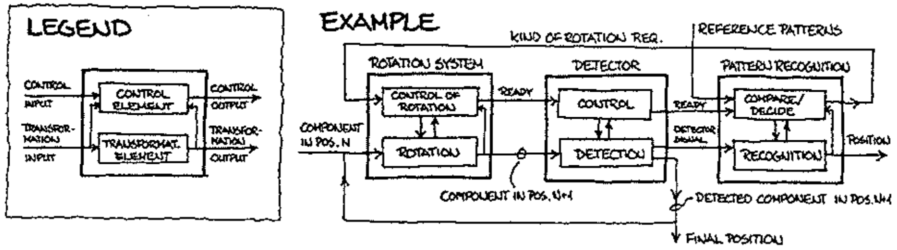


Figure 89 An elementary process model for mechatronic systems suggested by BERG ET AL 1989. The example shows a component feeding system

a control process, and thus they have two inputs and two outputs, one for the process object (material, energy or information), and one for control information. Inside each block, the interaction between control and transformation is symbolized. The control part may take both inputs and outputs of the process as a reference for the control task.

This model is more explicit about the influence of control on each subprocess of the system, and it forces the designer to think carefully about control information flows, but it does violate the Law of Vertical Causality by bringing the primary function and its control function on the same level.

On a more concrete level, models in the process domain include simulations of the process output and of sequences of operations. Such models are concerned with what happens to the process object itself rather than which effects may cause the transformations. When a medical engineering company simulates the display of a new piece of brain tomographic equipment on a computer, in order to discuss the necessary reproduction quality with doctors, then this is a process output simulation model. A similar type of model is found in the hearing aid industry, where designers try to determine by listening tests how the amplified signal should adapt to the individual characteristics of hearing losses. Both examples are models which are concrete enough for non-specialist users to participate in an evaluation process.

To conclude the discussion of design models in the process domain, Figure 90 indicates how the examples discussed are distributed in the plane of models.

The purpose functional structure

Especially the abstract type of models in the function domain is poorly represented in the literature, because the purpose function concept as such is more or less unknown. The symbolics used in this thesis for functional structures are adapted from ANDREASEN 1980 and HUBKA 1984. For examples, please see Figure 55 and Figure 76.

Functional models on a more concrete level are much more common, for instance experimental set-ups, fast prototyping of software programs, simulation models of the user interface, and prototypes. In functional models, the organs may differ from the ones chosen later in the design process.

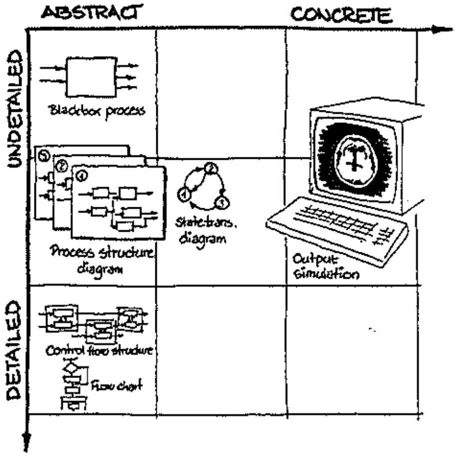


Figure 90 Mechatronic design models in the process domain

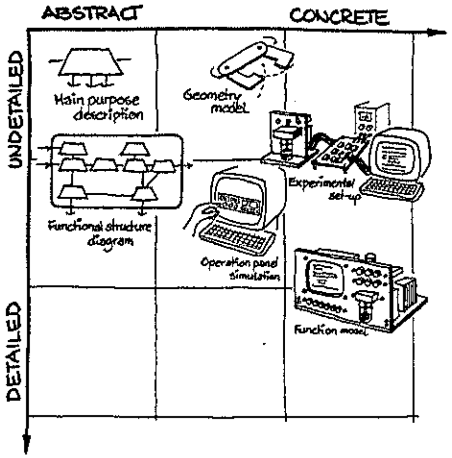


Figure 91 Mechatronic design models in the function domain

The organic structure

The organ domain invites the use of symbols, particularly in electronics, where the number of available component types is quite limited. Symbolic models range from very abstract to quite concrete representations of mechatronic systems. Some examples will be introduced here.

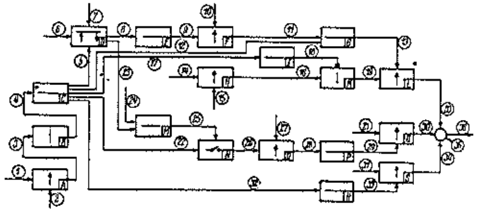
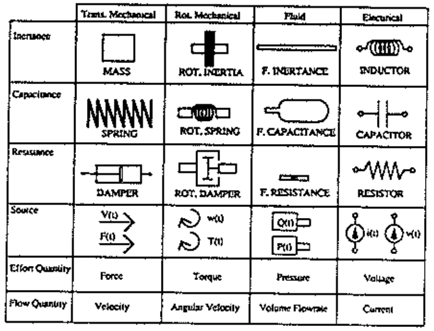


Figure 93 A symbolic model of the organs of a sewing machine, RÉGNAULT 1976

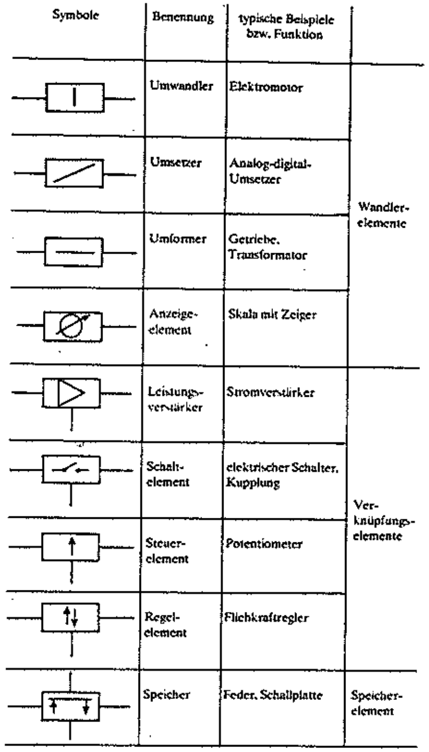
On the very abstract side, RÉGNAULT 1976 has suggested 9 basic symbols of organs for describing mechatronic systems (Geräte der Feinwerktechnik), Figure 92. His diagrams are based on direct mappings between organs and transformation functions, i.e. every organ is assumed to have one particular function, and the diagrams depict functional relations (couplings) between organs only, with no geometrical relations.

An application is shown in Figure 93. RÉGNAULT 1976 admits that his symbolic language can probably not be applied for the *synthesis* of new designs, but claims that it is suitable for analyzing existing systems, for explaining their structure, and for suggesting modified designs.



	Trans. Mechanical	Rot. Mechanical	Fluid	Electrical
Inertance	MASS	ROT. INERTIA	F. INERTANCE	INDUCTOR
Capacitance	SPRING	ROT. SPRING	F. CAPACITANCE	CAPACITOR
Resistance	DAMPER	ROT. DAMPER	F. RESISTANCE	RESISTOR
Source	V(t) F(t)	w(t) T(t)	Q(t) P(t)	i(t) v(t)
Effort Quantity	Force	Torque	Pressure	Voltage
Flow Quantity	Velocity	Angular Velocity	Volume Flowrate	Current

Figure 94 Symbols of organs used for modelling dynamic behaviour in four different technological areas, ULRICH & SEERING 1989



Symbole	Benennung	typische Beispiele bzw. Funktion	
	Umwandler	Elektromotor	Wandler-elemente
	Umsetzer	Analog-digital-Umsetzer	
	Umformer	Getriebe, Transformator	
	Anzeige-element	Skala mit Zeiger	
	Leistungs-verstärker	Stromverstärker	Ver-knüpfungs-elemente
	Schalt-element	elektrischer Schalter, Kupplung	
	Steuer-element	Potentiometer	
	Regel-element	Fliehkraftregler	
	Speicher	Feder, Schallplatte	Speicher-element

Figure 92 Symbols of 9 types of organs and their functional relations, RÉGNAULT 1976

When considering the dynamic behaviour of mechatronic systems, it is possible to apply an equivalent set of equations in the electronic, mechanical, fluid-mechanical and acoustic media, as discussed briefly in the paragraph on analogies in Chapter 5. The system is considered a network of idealized elements, which can easily be described using symbols, as in Figure 94.

Moving towards a more concrete level of models in the organ domain, two examples

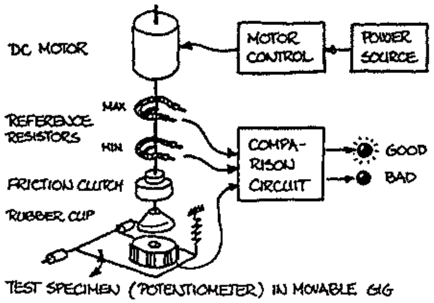


Figure 95 Pictograms for mechanics and blackboxes for electronics combined

can be found in Figure 31. One of them is a black-box diagram with each organ described verbally and related through arrows symbolizing flows of energy and information. The other applies pictograms of organs and indicates both functional and geometrical relations.

Mechanical engineers do not have a set of standardized organ symbols, because the number of different organ types is simply too large. Except for a small set of pictograms for bearings, gears, mechanisms etc. (see for instance TJALVE 1979-B), the designer has to develop his own symbols to depict the organ types needed.

Such pictogram type symbols for mechanical organs combined with black-box descriptions of electronics and software have proven extremely efficient for early pragmatic sketches of mechatronic design concepts, both as a tool for generating alternatives and as a means for communication, see Figure 95. One may argue that the mechanical and electronic organs are depicted on quite different levels of abstraction in this model type, but no good solution for modelling electronics on a more concrete and visual level than diagram symbols or blackboxes has not yet evolved.

An efficient way of modelling the spatial structure of organs using symbols has been suggested by TJALVE 1979-A, see Figure 62. Such symbols represent the rough shape and size of the organs, and they are well-suited for quickly suggesting geometrical arrangements of parts, independently of whether they are mechanical or electronic components.

Besides symbolic representations, one will find hardware models of the spatial arrangements of subsystems and components in the organ domain.

Also in the organ domain, we should discuss the *activity structure* as described by BOSMAN 1978. He has not suggested any suitable model type for this, but it seems important to devise tools for modelling the sequence and timing of effects exerted by both the mechatronic system and by its operator. In principle, all the model types of Figure 87 can be applied for this task.

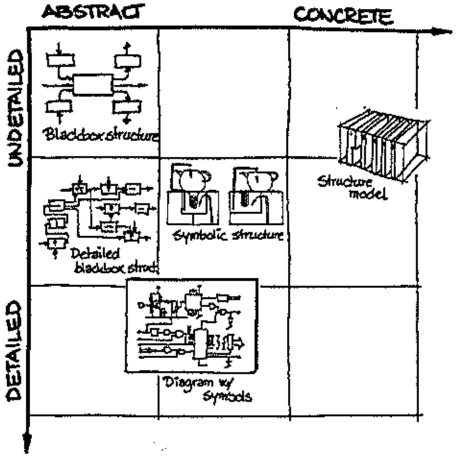


Figure 96 Mechatronic design models in the organ domain

7.2 Synthesis methods

Literature on design methods for the essential problem in mechatronics: technology allocation, is remarkably scarce. One of the few authors who discusses this topic is SALMINEN ET AL 1990-B, who draws on experience from several industrial development projects in the Finnish mechatronics programme. He realizes that several methods on different levels (for the total project, single design phases, individual design steps, particular technologies) are required for mechatronics, Figure 97. Salminen terms this idea 'metamethodics': The design engineer must acquire the knowledge of a set of different tools and then select and combine those which fit the task at hand. A major difficulty, he concludes, is that the existing methods have been developed independently of one another, which makes it very difficult to shift from one method to another.

In Figure 97 one will recognize the three level approach to product design already described in Section 3.3: Product development, product synthesis and problem solving. This thesis is only concerned with the second level: product synthesis, or in Salminen's terminology: methods applicable to certain steps, work phases and technologies.

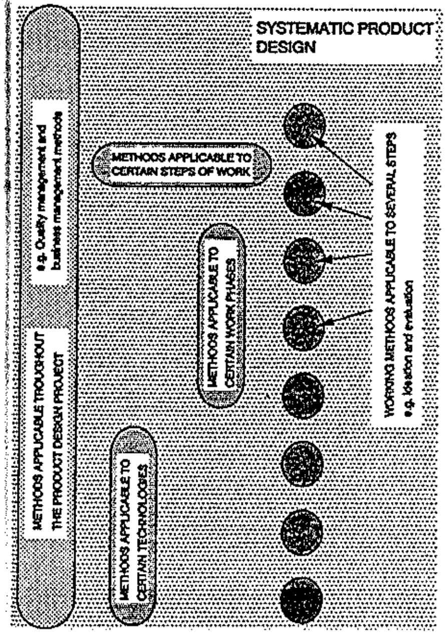


Figure 97 'Metamethodics': design methods on different levels, SALMINEN ET AL 1990-B

According to the Theory of Domains, two characteristic types of synthesis method can be distinguished, when keeping in mind that methods are instructions for how to propagate from one product model to another:

1 Methods *within one domain*. The purpose of such methods is to change the level of detail and/or of abstraction. For instance to 'formulate the blackbox process of a system' is a method to go from a concrete to an abstract model in the process domain.

2 Methods, which proceed *from one domain to another*. Their purpose is to take the design a step closer to its final realization. For instance a manufacturers' catalogue of electronic components takes the designer from the function domain to the parts domain.

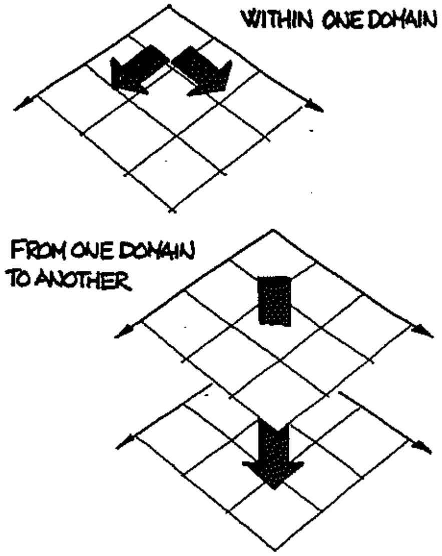


Figure 98 Two kinds of synthesis method distinguishable in the Theory of Domains

In the first type, even though the starting and ending point lie within the same domain, the designer is likely to do some thinking in other domains too.

In this section I will suggest promising design methods for mechatronics, and I will show how each method relates to the set of theorems formulated in Chapter 6.

One-state transformation structures

Based on Theorem 4 (substitution of multi-state transformations), we may suggest a design method for establishing the process structure of mechatronic systems:

1. Identify inputs and outputs of a blackbox process for the mechatronic system.
2. Decompose the blackbox process into a structure of subprocesses.
3. Identify those processes which are of multi-state type.
4. Deduce the number of states, and establish the state transition structure.
5. For each state of the system, organize a one-state process structure.

This is roughly the procedure which was followed in the example in Figure 73. The method operates within one domain. It aids the designer to extend the blackbox process to a detailed model of process structures and their state transition interactions.

What is the purpose of establishing the process structure? According to ROTH 1982 the designer has several options of creating design alternatives based on the process structure. They are to:

- Shift the systems boundaries.
- Change the sequence of subprocesses.
- Subdivide one process into subprocesses.
- Integrate several subprocesses into one.
- Introduce transducing and conduction elements.
- Move the information input or change the information carrier.
- Establish parallel processing branches.

Most of these options only need a slight change in wording to be directly applicable to creating alternatives also in the state transition structure of a mechatronic system.

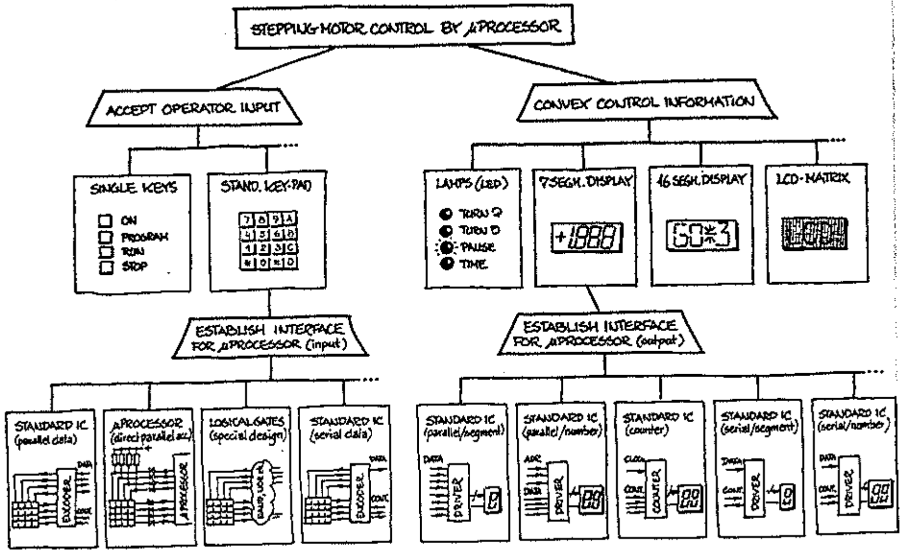


Figure 99 The function/means tree for the electronics of a simple stepping motor controller

The function/means tree

To establish a function/means tree is a powerful method for proceeding from the function to the organ domain for mechatronic systems. It is based on Theorem 1 (vertical causality) and Theorem 2 (complex of secondary functions), and has been suggested for mechanical systems by TJALVE 1979-A and ANDREASEN 1980.

1. Identify primary functions of the mechatronic system.
2. Suggest alternative means for the realization of each function.
3. For each means, check the complex of secondary functions for required subfunctions.
4. Repeat this procedure for each hierarchical level.

The method was illustrated in Figure 79. The example of Figure 99 shows that the function/means tree can also be established for designs which are mostly electronic.

It must be noted that this is not an instruction for a one-way transfer from the function domain to the organ domain. The function/means tree rather explains the necessity of jumping back and forth repeatedly between the formulation of functions and the search for alternative organs on different hierarchical levels.

The function/means structure also depicts a major difficulty in technology allocation: In order to avoid superfluous work, the designer wants to make the choice of alternative solutions as early as possible, i.e. to cut away branches right at the top. A qualified decision, however, needs sufficient knowledge of subsolutions on lower levels for all alternatives, i.e. knowledge in full depth of all branches. This dilemma has been treated in BUUR 1987, where the function/means tree method was applied for designing an electro-acoustic transducer.

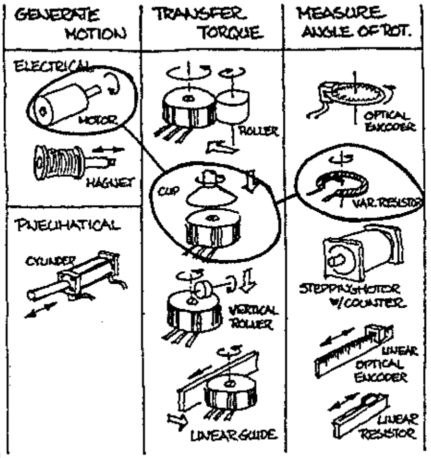


Figure 100 The morphological method applied to the design of a testing apparatus for hearing aid potentiometers

The morphological method

The morphological method developed by ZWICKY 1966 is widely applied in machine design. It exploits the fact that complex problems can often be divided into several characteristics or parameters, which may be altered independently, so that the total solution is a combination of subsolutions.

When used with subfunctions, the morphological method is well suited for mechatronic systems. It is a requirement that the subproblems, which are solved individually, do not depend on one another, i.e. the functions must belong to the same level of the function/means structure. This independence can be ensured by applying Theorem 2 (complex of secondary functions).

1. Identify the full set of primary functions required in the mechatronic system (this is most easily done by analyzing some preliminary solutions).
2. Check that the functions are independent by comparing with the complex of secondary functions.
3. Suggest alternative means to realize each function.
4. Generate full design concepts by forming all possible combinations of means for each function.

Of course not all combinations will be feasible, and some will require interface efforts to fit the means together. The method does however suggest a large number of new concepts. The morphological method helps the designer to proceed from the function domain to the organ domain.

The solution which is highlighted in the morphological box of Figure 100, was illustrated in Figure 95.

Interface specification

Theorem 6 (interface organs) invites the formulation of design methods for determining the interfaces between technological subsystems within the mechatronic system. HEINZL 1984 and CORDES 1984 have suggested a procedure for arranging the interface components of the border between a machine system and its electronic control. The method was illustrated in Figure 39 and Figure 41:

1 Determine the quality and quantity of information flow between the mechanical and electronic system.

2 Choose appropriate sensors and actuators for each task.

3 To establishing independent subsystems, group those sensors and actuators, which in a control sense must be linked.

The existence of a more general design method for determining external interface organs was indicated by KAJITANI 1986-A and GERHARD & LENART 1982:

1 Determine the type (material, energy, information), form and quantity of all flows between the mechatronic system and its surroundings: The human operator, the environment and other products.

2 Choose suitable interface organs (mechanical or electromechanical) to handle each input and each output of the system: Man/machine, environmental and systems interfaces.

3 For signal flows, consider electro/mechanical and analog/digital interfaces and examine options for integrating organs.

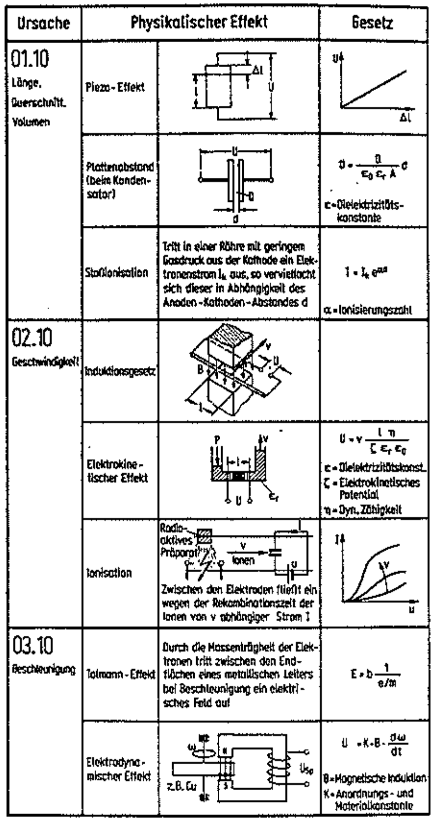
Catalogues of interface organs

The four types of organ capable of interfacing with electronic circuits are of central importance to mechatronics design: sensors, actuators, control elements and displays. Since the number of physical effects available for conversion to and from electrical signals is limited, this is an area well-suited for applying *design catalogues*.

Design catalogues are collections of solution principles for realizing commonly encountered subfunctions. They are based on the philosophy that many designers need to solve the same subproblems, and therefore it is profitable to make a standardized catalogue of solution alternatives available to them.

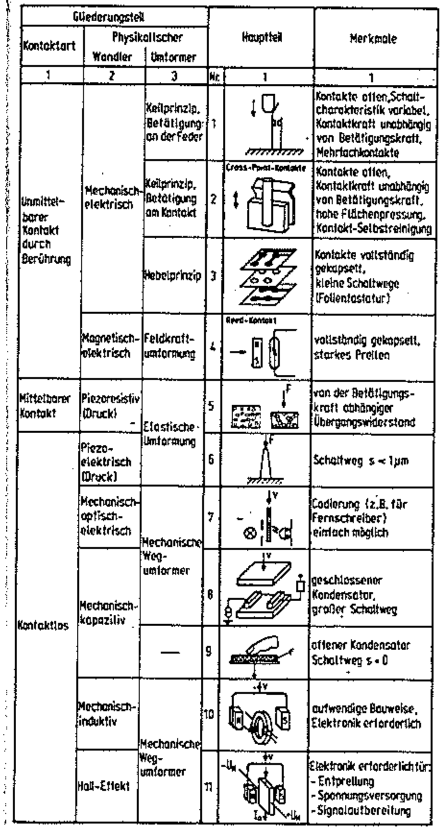
Major work on design catalogues for machine design has been carried out by KOLLER 1976 and ROTH 1982, and some of their results are directly applicable to interface functions in mechatronics, Figure 101.

The dilemma of compiling design catalogues is that they have to be fairly abstract to be generally applicable, and at the same time concrete enough to let the designer recognize his problem in the catalogue and understand the suggested solution principles.



Ursache	Physikalischer Effekt		Gesetz
01.10 Länge, Querschnitt, Volumen	Piezo-Effekt		
	Plattenabstand (beim Kondensator)		$U = \frac{Q}{\varepsilon_0 \varepsilon_r A} d$ ε = Dielektrizitätskonstante
	Stoßionisation	Tritt in einer Röhre mit geringem Gasdruck aus der Kathode ein Elektronenstrom I_k aus, so vervielfacht sich dieser in Abhängigkeit des Anoden-Kathoden-Abstandes d	$I = I_k e^{\alpha d}$ α = Ionisierungszahl
02.10 Geschwindigkeit	Induktionsgesetz		
	Elektrokinetischer Effekt		$U = v \frac{l\,\eta}{\zeta\,\varepsilon_r\,\varepsilon_0}$ ε = Dielektrizitätskonst. ζ = Elektrokinetisches Potential η = Dyn. Zähigkeit
	Ionisation	Radioaktives Präparat Ionen Zwischen den Elektroden fließt ein wegen der Rekombinationszeit der Ionen von v abhängiger Strom I	
03.10 Beschleunigung	Tolmann-Effekt	Durch die Massenträgheit der Elektronen tritt zwischen den Endflächen eines metallischen Leiters bei Beschleunigung ein elektrisches Feld auf	$E = b \frac{1}{e/m}$
	Elektrodynamischer Effekt	z.B. Cu	$U = K \cdot B \cdot \frac{d\omega}{dt}$ B = Magnetische Induktion K = Anordnungs- und Materialkonstante

Figure 101 An example of a design catalogue for transforming physical properties into electronic signals, compiled by KOLLER 1976



Gliederungsteil				Hauptteil	Merkmale
Kontaktart	Physikalischer Wandler	Physikalischer Umformer			
1	2	3	Nr.	1	1
Unmittelbarer Kontakt durch Berührung	Mechanisch-elektrisch	Keilprinzip, Betätigung an der Feder	1		Kontakte offen, Schaltcharakteristik variabel, Kontaktkraft unabhängig von Betätigungskraft, Mehrfachkontakte
		Keilprinzip, Betätigung am Kontakt	2	Cross-Point-Kontakte	Kontakte offen, Kontaktkraft unabhängig von Betätigungskraft, hohe Flächenpressung, Kontakt-Selbstreinigung
		Hebelprinzip	3		Kontakte vollständig gekapselt, kleine Schaltwege (Folientastatur)
	Magnetisch-elektrisch	Feldkraftumformung	4	Reed-Kontakt	vollständig gekapselt, starkes Prellen
Mittelbarer Kontakt	Piezoresistiv (Druck)	Elastische Umformung	5		von der Betätigungskraft abhängiger Übergangswiderstand
Kontaktlos	Piezoelektrisch (Druck)		6		Schaltweg s < 1 µm
	Mechanisch-optisch-elektrisch	Mechanische Wegumformer	7		Codierung (z.B. für Fernschreiber) einfach möglich
	Mechanisch-kapazitiv		8		geschlossener Kondensator, großer Schaltweg
		—	9		offener Kondensator Schaltweg s = 0
	Mechanisch-induktiv	Mechanische Wegumformer	10		aufwendige Bauweise, Elektronik erforderlich
	Hall-Effekt		11		Elektronik erforderlich für: - Entprellung - Spannungsversorgung - Signalaufbereitung

Figure 102 A design catalogue of physical effects for realizing an electric switch, compiled by ROTH 1985

Design catalogues of control elements and displays are not common, but examples can be found with ROTH 1985 and VDI/VDE 1985, see Figure 102.

An approach quite different from the very systematically arranged design catalogues of Koller and Roth is shown in Figure 103. This has more of the nature of a chart for inspiration which visualizes concrete interface organs (control elements). The purpose is to inspire the designer to widen his imagination for alternative solutions, not to pinpoint one particular principle to solve the problem.

Catalogues of sensor and actuator principles are more readily available in the literature.

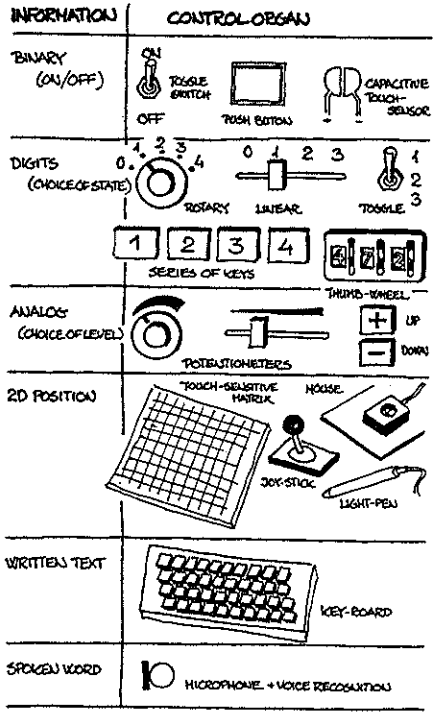


Figure 103 A map of control elements for electronics, compiled by the author to inspire the designer to consider alternative solutions

7.3 Design principles

There are good reasons for paying attention to design principles in mechatronics. Even though design principles change in response to the progress in technological developments, such pragmatic rules can help to 'move boundaries', i.e. to shift the emphasis between mechanics, electronics, and software, and thus to create alternative design concepts.

Rather than offer recommendations for how to design, I will concentrate on the core of design principles and discuss contrasting rules in the sense of SCHILLING 1984, e.g. 'integrating functions *vs.* differentiating functions'.

In fact I recommend the use of design principles in a way opposite to their original purpose: They have been suggested by experienced engineers to *narrow* the choice of solutions, but I will use them together with their contrasting principles to *extend* the field of solution alternatives.

Some such contrasting design principles can be directly deduced from the examples of design principles in Chapter 4. Her I will give a small selection of contrasting principles and their application.

Coupled mechanical movements *vs.* independent actuators

There has been a tradition in machine design to apply one central drive (motor) and to extract all the required motions in the system by means of mechanical gears and mechanisms. This gave the advantage of synchronization throughout the system, but it also meant that all relations between motions were controlled by the design of mechanical parts and therefore not very flexible.

Today, the trend is towards choosing an independent actuator for every motion required in the system and relate them through electronic control, to achieve flexibility and easy access to very complex motion patterns. This principle was illustrated by YAMAZAKI 1987 in Figure 3, on page 8.

Clearly, the multi-actuator approach is not preferable in all cases. In particular the demands for low cost and high safety will frequently favour mechanisms and gears. Mechatronics design needs careful weighing up of one approach against the other to find a good compromise both for subsystems and for the total system.

So in effect this principle illustrates the choice between realizing the *control function* of the required movements in mechanical technology or in electronics.

Mechanical conversion *vs.* direct drives and direct sensing

When fitting actuators and sensors to the physical reality of the mechanical system, the chain of converting movements may be long or short.

Electric motors may require mechanical gears to adjust torque and revolutions. Solenoids may require simple mechanisms to achieve the required linear displacement or direction of movement. Sensors may require membranes, mechanisms or gears to amplify (or attenuate) the physical property being measured.

This principle highlights the dilemma of choosing an actuator or sensor design to fit the required task as closely as possible. The *interface function* is a question of signal (or energy conversion), and it can be realized mechanically or electronically, i.e. on one side or the other of the electromagnetic component. A similar problem is found with operator controls and displays.

Multiplication of subsystems *vs.* multiple use of single one

When identical transformations have to be performed on several process objects, there is a choice between utilizing the same transformation system repeatedly, or multiplying the number of transformation systems for simultaneous, parallel processing.

The first case is seen clearly in assembly systems: A series of objects is moved past one active position to be assembled. The second case can be recognized for instance in telephone exchanges, where each subscriber has a separate electromechanical counter to ensure reliable registration of telephone charges.

The essence of the principle here is whether the transformation function should be executed in *serial* or *parallel* processes. An example of such considerations is shown in Figure 104.

Central control system *vs.* distributed intelligence

When microcomputers were still expensive, it was common to centralize all control functions in one processor. Now that microcomputers are more readily available, there is the option of creating autonomous subsystems, each of them with its own processor, which communicate intelligently. The choice seems to be largely a question of costs, execution speed and reliability. The cost of processing units for instance, is bound to be high in a distributed system, whereas wiring costs may be low, because the intelligent communication replaces one-wire-per function solutions.

Programmable control system *vs.* hardware logic control

The discussion of programmable versus hardware control is often dominated by arguments on flexibility: software provides an enormous flexibility for the designer to carry out modifications and improvements even to the last minute of the development project. Dedicated hardware logic (either in electronics or in mechanical technology) on the other hand will often display higher execution speed and reliability.

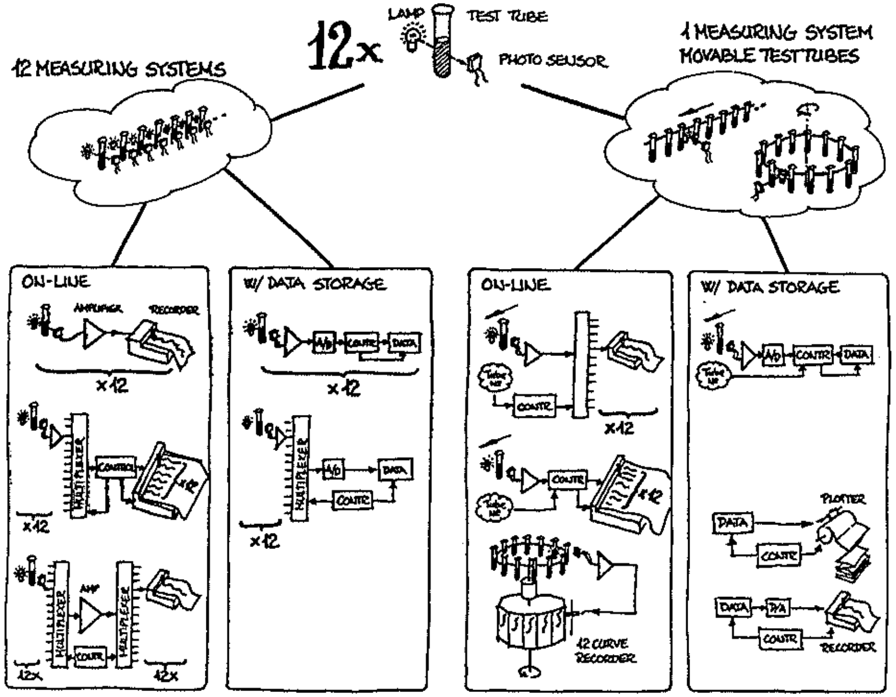


Figure 104 Serial vs. parallel processing: A system for monitoring the bacterial growth in 12 containers simultaneously

The technological development seems constantly to blur the boundary between programmable and not programmable. For instance such components as programmable logic devices (PLDs) and application specific integrated circuits (ASICs) provide great flexibility during design combined with the benefits of hardwired logic once the system is manufactured.

This principle is mainly concerned with the *hardware/software interfaces* of mechatronic systems.

Digitize signals close to the source
vs. close to the signal processing device

Conscious shifting of the D/A and A/D interfaces between 'early' and 'late' in the signal path, may disclose attractive alternatives in mechatronics design.

Digital signals are a prerequisite for microcomputer processing, but signals in the physical world exist for the most part in analog form, so most mechatronic systems require one or more conversion organs.

8 CONCLUSIONS

Mechatronics has developed fast, following advances in electronics and microprocessor technology. Design methodology research has had no time to catch up, even though mechatronics technology seems to influence not only design practice in companies, but also engineering education, project organization and company strategies.

The main objectives of this research were to define mechatronics, to determine the preconditions for product development, to establish a theory of mechatronic systems, and to suggest tools for conceptual design.

This chapter concludes the dissertation. It contains a summary of results and an evaluation of the research. Finally some suggestions for future research in fields related to mechatronics design are given.

8.1 Discussion of results

In this research, I have defined mechatronics by two kinds of integration of mechanics, electronics and software: functional and spatial. Intelligence and flexibility, though predominant properties of mechatronic systems, were found unsuitable as defining criteria because of their relative nature. It was argued that mechatronics engineering must be treated as an independent, product development oriented discipline, to exploit the full potential of this new technology.

The study in industry showed that successful mechatronic product development requires a set of typical strategy patterns, including technology management, product planning, market segmentation and development speed, and it was shown how these relate to the characteristics of mechatronics technology.

Interdisciplinarity is an absolute precondition for mechatronics design, and it can be achieved through engineering education, job-rotation, project organization and/or suitable office lay-out. It was established that the design methodology required specially for mechatronics belongs on the level of product synthesis, and that it is primarily necessary for concept design.

The review of the literature gave no clear indication of any theoretical understanding of mechatronic systems. Until the synergy between contributing authors is increased, we will probably not see any evolving consensus on the interpretation of the mechatronics concept or on design theory. Literature does however deal with a number of aspects which need to be considered in a mechatronic systems theory: The state transition structure, the role of information, the control functions, interface organs etc.

It has proved advantageous to acquire general theory from machine design literature and models for handling sequence and state transitions from software literature, in order to form a mechatronic systems theory.

It was found that a synthesis theory for mechatronics can be based directly on the Theory of Domains, provided it is extended to deal with the logical concept of state transitions.

A new theoretical basis for mechatronics design was formulated in a set of axiomatic statements and in six theorems. The theorems express essential conditions for the structured understanding of mechatronic systems, and they are directly applicable for formulating design methods.

A set of tentative design tools for mechatronics consisting of models, methods and design principles has been developed. The suggested design models are structured according to their level of abstraction and the number of details in each of the process, function and organ domains.

A problem which did not find any satisfactory solution, was how to combine concrete pictograms of mechanical principles with the abstract modelling of electronic and software principles in the organ domain.

Four design methods have been described in coarse step-by-step procedures, and the application of design catalogues for interface organs was recommended.

For creating alternative design concepts (alternative technology allocations), the application of contrasting design principles was suggested, and six examples of such principles, concerned with the central difficulties of mechatronics design, were presented.

Research evaluation and outlook

The novelty in this research lies in (1) the total, product development oriented approach to mechatronics, (2) the field study in Japanese industry, and (3) the scientific approach to mechatronics design methodology. It marks the first attempt to describe a mechatronic systems theory and a methodology for mechatronics design.

The research is based on a broad, but also superficial knowledge of product development practice in industry. It is superficial in two respects: Due to the study method (the outcome of an industry visit of a few hours is largely determined by coincidence) and due to the limited extent of the study (the number and selection of companies does not allow any quantitative analysis of data). The understanding of mechatronics design put forward in this thesis however, could not have been achieved through a less broad approach.

It has been my goal to give an if possible complete review of the literature on mechatronics design, to ensure the best academic basis for the research, but there may still be, for instance, French and Russian contributions of which I am not aware.

The verification of the theoretical results constitutes a major obstacle in this work. Verification has been limited mostly to logical reasoning: The proposed theory permits the explanation of all observed phenomena in the literature and in industry, and it covers such well-accepted paradigms as system models, interface thinking and design principles.

As for verification by acceptance, the pedagogic formulation of tools has not yet been pursued to a level where designers in industry would agree to experiment with their use. Work is under way to complete an application oriented presentation of these tools in a pedagogic form suitable for engineering education and post graduate training.

In the light of the rather poor results achieved with introducing design methods into industry in the machine design field, the question must be asked: Is it all worth the effort? There are some indications that the attitude towards design methodology is different in mechatronics - and changing:

1 The need for 'common language' in mechatronics design is generally recognized in industry.

2 The movement towards formal software methodology will force mechanical and electronics designers to apply a structured approach.

3 The increased application of CAD will have a similar effect.

4 There is a growing understanding that high quality in products can only be achieved through the use of methodical procedures.

8.2 Topics recommended for future research

In the course of this research, the following areas have proved in want of more thorough investigation, if the method of mechatronic product development is to be substantially improved:

1 Technology management for mechatronics Rapid technological advances make it crucial for the mechatronics company to ensure that the right new technologies are acquired, and that the obsolete ones are phased out at the appropriate moments. There are very few methods available yet to assist in this task.

2 Evaluation of mechatronic design concepts Because of the required interdisciplinary knowledge, early decisions on mechatronic solutions are difficult. There is a need for clarification of the relations between mechatronic design models and evaluation criteria.

3 Man/machine systems design Not only the design of the human interface but also the precise assessment of human needs and preferences is becoming increasingly important for mechatronics design. I believe it is possible to develop a set of tools for this task, and to suggest design models which can support a 'fast prototyping' approach.

4 Linking mechatronics and software tools Presently there is a movement in software engineering towards formalized methods for specification and design. Even though they are not capable of handling spatial relations and physical effects, it will be profitable to investigate the application of software design methods to mechatronics design.

5 The spatial integration i mechatronics It is likely that design tools can be developed for the task of physically integrating mechanics and electronics in the same device. This is particularly interesting for fields like sensor and actuator design, electro-hydraulics and micromechanics.

Bibliography

Abegglen, J C and Stalk, G: Kaisha - the Japanese corporation. Charles E Tuttle Comp, Tokyo 1987.

Adolphs, D: Münzfernsprecher NT 2000 mit Mikrorechner. Feinwerktechnik & Messtechnik vol 88 no 6, 1980.

Almássy, G: Neue Aufgaben in der mechanischen Konstruktion. Conf Mech Aspects Electronic Design (Constronic '80), Budapest 1980.

Ammon, P and Wer, O: Hochintegrierte, kundenspezifische Schaltungen bringen technische und wirtschaftliche Vorteile. Feinwerktechnik & Messtechnik vol 94 no 3, 1986.

Amstutz, D: Elektronik in Haushaltgeräten: ASICs lassen grüssen. Schweizer Maschinenmarkt no 10 1987.

Andreasen, M M: Syntesemetoder på systemgrundlag. Dissertation, Lunds Tekniska Högskola 1980 *(in Danish)*.

Andreasen, M M; Kähler S and Lund, T: Design for assembly. IFS Publ/Springer-Verlag, London 1982.

Andreasen, M M and Hein, L: Integrated product development. IFS Publications Ltd, Springer-Verlag, London 1987.

Andrews, G R and Schneider, F B: Concepts and notations for concurrent programming. Computer Surveys vol 15 no 1, March 1981.

Bajorek, C H; Thompson, D A and Lovgren, J L: Trends in recording technologies and role of micromechatronics in data storage. Int Conf on Adv Mechatronics, Jpn Soc Mech Eng Tokyo 1989.

Bálint, L and Haslinger, L: The role of physical design, packaging and engineering design in integrated CAD-CAM-CAT systems of the electronics industry. Conf Mech Aspects Electronic Design (Constronic '80), Budapest 1980.

Barney, G C: Intelligent instrumentation. Prentice-Hall, London 1985.

Baumann, H G et al: Festlegung der logischen Wirkzusammenhänge beim systematischen Projektieren und Konstruieren komplexer technischer Systeme. Metallwissenschaft und Technik vol 33 no 2, February 1979.

Becker, R and Neumann, K: Systematische Konzepierung des konstruktiven Aufbaus elektronischer Geräte. Feingerätetechnik vol 32 no 6, 1983.

Berardinis, L A: Automakers move to multiplexing. Machine Design, June 8, 1989.

Berg, S; Christensen, B and Andreasen, M M: Mechatronics modelling in a technology development project regarding flexible component feeders. Inst for Eng Design, DTH Lyngby 1989 *(in Danish)*.

Bersoff, E H: Elements of software configuration management. IEEE Trans Software Eng vol SE-10 no 1, 1984.

Boehm, B W: Software engineering economics. IEEE Trans Software Eng vol SE-10 no 1, January 1984.

Boehm, B W; Gray, T E and Seewaldt, T: Prototyping versus specifying: A multiproject experiment. IEEE Trans Software Eng vol SE-10 no 3, May 1984.

Boehm, B W: Understanding and controlling software cost. IFIP Conf Inf Processing, Elsevier Science 1986.

Bosman, D: Systematic design of instrumentation systems. J Phys E Sci Instrum vol 11 no 2, February 1978.

Bradley, D A and Dawson, D: Mechatronics - an integrating approach in engineering design. Int Conf Mechatronics, IMechE Cambridge 1990-A.

Bradley, D A and Dawson, D: Achieving integration in engineering design education: The role of mechatronics. Mechatronic Systems Eng, vol 1 no 1, 1990-B.

Brunthaler, S: Konstruktion von Anwendersoftware in Analogie zum Methodischen Konstruieren. Konstruktion vol 37 no 7-12, 1985.

Bryan, W and Siegel, S: Making software visible, operational, and maintainable in a small project environment. IEEE Trans Software Eng vol SE-10 no 1, 1984.

Brümmer, H: Elektronische Gerätetechnik. Vogel-Verlag, Würzburg 1980.

Buschmann, H and Weissmantel, H: Praktische Entwicklungsmethodik: Ausbildung in kleinen Gruppen zu Ingenieuren der Feinwerktechnik. 1 Conf Mechanica Fina, Sao Paulo 1984.

Buur, J: Integrated PVDF handset for telephones. MSc thesis, Inst for Eng Design, DTH Lyngby 1984 *(in Danish)*.

Buur, J: Systematische Versuche bei der Konstruktion eines piezoelektrischen Wandlers. Feinwerktechnik & Messtechnik vol 95 no 8, December 1987.

Böhme, L: Feingeräte unter dem Einfluss der Mikroelektronik. Feingerätetechnik vol 27 no 5, May 1978.

Cooke, P: Electronics design demands a new approach. Engineering vol 224 no 4, 1984.

Cordes, C: Entwurf der Prozesssteuerung bei Geräten ausgehend vom Benutzerdialog. VDI-Berichte no 564, 1984.

Cordes, C: Zustandsgraphen und Zustandsmatrizen beim Entwurf von Gerätesteuerungen. Dissertation, Technische Universität München 1986.

Cramer, B: Die Bedeutung der Feinwerktechnik für die Kommunikationsdienste der Zukunft. 1 Conf Mechanica Fina, Sao Paulo 1984.

D'Angelo, R and Ufer, W: Neue Spiegelreflexkamera PRAKTICA BX20 mit teil- und vollautomatischer Belichtungssteuerung. Feingerätetechnik vol 37 no 4, April 1988.

Darby, B J and Orton, D W R: Structured approaches to design. IEEE Proc vol 133 Pt E no 3, May 1986.

Deasley, P J: Control: the integrating factor in mechatronics. Mechatronic Systems Eng, vol 1 no 1, 1990.

Detter, H: Mechatronic - die Symbiose zwischen Elektronik und Mechanik. 1 Conf Mechanica Fina, Sao Paulo 1984.

Dijkstra, E W: Structured programming. *In:* Buxton, J N: Software engineering techniques, Rom 1969.

Dinsdale, J: Mechatronics - the European dimension. IMechE Conf What is mechatronics? UK 1988.

Engler, P: Feinwerktechnik. Int Conf Eng Design (ICED), Dubrovnik 1990.

Evans, B: Japanese-style management, product design and corporate strategy. Design Studies vol 6 no 1, January 1985.

Fass, K G et al: Ausbildung von Ingenieuren der Fachrichtung Feinwerktechnik. Feinwerktechnik & Messtechnik vol 89 no 6, 1981.

Feiertag, R: Die Feinwerktechnik in Europa. 1 Conf Mechanica Fina, Sao Paulo 1984.

Finkelstein, L and Finkelstein, A C W: Review of design methodology. IEE Proceedings vol 130 Pt A no 4, June 1983.

Franksen, O I: The nature of data - from measurements to systems. BIT no 25, 1985.

French, M J: Do we really need mechatronics? IMechE Conf What is mechatronics? UK 1988.

French, M J: The partition of functions in mechatronic design. Int Conf Mechatronics, IMechE Cambridge 1990.

Frölich, B and Schlottmann, K: Systematische Konstruktion elektronischer Geräte. VDI-Berichte no 460, 1982.

Fujiwara, H: Problems in the age of revolution - The softening evolution of machine industry. Koyo Engineering Journal no 126, 1984 *(in Japanese)*.

Furchert et al: Entwicklung der Mikroelektronik und den Einfluss auf die Gerätetechnik. Feingerätetechnik vol 28 no 5, May 1979.

Gaskell, C S: The impact of modern circuit technologies on electronic instrument design. Electron Supplement, October 1976.

Gerhard, E and Lenart, C: Physikalisch-technische und gerätetechnische Darstellung feinwerktechnischer Produkte. VDI-Berichte no 460, 1982.

Gjerløv-Knudsen, O et al: Fælleskursus i industriel design. Skolen for Brugskunst, Copenhagen 1989 *(in Danish)*.

Gregory, G: Japanese electronics technology: enterprise and innovation. John Wiley & Sons, Chichester 1986.

Hall, A D: A methodology for systems engineering. D van Nostrand Comp, Princeton New Jersey 1962.

Hanson, M: Integration of advanced microcomputer system design in mechanical engineering education. Int Conf Mechatronics, IMechE Cambridge 1990.

Harper, C A: Handbook of Electronics Packaging. McGraw-Hill, New York 1967.

Heath, C: Designing electronic circuits. Engineering Design Education, Design Council, London 1986.

Heering, E: Software Engineering. Vieweg, Braunschweig-Wiesbaden 1984.

Hein, L: A product development approach to mechatronics. Mechatronic Systems Eng, vol 1 no 1, 1990.

Heinzl, J: Entwicklungsmethodik für Geräte mit Steuerung durch Mikroelektronik. VDI Berichte no 515, 1984.

Heinzl, J (ref): Mikroelektronik verlangt Koordination - methodisch Entwickeln. Hard and Soft, June 1985.

Herrmann, G: Das Pflichtenheft als Aufgabenstellung für die Entwurfsphase. VDI-Berichte no 460, 1982.

Herzog, R E: ME/EE interface: The disappearing barrier. Machine design, April 4, 1974.

Hildebrand, S: Einführung in die feinmechanische Konstruktion. Carl Hanser Verlag, München 1969.

Holmberg, K et al: The mechatronic design concept and its application on robot gripper design. Int Conf Eng Design (ICED), Harrogate 1989.

Horner, J F: Synergistic trends in diverse capability instrument design. Wescon Technical Papers vol 17 no 18, 1973.

Hubka, V: Theorie der Maschinensysteme. Springer-Verlag, Berlin 1973.

Hubka, V: Theorie der Konstruktionsprozesse. Springer-Verlag, Berlin 1976.

Hubka, V: Theorie technischer Systeme. Springer-Verlag, Berlin 1984.

Hubka, V; Andreasen, M M; Eder, W E: Practical studies in systematic design. Butterworth & Co Publishers, UK 1988.

Hubka, V: Spezielle Konstruktionswissenschaft für Fachgebiete. Int Conf Eng Design, Dubrovnik 1990.

Hurst, K S and James, R D: Feasibility of an automated materials testing facility for British Steel at Scunthorpe. Int Conf Mechatronics, IMechE Cambridge 1990.

IRDAC: Opinion on R&D needs in the field of mechatronics. Industry R&D Advisory Committee of the Comm of the European Communities, Bruxelles 1986.

Isermann, R: Bedeutung der Mikroelektronik für die Prozessautomatisierung (Teil 3). Elektrotechnische Zeitschrift (etz) vol 106 no 12, 1985.

Ishii, T: Mechatronics and Japanese society. /in McLean, M: Mechatronics developments in Japan and Europe. Technova, Quorum Books, London 1983.

Ishii, T: Future trends in mechatronics. Int Conf on Adv Mechatronics, Jpn Soc Mech Eng Tokyo 1989.

Itao, K: Mechanism design for mechatronization. Machine Design vol 28 no 8, July 1984 *(in Japanese)*.

Itao, K: Mechatronics and mechanism control. J Jpn Soc Prec Eng vol 52 no 7, July 1986 *(in Japanese)*.

Itao, K: Mechatronic mechanisms. Precision machine elements (2). Corona Publ, Tokyo 1987 *(in Japanese)*.

Itao, K and Nishida, Y: Present status and future of opto-mechatronics. Int conf on Adv Mechatronics, Jpn Soc Mech Eng, Tokoy 1989.

Jackson, M: System development. Prentice-Hall, 1983.

Jones, J C: Design methods - seeds of human futures. John Wiley & Sons, New York 1970.

JSME: Mechatronics. Mechanical engineers' handbook (C4). Jpn Soc Mech Eng, Tokyo 1989 *(in Japanese)*.

JTECH: Panel report on mechatronics in Japan. Japanese technology evaluation program, US Department of Commerce, Washington DC 1985.

Just, E: Modellbildung und Rechenmethoden in der Gerätetechnik. Feingerätetechnik vol 30 no 11, 1981.

Jørgensen, T: Design og produktion af miniatureelektronik - analyse af 10 østasiatiske konsumprodukter. Elektronikcentralen ECR-191, Copenhagen 1986 *(in Danish)*.

Kajitani, M: Advanced information society and mechatronics. J Jpn Soc Mech Eng vol 89 no 810, May 1986-A *(in Japanese)*.

Kajitani, M: A concept of mechatronics. J Jpn Soc Prec Eng vol 52 no 7, July 1986-B *(in Japanese)*.

Kajitani, M: Design education in the age of mechatronics. J Jpn Soc Des Draft vol 23 no 2, February 1988. *(in Japanese)*.

Kajitani, M: A concept of mechatronics. J Robotics and Mechatronics vol 1 no 1, June 1989.

Keilwerth, H et al: Feinwerktechnik - Situationsanalyse und Denkanstöße für Innovation. Feinwerktechnik & Messtechnik vol 88 no 6, September 1980.

Kestenbaum, R: What is a 'systems' approach? It's just the name of the game. Electronics, April 1969.

Kieras, D E: What people know about electronic devices: A descriptive study. University of Arizona, Technical Report no 12, 1982.

Kimm, R el al: Einführung in Software Engineering. Walter de Gruyter, Berlin 1979.

Kisaka, S: The role of mechanical engineering in the progress of electronics. J Jpn Soc Mech Eng vol 89 no 806, January 1986 *(in Japanese)*.

Klaus, G: Wörterbuch der Kybernetik. DietzVerlag, Berlin 1968.

Knetsch, W: Die industrielle Anwendung der Mikroelektronik in der Bundesrepublik Deutschland, Frankreich, Grossbritannien. Markt & Technik Verlag, 1985.

Kohler, C: Mikroelektronik-kompatible Sensoren. Feinwerktechnik & Messtechnik vol 94 no 7, 1986.

Koller, R: Konstruktionsmethode für den Maschinen- Geräte- und Apparatebau. Springer-Verl, Berlin 1976.

Komischke, M et al: Sensoren gekoppelt mit Mikroprozessoren - Möglichkeiten der Vorverarbeitung und Sensorschnittstelle.
Elektrotechnische Zeitschrift (etz) vol 105 no 15, 1984.

Komoriya, K and Tachi, S: Research and development of mechatronics. J Jpn Soc Prec Eng vol 49 no 11, November 1983 *(in Japanese)*.

Krapp, H: Rechnerunterstützte Elektronik-Entwicklung: Den Anschluss nicht verpassen.
Elektronik vol 34 no 5, 1985.

Krause, W: Konstruktionsausbildung in der Gerätetechnik. Feingerätetechnik vol 28 no 10, October 1979.

Krause, W: Gerätekonstruktion.
VEB Verlag Technik, Berlin 1986 (2 ed).

Kurosawa, T: Development and needs for mechatronics technology. J Jpn Soc Prec Eng vol 49 no 11, November 1983 *(in Japanese)*.

Lauruschkat, H: Struktur, Aufgaben und Arbeitsweise der VDI/VDE-Gesellschaft Feinwerktechnik.
Feinwerktechnik & Messtechnik vol 93 no 7, 1985.

Lehmann, M: Entwicklungsmethodik für die Anwendung der Mikroelektronik im Maschinenbau.
Konstruktion vol 37 no 9, 1985.

Løvengreen, H et al: Metodikker og værktøjer til konstruktion af programmel (KOMET). Dansk Datamatik Center DDC 05, Copenhagen 1980 *(in Danish)*.

M'Pherson, P K: Systems engineering: An approach to whole-system design. The Radio and Electronic Engineer vol 50 no 11/12, November/December 1980.

Matisoff, B S: Handbook of electronics packaging design and engineering.
Van Nostrand Reinhold Company, New York 1982.

Meyer, B: On formalism in specifications.
IEEE Trans Software vol 2, 1985.

Meyer, H F: Elektronisch-pneumatische Regelung im Druck- und Wegbereich. Techn Rundschau no 5, 1986.

Meyer, K: Die Feinwerktechnik in Dänemark.
Feinwerktechnik & Messtechnik vol 86 no 6, 1978.

Meyer, K et al: Uddannelse af apparatingeniører - et forslag. Dansk Forening for Apparatteknik, Copenhagen 1980 *(in Danish)*.

Meyer, K: Ausbreitung der feinwerktechnischen Ingenieurausbildung in Dänemark und Einsatzgebiete an der Dänischen Technischen Hochschule.
12. Int Kolloquium der Feinwerktechnik, Munich 1988.

Milne, J S and Fraser, C J: Development of a mechatronics learning facility.
Mechatronic Systems Eng, vol 1 no 1, 1990.

Morgenbrod, H and Mrva, : Die zunehmende Bedeutung der Software.
Elektrotechnische Zeitschrift vol 106 no 13, 1985.

Müller, H: Konstruktive Gestaltung und Fertigung in der Elektronik. Vieweg & Sohn, Braunschweig 1981.

Mäder, K: Einfluss der Mikroelektronik auf die konstruktive Entwicklung von Seriendruckern.
Feingerätetechnik vol 30 no 12, 1981.

Naur, P: Formalization in program development.
BIT vol 22 no 4, 1982.

Nielsen, A: Finmekanik og apparatkonstruktion.
Elektronik no 10, October 1976 *(in Danish)*.

Noltingk, B E: Creative instrument design.
J Phys E Sci Instrum vol 15 no 4, April 1982.

Nüchel, W and Piller, U: Entwurf Mikrocomputergesteuerter produkte (Part 3 - 8).
Elektrotechnische Zeitschrift (etz) vol 105 no 14 and 17, 1984; vol 106 no 1, 3, 6, 10 and 12, 1986.

Pahl, G and Beitz, W: Konstruktionslehre.
Springer-Verlag, Berlin 1977.

Petrik, O: Methodical problems of instrument designers' education. Conf Mech Aspects Electronic Design (Constronic '84) 1984.

Preston, M E et al: Mechatronics applied to the manufacture of knitted garments. Conf Mechatronics in Products and Manufacturing, Lancaster 1989.

Prince, J L et al: The role of universities in electronic packaging engineering.
Proc IEEE vol 73 no 9, September 1985.

Pöhls, A and Roth, W: Benutzerfreundliche CAD-Werkzeuge zur Mikroelektronikentwicklung.
Elektrotechnische Zeitschrift vol 106 no 13 1985.

Rasmussen, K: Analogier mellem mekaniske, akustiske og elektriske systemer. Polyteknisk Forlag, Lyngby 1970 *(in Danish)*.

Ravn, A P; Rischel, H og Mortensen, B G: Konstruktion af indlejrede, tidstro systemer.
Instituttet for Datateknik, Technical University of Denmark 1986 *(in Danish)*.

Régnault, G: Symbolschrift für die Feinwerktechnik (part 1-3).
Feinwerktechnik & Messtechnik vol 84 no 6-8, 1976.

Reuter, E: Laufwagenpositionierung bei Druckern und Plottern.
Feinwerktechnik & Messtechnik vol 97 no 10, 1989.

Richter, A and Aschoff, H J: Über die funktionsorientierte konstruktive Gestaltung von signalverarbeitenden Geräten nach statischen und dynamischen Gesichtspunkten. Feinwerktechnik vol 75 no 11, 1971.

Riescheberg, G and Woitasch, H J: Signaltechnische Analyse informationsmechanischer Konstruktion.
Feingerätetechnik vol 25 no 6-7, 1976.

Rietdijk, J A: Ten propositions on mechatronics.
Mechatronic Systems Eng, vol 1 no 1, 1990.

Rodenacker, W G and Clausen, U: Regeln des Methodischen Konstruierens I.
Krauskopf-Verlag, Mainz 1973.

Rodenacker, W G: Methodisches Konstruieren.
Springer-Verlag, Berlin 1976 (2 ed).

Ropohl, G: Flexible Fertigungssysteme.
Dissertation, Krausskopf, Stuttgart 1969.

Ross, D T: Applications and extensions of SADT.
IEEE Trans Computer vol 18 no 4, April 1985.

Roth, K: Kennzeichnende Merkmale feinwerktechnischer Konstruktionen.
VDI-Zeitschrift vol 105 no 22, August 1963.

Roth, K: Die Feinwerktechnik im Blickpunkt ihrer Nachbargebiete.
Industrielle Organisation vol 37 no 4, 1968.

Roth, K: Konstruieren mit Konstruktionskatalogen.
Springer-Verlag, Berlin 1982.

Roth, K: Anzeigen, Bedienelemente, Aufnehmer und Getriebe in zeitgemässen Feingeräten.
Feinwerktechnik & Messtechnik vol 93 no 6, 1985.

Roth, K: Modellbildung für das methodische Konstruieren ohne und mit Rechnerunterstützung.
VDI-Z vol 128 no 1/2, 1986.

Roth, K: Die Information in der Konstruktionsmethodik: Bedeutung wird immer grösser.
Schweizer Maschinenmarkt no 16, 1988.

Rüchardt, H: VLSI-Design-Szenario.
Technische Rundschau no 28, 1985.

Rzevski, G: Engineering design and artificial intelligence. The Open University, Milton Keynes 1990.

Rzevski, G: On the design of intelligent machines: Critical issues and problems.
Int Conf on Design Productivity, Honolulu 1991.

Röhrs, G: Ein allgemeines Modell des Geräteaufbaus.
Feingerätetechnik vol 29 no 6, 1980.

Salminen, V and Verho, A J: Multi-disciplinary design problem in mechatronics and some suggestions to its methodical solution in conceptual design phase.
Int Conf Eng Design (ICED), Harrogate 1989.

Salminen, V; Ropponen, T and Verho, A: Multidisciplinary problems in mechatronics and some solutions.
Int Symp Robotics and Manufact, Vancouver 1990-A.

Salminen, V et al: Metamethodics in the task definition and conceptual design phases.
Int Conf Eng Design (ICED), Dubrovnik 1990-B.

Salminen, V; Tanskanen, K and Verho, A: Finnish mechatronics approach - experiences in adaptation to educational, research and industry purposes.
Int Conf Mechatronics, IMechE Cambridge 1990-C.

Saupe, V and Lämmel, J: Ein Modell zur Labor- und Experimentautomatisierung. Feingerätetechnik vol 35 no 7, July 1986.

Schaudel, D E: Gehäuse von Sensoren - nur Verpackung? VDI-Berichte no 460, 1982.

Schilling, M: Konstruktionsprinzipien der Gerätetechnik. Conf Mech Aspects Electronic Design (Constronic '84), Budapest 1984.

Schumann, J and Gerisch, M: Softwareentwurf.
VEB Verlag Technik, Berlin 1984.

Schweitzer, G: Mechatronics - a concept with examples on active magnetic bearings.
Int Conf Mechatronics, IMechE Cambridge 1990.

Skelboe, S: Selected techniques for efficient software development for microcomputers.
Elektronikcentralen ECR-114, Copenhagen 1981.

Smith, L: Hierarchical design structures simplify system development. EDN vol 28 no 19, 1983.

Sneed, H: Software Entwicklungsmethodik.
Verlagsgesellschaft R Müller, Köln 1986.

Spannehl, H P: Entwicklungsablauf am Beispiel eines elektromechanischen Geräts mit Steuerung durch Mikroelektronik. VDI-Berichte no 564, 1984.

Spannehl, H P (ref): Systematische Entwicklung vom Energie- zum Signalfluss. Hard and Soft, April 1985.

Sperlich, H: Struktur, Gestalt und Funktion technischer Gebilde. Feingerätetechnik vol 32 no 5, 1985.

Stone, M: Competing with Japan - the rules of the game. Long Range Planning, vol 17 no 2, 1984.

Stute, G: Translatorische und rotatorische Aktoren. VDI-Berichte no 460, 1981.

Takeuchi, H and Nonaka, I: The new product development game. Harward Business Review, January/February 1986.

Takeuchi, Y: Mechatronics and software. J Jpn Soc Prec Eng vol 52 no 7, July 1986 *(in Japanese)*.

Tayefeh-Emamverdi, M: Methodische Konzeptentwicklung in der Elektrotechnik, Teil 1, 2.
Bull SEV/VSE vol 72 no 13 and 17, 1981.

Tayefeh-Emamverdi, M: Problemformulierung und Funktionsstruktur im Konstruktionsprozess.
Int Conf Eng Design (ICED), Copenhagen 1983.

Tayefeh-Emamverdi, M: Problemlösen beim Entwickeln von produkten im Maschinenbau und in der Elektrotechnik. Int Conf Eng Design, Hamburg 1985.

Taylor, G T: The mechatronics curriculum for the modern engineer. Int Conf Mechatronics, IMechE Cambridge 1990.

Thomass, W: Mikroelektronische Funktionseinheiten in elektrischen Haushalt-Kleingeräten.
Feingerätetechnik vol 35 no 3, March 1986.

Tjalve, E: A short course in industrial design.
Newnes-Butterworth, London 1979-A.

Tjalve, E et al: Engineering graphic modelling.
Newnes-Butterworths, London 1979-B.

Tovey, M: Thinking styles and modelling systems.
Design Studies vol 7 no 1, January 1986.

Uhden, I: Strukturierung und Modularisierung der Software bei Gerätesteuerungen.
VDI-Berichte no 564, 1984.

Ulrich, K T and Seering, W P: Synthesis of schematic descriptions in mechanical design.
Res Eng Design vol 1 no 1, 1989.

Valbjørn, K et al: Mekatronik i Jernet. Jernets Arbejdsgiverforening, Copenhagen 1989 *(in Danish)*.

VDI-Richtlinie 2222: Konstruktionsmethodik, Konzipieren technischer Produkte, Blatt 1.
Beuth-Verlag, Berlin-Köln 1977.

VDI-Richtlinie 2221: Methodik zum Entwickeln und Konstruieren technischer Systeme und Produkte.
Beuth-Verlag, Berlin-Köln 1986.

VDI/VDE-Richtlinie 2422: Entwicklungsmethodik für Geräte mit Steuerung durch Mikroelektronik.
Unpublished draft 1985.

Vermeier, D and Nijssen, G M: A procedure to define the object type structure of a conceptual schema.
Information Systems vol 7 no 4, 1982.

Vorbach, G: Einfluss der Mikroelektronik auf Entwicklung, Fertigung und Servicekonzept feinwerktechnischer Geräte. VDI-Berichte no 460, 1982.

Ward, P T and Mellor, S J: Structured development for real-time systems, vol 1-3. Prentice-Hall, 1985.

Wassell, J H: The basic knowledge required for design engineering. Colloq Electrical and Electronic Eng. Design - Education and training for tomorrow.
IEE Design Committee, Digest no 22 1980.

Weber, C: Neue Aspekte der Analyse und Synthese dynamischer Prozesse auf der Basis der Konstruktionsmethodik. Konstruktion vol 39 no 10, 1987.

Weismeier, A and Schwerdt, P: Anwendung faseroptischer Sensoren im Kraftfahrzeug.
VDI-Berichte no 515, 1984.

Westbrook, M H: Automotive electronics.
IEE Proc vol 133 Pt A no 4, June 1986.

Wheeler, G J: The design of electronic equipment.
Prentice-Hall, New Jersey 1972.

Wilson, P M and Greaves, J: Forward engineering - a strategic link between design and profit.
Mechatronic Systems Eng, vol 1 no 1, 1990.

Wolf, H: Gegenwart und Zukunft mikroelektronikkompatibler Sensoren. VDI-Berichte no 421, 1981.

Wolters, K: Perspektiven der Mikroelektronik.
VDI-Berichte no 515, 1984.

Yamazaki, K and Suzuki, H: A methodology of microprocessor control system development for machine tool applications. 11th North American Manufacturing Research Conference 1983.

Yamazaki, K et al: Methodology of education and R&D in "Mechatronics".
Int J Appl Engng Ed vol 1 no 1, 1985.

Yamazaki, K: Trends of ASIC technologies for mechatronics control and some results obtained in the development work. Japan Productivity Center 1987.

Yamazaki, K: Mechatronics engineering education.
Computer Graphics, (to be printed) 1989.

Zwick, L: Die Feinwerktechnik in der Technik.
Feinwerktechnik vol 75 no 8, 1971.

Zwicky, F: Entdecken, Erfinden, Forschen im morphologischen Weltbild. München 1966.

Publications completed during this research:

Articles:

Buur, J: Systematische Versuche bei der Konstruktion eines piezoelektrischen Wandlers. Feinwerktechnik & Messtechnik vol 95 no 8, December 1987. (Also 11. Int Koll der Feinwerktechnik, Lausanne 1986)

Buur, J and Andreasen, M M: Design models in mechatronic product development. Design Studies vol 10 no 3, July 1989. (Design Studies Award for best paper 1989). First published in: Udvikling af Apparater, DFA-Temadag, October 1986 *in Danish*.

Buur, J: New telephones in Japan.
Japan Forum no 3, December 1987 *(in Danish)*.

Buur, J: Positioning mechatronics design between mechanics, electronics and software. Int conf on Adv Mechatronics, Jpn Soc Mech Eng, Tokoy 1989.

Buur, J: A framework for mechatronics design methodology. Int Conf Eng Design (ICED), Harrogate 1989.

Buur, J: Mechatronics design in Japan as seen from Europe. Comparison of product cycles, life span, and functions. Nikkei Mechanical, April 30, 1990.
(in Japanese)

Buur, J: Design methods in Japan. Research, education and industrial application from a European viewpoint. Int Conf Eng Design (ICED), Dubrovnik 1990.

Buur, J: Mechatronics in Japan. Strategies and practice in product development. Int Conf Mechatronics, IMechE Cambridge 1990.
(Also Workshop Comp Integr Design, Bruxelles 1990 and 13. Int Koll der Feinwerktechnik, Lyngby 1990)

Buur, J: Does Mechatronics need a special design attitude? IMechE/SERC Research Conf on Mechatronics, St Albans 1990.

Buur, J: Industrial design methods in Japan.
Design Danmark, Danish Design Centre (to be published) 1991.

Buur, J: Mechatronics product development - the Japanese threat.
IEE Colloq on Mechatronics, London 1991.

Report:

Buur, J: Mechatronics design in Japan. A study of Japanese design methods and working practice in Japanese companies. Technical University of Denmark, Lyngby 1989.

Video movie:

Buur, J: Focusing on competition and human resources - the Japanese success. Interviews on product development in Japanese mechatronics industry.
Technical University of Denmark, Lyngby 1989.
(32 min, versions in Danish and English)

Progress reports (in Danish):

Buur, J: Apparatkontruktion 1: Project introduction.
Inst for Eng Design, Lyngby 1985.

Buur, J: Apparatkontruktion 2: Problem analysis.
Inst for Eng Design, Lyngby 1986.

Buur, J: Apparatkontruktion 3: Software design and the design concept. Inst for Eng Design, Lyngby 1986.

Buur, J: Apparatkontruktion 4: Electronics design and systems theory. Inst for Eng Design, Lyngby 1987.

Buur, J: Apparatkontruktion 5: Mechatronics design in Japan - literature study.
Inst for Eng Design, Lyngby 1987.

Buur, J: Apparatkontruktion 6: Mechatronics design in Japan - first impressions.
Inst for Eng Design, Lyngby 1988.

Appendix A

POSITIONING MECHATRONICS DESIGN BETWEEN MECHANICS, ELECTRONICS AND SOFTWARE

Jacob Buur
Department of Mechanical Engineering
University of Tokyo, Japan

Institute for Engineering Design
Technical University of Denmark
Bygn. 421, 2800 Lyngby, Denmark.

ABSTRACT

Mechatronics design means integrating mechanics, electronics and software in one system. This activity is difficult both to perform and to describe, because it needs an overall knowledge of all three technologies. The paper seeks to position mechatronics design among the three traditional fields of engineering, and also to relate it to existing design methodology. Thus, the author hopes to create a platform for development of new design methods for the mechatronics engineer.

1. INTRODUCTION

A designer of Canon Inc. described the recent development of mechatronics like this: '10 years ago it was comparably easy to explain the functions of a camera to a young engineer, even though the mechanisms were quite complex. Today it is nearly impossible, since he needs to know not only mechanics and optics, but also electronics and software.'

The design of mechatronic systems is shifting from merely adding microprocessor control to existing machines towards a total approach: creating new, innovative mixtures of mechanics, electronics and software. The decision on the design concept, i.e. this combination of principles from the three fields, is usually made at an early stage of the development project and by very few designers.

So far, we have not seen design tools, which may aid the designers in suggesting and evaluating such total combinations of mechanics, electronics and software. Design methods exist to some extent in each discipline, but neither seem capable of bridging the gap to the other technologies.

The purpose of this paper is to point out the characteristics of mechatronics design in relation to existing design methodology and to mechanics, electronics and software development. It is a necessary step prior to suggesting specific design methods for handling mechatronics concept design.

This study is part of a Ph.D. programme at The Institute for Engineering Design of The Technical University of Denmark. It is based on approximately 40 interviews with designers in both Danish and Japanese companies of mechatronics.

2. MECHATRONICS IN RELATION TO EXISTING DESIGN METHODOLOGY

Product development is a complex activity which may be regarded from many different points of view: planning, organization, creativity, methodology, task assignments etc. In a design methodical sense, it is expedient to describe product development on three levels of resolution, ANDREASEN and HEIN [1]:

1. problem solving
2. product synthesis
3. product development.

The activities on each level may be divided into phases, and we can attach design methods to each phase.

2.1 The designer: Problem solving

The model of 'General Problem Solving', as shown in Fig. 1, is a sequence of 5 activities applicable to problems of any type during design work JONES [2]. It implies that evaluating a number of problem solutions will always yield a better result, than regarding only one, intuitively found solution. In order to limit the field of possible solutions, the problem should be defined in advance, and criteria should be determined for the evaluation of alternatives.

Fig. 1 The model of General Problem Solving [2]

Problem solving may be regarded as en elementary activity to be applied to every sub-problem and in every iteration cycle of design work. Naturally the number of necessary alternatives and the care taken in evaluation will be determined by the priority of the problem and the degree of innovation (i.e. how 'new' and how difficult to solve the problem is).

In each phase of general problem solving, a number of design methods and tools may be applied, e.g. methods for creative and systematic idea generation (brainstorming, morphological methods, hierachical methods), for evaluation (point scale met-

The International Conference on ADVANCED MECHATRONICS, Tokyo 1989 pp.189-194

hods, pair-wise comparison), and CAD tools for specification of geometry and structure. The main purpose of these methods is to encourage the designers imagination, and to prevent 'human blocks' due to preoccupations and limited knowledge.

2.2 The technical system: Product synthesis

On the level of product synthesis we concentrate on the technical system itself. The theory of properties proposed by HUBKA [3] states that a machine is completely defined by a set of **basic design properties**, which are:

for the total product	**structure** (i.e. elements, relations)
for each element	**form** **dimensions** **material** **surface quality**

The basic design properties determines all other properties of the product, e.g. function, price, quality, appearance, ergonomics. They are the only variables, which the designer can manipulate directly, and we can regard product synthesis as a progress towards determining these basic properties.

The model of 'Product Synthesis' shown in Fig. 2 is derived from a theory of machine systems and applicable for the design of mechanical products, TJALVE [4]. It is not an algorithm of activities leading automatically to a final design, if only performed in the right sequence. It rather indicates stages or domains characteristic of machine systems, within which design work must be completed in order to determine the basic design properties, ANDREASEN [5].

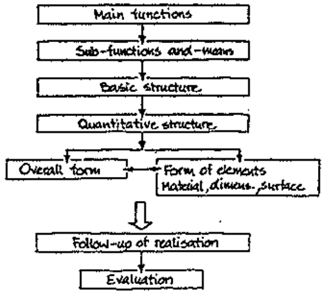


Fig. 2 The model of Product Synthesis

The human designer has the ability of freely jumping back and forth between domains in his mind in an iterative sequence. He may for instance think of the abstract function of 'supporting a rotating shaft' and simultaneously consider the very concrete properties of a ball bearing. Similarly in electronics design: he may think of the function of 'converting analog to digital signals' and simultaneously consider price and specifications of a SMD-type component. This makes the suggestion of a detailed design algorithm on the level of product synthesis unrealistic.

The knowledge of domains in product synthesis, however, permits us to develop design methods attached to one domain or to the transition from one domain to another. Using a catalogue of electronic components for example, is a method for proceeding from an abstract description of function to a physical realization.

2.3 The company: Product development

The level of product development reflects the total activity of a company. In fact, the goal of product development is not the product itself, but rather the successful business, it creates for the company. Therefore it is insufficient to concentrate only on product design, we must consider also market research and the establishment of production and sales.

As shown in Fig. 3, it is possible to describe activities in marketing, product design, and production that should be performed simultaneously to ensure a successful product development project, ANDREASEN and HEIN [1].

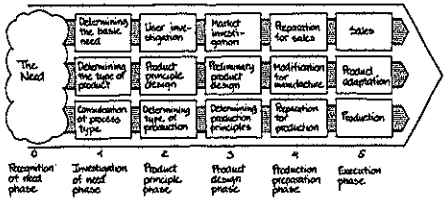


Fig. 3 The model of Integrated Product Development

On this level, we can explain the restrictions laid upon the product design by the customers and competitors (the product must be sell-able) and by production (the product must be produce-able). The starting point of the model of 'Integrated Product Development' is a rather undefined situation of need to be examined. The process is then divided into 5 phases to be completed in sequence by joint forces of marketing, product design and production (symbolized by the three arrows).

The amount of work to be fulfilled in each phase depends on the nature of the product: is it a totally new product type, a revised model, or production rationalizing? Special attention should be paid to the transition points between phases, as these are indicators of the progress and direction of the total project.

Working methods may be attached to the phases of integrated product development, e.g. specification methods in the 'investigation of need' phase and design review methods in the transition points.

2.4 Mechatronic product development

The three levels described above are related to respectively, the individual designer's way of working (problem solving), the technology of the product itself (product synthesis), and the situation of the company (product development). Considering mechatronic products, there are no differences from 'pure' machine design on the first and third level.

However we must expect difficulties, if we try to apply terms and methodology from machine design theory to mechatronics on the level of product synthesis. This is the level to be examined in the following.

3. DESIGN CHARACTERISTICS OF MECHANICS, ELECTRONICS AND SOFTWARE

There are significant differences between designing mechanics, electronics and software, not only in the technical skills required, but the very substance of design problems differ. The following sections compare design aspects of the three fields, and main points are summed up in Fig. 5.

3.1 Functions

The term function is used in all three fields. In machine design it is possible to describe the abstract function of a product as a series of transformations of material, energy or information, PAHL and BEITZ [6], ROTH [7] and others. However, designers do not consequently think of transformations, when using the word 'function'. They also equal the term with 'effects needed in a machine'. Indeed, there seems to be a duality between **transformation functions** and **purpose functions**, as stated in the design theories of HUBKA [3] and ANDREASEN [5].

In mechanical systems, information can not exist independently, but must be attached to the substance of either material (a punched card carries information) or energy (hydraulic pressure may carry information). In machine design, the handling of energy and material is emphasized compared to information aspects.

In electronics design, function may be completely described as transformations of electric properties (voltage level, currency, frequency etc.). In principle, they are transformations of information attached to energy, but the circuit designer will usually ignore the energy aspects and only regard the flow of signals. Later, energy aspects will pop up as a recurring nuisance: heat dissipation, non-ignoreable resistance in conductors, emitted electrical noise etc.

In software design, the abstract function of a program can be described as transformations of data and logical relations ('if... then do...'-type) between transformations. On each level of the program it is possible to distinguish between data to be transformed and control data. In software, data (information) can be handled independently of energy representation, even though it will be tied to electric properties, once the program is implemented in electronic hardware.

3.2 Concept design

Problems in mechanical concept design are usually 'open' that is, a large number of alternative solutions exist. Every problem is considered 'new' (the designer looks for unknown solutions), because only few solutions are 'recycled' in machine design. General solution catalogues of physical effects exist, ROTH [7], but they are not widely spread, as their use demand a high ability of abstraction from the designer, to fit the general principle of the catalogue with the actual design problem. Mechanical design requires great skills in creating and evaluating alternative solutions to individual problems.

In this aspect, electronics design is quite different. The innovation in circuit design lies in combining existing components and modules for a specific application. Electronic engineers seem to have a common fund of standard solutions to circuit problems to be found in literature, technical magazines, and component manufacturer's instructions. A thorough knowledge of the state-of-the-art of existing components and standard solutions is required for good electronics design.

In software, the main problem is to create a total structure of all the necessary transformations and bits of data. This structure is usually unique for a given product, where as program modules (e.g. algorithms) in principle could be used again in other designs. But for the time being, a standardized way of specifying the function and input/-output of modules does not exist, which means that categorizing is difficult, and recycling is severely limited. So in software also, problems are generally regarded as being 'new', requiring previous unknown solutions.

3.3 Physical realization

In machine design, the phases of concept design and embodiment design are closely intertwined. Except for the limited number of standard machine elements used, the mechanical designer has to specify every single part to be manufactured exclusively for his product. He specifies the **basic design properties** (form, dimensions, material and surface quality) in every parts drawing and the total structure in assembly drawings.

Realizing the electronic circuit diagram (called 'electronics packaging design') is in fact very similar to mechanical design. In industry it is sometimes carried out by electronics packaging engineers, who have specialized in electronics production technologies: printed circuit board, surface mount technology, connection and cabling techniques, hybrid circuits, IC-technology. Compared to machine design, the number of technologies is rather limited. Having specified the types of standard components to be used, the electronic packaging technologies and physical lay-outs will be detailed in drawings very similar to machine drawings. Usually, only the lay-outs and some specially designed components (e.g. custom specified IC) will be unique for the electronic product.

Software has no true production phase. While programming, the designer works directly on the media, and when completed, the program only has to be copied into the applied memory type (e.g. ROM's).

3.4 Design modelling

During the design process, design models serve different purposes, e.g. verifying functions, communicating ideas, documenting. Fig. 4 shows examples of design models in mechanics, electronics and software, BUUR and ANDREASEN [8].

Since mechanical design is 3-dimensional even in the concept phase, the designer needs perspec-

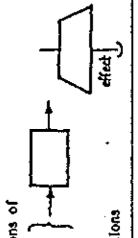
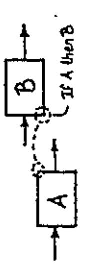
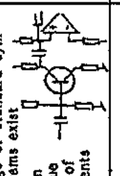
	MECHANICS	ELECTRONICS	SOFTWARE
FUNCTIONS	Transformations of material energy information Purpose functions 	Transformation of information attached to energy (electrical signals) 	Transformations of information Logical relations between transformations 
CONCEPT DESIGN	**Function principles and organic structure** 1. Each problem appears to be 'new' 2. Many alternative solutions exist for each sub-problem	**Module structure and circuit design** 1. Standard circuit concepts are available for most sub-problems 2. Large choice of standard components 	**Program structure and algorithms** 1. A set of basic operations exists 2. Recycling of algorithms is not common 3. Each problem appears to be 'new'
REALIZING THE CONCEPT	**Embodiment design and element geometry** 1. Large choice of production technologies 2. All elements must be specified (form, dimensions, material, surface) 	**Electronic packaging design** 1. Limited number of circuit technologies (PCB, hybrid circuits, IC etc.)	**Coding of the program** 1. There is no production phase but for program copying.
DESIGN MODELLING	1. For function modelling, elements have to be specially manufactured 2. Models on a concrete level (e.g. sketches, mock-ups, function models) are easily understood by non-specialists	1. A recognized language of standard symbols for circuit diagrams exist 2. Function modelling on breadboard is easy due to the large number of off-the-shelf components 	1. Until the coding phase, only graphical models are possible (paper or screen) 2. Checking of program function is easy by trial-runs on a standard computer 3. Program function is difficult to explain to non-specialists
DESIGN METHODS	1. Design methods exist 2. Methods are only to some extent accepted in industry	1. No methods for (analog) circuit design 2. Only very few methods available for packaging design 3. In industry, a bread-board approach is common, methods are seldom used	1. A number of methods exist 2. The necessity for design methods is generally recognized in industry
COMPUTER AIDS	1. Geometric models of present 2D and 3D systems can be applied only in the final design phases 2. Parametric design is possible for well-known, standardized design procedures	1. Circuit generation for logic design 2. PCB, hybrid and IC lay-out & simulation 3. Silicon compilers generate IC lay-out 4. Prototype simulation is possible	1. None for early design phases 2. Higher-level programming languages 3. Program language compilers 4. Graphical documentation systems exist

Fig. 5 Major characteristics of mechanical, electronical and software design

tive sketches and mock-ups to support his spatial perception. This is not the case with electronic circuits and software, both being basically 2-dimensional. On the other hand, sketches and 3D models belong to a rather concrete level of abstraction (close to the 'real world'), and this make them a potent means of communication with technical non-specialists: product users, managers, the marketing department etc.

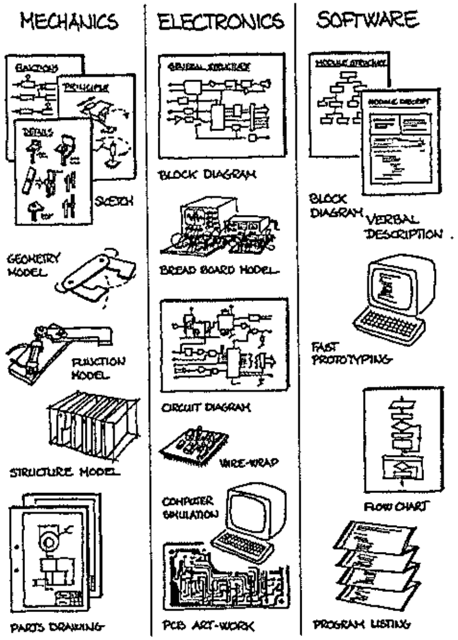


Fig. 4 Typical design models used when developing mechanics, electronics and software

Function modelling in mechanical design is limited by the fact that every part must be manufactured specially in a model workshop. Modifications are only accomplished slowly, as they require machining of altered or new parts.

Electronics engineers have an internationally accepted model tool for circuit design: the circuit diagram symbols. They are powerful for the purpose of discussion among specialists and for specification of the design on a principal level. For non-specialists they are too abstract to give any real understanding of function.

Function modelling in electronics is simple, since mostly standard components are used. Modifications are never farther away than the soldering iron and the component drawers. Also 'human interface' components that is, switches, buttons, lamps, displays etc. are easily available as standard parts, so the breadboard model may be turned into a function model for discussions with non-specialists (e.g. operation procedure and user feedback).

In software engineering, design modelling is a weak point. Software design is very abstract, and until entering the programming phase, the designer has nothing but verbal formulations and diagrams to describe the structure and function of the design. Even during programming, work may only be traced through program listings and input/output data.

This naturally causes frictions between software designers and non-specialists, as the function of the system can not be tested, until the design is nearly finished, and the program is installed in hardware - often far too late to make any serious alterations.

There have been some approaches to solve this problem, and they may generally be placed in two groups:

1. 'fast prototyping' that is, function modelling of early, rough ideas of the program, to get quick response from users, BOEHM [9], and
2. formalized design specification that is, applying mathematical tools in the specification phase to provide the designer with a complete and unambiguous basis for his work, MEYER [10] and NAUR [11]. However, the basic problem of providing models that are understandable for non-specialists, has not yet been solved here.

Even using fast prototyping, there is a necessity for the designer to make extensive use of graphics in order to communicate the function of his program to non-specialists (which may easily include the electrical and mechanical engineers of his own project team). As a paradox, software engineering with its high level programming languages and standardized computer hardware is by far the easiest of the three fields for function modelling with the possibility of trying alternative solutions directly from the keyboard.

3.5 Design methods

In this context, I will understand a **design method** as being a description of activities that supports the designer in proceeding from one stage of the design process to another. Or, in other words, from one design model to another. On the level of product synthesis, design methods must reflect the very nature of the product to be designed, according to the three level approach.

In machine design, a number of design methods for creating functional principles and transforming them into the physical structures of machine parts exist, particularly in German speaking countries, e.g. PAHL and BEITZ [6], ROTH [7]. Though introduced up to 20 years ago however, these methods are only partially accepted and employed by industry.

In electronics design, we must distinguish between circuit design and electronics packaging design. The easy access to function modelling of circuits has lead many electronics engineers to a breadboard approach to design: realize the first, intuitive idea in hardware and simply adjust component values and ad circuit fragments until design specifications can be met. It may have been a sufficient working method 15 years ago, but today's electronic products are far too complex for this, COOKE [12]. Structuring on block diagram level has gained importance, but hardly any methods exist for this activity.

In electronics packaging design, a few design methods for deciding on packaging technologies and module structure (circuit grouping) exist, HARPER [13]. Also methods for overcoming EMC problems and for high frequency design are available. In this phase, methods for embodiment design of mechanical products may well be applied, e.g. form concept variation methods as proposed by TJALVE [4]. Specific layout methods for PCB, thick-film and IC are well described and mostly computerized.

In the 1960's and 70's a number of failures in software programs proved that a more systematic approach to design was necessary than the 'direct-spaghetti-programming method'. Since then, design methods have been developed rather intensely. No single methodology has achieved overall acceptance, as the existing methods favor different types of software systems, e.g. data or function oriented, sequential or concurrent programming.

3.6 Computer Aids

The extent, to which computer aids have been developed for design, varies with the types of problems in the three fields.

In machine design, CAD can so far only be applied for embodiment design in the last design phases or for standardized design procedures, where each step is well defined. In CAD systems, the design is represented in a 2- or 3-dimensional geometric model, which prevents the use of the systems until dimensions and form of parts are considered. On the other hand computer tools certainly exist for solving a variety of detail problems in machine design: finite element analysis etc.

Electronic circuit design is fundamentally 2-dimensional, and the diagram symbols are well suited for computerization. A range of CAD systems are available for transforming logical statements into circuits, for simulating functions, or for merely documenting the design.

Electronics packaging design makes extensive use of CAD: generating PCB, thick-film or IClayout from the diagram, simulating and testing. In fact, CAD programs are indispensable in IC design, and they offer the possibility to shortcut the prototype stage by jumping directly from circuit diagram (or in the case of silicon compilers: functional specifications) to the first production series. For the purpose of function modelling or prototyping, the computer may instead simulate the presence of a programmed single chip microprocessor or a custom designed IC component in the actual hardware.

In software design, surprisingly few computerized design tools may be found. On the detailed design level, compilers may automatically transform high level programming languages into assembler or machine code needed for the chosen micro processor, but on the more abstract levels of systems structure design, computer aids are with few exceptions merely graphic documentation systems that help the designer in keeping track of and updating the huge number of working documents.

4. CONCLUSIONS

The difficulty of suggesting new, mechatronic design concepts are not just due to the gaps of understanding between engineers educated in mechanics, electronics and software design. The very substance of the three technologies are different.

When developing design methodology for mechatronics, we must concentrate on the level of product synthesis (i.e. design of technical systems). This is because the designer's problem solving and the company's product development activities do not differ from those applied to machine design, where design methods already exist.

There are two major obstacles in creating mechatronics design methodology:

(1) abstract functions of systems are understood and described very differently in mechanics, electronics and software, and that

(2) the means for describing design concepts (i.e. conditions of design modelling) vary greatly.

REFERENCES

[1] Andreasen, M.M. and Hein, L.: Integrated Product Development. IFS Publications Ltd, UK, Springer-Verlag 1987.

[2] Jones, J.C.: Design Methods - seeds of human futures. London Wiley 1970.

[3] Hubka, V.: Principles of Engineering Design. Butterworth & Co Publishers Ltd, UK 1982.

[4] Tjalve, E.: A Short Course in Industrial Design. Newnes-Butterworth & Co Publishers Ltd, UK 1979.

[5] Andreasen, M.M.: Syntesemetoder paa systemgrundlag. Diss. Lunds Tekniska Hogskola, Sweden 1980 (in Danish).

[6] Pahl, G. and Beitz, W.: Engineering Design. The Design Council, London 1984.

[7] Roth, K.: Konstruieren mit Konstruktionskatalogen. Springer-Verlag, Heidelberg 1982.

[8] Buur, J. and Andreasen, M.M.: Design Models in Mechatronic Product Development. Design Studies, 1989 (to be published).

[9] Boehm, B.W, Gray, T.E. and Seewaldt, T.: Prototyping versus specifying: A multiproject experiment. IEEE Trans. on Software Eng. Economics. Vol. SE-10, Nr.3 May 1984.

[10] Meyer, B.: On Formalism in Specifications. IEEE Software. Vol.2, 1985.

[11] Naur, P.: Formalization in Program Development. BIT. Vol.22, Nr.4 1982.

[12] Cooke, P.: Electronics Design Demands a New Approach. Engineering. Vol.224, Nr.4 1984.

[13] Harper, C.A.: Handbook of Electronics Packaging. McGraw-Hill, New York 1967.

Appendix B

Design models in mechatronic product development

Jacob Buur and Mogens Myrup Andreasen

Institute for Engineering Design, Bygn. 421 Technical University of Denmark, DK-2800 Lyngby, Denmark

Design models are important tools in mechatronics design. Based on well-known types of design models, like flow chart, design sketch, function model and prototype, this article describes important terms associated with modelling and proposes a model morphology as a convenient system for categorizing. In the early phases of product development, modelling is a way of buying information of the final product, and thereby diminishing the risk of making false decisions. However, mechatronics design is a complex activity involving the different fields of mechanics, electronics and software, and the article suggests that new design models describing the total mechatronic product in the initial design phases are needed. Based on the model morphology, it is possible to discuss properties of such models.

Keywords: design models, design process, mechatronics

The design of a 'mechatronic product' is a combination of mechanical, electronic and software engineering. This makes it a difficult task for the designer to maintain a total view of the product. The total view is important, since choosing any particular design principle in one field has consequences for the properties of the total product, e.g. ease of operation, marketing possibilities, production price and development cost.

Likewise, the communication within the project group is affected by the difference in engineering fields: specialized education makes it difficult to understand problems in the neighbouring field.

For the designer, design modelling is an important tool, because it permits him to describe, visualize and sculpture his thoughts for the benefit of both himself and others. As indicated in a survey carried out in Danish mechatronics industries, the success of a mechatronic design project does not only depend on the specialized skills of the designers but perhaps even more on their abilities to communicate and visualize their ideas to the rest of the project group.

This article describes a theory and a model morphology for better understanding of the modelling activity in mechatronics design. Also, the need for design models in the early phases of product development is discussed. This work is part of a PhD study at the Institute for Engineering Design of the Technical University of Denmark, with the goal of developing a design theory for mechatronics.

MODEL THEORY

The terms *property* and *model* are central to any theory of modelling. We will base this study on the definitions of the WDK school of design theory, Hubka[1]:

A *property* is any attribute or characteristic of an object.

Properties of products are, for example: performance, size, colour, reliability, costs. Properties represent the quality of the product in manufacturing, in marketing, in daily use etc. A product is designed with the purpose of possessing certain properties, and these will be prescribed as requirements in the design specification. The fundamental activities of design work are the generation of design alternatives and the evaluation according to *properties*.

Some properties are quantifiable, e.g. weight, speed, energy consumption. These can be measured directly and objectively. Others may be still quantifiable, but more difficult to measure directly, e.g. reliability, lifetime, ease of assembly, noise nuisance. Properties like appearance and ease of operation are not quantifiable. They can only be evaluated subjectively.

A *model* reproduces properties of an object.

The product designer creates models of the not yet finished product, e.g. mathematical formulae, verbal descriptions, sketches, function models. He does so in order to gain information about product properties: 'Is it strong enough?' 'Does it work?' 'How is the performance?' 'What does it look like?' Thus, the *object* is the product or the designer's idea of the product to be designed.

Design is a propagation from model to model, and the designer utilizes a large number of different 'product reproducing models' in order to design the product, Roth[2]. The design models vary in terms of the properties they reproduce, degree of abstraction, number of details, finish etc.

Figure 1 illustrates the relations between object and model. The model has a set of properties in common with the object, the *modelled properties*. For example, a function model displays the *functional* properties of the product, but not appearance and ease of operation.

But the model also holds properties that do not belong to the product, and which are irrelevant for modelling itself, e.g. a circuit diagram has the property of *structure* in common with the electronic circuit of the product, but it is drawn on paper and does not work.

EXAMPLES OF DESIGN MODELS

Technical universities extensively teach the use of verbal

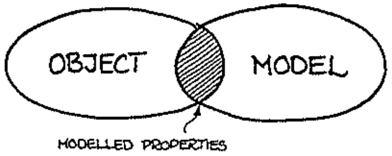


Figure 1. Relation between object and model

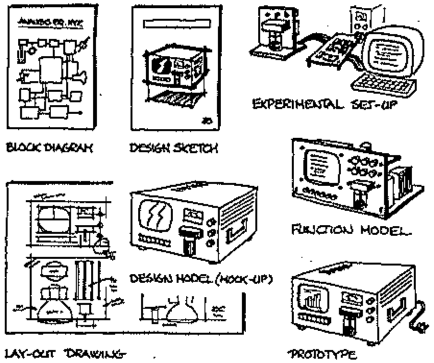


Figure 2. Design models of mechatronic products

models and formal, mathematical models, e.g. for strength analysis, dynamics and control theory. However, graphical models (sketches, diagrams) and three-dimensional ('hardware') models are often neglected, even though they may be of equal importance to the design engineer. In the following, we will concentrate on such types of models.

Figures 2 and 3 show design models used in mechatronics design. The graphics should not be interpreted as a complete registration of possible model types; rather, the examples were selected to indicate the broadness of design models.

The models sketched in Figure 2 reproduce *total aspects* of the product: abstract functional structure, function principles, appearance, form design, ease of operation, manufacturing properties etc.

Figure 3 shows design models from the traditional engineering fields of mechanics, electronics and software. A most interesting difference between these tools is that the software engineer has nothing but paper and computer display for design modelling, whereas the electronics engineer and, in particular, the mechanical engineer have a variety of hardware ready for building 3-D models.

MODEL MORPHOLOGY

To apply design models rationally requires a thorough understanding of modelled properties and modelling purpose. The main difficulty is to choose a model type that models just the necessary number of product properties at the present stage of design.

Including too many modelled properties in a model at an early design stage (e.g. the appearance of a function model), means that the designer will spend too much time and effort making the model. On the other hand, trying to evaluate more properties than the ones intended in the model (e.g. evaluating ergonomics of a design sketch) means that the designer may get a false impression of the solution to be examined.

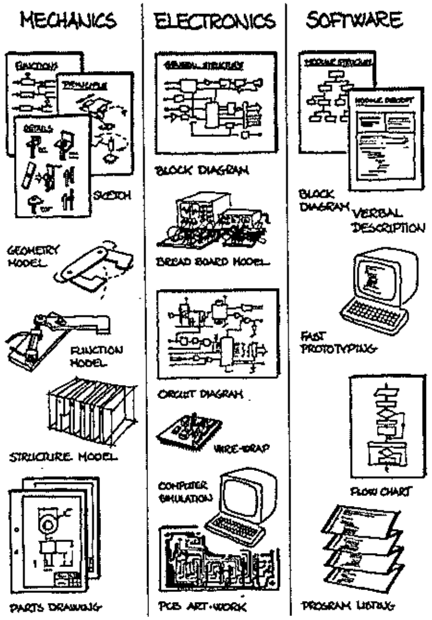


Figure 3. Design models in mechanics, electronics and software engineering

For a general explanation of the design modelling activity, we would like to propose the morphological schematic shown in Figure 4, based on Tjalve[3] and Andreasen[4]. Using this, it is possible to describe precisely the purpose of the intended modelling and the characteristics of a suitable design model type.

It is not a method that will automatically generate an optimal design model. Rather, it should be seen as a pedagogical tool, which makes the designer consider some important aspects before building a design model.

This model morphology is divided into two sections. The first describes the activity of design modelling, and the second describes the design model itself.

The modelling activity

Four aspects are important when discussing the design modelling activity: the object, the modelled properties, the purpose of modelling and the model user.

The *object* is the product to be designed or rather the designer's image of the product, e.g. the technological principles, the user interface, the structure of parts, the total form. At times, the designer will also want to model the product surroundings or the problem to be solved by the product. When designing hospital equipment, for instance, it may be helpful to model physical surroundings of an operating room and the sequences of operations as performed by surgeons and nurses.

The *modelled properties* are the product properties to be reproduced by the model. The properties in Figure 4 are arranged according to the 'life-cycle' of the product, Tjalve[5]: design – manufacturing – marketing – use – destruction. The emphasis on properties differs from product to product, but properties often modelled include: function, strength, ergonomics, production, reliability.

The *purpose* is what the designer wants to do with the modelled properties. It may be any of the basic operations of (engineering) design:

- define
- generate (ideas)
- describe
- verify
- evaluate

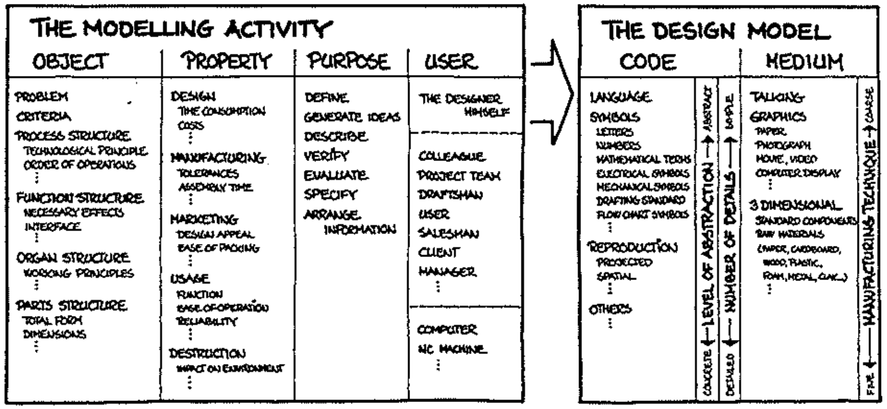


Figure 4. Morphology of design modelling characteristics

- specify
- arrange (information)

These are the operations that the designer performs again and again during the design process. For example, design sketches are used to *describe* and *generate* new ideas. A function model is used to *verify* or *evaluate* product function. A technical drawing will *specify* form, dimensions and manufacturing processes of machine parts.

The *user* of the design model may be either the designer himself, other persons (e.g. colleagues, product users, managers), or machines/computers. For the designer himself, there are only few restrictions on how to make the design model, as he will understand almost anything created by himself. Models intended for others however, need to be finished in a 'model language' or code understandable to those.

McKim[6] has carried out extensive work on the influence of sketching (graphical modelling) on the designer's imaginative power. He shows how it is possible for the designer to gain some control of his creative thought process by consciously following the cycle of seeing, imagining and drawing.

Also concerned with graphical models, Tovey[7] describes three principal functions of design drawing (purpose, model user):

- 'to facilitate the design process
- to externalize the process and thus allow others to participate in it such as design managers, other members of a design team or the client; and
- to communicate the completed design proposal to others this will be the client, marketing managers, production engineers, etc and so forth.'

Models or internal representations of design in computer programs are still gaining importance. Attempts to drag CAD 'upstream' in the design process call for models other than the strict geometrical models used today. Roth[8] describes the difference in characteristics of design models used by man and by computers: the 'Product reproducing models' used in manual design do not need to be true-to-scale and complete, as their main purpose for the designer is to create associations to knowledge, which is 'stored' elsewhere (in the designer's mind, in handbooks etc.). Contrary to this, computers have to extract all necessary data from the internal product model.

The design model

The design model may be regarded as part of a communication process as in Figure 5. It is a way of transferring information from a sender (the designer) to a receiver (the model user). The information (e.g. ideas, thoughts) is coded by the sender and again decoded by the receiver. The code must be well understood by both to ensure an optimal information transition, but still the sender may expect loss of information and noise.

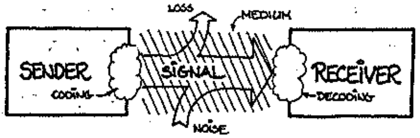


Figure 5. General communication process

The *code* of the model (or 'modelling language') is, for instance, human language, symbols, drafting standards, projections.

Outside the human brain, information can only exist in a medium, e.g. writing on paper, sound in air. The *medium* of the model is often graphical (e.g. paper, computer display, photograph) or three-dimensional (e.g. made of standard components and raw materials). Also, when making a model, the designer will decide the model *manufacturing technique*; that is, whether the medium is processed coarsely (i.e., cheap and fast) or very carefully (i.e. expensive, time consuming).

Two other characteristics are important, when describing a design model: the *degree of abstraction* and the *number of details* (or complexity). Figure 6 illustrates these terms.

The upper left corner shows a rough, two-dimensional sketch of a pulley for a conveyer belt. To make the model less abstract, the designer draws a perspective drawing or makes a hardware model. To make it more detailed, the designer must add design considerations, e.g. divide the form into single, processable parts, specify dimensions and material.

Examples

We would like to give some examples of how the modelling activity associated with well-known design model types may be described using the model morphology of Figure 4.

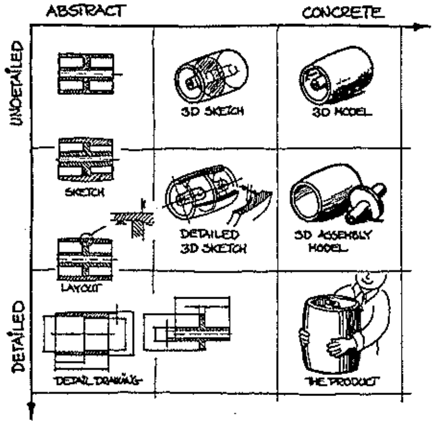


Figure 6. Degree of abstraction and number of details reproduced by a design model

A *function model* is a model of the 'working principles' of the product. Its purpose is to 'verify' ('Does it work?') or to 'evaluate' ('How is the performance?') the 'function properties'. It will often be used only by the 'designer himself' and by the 'project team'. The model is a 'spatial reproduction' composed of 'standard components' and 'raw materials'.

A *circuit diagram* models the *structure of electronic parts* of the product. The purpose is, for example, to *specify manufacturing properties* (e.g. types of components to be used, assembly) to be used by the *manufacturing department*. It is drawn on *paper* using *electrical symbols*.

The *design specification* may also be regarded as a model reproducing the *problem* and the *criteria* for the product to be designed. The purpose is to *specify* product properties concerning *manufacturing, marketing, usage* and *destruction*, but it may also be used for *evaluating* properties like expected *costs* and *time consumption* of the design project. The design specification is written on *paper* using *language* and *symbols*.

MODELLING: A WAY OF BUYING INFORMATION

By deciding the properties of the product, the designer establishes the size of the business that this product can achieve for the company. According to Andreasen and Hein[9], we may roughly divide product properties into two categories: the 'functional properties' (functionality, operation, appearance etc.), which determine the achievable sales price and volume, and the 'production properties' (raw materials, manufacturing processes etc.), which determine the production costs. By handling at the same time principles, structure and details of the product, the designer will try to make the gap between sales turnover and production costs as wide as possible, thus optimizing the business potential.

As illustrated in Figure 7, the main part of the project costs are allocated in the early phases of product development, at a time when few resources (manpower, money) are actually spent on the project. When completing the design phase, a high percentage of the cost may be allocated, although only very few percent are spent until then. In other words, the designer has to make several very important decisions for which he does not see the full consequences until much later.

During the early design stages, every decision and choice of solution is based on incomplete information, since the knowledge of the design problem grows with the progress of the project. Design modelling is one tool for buying information in order to reduce the risk of the product development not succeeding. By making models, the designer tries to gain information about the relations between decisions and consequences. In software engineering literature, there is a growing recognition of the importance of 'fast prototyping', that is, early function modelling of the main program ideas in order to obtain a quick response from users etc. Boehm[10] gives five conditions for buying information by modelling ('investigations'), which have general value also for other fields than software design:

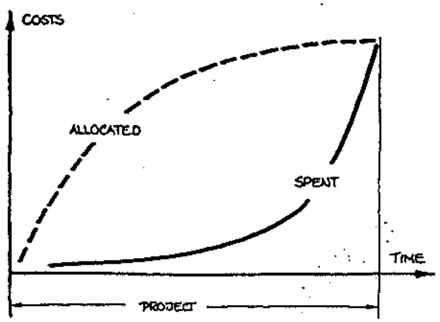


Figure 7. Relations between allocated and actually spent costs during a design project

> Condition 1: There exist attractive alternatives whose payoff varies greatly, depending on some critical states of nature.
>
> Condition 2: The critical states of nature have an appreciable probability of occurring.
>
> Condition 3: The investigations have a high probability of accurately identifying the occurrence of the critical states of nature.
>
> Condition 4: The required cost and schedule of the investigations do not overly curtail their net value.
>
> Condition 5: There exist significant side benefits derived from performing the investigations.

Thus, one should be cautious of using design models if no alternative solutions exist, if the benefit is expected to be small or if modelling in time or costs is more expensive than the obtained benefits.

An example of modelling being too expensive, is the production of custom-designed IC: the designer will choose to jump directly to a small series production, since making a single prototype IC for evaluation of function is far too expensive. Instead, other tools have been developed for this purpose: computer simulation and in-circuit emulators.

Often the *time consumption* of modelling will be much more critical than the costs. The design model is part of a creative design process, and it should preferably be finished immediately after the designer has felt the need. So, queues in the model workshop are fatal for creativity!

The *side benefits* of modelling may be extremely valuable: knowledge of new technologies, of practical handling of materials, of production difficulties (assembly, required precision) etc. Boehm[10] mentions training, team building and customer relations as side benefits.

- design costs and time consumption
- manufacturing costs and investments
- marketing properties (customer appeal, innovative value)

The *purpose* of such models should be to:

- generate and describe new (design concept) ideas
- evaluate roughly the consequences.

Therefore, flexibility is an important feature.

As *user* of design concept models, we may consider two different categories:

- colleagues (mechanical/electronic/software designers)
- non-specialists (marketing, production, managers, users)

Thinking of software development, even designers in the same project group may have to be regarded as 'non-specialists' in the sense that they have had no training in understanding programming possibilities and restrictions.

There is certainly not a single model type that will yield to all these specifications. Rather, we are looking for a family of models, of which we have already seen some elements in use in industry. However, a brief discussion of the model characteristics of code and medium may indicate possibilities.

Code: Mechanical, electrical and software engineers do not have a common language of symbols (except mathematical symbols) to describe function. Thus, it may be necessary to develop new symbols for the 'colleague-type' models. Models for non-specialists need an easy understandable code, and this may not be possible at all with symbols.

Medium: To fulfil the purpose of generating design concept alternatives, the models must be flexible (easy to modify) or fast to make. In this respect, graphical models (sketches, computer graphics) seem to have advantages. Models for non-specialists need to be concrete, and should make good use of graphics and three-dimensional hardware.

A key point when discussing mechatronic design concept models is the understanding of 'software'. The term is commonly used for computer programs and their documentation but, in order to derive combined mechatronic models of function, we need to reconsider the logical contents in machines, the 'mechanical software'. When replacing mechanical principles of a machine with microprocessors and programs, the designer is forced to formulate logical relations and algorithms that were formerly built into the mechanical system. These logical relations are seldom expressed explicitly in machine design, but they must be included when evaluating alternative mechatronic design concepts.

CONCLUSIONS

Used with care, design modelling is a powerful tool for buying information and thus diminishing risk in mechatronic product development. It is possible to describe precisely the design modelling activity and model type by the introduced model morphology.

The design concept of a mechatronic product is a complex blend of mechanical, electronic and software technologies, which is established in the early stages of a design project. The well-known function model is not sufficient to describe and discuss the design concept, and we see the need for a new family of early design concept models in mechatronics design.

These must be flexible models, to be used for generating design concept alternatives and for roughly evaluating the consequences for development, manufacturing and marketing in the early stages of a project. Further, there is a need for a model language or a model type that could improve communication both between mechanical, electrical and software engineers, and between the project team and, for example, managers and users.

REFERENCES

1 **Hubka, V** *Terminology of the science of design engineering in 6 languages* Schriftenreihe WDK 3, Heurista (1980)

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

3 **Tjalve, E et al.** *Engineering graphic modelling* Newnes–Butterworths, London (1979)

4 **Andreasen, M M** *Syntesemetoder paa systemgrundlag* (in Danish) Diss. Lunds Tekniska Högskola, Sweden (1980)

5 **Tjalve, E** *A short course in industrial design* Newnes–Butterworths, London (1979)

6 **McKim, R H** *Thinking visually: a strategy manual for problem solving* Wadsworth (1980)

7 **Tovey, M** 'Thinking styles and modelling systems' *Design Studies* Vol 7 No 1 (January 1986)

8 **Roth, K** 'Modellbildung für das methodische Konstruieren ohne und mit Rechnerunterstützung' *VDI Z.* Bd 128 No 1/2 (January 1986)

9 **Andreasen, M M and Hein, L** *Integrated product development* IFS Publications Ltd/Springer Verlag, Bedford (1987)

10 **Boehm, B W** 'Software engineering economics' *IEEE Trans. Software Eng.* Vol SE–10 No 1 (January 1984)

11 **Finkelstein, L and Finkelstein A C W** 'Review of design methodology' *IEE Proceedings* Vol 130 Pt A No 4 (June 1983)

12 **Heinzl, J** 'Entwicklungsmethodik für Geräete mit Steuerung durch Mikroelektronik' *VDI Berichte* No 515 (1984)

electronic products are far too complex for this, COOKE [12]. Structuring on block diagram level has gained importance, but hardly any methods exist for this activity.

In electronics packaging design, a few design methods for deciding on packaging technologies and module structure (circuit grouping) exist, HARPER [13]. Also methods for overcoming EMC problems and for high frequency design are available. In this phase, methods for embodiment design of mechanical products may well be applied, e.g. form concept variation methods as proposed by TJALVE [4]. Specific layout methods for PCB, thick-film and IC are well described and mostly computerized.

In the 1960's and 70's a number of failures in software programs proved that a more systematic approach to design was necessary than the 'direct-spaghetti-programming method'. Since then, design methods have been developed rather intensely. No single methodology has achieved overall acceptance, as the existing methods favor different types of software systems, e.g. data or function oriented, sequential or concurrent programming.

3.6 Computer Aids

The extent, to which computer aids have been developed for design, varies with the types of problems in the three fields.

In machine design, CAD can so far only be applied for embodiment design in the last design phases or for standardized design procedures, where each step is well defined. In CAD systems, the design is represented in a 2- or 3-dimensional geometric model, which prevents the use of the systems until dimensions and form of parts are considered. On the other hand computer tools certainly exist for solving a variety of detail problems in machine design: finite element analysis etc.

Electronic circuit design is fundamentally 2-dimensional, and the diagram symbols are well suited for computerization. A range of CAD systems are available for transforming logical statements into circuits, for simulating functions, or for merely documenting the design.

Electronics packaging design makes extensive use of CAD: generating PCB, thick-film or IClayout from the diagram, simulating and testing. In fact, CAD programs are indispensable in IC design, and they offer the possibility to shortcut the prototype stage by jumping directly from circuit diagram (or in the case of silicon compilers: functional specifications) to the first production series. For the purpose of function modelling or prototyping, the computer may instead simulate the presence of a programmed single chip microprocessor or a custom designed IC component in the actual hardware.

In software design, surprisingly few computerized design tools may be found. On the detailed design level, compilers may automatically transform high level programming languages into assembler or machine code needed for the chosen micro processor, but on the more abstract levels of systems structure design, computer aids are with few exceptions merely graphic documentation systems that help the designer in keeping track of and updating the huge number of working documents.

4. CONCLUSIONS

The difficulty of suggesting new, mechatronic design concepts are not just due to the gaps of understanding between engineers educated in mechanics, electronics and software design. The very substance of the three technologies are different.

When developing design methodology for mechatronics, we must concentrate on the level of product synthesis (i.e. design of technical systems). This is because the designer's problem solving and the company's product development activities do not differ from those applied to machine design, where design methods already exist.

There are two major obstacles in creating mechatronics design methodology:

(1) abstract functions of systems are understood and described very differently in mechanics, electronics and software, and that

(2) the means for describing design concepts (i.e. conditions of design modelling) vary greatly.

REFERENCES

[1] Andreasen, M.M. and Hein, L.: Integrated Product Development. IFS Publications Ltd, UK, Springer-Verlag 1987.

[2] Jones, J.C.: Design Methods - seeds of human futures. London Wiley 1970.

[3] Hubka, V.: Principles of Engineering Design. Butterworth & Co Publishers Ltd, UK 1982.

[4] Tjalve, E.: A Short Course in Industrial Design. Newnes-Butterworth & Co Publishers Ltd, UK 1979.

[5] Andreasen, M.M.: Syntesemetoder paa systemgrundlag. Diss. Lunds Tekniska Hogskola, Sweden 1980 (in Danish).

[6] Pahl, G. and Beitz, W.: Engineering Design. The Design Council, London 1984.

[7] Roth, K.: Konstruieren mit Konstruktionskatalogen. Springer-Verlag, Heidelberg 1982.

[8] Buur, J. and Andreasen, M.M.: Design Models in Mechatronic Product Development. Design Studies, 1989 (to be published).

[9] Boehm, B.W, Gray, T.E. and Seewaldt, T.: Prototyping versus specifying: A multiproject experiment. IEEE Trans. on Software Eng. Economics. Vol. SE-10, Nr.3 May 1984.

[10] Meyer, B.: On Formalism in Specifications. IEEE Software. Vol.2, 1985.

[11] Naur, P.: Formalization in Program Development. BIT. Vol.22, Nr.4 1982.

[12] Cooke, P.: Electronics Design Demands a New Approach. Engineering. Vol.224, Nr.4 1984.

[13] Harper, C.A.: Handbook of Electronics Packaging. McGraw-Hill, New York 1967.

Appendix B

Design models in mechatronic product development

Jacob Buur and Mogens Myrup Andreasen

Institute for Engineering Design, Bygn. 421 Technical University of Denmark, DK–2800 Lyngby, Denmark

Design models are important tools in mechatronics design. Based on well-known types of design models, like flow chart, design sketch, function model and prototype, this article describes important terms associated with modelling and proposes a model morphology as a convenient system for categorizing. In the early phases of product development, modelling is a way of buying information of the final product, and thereby diminishing the risk of making false decisions. However, mechatronics design is a complex activity involving the different fields of mechanics, electronics and software, and the article suggests that new design models describing the total mechatronic product in the initial design phases are needed. Based on the model morphology, it is possible to discuss properties of such models.

Keywords: design models, design process, mechatronics

The design of a 'mechatronic product' is a combination of mechanical, electronic and software engineering. This makes it a difficult task for the designer to maintain a total view of the product. The total view is important, since choosing any particular design principle in one field has consequences for the properties of the total product, e.g. ease of operation, marketing possibilities, production price and development cost.

Likewise, the communication within the project group is affected by the difference in engineering fields: specialized education makes it difficult to understand problems in the neighbouring field.

For the designer, design modelling is an important tool, because it permits him to describe, visualize and sculpture his thoughts for the benefit of both himself and others. As indicated in a survey carried out in Danish mechatronics industries, the success of a mechatronic design project does not only depend on the specialized skills of the designers but perhaps even more on their abilities to communicate and visualize their ideas to the rest of the project group.

This article describes a theory and a model morphology for better understanding of the modelling activity in mechatronics design. Also, the need for design models in the early phases of product development is discussed. This work is part of a PhD study at the Institute for Engineering Design of the Technical University of Denmark, with the goal of developing a design theory for mechatronics.

MODEL THEORY

The terms *property* and *model* are central to any theory of modelling. We will base this study on the definitions of the WDK school of design theory, Hubka[1]:

A *property* is any attribute or characteristic of an object.

Properties of products are, for example: performance, size, colour, reliability, costs. Properties represent the quality of the product in manufacturing, in marketing, in daily use etc. A product is designed with the purpose of possessing certain properties, and these will be prescribed as requirements in the design specification. The fundamental activities of design work are the generation of design alternatives and the evaluation according to *properties*.

Some properties are quantifiable, e.g. weight, speed, energy consumption. These can be measured directly and objectively. Others may be still quantifiable, but more difficult to measure directly, e.g. reliability, lifetime, ease of assembly, noise nuisance. Properties like appearance and ease of operation are not quantifiable. They can only be evaluated subjectively.

A *model* reproduces properties of an object.

The product designer creates models of the not yet finished product, e.g. mathematical formulae, verbal descriptions, sketches, function models. He does so in order to gain information about product properties: 'Is it strong enough?' 'Does it work?' 'How is the performance?' 'What does it look like?' Thus, the *object* is the product or the designer's idea of the product to be designed.

Design is a propagation from model to model, and the designer utilizes a large number of different 'product reproducing models' in order to design the product, Roth[2]. The design models vary in terms of the properties they reproduce, degree of abstraction, number of details, finish etc.

Figure 1 illustrates the relations between object and model. The model has a set of properties in common with the object, the *modelled properties*. For example, a function model displays the *functional* properties of the product, but not appearance and ease of operation.

But the model also holds properties that do not belong to the product, and which are irrelevant for modelling itself, e.g. a circuit diagram has the property of *structure* in common with the electronic circuit of the product, but it is drawn on paper and does not work.

EXAMPLES OF DESIGN MODELS

Technical universities extensively teach the use of verbal models and formal, mathematical models, e.g. for strength analysis, dynamics and control theory. However, graphical models (sketches, diagrams) and three-dimensional ('hardware') models are often neglected, even though they may be of equal importance to the design engineer. In the following, we will concentrate on such types of models.

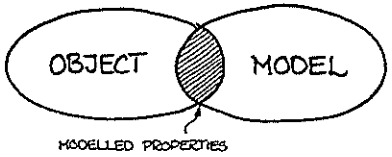


Figure 1. Relation between object and model

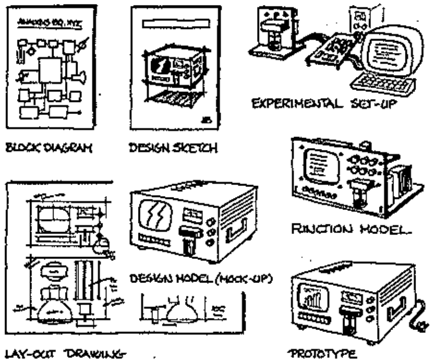


Figure 2. Design models of mechatronic products

Figures 2 and 3 show design models used in mechatronics design. The graphics should not be interpreted as a complete registration of possible model types; rather, the examples were selected to indicate the broadness of design models.

The models sketched in Figure 2 reproduce *total aspects* of the product: abstract functional structure, function principles, appearance, form design, ease of operation, manufacturing properties etc.

Figure 3 shows design models from the traditional engineering fields of mechanics, electronics and software. A most interesting difference between these tools is that the software engineer has nothing but paper and computer display for design modelling, whereas the electronics engineer and, in particular, the mechanical engineer have a variety of hardware ready for building 3-D models.

MODEL MORPHOLOGY

To apply design models rationally requires a thorough understanding of modelled properties and modelling purpose. The main difficulty is to choose a model type that models just the necessary number of product properties at the present stage of design.

Including too many modelled properties in a model at an early design stage (e.g. the appearance of a function model), means that the designer will spend too much time and effort making the model. On the other hand, trying to evaluate more properties than the ones intended in the model (e.g. evaluating ergonomics of a design sketch) means that the designer may get a false impression of the solution to be examined.

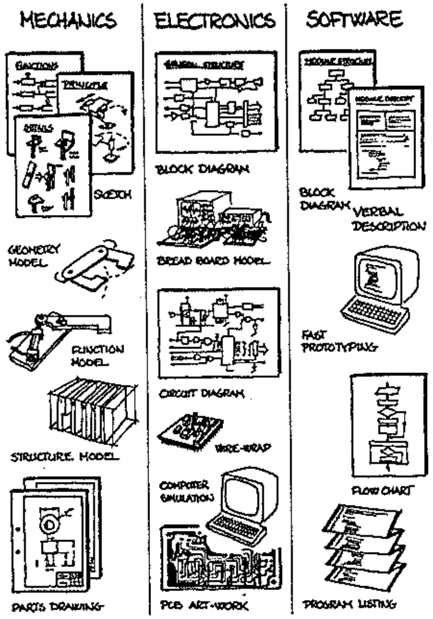


Figure 3. Design models in mechanics, electronics and software engineering

For a general explanation of the design modelling activity, we would like to propose the morphological schematic shown in Figure 4, based on Tjalve[3] and Andreasen[4]. Using this, it is possible to describe precisely the purpose of the intended modelling and the characteristics of a suitable design model type.

It is not a method that will automatically generate an optimal design model. Rather, it should be seen as a pedagogical tool, which makes the designer consider some important aspects before building a design model.

This model morphology is divided into two sections. The first describes the activity of design modelling, and the second describes the design model itself.

The modelling activity

Four aspects are important when discussing the design modelling activity: the object, the modelled properties, the purpose of modelling and the model user.

The *object* is the product to be designed or rather the designer's image of the product, e.g. the technological principles, the user interface, the structure of parts, the total form. At times, the designer will also want to model the product surroundings or the problem to be solved by the product. When designing hospital equipment, for instance, it may be helpful to model physical surroundings of an operating room and the sequences of operations as performed by surgeons and nurses.

The *modelled properties* are the product properties to be reproduced by the model. The properties in Figure 4 are arranged according to the 'life-cycle' of the product, Tjalve[5]: design – manufacturing – marketing – use – destruction. The emphasis on properties differs from product to product, but properties often modelled include: function, strength, ergonomics, production, reliability.

The *purpose* is what the designer wants to do with the modelled properties. It may be any of the basic operations of (engineering) design:

- define
- generate (ideas)
- describe
- verify
- evaluate

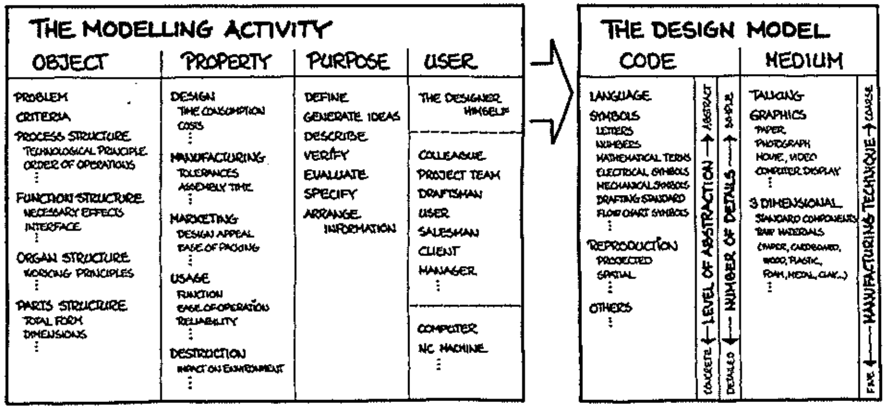


Figure 4. Morphology of design modelling characteristics

- specify
- arrange (information)

These are the operations that the designer performs again and again during the design process. For example, design sketches are used to *describe* and *generate* new ideas. A function model is used to *verify* or *evaluate* product function. A technical drawing will *specify* form, dimensions and manufacturing processes of machine parts.

The *user* of the design model may be either the designer himself, other persons (e.g. colleagues, product users, managers), or machines/computers. For the designer himself, there are only few restrictions on how to make the design model, as he will understand almost anything created by himself. Models intended for others however, need to be finished in a 'model language' or code understandable to those.

McKim[6] has carried out extensive work on the influence of sketching (graphical modelling) on the designer's imaginative power. He shows how it is possible for the designer to gain some control of his creative thought process by consciously following the cycle of seeing, imagining and drawing.

Also concerned with graphical models, Tovey[7] describes three principal functions of design drawing (purpose, model user):

- '• to facilitate the design process
- to externalize the process and thus allow others to participate in it such as design managers, other members of a design team or the client; and
- to communicate the completed design proposal to others this will be the client, marketing managers, production engineers, etc and so forth.'

Models or internal representations of design in computer programs are still gaining importance. Attempts to drag CAD 'upstream' in the design process call for models other than the strict geometrical models used today. Roth[8] describes the difference in characteristics of design models used by man and by computers: the 'Product reproducing models' used in manual design do not need to be true-to-scale and complete, as their main purpose for the designer is to create associations to knowledge, which is 'stored' elsewhere (in the designer's mind, in handbooks etc.). Contrary to this, computers have to extract all necessary data from the internal product model.

The design model

The design model may be regarded as part of a communication process as in Figure 5. It is a way of transferring information from a sender (the designer) to a receiver (the model user). The information (e.g. ideas, thoughts) is coded by the sender and again decoded by the receiver. The code must be well understood by both to ensure an optimal information transition, but still the sender may expect loss of information and noise.

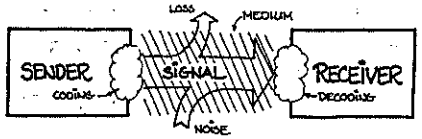


Figure 5. General communication process

The *code* of the model (or 'modelling language') is, for instance, human language, symbols, drafting standards, projections.

Outside the human brain, information can only exist in a medium, e.g. writing on paper, sound in air. The *medium* of the model is often graphical (e.g. paper, computer display, photograph) or three-dimensional (e.g. made of standard components and raw materials). Also, when making a model, the designer will decide the model *manufacturing technique*; that is, whether the medium is processed coarsely (i.e., cheap and fast) or very carefully (i.e. expensive, time consuming).

Two other characteristics are important, when describing a design model: the *degree of abstraction* and the *number of details* (or complexity). Figure 6 illustrates these terms.

The upper left corner shows a rough, two-dimensional sketch of a pulley for a conveyer belt. To make the model less abstract, the designer draws a perspective drawing or makes a hardware model. To make it more detailed, the designer must add design considerations, e.g. divide the form into single, processable parts, specify dimensions and material.

Examples

We would like to give some examples of how the modelling activity associated with well-known design model types may be described using the model morphology of Figure 4.

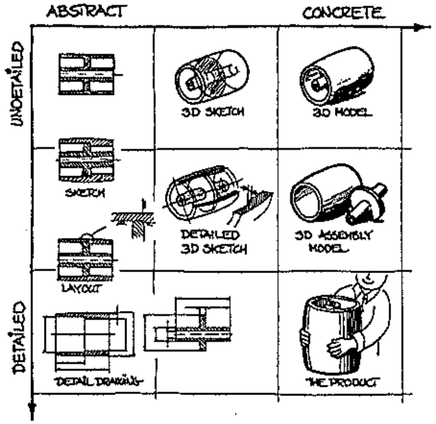


Figure 6. Degree of abstraction and number of details reproduced by a design model

A *function model* is a model of the 'working principles' of the product. Its purpose is to 'verify' ('Does it work?') or to 'evaluate' ('How is the performance?') the 'function properties'. It will often be used only by the 'designer himself' and by the 'project team'. The model is a 'spatial reproduction' composed of 'standard components' and 'raw materials'.

A *circuit diagram* models the *structure of electronic parts* of the product. The purpose is, for example, to *specify manufacturing properties* (e.g. types of components to be used, assembly) to be used by the *manufacturing department*. It is drawn on *paper* using *electrical symbols*.

The *design specification* may also be regarded as a model reproducing the *problem* and the *criteria* for the product to be designed. The purpose is to *specify* product properties concerning *manufacturing, marketing, usage* and *destruction*, but it may also be used for *evaluating* properties like expected *costs* and *time consumption* of the design project. The design specification is written on *paper* using *language* and *symbols*.

MODELLING: A WAY OF BUYING INFORMATION

By deciding the properties of the product, the designer establishes the size of the business that this product can achieve for the company. According to Andreasen and Hein[9], we may roughly divide product properties into two categories: the 'functional properties' (functionality, operation, appearance etc.), which determine the achievable sales price and volume, and the 'production properties' (raw materials, manufacturing processes etc.), which determine the production costs. By handling at the same time principles, structure and details of the product, the designer will try to make the gap between sales turnover and production costs as wide as possible, thus optimizing the business potential.

As illustrated in Figure 7, the main part of the project costs are allocated in the early phases of product development, at a time when few resources (manpower, money) are actually spent on the project. When completing the design phase, a high percentage of the cost may be allocated, although only very few percent are spent until then. In other words, the designer has to make several very important decisions for which he does not see the full consequences until much later.

During the early design stages, every decision and choice of solution is based on incomplete information, since the knowledge of the design problem grows with the progress of the project. Design modelling is one tool for buying information in order to reduce the risk of the product development not succeeding. By making models, the designer tries to gain information about the relations between decisions and consequences. In software engineering literature, there is a growing recognition of the importance of 'fast prototyping', that is, early function modelling of the main program ideas in order to obtain a quick response from users etc. Boehm[10] gives five conditions for buying information by modelling ('investigations'), which have general value also for other fields than software design:

> Condition 1: There exist attractive alternatives whose payoff varies greatly, depending on some critical states of nature.
>
> Condition 2: The critical states of nature have an appreciable probability of occurring.
>
> Condition 3: The investigations have a high probability of accurately identifying the occurrence of the critical states of nature.
>
> Condition 4: The required cost and schedule of the investigations do not overly curtail their net value.
>
> Condition 5: There exist significant side benefits derived from performing the investigations.

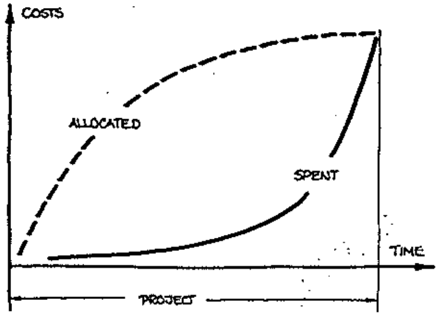


Figure 7. Relations between allocated and actually spent costs during a design project

Thus, one should be cautious of using design models if no alternative solutions exist, if the benefit is expected to be small or if modelling in time or costs is more expensive than the obtained benefits.

An example of modelling being too expensive, is the production of custom-designed IC: the designer will choose to jump directly to a small series production, since making a single prototype IC for evaluation of function is far too expensive. Instead, other tools have been developed for this purpose: computer simulation and in-circuit emulators.

Often the *time consumption* of modelling will be much more critical than the costs. The design model is part of a creative design process, and it should preferably be finished immediately after the designer has felt the need. So, queues in the model workshop are fatal for creativity!

The *side benefits* of modelling may be extremely valuable: knowledge of new technologies, of practical handling of materials, of production difficulties (assembly, required precision) etc. Boehm[10] mentions training, team building and customer relations as side benefits.

NEW DESIGN CONCEPT MODELS NEEDED

Figure 8 shows types of often-used hardware models in the rough sequence of a mechatronic product development project. The purpose and the modelled properties of each design model are briefly described.

The relations between mechanics, electronics and software in a mechatronic product are laid down in the early phases of product development. By choosing function principles, the designer decides which parts of the product (of the total product function) should be realized by using mechanical, electronic or software means. In agreement with Finkelstein[11] we will use the term *design concept* for this allocation of technologies or the overall idea of working principles.

Establishing the design concept is a task that requires broad knowledge of technologies within all three fields in order to create the largest possible business for the company. The choice of design concept, however, is not a purely technical matter, as it is closely related to company strategy (e.g. product families, corporate image, know-how building).

Poor communication between designers of specialized fields, though, means that companies tend to divide the product into independent, well-defined design tasks of respectively mechanics, electronics and software as early as possible. By doing so, there may be a substantial risk of sub-optimizing the individual modules on the premises of technology, rather than optimizing the total product and the business potential of the project.

We believe that new model types are needed for this phase of creating the design concept in mechatronics design. One could argue that this problem of the design concept seems to be associated only with development of new product ideas, whereas most design projects in industry consist of updating or modifying existing products. But, in fact, updating or modifying products usually means altering the design concept because of new technological achievements by shifting the weight from, for example, mechanics to electronics, from analogue to digital electronics, from hardware to flexible software, and from general microprocessor solutions to dedicated custom ICs.

But does not the function model of Figure 8 reflect the design concept of the product, then? It does, but the following circumstances emphasize the need for new design concept model types in the early phases of mechatronics design.

- The function model will mostly be the first time for mechanics, electronics and software to meet. It is the first occasion for the total design concept to be tested.
- The function model is made at a comparatively late stage of the project, which means at a time when most important decisions regarding the design concept have already been taken.
- In many companies, the function model has a status of project key-point, at which the project group presents results to the management level. Thus, real alternative concepts hardly exist at this time, and only details will be altered.

Figure 9 shows examples of models used at the early stages of discussing the design concept in industry projects. They have in common that they are all made long before a function model could be realized, and they are designed with flexibility in mind, because their most distinguished purpose is to aid the designer in generating and evaluating alternative ideas.

The first group of examples use computers for simulating the abstract idea of *product function in real surroundings*. First, an algorithm for automatic control of a central heating system, tested in a villa fully equipped with heating. Second, a concept of intelligent communication between coin-operated telephone and telephone exchange, operated on the actual telephone network.

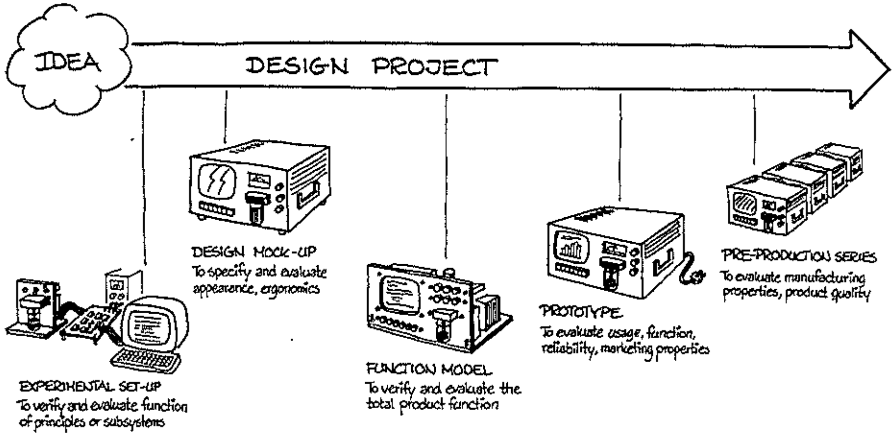


Figure 8. Typical hardware models in a mechatronic product design project

The next group also contains, in a way, models concerned with the 'real surroundings' of the product: the *user interface*. Three examples: the first is simply an operation manual based on the pure imagination of the product, written and evaluated before any hardware model exists. This may yield valuable information about the situation of the user and the logical dialogue between product and user.

The second uses computer graphics to create a two-dimensional image of the product front panel, simulating response from displays and indicator lamps. An easy-touch panel gives the user the possibility of really trying directly push-button type inputs to the simulator. Creating alternative designs is merely a question of redefining computer graphics.

The third example is building an empty (cardboard) mock-up of the product with operation elements and displays connected to a computer. This model type may simulate a full operation dialogue and also allow for the evaluation of ergonomic features of operation. Again, model changes are a matter of reprogramming the computer.

The last group of Figure 9 is a block diagram type suggested by Heinzl[12] for modelling the *functional structure* of a machine with electronic control. It illustrates that an electronic control system can only interact with the machine by means of sensors (input) and actuators (output). Likewise, interaction with man is possible only through operation controls (input) and indicators (output). By thinking in these elements, the designer becomes conscious of the borderline between electronics and mechanics, and between electronics and operator.

SPECIFYING NEW MODEL TYPES

Using the headings of the model morphology of Figure 7, it is possible to describe characteristics of such new model types, which are attractive in the early stages of mechatronic product development.
The *objects* we need to model are:

- the functional structure
- the applied principles of mechanics/electronics/software
- the interfaces between the three technologies.

This is what is special about mechatronics design: we need to look for solutions in all three fields, and the design concept will be a blend of technologies with complex interfaces between subsystems.

As a basis for allocating technological principles to the different functions, we need to model the abstract functional structure of the product, and we need to do so without the model itself favouring a particular technology (e.g. a flowchart favours software type solutions).

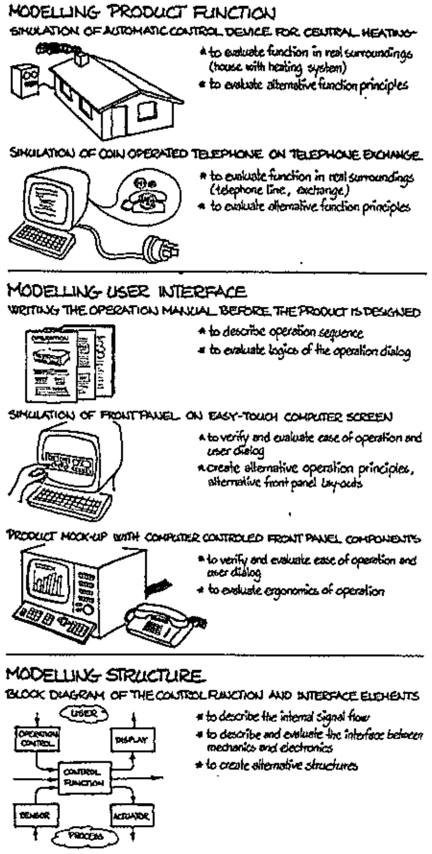


Figure 9. Examples of design models used for discussing the design concept in industry projects

This is possible for machine systems (Roth[2], Andreasen[4] and others), but a single model type for describing functional structure of mechatronic systems has yet to be developed.

Likewise, in the phase of discussing principles to be applied in the design concept, the designer should be able to jump freely between mechanical, electronic and software technologies, but so far a single, practical model language to describe, for example, the combination of a mechanism, a stepmotor, a microprocessor and a control algorithm, does not exist. The *modelled properties* of design concept models are not so much function itself (does it work?) at this stage, rather they are:

- the relations between product and surroundings (does this concept solve the problem? user properties?)

- design costs and time consumption
- manufacturing costs and investments
- marketing properties (customer appeal, innovative value)

The *purpose* of such models should be to:

- generate and describe new (design concept) ideas
- evaluate roughly the consequences.

Therefore, flexibility is an important feature.

As *user* of design concept models, we may consider two different categories:

- colleagues (mechanical/electronic/software designers)
- non-specialists (marketing. production, managers, users)

Thinking of software development, even designers in the same project group may have to be regarded as 'non-specialists' in the sense that they have had no training in understanding programming possibilities and restrictions.

There is certainly not a single model type that will yield to all these specifications. Rather, we are looking for a family of models, of which we have already seen some elements in use in industry. However, a brief discussion of the model characteristics of code and medium may indicate possibilities.

Code: Mechanical, electrical and software engineers do not have a common language of symbols (except mathematical symbols) to describe function. Thus, it may be necessary to develop new symbols for the 'colleague-type' models. Models for non-specialists need an easy understandable code, and this may not be possible at all with symbols.

Medium: To fulfil the purpose of generating design concept alternatives, the models must be flexible (easy to modify) or fast to make. In this respect, graphical models (sketches, computer graphics) seem to have advantages. Models for non-specialists need to be concrete, and should make good use of graphics and three-dimensional hardware.

A key point when discussing mechatronic design concept models is the understanding of 'software'. The term is commonly used for computer programs and their documentation but, in order to derive combined mechatronic models of function, we need to reconsider the logical contents in machines, the 'mechanical software'. When replacing mechanical principles of a machine with microprocessors and programs, the designer is forced to formulate logical relations and algorithms that were formerly built into the mechanical system. These logical relations are seldom expressed explicitly in machine design, but they must be included when evaluating alternative mechatronic design concepts.

CONCLUSIONS

Used with care, design modelling is a powerful tool for buying information and thus diminishing risk in mechatronic product development. It is possible to describe precisely the design modelling activity and model type by the introduced model morphology.

The design concept of a mechatronic product is a complex blend of mechanical, electronic and software technologies, which is established in the early stages of a design project. The well-known function model is not sufficient to describe and discuss the design concept, and we see the need for a new family of early design concept models in mechatronics design.

These must be flexible models, to be used for generating design concept alternatives and for roughly evaluating the consequences for development, manufacturing and marketing in the early stages of a project. Further, there is a need for a model language or a model type that could improve communication both between mechanical, electrical and software engineers, and between the project team and, for example, managers and users.

REFERENCES

1 **Hubka, V** *Terminology of the science of design engineering in 6 languages* Schriftenreihe WDK 3, Heurista (1980)

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

3 **Tjalve, E et al.** *Engineering graphic modelling* Newnes-Butterworths, London (1979)

4 **Andreasen, M M** *Syntesemetoder paa systemgrundlag* (in Danish) Diss. Lunds Tekniska Högskola, Sweden (1980)

5 **Tjalve, E** *A short course in industrial design* Newnes–Butterworths, London (1979)

6 **McKim, R H** *Thinking visually: a strategy manual for problem solving* Wadsworth (1980)

7 **Tovey, M** 'Thinking styles and modelling systems' *Design Studies* Vol 7 No 1 (January 1986)

8 **Roth, K** 'Modellbildung für das methodische Konstruieren ohne und mit Rechnerunterstützung' *VDI Z.* Bd 128 No 1/2 (January 1986)

9 **Andreasen, M M and Hein, L** *Integrated product development* IFS Publications Ltd/Springer Verlag, Bedford (1987)

10 **Boehm, B W** 'Software engineering economics' *IEEE Trans. Software Eng.* Vol SE–10 No 1 (January 1984)

11 **Finkelstein, L and Finkelstein A C W** 'Review of design methodology' *IEE Proceedings* Vol 130 Pt A No 4 (June 1983)

12 **Heinzl, J** 'Entwicklungsmethodik für Geräete mit Steuerung durch Mikroelektronik' *VDI Berichte* No 515 (1984)