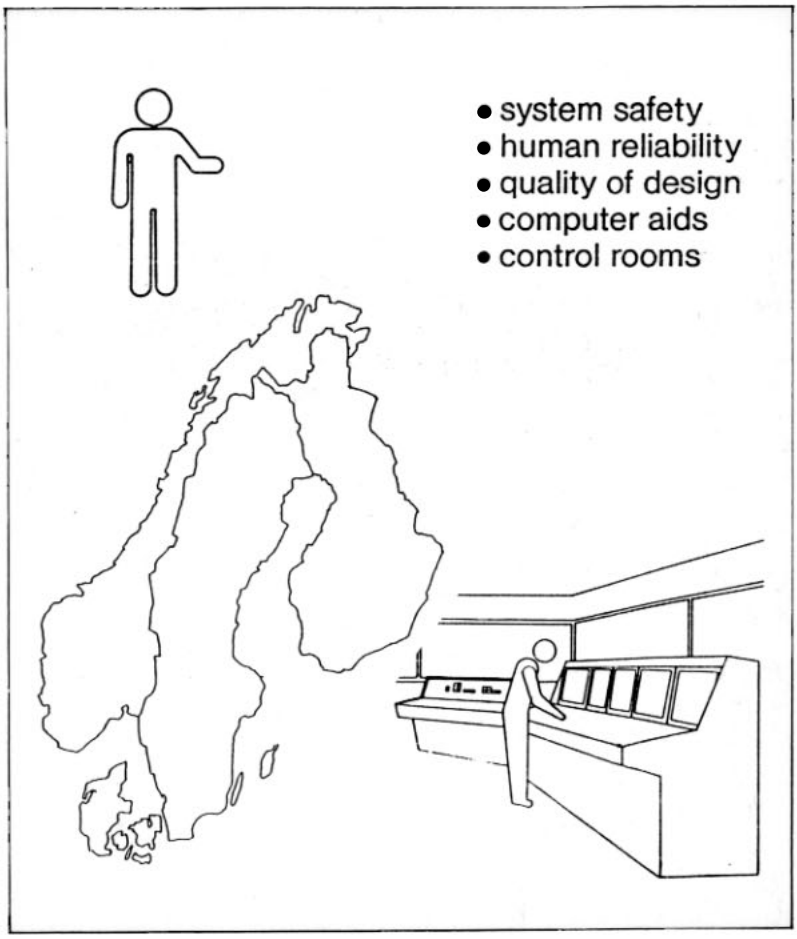
NORD

September 1985

The Design Process and the Use of Computerized Tools in Control Room Design



nka

Nordic
liaison committee for
atomic energy

nka

Nordisk kontaktorgan for atomenergispørgsmål | Nordiska kontaktorganet för atomenergifrågor | Pohjoismainen atomienergia-yhdyselin | Nordic liaison committee for atomic energy

NKA/LIT(85)4

The Design Process and the Use of Computerized Tools in Control Room Design

Final Report from the Nordic LIT-3.1 Project

B. Wahlström, R. Heinonen, J. Ranta, J. Haarla

Technical Research Centre of Finland

Electrical Engineering Laboratory

SF-02150 Espoo, Finland

September 1985

ABSTRACT

Control room design has proven an important component when the safety and availability of a complex industrial process plant are considered. Many control room deficiencies can be traced back to oversights and other errors during the design process. The introduction of powerful computers and software for computer-aided design (CAD) offers one possibility when tools for improving the quality of design are being selected.

The report gives a broad assessment of problems of design and the benefits of using computer-aided design. One proposal for a structure of a computer-aided design system is considered in more detail. In this system special emphasis has been laid on dealing with requirements during the design process. A demonstration system has been built and sample system user dialogues are described. The report is the final report of the LIT3.1 project of the Nordic cooperation on human reliability in the energy production field.

Key words: Computer-aided design, control room design, specification languages, human reliability, control and instrumentation, nuclear power, energy production, man-machine interface, Nordic co-operation, Denmark, Finland, Norway, Sweden.

This report is a part of the safety programme sponsored by NKA, the Nordic Liaison Committee for Atomic Energy, 1981-1985. The project work has partly been financed by the Nordic Council of Ministers.

LIST OF CONTENTS

SUMMARY

The control room design and the automation concept used in a complex industrial process could be decisive in preventing a minor event from developing into a major disaster in safety or in economic terms. Minor deficiencies in the design could increase the likelihood of human errors in operation and maintenance and every effort should therefore be made to achieve the best possible man-machine interface. The designer of the control room and the automation concept to be used should therefore be given appropriate design tools and be supported by an efficient organization.

The development of computer-aided design (CAD) systems in engineering and technical design has been proceeding at a rapid pace. The system can not only increase the productivity of the designer, but more imprortantly, improve the quality of the design. The computerization of the design data base also provides new possibilities for the integration of computerized design tools and for the presentation of design information. The technical development with commercial availability of new equipment and software, makes it necessary to re-evaluate how the design of control rooms and the automation concept used should be carried out in the future.

The design procedure may be seen as a series of consecutive decisions to match technical solutions to specified requirements. The growing complexity of modern industrial plants with an increasing number of interactions between the parts of the plant puts heavier demands on the designer of the control and instrumentation system. The management of the design data base, especially in the case of design changes, has also been becoming increasingly difficult due to the interactions and the large amount of information to be handled. The

introduction of computerized design tools is the obvious solution to many of the problems in design.

The specifications of computerized design tools should always be tailored to the design procedure defined in terms of information flows and information users. This means that an idealized design procedure should be broken down into a sufficient degree of detail to facilitate the selection of suitable information representations and data structures. One such model of a control and instrumentation design project has been developed using the structured analysis and design technique method.

In addition to the design tools, attention given to the design organization is another means of reducing the likelihood of hidden errors in the design. It is important to set up an appropriate procedure for a continuous review of the design quality to make it possible to spot and correct errors as early as possible. In the design organization it is also important to collect experience gained in earlier design projects in order to be able to make recommendations on organizational practices to be used. The report gives a general review of typical problems of design that have been observed in a number of projects in the field of computerized control and instrumentation.

Human errors in design will lead to hidden deficiencies in the control and instrumentation. These deficiencies may suddenly be observed when the plant is brought to a new operational regime. The design tools developed should be based on a consideration of the seriousness and the frequency of errors a designer makes in his work. This means human error data has to be collected before the design tools can be optimized. A classification scheme for design errors has been developed, which could be used to collect data on the importance of different types of errors.

The computerization of the control and instrumentation systems of complex industrial plants places new demands on design practices. The total design period has been shortened and there has been a shift from hardware design to software design. Computers are also being increasingly used in different phases of the tendering, the specification and the detailed design of the control and instrumentation systems. This means that the design data bases of future plants to a large extent will be computerized. This will also provide new opportunities for the presentation of design information during the design and operation. The concept of a designer's work has been developed in the project and proposed solutions seem feasible in a projection of the expected technical development.

Present design practices rely to a large extent on natural language descriptions. These are, however, not suitable for computerization. This means that a formal design language has be constructed to serve as a core of a CAD system. The formal language should include all the concepts manipulated during the design procedure but should also be easy to extend later. The most obvious solution is to define a "metalanguage", which is used to specify the design language to be used. This concept of specifying tools for the construction of the design tools to be used in a specific design project has been elaborated in the report and the present technical development seems to proceed in such a direction.

The task of the designer is to construct a model of his vision of the plant. This may be used as a blue print for the construction of the control and instrumentation system. This means that the design effort corresponds to the modelling of this vision in terms of the formal design language. The design language could thus be seen as a system modelling language (SML), enabling the designer to specify his design on a high level of abstraction. The

verification of the design and the production of low level descriptions may then be done automatically using special software packages operating on the high-level descriptions. The SML concept has been elaborated in the report and it seems to be a viable approach for the construction of computerized design tools.

The prototype of the SML concept has been built as a demonstration system to assess the difficulties in the formalization of the design data base and in constructing the computerized design tools. The demonstration is based on an assessment of typical errors made during the design procedure. One of the key features of the demonstration is the establishment of an accurate and traceable connection between the requirements and the technical solutions of the design. Another feature is the possibility to build a rich and well-specified set of associations between different pieces of information in the design data base.

The report concludes that there are both ample reasons for improving the quality of the design as well as possibilities of doing so. There is also a clear consensus that CAD methods should be used. The main question is what functions it is feasible to include in the computerized system. The development of a specific CAD system for some particular type of design within the control and instrumentation field is a large task which has to be carried out separately. The different concepts and approaches developed in the report could serve as the basis when future tools for computer aided design (CAD) are developed.

SAMMANFATTNING (Summary in Swedish)

Vid en komplex industrianläggning kan kontrollrummets utformning eller graden av automatisering vara det som avgör om en relativt obetydlig händelse utvecklas till en betydande olycka med säkerhetsrelaterade eller ekonomiska konsekvenser. Små förbiseenden under planeringsprocessen kan medföra brister i kontrollrum och instrumentering som ökar sannolikheten för tekniska fel och som bidrar till risken för mänskligt felhandlande. Det är därför uppenbart att de personer som svarar för planering och konstruktion bör ha tillgång till bästa möjliga verktyg och att de bör ha stöd av en effektiv organisation.

Planerings- och konstruktionsprocessen kan uppfattas som en serie av på varandra beroende beslut där tekniska lösningar skall anpassas till uppställda systemkrav. Moderna processanläggningar har blivit allt mera komplicerade, framför allt beroende på ett allt större antal kopplingar införs mellan olika system och komponenter. Likaså uppstår, t.ex. när ändringar i konstruktionen måste göras, problem med stora datamängder och komplicerade beroendeförhållanden mellan olika delar av processen och instrumenteringen. Komplexiteten ställer ökade krav på de personer som planerar kontrollrum och instrumentering. Den uppenbara lösningen av många av dessa problem är att i större utsträckning använda datoriserade system som verktyg vid planering och konstruktion.

Utvecklingen av sådana datoriserade planeringssystem, d.v.s. CAD-system (Computer aided design) har skett i rask takt. CAD-systemen ökar effektiviteten av det arbete som görs under planering och konstruktion men viktigare är att systemen också höjer kvalitén på planeringsarbetet. Att planeringsprocessen är datoriserad innebär att också slutprodukten av arbetet design databasen, d.v.s. system- och konstruktionsbeskrivningar, ritningar, etc. kommer att vara datoriserad. Detta betyder att nya möjligheter öppnar sig för att söka informationen databasen och att den kan användas för automatiserade dokumenterings- och konstruktions-verktyg. Den tekniska utvecklingen och kommersiellt tillgängliga system kommer också att göra det möjligt att i framtiden omvärdera hur planeringen av kontrollrum och automation skall genomföras.

Datoriserade system bör alltid anpassas till behoven hos sina användare. Detta betyder för planerings- och konstruktionsprocessen att olika användare och informationsflöden mellan dessa måste identifieras. Man måste sålunda beskriva en idealiserad planerings- och konstruktionsprocess för att göra det möjligt att specificera olika datarepresentationer och lämpliga sätt att struktura den information som behandlas. En sådana modell av ett planerings och konstruktionsprojekt av kontrollrum och instrumentering för en processanläggning har utarbetats i LIT-3.1 projektet och beskrivs i rapporten.

Det sätt på vilket planerings- och konstruktionsprocessen genomförs bidrar på ett avgörande sätt till kvaliteten på konstruktionen. Det är viktigt att använda sig av väl definierade procedurer för att övervaka kvaliteten av arbetet. På detta sätt kan man hitta och korrigera planeringsfel så tidigt som möjligt. Det är också viktigt att utnyttja erfarenheter samlade från tidigare planerings- och konstruktionsprojekt. Rapporten ger en översikt över problem som har observerats i ett antal projekt med anknytning till planering och konstruktion av datoriserade system för kontrollrum och instrumentering.

Olika fel som görs i planerings- och konstruktionsskedet av en industrianläggning ger upphov till dolda fel som plötsligt observeras exempelvis när processen kommer in i ett nytt driftläge. De planerings- och konstruktionsverktyg som används skall vara baserade på en uppskattning av vilka typer av fel som vanligtvis görs och hur ofta dessa förekommer. För att kunna optimera designverktygen bör således data om mänskligt felhandlande under planering och konstruktion samlas in. Rapporten föreslår ett schema, som kan användas för att samla data om olika typer av designfel.

Datorisering i kontrollrum och instrumentering i komplexa industrianläggningar kommer att föra med sig nya behov för system och rutiner av planerings- och konstruktionsarbetet. Den totala tiden för detta arbete har blivit kortare och man kan se att proportionen av program varo planering vuxit kraftigt. Datorer används i ökande grad i olika skeden av offertgivning, specifikation, planering och konstruktion av de system som

installeras i kontrollrummen och i instrumenteringen. Detta innebär att konstruktions databasen för framtida anläggningar till stor del kommer att vara datoriserad. Dessa system kan även användas under driften för att presentera information för operatörer eller underhållspersonal. De koncept för en konstruktörs arbetsbänk som har utarbetats i projektet och presenteras i rapporten är också sannolika beaktande den tekniska utvecklingen.

Det är idag vanligt att planerings- och konstruktionsarbetet stöder sig på beskrivningar avfattade på vanligt språk. Vanlig text är dock svår och ibland omöjlig att behandla med datorer eftersom en text ofta måste förstås för att kunna behandlas rätt. Det är därför nödvändigt att konstruera ett formaliserat språk, som täcker in de begrepp som kommer att användas under planerings- och konstruktionsarbetet. För att erhålla tillräcklig flexibilitet, bl.a. för att senare kunna bygga på språket, är den naturligaste lösningen att definiera ett metaspråk. d.v.s. ett språk som används för att bygga upp det formaliserade språket. Denna metod, att snarare specificera ett verktyg med vilket planeringshjälpmedel konstrueras än att specificera planeringshjälpmedlen, har använts inom området för programvara och liknande lösningar före slås för planerings- och konstruktionsprocessen.

Designerns uppgift är att bygga en modell av sin vision av konstruktionen och att göra det på ett sådant sätt att denna modell kan användas som en ritning för alla faser i konstruktionsprocessen. I planeringsarbetet beskrivs således konstruktionen med de termer som ingår i det formaliserade designspråket som används. Designspråket kan därför ses som ett systemmodelleringsspråk (SML) som gör det möjligt för konstruktören att beskriva konstruktionen i abstrakta termer. Denna beskrivning kan användas för att verifiera konstruktionen och generera andra beskrivningar som kan användas för bl.a. dokumentering. SML konceptet har utvecklats i projektet. Bedömningen är att konceptet är en användbar bas för datoriserade planerings- och konstruktionshjälpmedel. En prototyp av SML konceptet har byggts för att demonstrera användbarheten och uppskatta det arbete som behövs för att formalisera en konstruktions

databas. Demonstrationen bygger på en uppskattning av vilka hjälpmedel som är de viktigaste i planerigs- och konstruktionsarbetet. Den nuvarande versionen av SML gör det möjligt att upprätta en noggrann och dokumenterad koppling mellan kravspecifikationer och tekniska lösningar. Den ger också möjlighet att bygga upp ett rikt nät av kopplingar mellan olika objekt i design databasen.

Slutsatsen från rapporten är att det både finns anledning till och är möjligt att höja kvaliteten på det arbete som görs under planerings- och konstruktionsarbetet för kontroll rum och instrumentering. Det är också en självklarhet att man bör använda sig av datorer i detta arbete. Frågan är då vilka funktioner som skall ingå i de datoriserade planerings- och konstruktionshjälpmedlen. Den lösning som föreslagits i rapporten separerar mellan utvecklingen av datoriserade hjälpmedel och utvecklingen av verktyg som kan användas för att utveckla hjälpmedlen. Det visar sig att de koncept och lösningar som har tagits fram är lämpliga för att utveckla de praktiska system som sedan kan användas i planerings- och konstruktionsprocessen.

PREFACE

The safety of nuclear power stations and other complicated industrial processes depends on an accurate and well-timed execution of tasks during the operation. There is, however, always the possibility that human errors either directly or indirectly can initiate an unwanted course of events. The general problem is then to decrease the probability of human errors and to increase the probability of their detection. This is in principle made possible by careful task design and by giving the human operator appropriate training. This means in practice that one should consider the tools of the operator, the organization he is working in, and the training he is given. All these aspects have been considered in the Nordic LIT-research programme, which was started in 1981 and has been running to 1985.

In particular, the Nordic LIT-research programme has addressed the following questions:

- human errors in test and maintenance (LIT-1)
- safety oriented organizations and human reliability (LIT-2)
- computer-aided design of control rooms and plant automation (LIT-3.1)
- computer-aided operation and experimental validation (LIT-3.2 and LIT-3.3)
- planning and evaluation of operator training (LIT-4)

The selection of these fields of research was based on experience from an earlier phase of the Nordic cooperation (cf. the reference Wahlström, Rasmussen, 1983).

The Nordic LIT-research programme involved a total effort of about 40 man-years of qualified researchers in Denmark, Finland, Norway and Sweden. The research programme has been financed partly by project funds from the Nordic Council of Ministers and partly by funds from the different participating organizations. The LIT-research programme was initiated by the Nordic Liaison Committee for Atomic Energy (NKA) as a part of the Nordic cooperation in the field of safety in the energy production field. The following organizations have been financing and have also been directly involved in the LIT-research programme:

Risø National Laboratory, Roskilde, Denmark
Technical Research Centre of Finland (VTT), Espoo, Finland
Institute for Energy Technology (IFE), Halden, Norway
Swedish Nuclear Power Inspectorate (SKI), Stockholm, Sweden
Swedish State Power Board, Vällingby, Sweden

The LIT-programme is reported in the following final reports:

- The human component in the safety of complex systems; LIT programme summary report NKA/LIT(85)1

- Human errors in test and maintenance of nuclear plants; LIT-1 final report NKA/LIT(85)2

- Organizations for safety; LIT-2 final report, NKA/LIT(85)3

- The design process and the use of computerized tools in control room design; LIT 3.1 final report, NKA/LIT(85)4

- Computer aided operation of complex systems LIT-3.2 & 3.3 final report, NKA/LIT(85)5

- Training diagnostic skills for nuclear power plants; LIT-4 final report, NKA/LIT(85)6

1. INTRODUCTION

The design activity may be seen as the process where the designer brings his vision of the plant to a point from where it can be built. The operator of the plant will in a way reconstruct this vision into an internal model which, however, will be based both on formal training and experience gained. The actual design will naturally bear a resemblance to the internal models of both the designers and the operators, but there may also be important differences (cf. Figure 1).

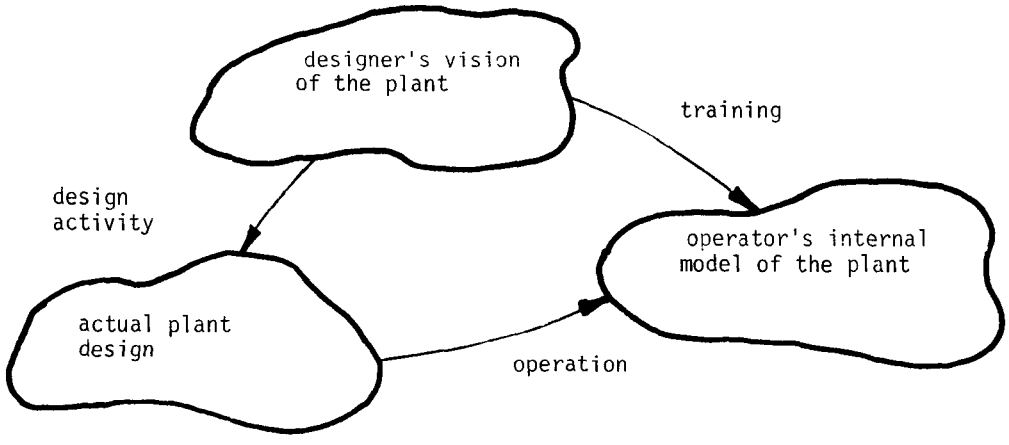


Figure 1. The internal models of the designer and the operator and their relations to actual plant design

The design of the control and instrumentation is an important part of the design and construction of industrial production systems (cf. the reference Hanes, O'Brien, Disalvo, 1982). The control room and the automation concept are developed as a part of the design of the control and instrumentation. The control room and the automation concept have great bearing on the operability of the plant and can thus be decisive in plant economics. The design of control rooms and the automation concept for different industrial processes has a long tradition, where the development has evolved in small steps, and it has been possible to collect experience prior to full-scale implementations. The advent of cost-effective computers that can easily be reprogrammed has initiated a rapid development with a corresponding need for a more general evaluation of the whole design process.

The quality of design also has an influence on the safety of potentially dangerous processes, such as nuclear power and various chemical processes. A small design deficiency could be the trigger which through a series of events might cause a major accident (cf. the reference Rubinstein, Mason, 1979). When we consider the safety of such processes, it is clear that all possible means should be employed to ensure high design quality. One of the problems then is to define the design quality, to select the best approach for the design and to measure the resulting quality.

It is evident that the design quality should rely on some measure of operability, where the main objectives of the plant are taken into account. The definition of design quality is then related to the requirements placed on the process. However, some man-oriented measures also need to be included in the design quality, because the control room and the automation should give the operator a simple and easy process.

Even in the case where some accepted measure for the quality of design has been defined, the main difficulty during the design is to relate any design decision to such a quality measure. Another difficulty of the design process involves the long chains of interrelated design decisions, where the final outcome is difficult to judge before the design has advanced to some level of concretization.

With the introduction of computers in other fields of production, systems are now available to increase the productivity of designers. One of the arguments for the introduction of computer-aided design (CAD) systems has been that the dull and monotonous tasks in the design could be automated, leaving for the designer the tasks where human creativity is needed. This is certainly true provided that a proper division of tasks between the designer and the computer is found. This proper division is, however, problematic and fears have been raised that the CAD systems will tend to proletarize the jobs of the designer. Consequently, the introduction of new technology makes it necessary to consider the problems of design on a broader basis.

Software has been playing an increasingly important role in the design of control and instrumentation. The design of control rooms and automation concepts are also to a large extent realized by software using intelligent display terminals and computers. It is therefore possible to use in the control room design similar kinds of tools that have been developed in the field of software engineering.

The LIT-3.1 project on "Computer-aided design of control rooms and plant automation" has been a part of the Nordic cooperation in the safety field. The main goals of the LIT-3.1 project have been to investigate the applicability of computer aided design (CAD) methods to the design of

control rooms and instrumentation for industrial installations, such as nuclear power plants, chemical processes and off-shore facilities. The angle of approach included design quality and possibilities of human errors in the design. The project was divided into the following main parts:

- analysis of present design methods
- construction of an idealized design model
- analysis of information needs and data flow during the design process
- specification and construction of a demonstration system
- experiments with and assessment of the demonstration system.

The LIT-3.1 project is related to the other LIT projects with respect to the following subjects:

- the design of the control rooms and the plant automation has to also cater for the interface with personnel outside the control room (LIT-1)
- the design organization is an important determining factor when the quality of design is considered (LIT-2)
- new approaches to information display have to be considered when future design systems are to be built. The use of new methods (e.g. multilevel flow models) could have an important impact on the design process when the designer has to transfer an understanding to the operators of the plant (LIT-3.2 and 3.3).
- an interface with the design data base could serve as a training device for the training of operators (LIT-4).

2 THE DESIGN PROCESS

Each new plant or system has to be designed before it can be built. The design process is in this context understood to be separated from the construction process although this seldom is the case in reality. It is also clear that the design process will vary considerably depending on the target of the design, national practices and the organization responsible for the design. The consideration of the design process below is based on an idealized model (cf. Ranta, 1985), which has borrowed characteristics from many different fields and countries. The section contains the following parts:

* On the design of complex systems
* Problems of design
* The design organization
* Human errors in design
* An idealized model of the design process
* Actual design practices

2.1 On the design of complex systems

Design in general may be seen as the process where a vision of a designer is brought to a level of concreteness from where a construction process can be started. In the design process the designer is considering alternative designs, evaluating them, settling for one and

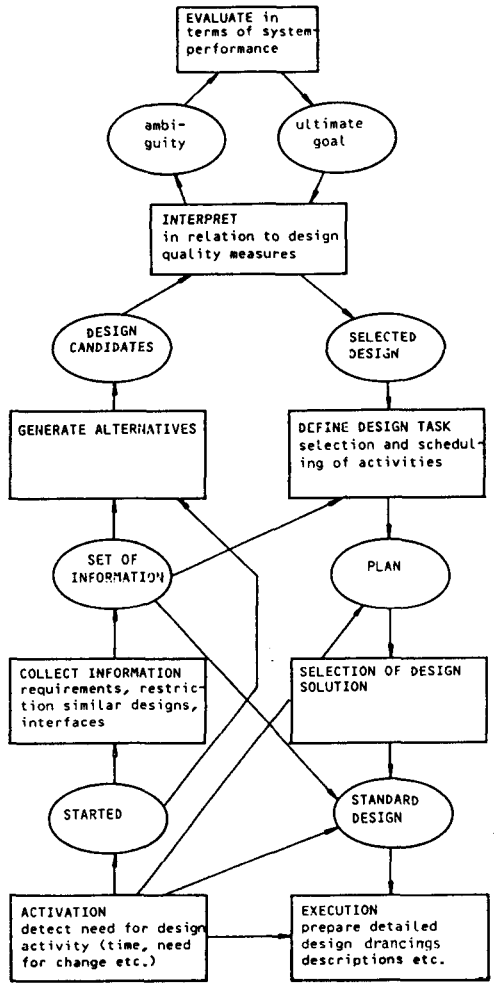


Figure 2.1 A description of the decision making tasks of the designer. (Adopted from Rasmussen, 1976).

documenting it as a part of the design. The design effort may be seen as an information processing task (cf. figure 2.1) where the designer's different information processing activities are separated by states of knowledge. Two different modes of mental processing may be identified; one knowledge-based, where the long route is followed, and the other rule-based, which is characterized by rapid associations between states of knowledge.

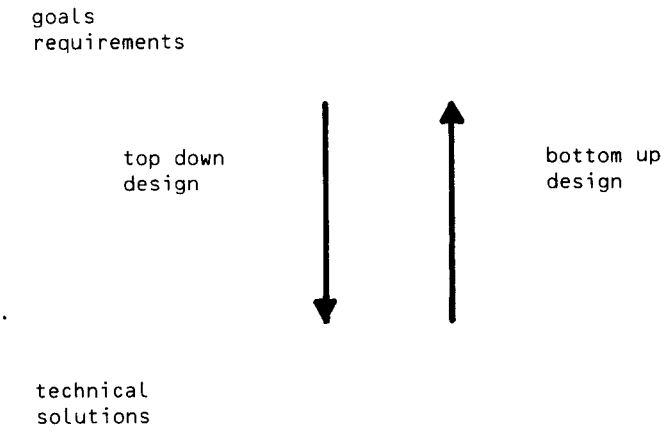


Figure 2.2 The design means that goals and requirements have to be matched with technical solutions. This can be done in two ways either using top down or bottom up design methods.

The design process in general can be seen as a search for technical solutions consistent with the requirements imposed on the design (cf. Fig. 2.2). The design selected evidently has some relation to the design requirements, i.e. the requirements give the reasons for the choice of design solutions. The search for design solutions could in principle be carried out either starting from the goals and requirements or from the technical solutions. In the

top-down design approach the goals and requirements are broken down in several steps to specifications for which technical solutions can be selected. In the bottom-up approach the design is started from technical solutions which are revised in order to meet the design requirements. Actual design projects are usually carried out as a continuous alternation between both approaches.

The design of complex systems is a problem solving task where a tree of design decisions is built (cf. figure 2.3.), which gradually bridges the gap between the requirements and the technical solutions. Actual design is often carried out by applying general schemata, which are expanded locally to become building blocks in the design process. As in many problem solving tasks, it is difficult to assess the quality of design before it has been advanced to some level of concreteness. This means that it might be necessary to re-do large parts of the design when incompatibilities between requirements and technical solutions are observed.

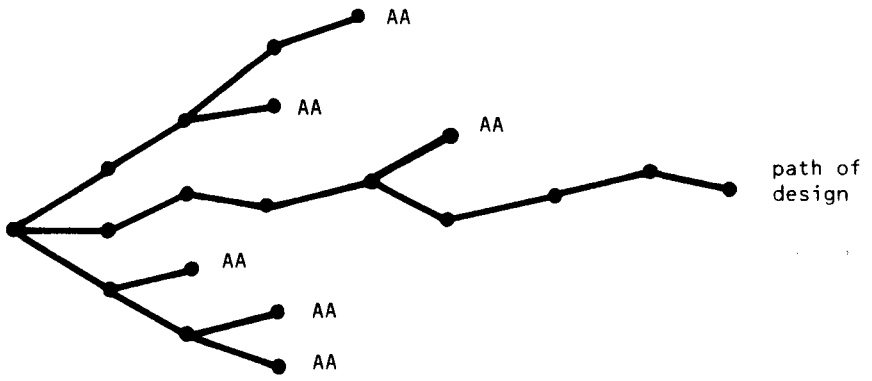


Figure 2.3 The design process can be viewed as a tree of consecutive design decisions. (AA alternative abandoned).

In the design of complex systems one has to find ways to cope with the complexity (cf. Rasmussen, Lind, 1982). A division of the system into parts makes the parts easier to manage but makes it necessary to consider also the interactions between the parts. It is therefore important to select a proper division of the system, so that the interactions between the parts are minimized.

In the design process, early decisions will influence the later design stages not only by giving more guidance on acceptable alternatives, but also by imposing additional restrictions on the design choices. This means that the freedom of the designer to use different solutions will decrease with time as the design proceeds. If some large design modifications have to be introduced, they will have more influence on the later stages of the design than in the earlier stages. This means that more effort is needed in carrying out a design modification in the later stages than in the earlier stages. This fact can be illustrated with the conceptual graph in Figure 2.4.

In the design and construction of large and complex industrial plants there is a definite need for standardization. Standardization of components and design solutions makes it easier to plan the maintenance of the plant and produces higher efficiency. Standardization also tends to rationalize the design process by providing guidance to the individual designers. Standardization makes it easier to understand the plant and to document the construction. Standardization, however, also introduces additional restrictions by limiting the design freedom and may thus involve higher costs. On the other hand one must consider the trade-off between the costs of the initial construction and plant lifetime costs.

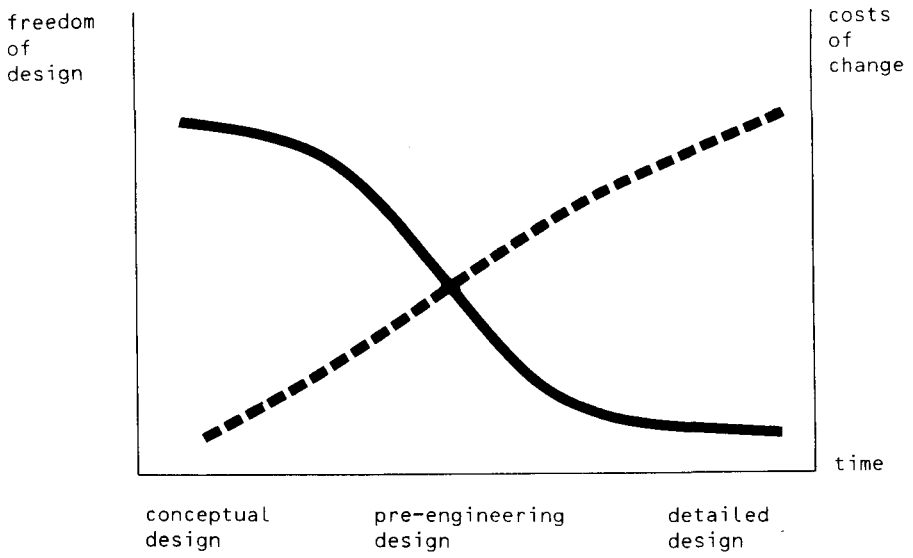


Figure 2.4 The freedom in the design decisions will decrease gradually with time and the costs of changes introduced will increase correspondingly (cf. EPRI, 1984, p. 5).

The use of standards may be introduced from different levels, such as:

- international standards
- national standards
- recommended practices
- company practices
- project practices

International standards, national standards and recommended practices have usually such a status that they are enforced by agreements for the design projects. The safety rules which are applied e.g. in the field of nuclear power, are on the other hand mandatory and hold for different licencing procedures. In forming company and project practices there is more freedom, but it is advantageous to arrive at such practices as the result of a deliberate design effort.

2.2 Problems of design

The increasing complexity of industrial plants means that more effort is needed simply to maintain the design data base. The increasing complexity also means that there is a larger number of relationships between the different objects manipulated during the design. The relationships between the objects are actually many to many, where one object serves many purposes and one purpose is satisfied by many objects. The many to many mappings between goals, functions and equipment has been considered by Lind (1983). It is desirable for the designer to have easy access to all design data related to the objects he is working with. The complexity of the design data base also makes it more difficult to carry out design changes because it is easy to forget to carry out the change in some part of the data base.

Design is often iterative, which means that the decisions needed to complete a part of the design process are mutually interdependent. In such cases it may be necessary to carry out the design process in several rounds of iterations before an acceptable solution is reached.

It is always difficult to assess the quality of design before it has been completed. By this time, however, it may be difficult to change the design and far more difficult to find an acceptable solution for re-design. It is also very difficult to ensure that all important requirements are considered because the operation of any new system will always bring in new operating experience, which in turn may introduce new requirements.

The effort to design a large industrial plant has to be divided between a large number of designers or groups of designers, all having their own areas of responsibility. This means that there will be much interaction between the

different designers or groups of designers when they are working towards common design goals. This division of effort between many persons is a potential source of problems because different persons may interpret the requirements in different ways with the result that there are incompabilities in the design. Areas of design on the borderline of the responsibility for two different designers or groups may also be forgotten with a resulting lack of completeness in the design.

The largest problem of the design process is the need to foresee the implications of different decisions. The problem is illustrated by the possibility that a chain of reasonable design decisions could lead to a completely unacceptable design solution. Each design project needs some accepted practices for ensuring the quality of the design. In the simplest form it could be regular design reviews where one would especially address the interfaces between different areas of the design. The governing rule for each design project is that design errors should be spotted as early as possible. This means actually that there will not be a single body responsible for the quality of design, but the quality should be considered at all levels.

The documentation of the design as it proceeds is one of the most important tasks because during design continuous reference is made to earlier design decisions. The documentation is also the main path for transferring knowledge and intentions from the design process to the operational phase of the plant.

Different design alternatives evaluated before a design decision is made are usually not documented. The dependence of a design solution on the design requirements is rarely documented, this meaning that it could be quite difficult for an outsider to understand why the system is

designed as it is. This, combined with the fact that the original designer is seldon available to explain the details when the design has been completed, makes it increasingly difficult to carry out changes in the design after the plant has been in operation for some period.

As mentioned above, the main problem in the design process is related to the unavoidable need for making changes. It is clear that a common aim is to minimize the need for late changes because they are bound to introduce cost and timetable overruns for the project. It is, however, also clear that it is impossible to consider all the details beforehand because each project is in a way unique. If a design change is carried through, there should be means to ensure that all the objects influenced by the change are considered for a possible re-design. It is also important to ensure that there is no possibility of the designers working with designs that are outdated as a result of some previous change.

2.3 The design organization

The organization is regarded as a set of formal and informal rules aiding individuals and groups of individuals in carrying out their tasks (cf. Lindqvist, Rydnert, Stene, 1984). The organization thus provides certain resources for maintaining the functions necessary for fulfilling the goals of the organization.

The design of a new plant or new system is always in some way unique and is therefore usually carried out as a project restricted with respect to both time and money. The organization responsible for the design is therefore established for this sole purpose, and the project organization is broken up when the design has been finished. The limited lifetime of the design organization is an important characteristic that makes it very different from e.g. the operational organization of a

plant. Different arrangements in the case of vendor organizations and turn-key contracts may change the situation, though in this case the transfer of technology to the operational organization from the vendor organization may pose a problem.

The design organization is established by a mother organization, which will either lend expertise to the design organization or hire it from outside. Depending on the type of design project under consideration, the mother organization could be the final user of the system to be designed, the vendor of the system, or a consultant. In practice this means that the design is usually carried out as some sort of cooperation between the vendor and the user of the system. This arrangement makes it possible to carry out iterations between user requirements and vendor solutions in an efficient manner, provided that the vendor and the user are able to express themselves in a common language.

The limited lifetime of project organizations means that there is an inherent difficulty in accumulating knowledge on how projects should be carried out, because the experience collected once the project is terminated will be scattered around to different places. The limited lifetime of a design project also has the consequence of placing a very strong emphasis on the documentation of the design. Generally all information on the why's, what's, and how's, i.e. the reasons for and the selected solutions to the design have to be transformed into a written form. The test operation of a new design naturally leaves some overlap when the design organization and the operations are working together, but this does not change the fact that it is difficult to get some of the designers to explain the details of their design after the plant has been in operation for some years. The documentation also has to be adapted for different classes of personnel such as operations, maintenance, etc.

The problems of the design organizations have naturally been implicitly known and they have been at least partially solved by different practices used in the design projects. One of the solutions is to establish organizations specializing in design. This has been practised for many of the vendor organizations e.g. in the field of nuclear power. Another such practice is to start the design as a pre-project, where the complete design project in a way is simulated. The pre-project makes it possible to survey the possible difficulties of the design project in advance and thus to allocate enough resources for the solution of the problems. Many of the crucial design decisions will actually be made as a part of the pre-project, which thus guides the design project. The pre-project also makes it possible for the design organization to build up its informal parts to some extent before the main project is launched. The breaking up of the design into smaller design tasks, and the preparation of a schedule for the whole design are also important parts of the pre-project. The budget for the design project is usually settled during the pre-projectstage.

When we consider the motivational aspects of a designer, one often-cited advantage in working with design is the possibility to create. This means that design tasks often appeal to creative people, and this is a fact which has to be considered when systems with strick safety requirements are constructed. The safety requirement makes it necessary to standardize the design to a large extent, i.e. to make the design more a routine task. There is also the requirement of precion right down to detail, which makes the design more a tedious time-consuming clerical task than a task for a creator working on a large scale. It may even be so that the two aspects require different personal design styles difficult to combine in one person. The two aspects follow from the design phases to some extent, where conceptual design requires more creativity than the

detailed engineering. In selecting people for design projects, one should be aware of the contradiction between the two aspects in order to avoid differences in expectations.

The design organization should naturally give the designers as much support as possible in their work in a well-timed and efficient manner. This means that enough resources should be allocated to building the design tools and setting up the information system of the design effort. The aim of the information system is to give the designers all the information they need without overloading them with unnecessary information.

In any design project there are always things that have not been foreseen and which require a solution on the spot. The human mind is, however, very error-prone when a decision has to be taken in a new situation and under stress. Different fixes for problems encountered during design therefore deserve special consideration. When need for design change is detected in the middle of a design project, it has to be handled very carefully. If a change is suggested, it has to be evaluated with respect to all the original design requirements i.e. the design has to be tracked back to find a level of invariance. After the change, it is very important that also all the completed design tasks influenced are subjected to re-design; that all the designers concerned are informed; and that all changes are introduced in the final documentation. If those steps are not carried out properly, the design will contain contradictions or be poorly documented, i.e. there will be a deficiency in the design.

Large design changes should generally be avoided during design projects, and one way to cope with the need for changes is to freeze the design at some instant. Freezing the design can in practice be done at several levels after main design activities, where design goals, functional design, and detailed design are frozen before the system

is issued for construction. Freezing the design is an administrative necessity in order to make it possible to complete the design project on time. The design change is then carried out as a separate project after the original design and construction have been completed.

2.4 Human errors in design

The definition of a human error has always to be made in relation to some accepted standard of performance (cf. Rasmussen, 1978). A human error is then an act which is outside the required performance limits. In investigating human errors it is, however, important to note that they often may be seen as consequences of the task characteristics, which means that a more constructive approach is to consider the task design as a cause of error rather than to put the blame on the person carrying out the task. It is, however, clear that a causal explanation of human errors is very difficult to construct as there are usually several causes contributing to an observed error (cf. Rasmussen, 1981). In the case of errors in the design, the error made is not the only one because it should have been observed either in the design reviews or during the test operation.

In investigating human errors and ways to decrease their probability, it is important to collect data on incidents and accidents. One proposal for such a data collection system has been given in Rasmussen et al. (1981). Such data have, however, to be categorized to enable the investigators to see the patterns. When patterns in the data have been identified, more general models of human behaviour can be suggested and tested on new data. This indicates that both aspects, the collection of data, and the construction of theoretical models, are necessary for the understanding of human errors (cf. Hollnagel, Pedersen, Rasmussen, 1981).

A broad categorization can be used for human errors:

- errors induced by the task characteristics
- errors due to human variability
- intentional errors

This division is not unambiguous; there are errors which could be attributed to two or even all three categories. With consideration to the design task and the design tools, the first category is the most important. The errors due to human variability include clumsiness, occasional lapses of attention, etc., and they should be taken care of by the quality control of the design process. The intentional errors, such as different kinds of sabotage, are not here a matter of investigation.

The design errors are hidden in the construction and are thus different from human errors made during the operation. An operational error could be the immediate initiator of some unwanted chain of events, but the design error will only introduce some hidden deficiency in the system. The design error can, however, unexpectedly cause either a technical failure in the system or a human error when the plant is brought into an operational regime where it has not been before. The existence of a design error is an indication of incomplete or erroneous testing of design. The complexity of present industrial processes makes it on the other hand completely impossible to carry out a complete testing programme. The definition of the test programme of the completed design is therefore also an important part of the design process.

Different models have been constructed to explain the causes behind human errors. The models can be used to suggest schemes for the collection of human errors, which again can be used to validate the models. In appendix 1 such a model is suggested with a corresponding scheme for the categorization of design errors. The model and the categorization scheme are based on the model considered in Rasmussen et al (1981).

The classification system should be seen as a conceptual model of human decision making during design rather than as a scheme for the collection of human error data. This conceptual model can be used to make a subjective judgement on the relative importance of the different error categories, and can thus serve as a basis for the construction of design tools. The conceptual model may also be used to develop a method for the collection of human error data during design.

2.5 An idealized model of the design process

The design process proceeds from the consideration of general requirements to the generation of technical solutions. Using the concept of a dimension of abstraction, the design proceeds from a more abstract concept to the concrete realization. Rasmussen (1979) discusses the dimension of abstraction in terms of mental models of a complex system. The design can be visualized as a two-way search, where the requirements provide the reasons in a top-down manner and the technical solutions are built on the physical basis in a bottom-up construction process (cf. Figure 2.5).

LEVELS OF ABSTRACTION

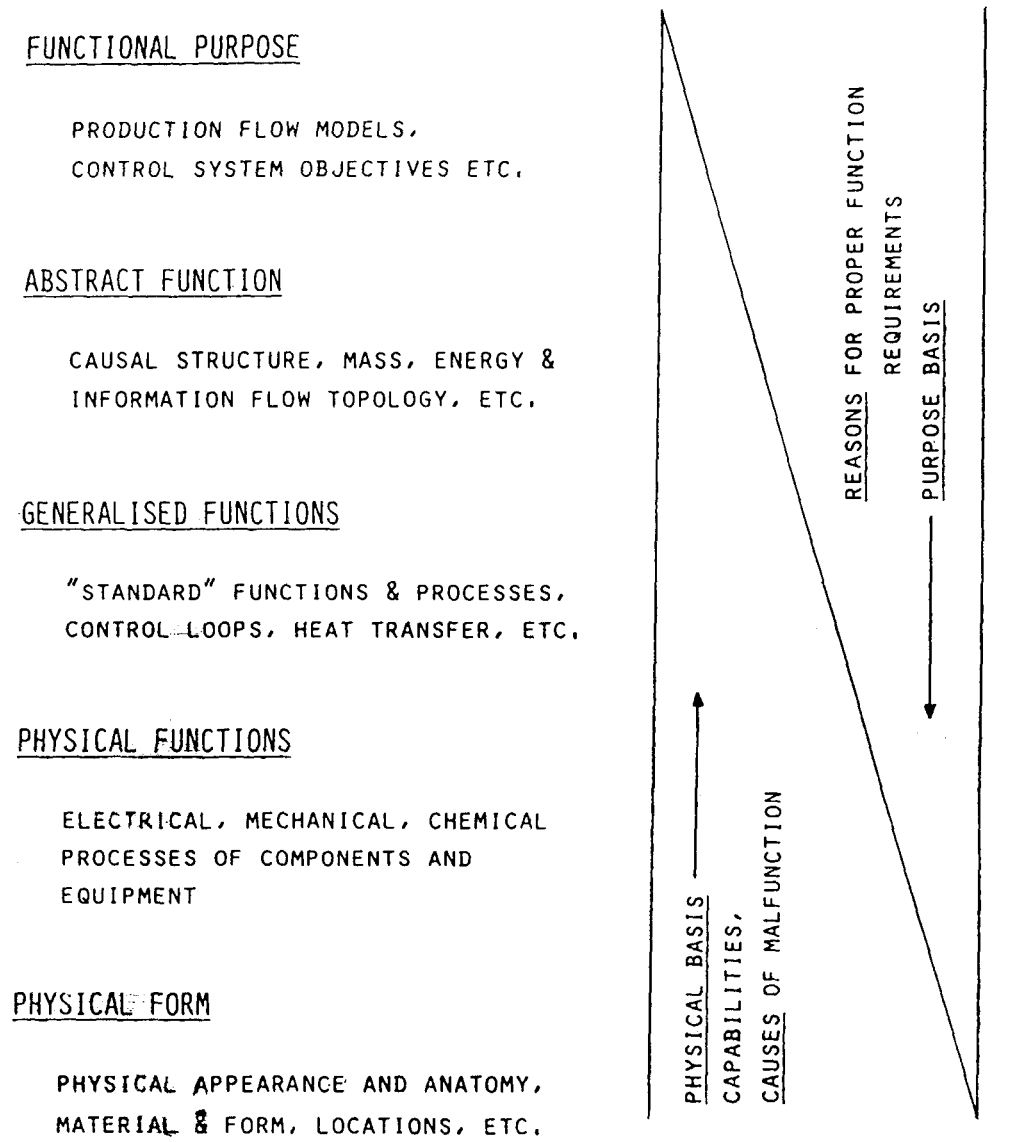


Figure 2.5 The abstraction hierarchy used for representation of functional properties of a technical system (from Rasmussen, Lind 1981).

As a simplification of this scheme, the design process can be described by a model (cf. Figure 2.6), in which the design process proceeds through three different abstraction levels. The uppermost level concerns the goals and requirements of the design process, and it states why the system is needed, which problems are solved by it, and how the system interacts with its environment. The intermediate level transfers the goals and requirements into the functions of the system and it gives the functions the requirements necessitated, why these functions are needed, which criteria a function fulfills, and how a function can be realized in practice. The lowest level deals with the technical realization of the system, and it tells how the system is functioning, and what technical solutions have been used.

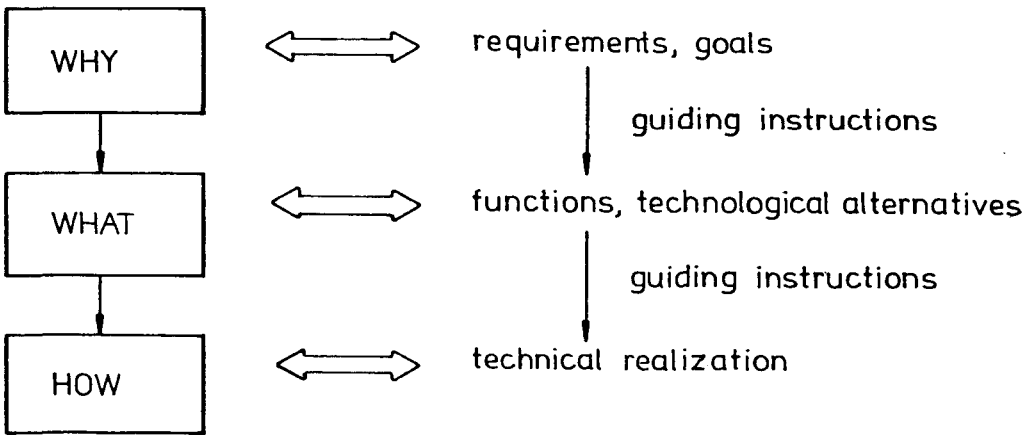


Figure 2.6 The three-level model of the design process.

At the same time as the design proceeds from abstract to concrete, the attention of the design shifts from the whole to parts. This shift corresponds to what can be termed another dimension related to the level of aggregation and decomposition of the system to be designed.

The design process and the relation of the design activities to the design dimensions are illustrated in Figure 2.7. The arrows indicate the temporal ordering of the design activities. The design thus proceeds in the dimension of abstraction from the why's through the what's and the how's. In the dimension of aggregation the what's on the system level are used to generate the why's on the subsystem level, and so on.

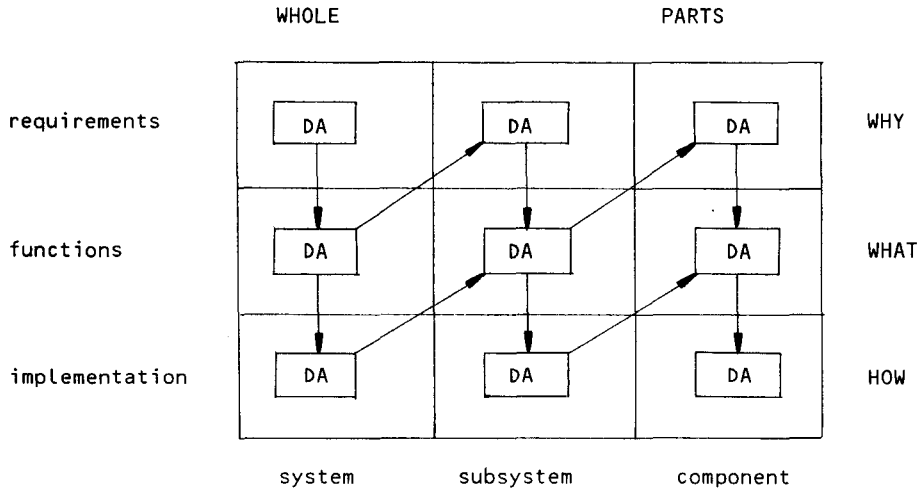


Figure 2.7 The design process broken down with respect to the two dimensions of design. Arrows indicate the temporal ordering between design activities (DA).

The two dimensions of the design, the level of abstraction and the level of aggregation, should be reflected in the design data base to give the designer easy access to data at different levels. A continuous updating of the design data base as the design process proceeds is important, because documentation has to be done when the design decisions are made. The designer should even be forced to give a track of his thought when a requirement is considered and broken down to subrequirements or to a specific technical solution.

The computer-aided design (CAD) system should be adapted to the design process and in tailoring the CAD system it is important to consider the design process as a whole and to investigate when different data items are generated and when they are used. Such an analysis of the information flow during the design will then form the basis for the application of the CAD system to the specific design problem.

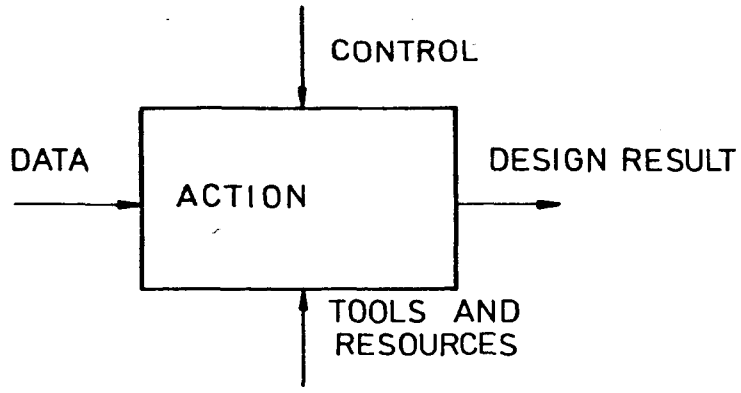


Figure 2.8 A model of the design action, which is used as the building block for the SADT diagrams.

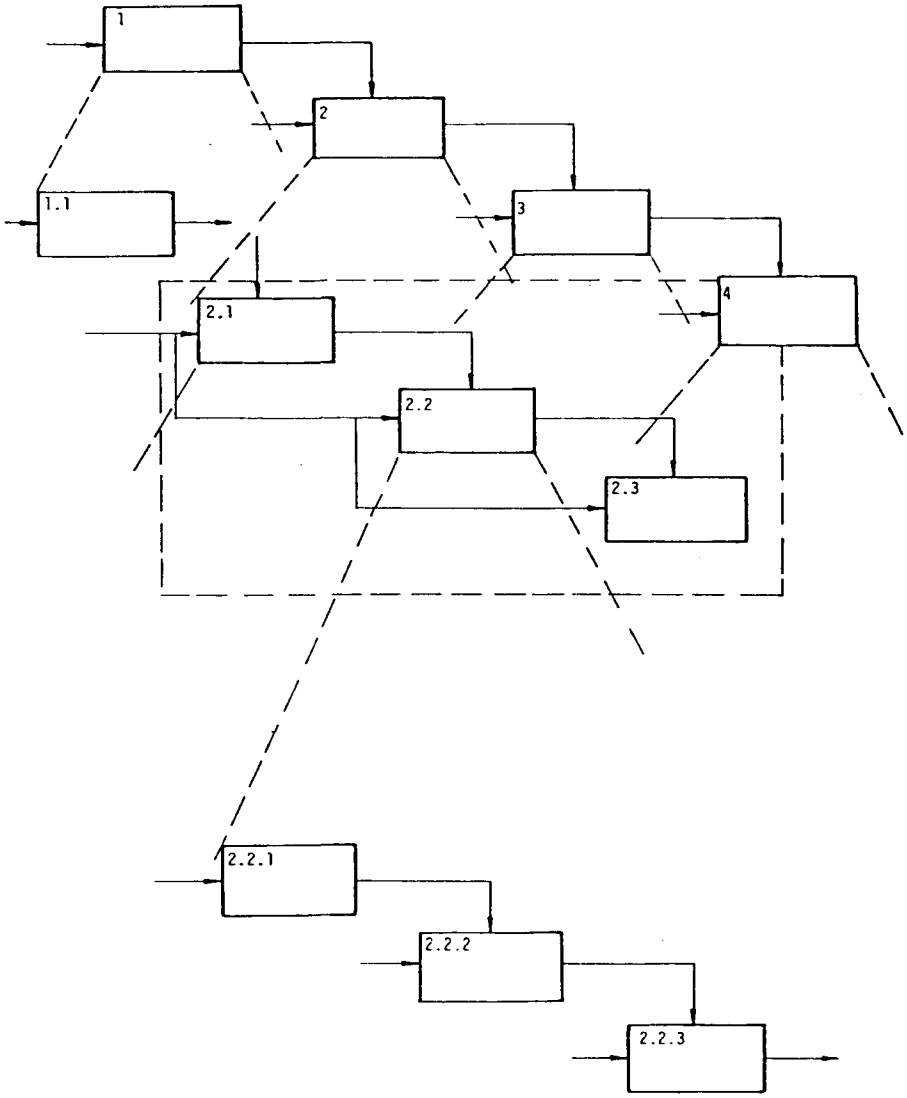


Figure 2.9 The decomposition of the design process.

One possibility to analyse the design process is the so-called structured analysis and design technique (SADT) (cf. Jackson, 1982), in which the design process and its subtasks are described through actions, Figure 2.8. The actions process input data with respect to the control by utilizing certain tools and resources, and they produce as an output the design task of the subtask. The starting point of the analysis is as general as possible, e.g. the basic action could be the complete design process of the control and instrumentation of the plant. The basic action is then decomposed into a fine structure or subactions according to a given rule in the design practice used. The results are structures as shown in Figure 2.9. We can see that it is also possible to follow how control influencing a certain task acts on the subtasks. In the same way we can also analyse the utilization of resources and tools within the subtasks.

Control and instrumentation design can be presented as in Figure 2.10 (cf. Ranta, 1983). The fine structure of the requirement specification is found by decomposing the task into subtasks:

- analysis of requirements originating in law, authorities and social norms,
- analysis of goals related to the plant safety and availability,
- analysis of goals related to the plant economy, product quality and organization

In the same way the functional specification can be divided into subtasks, which can be

- definition of control room functions,
- definition of controls and sequences,
- definition of interlockings and protections,
- definition of reports and alarms,
- analysis of realization possibilities.

These subtasks can further be divided into the sub-subtasks, and so on. The design process is described in more detail by Ranta, 1985.

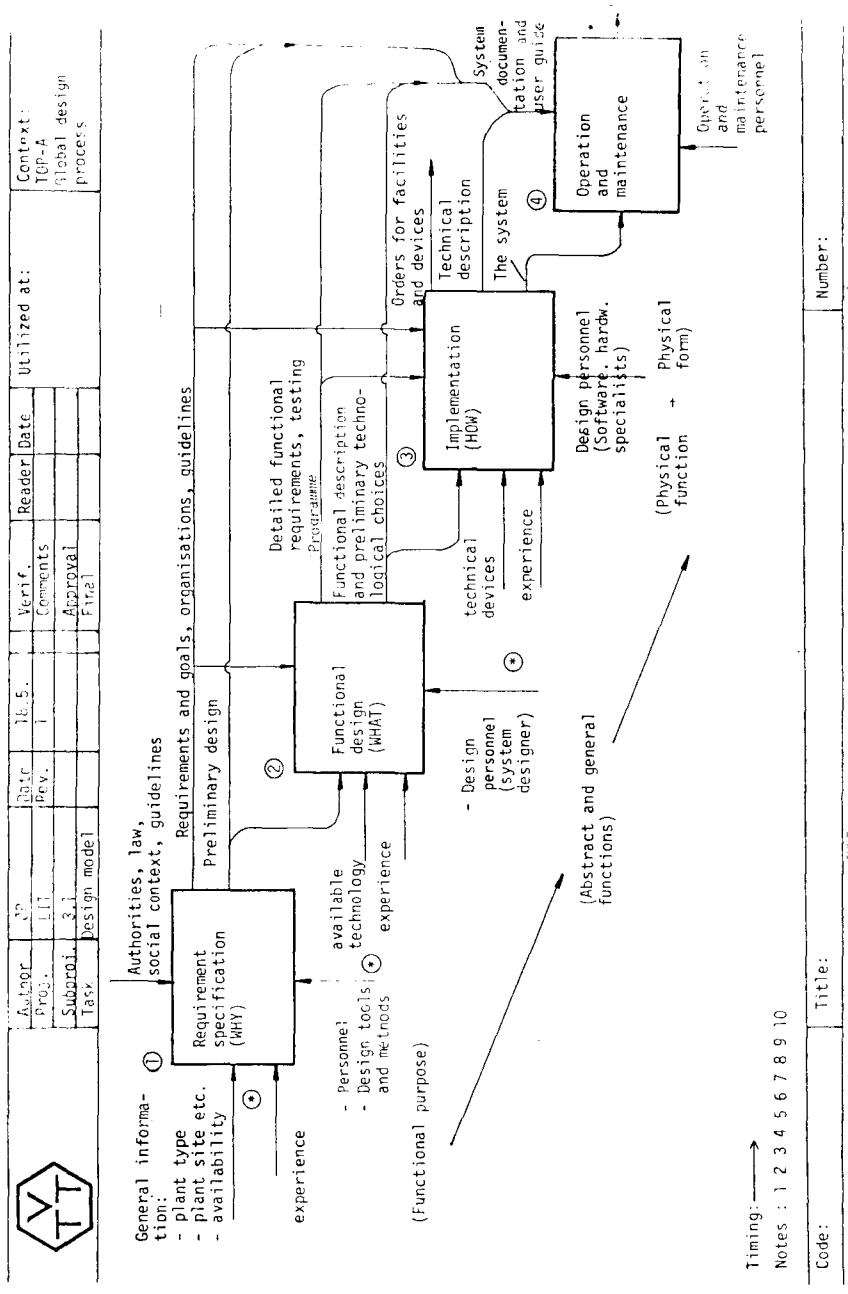


Figure 2.10 The control and instrumentation design visualized with the SADT action diagram. The actions can be further decomposed for a more accurate description.

The formal analysis of the design process helps to specify the design tools to be used by different categories of designers. The formal analysis may also be used as a check-list to structure the design process in terms of:

- subtasks of the design process,
- necessary resources and skills required,
- schedules and check-points,
- standards, guidelines and agreed practices to be used,
- documentation principles,
- supervision, control and evaluation methods to be used.

2.6 Actual design practices

Commonly used design practices do not always follow the idealized model. Where a standarized design has been adopted, in the projects there is not always the same need to consider all the details of requirements and the technical solutions as in a completely new design. Here case the design project tends to be more construction-oriented, and the design concentrates on changes from standardized design.

The commercial interaction between the parties involved in a design project will also bring new aspects into the design. The preparation of the different tendering documents here will actually be a part of the design process. The tendering documents also have a legal status and this means that they are not written for the designers only. The legal status of the documents implies that changes have to be agreed on and documented on different organizational levels, which means that the design data base tends to be distributed into minutes of meetings and other documents.

The administration of a design process and construction project is a task which grows rapidly with the size of the plant. Organizations specializing in design and construction nave therefore developed their own routines for producing, circulating, and approving technical design documents. The division into different offices and departments also offers a model of the design process as it is carried out.

The design process can be visualized in different ways, of which the PERT chart perhaps is the most common. The PERT chart maps the time for ordering different design tasks and their relations. The PERT chart is ideal from a project management point of view, because the chart can be used to monitor the progress of the design with respect to an agreed schedule. The PERT chart can also be used for resource allocation and to calculate a critical path of tasks, which will determine the length of the design project.

3 CONTROL ROOM DESIGN

Control room design has to be based on a consideration of the resources and limitations of the human operator. This means that behavioural science should be combined in the design process and the operability of control room should be the guiding quality measure. It is, however, not possible to embark on thorough investigations on the relative merits of different solutions as a part of a design project. This means that the considerations regarding the human operator should be built into the design tools and the design organization. The section contains the following parts:

* Design of control and instrumentation
* Elements of control room design
* Quality measures in control room design
* Documenting the reasons in design
* Display of design information in the control room

3.1 Design of control and instrumentation

The control and instrumentation of an industrial plant encompass the collection of measuring signals, cabling, control equipment, control rooms, process computers and interfaces to the final control elements. Control and instrumentation design is based on the defined operability criteria of the plant. The operability criteria include requirements on speed and accuracy of the control as defined for different operational states of subprocesses and components. General requirements on the plant in terms of safety, availability and economy also introduce additional requirements on control and instrumentation.

Design control and instrumentation use the process diagrams and system descriptions of the plant. In the design process the requirements are interpreted in terms of functional specifications, which steer the selection of technical solutions. The control and instrumentation design will, for instance, produce the following types of diagrams:

- process and instrumentation diagrams (PI-diagrams)
- control diagrams covering continuous control, sequence automatics, protection logic and interlocks
- cabling diagrams
- software flow-sheets and listings

Control and instrumentation design depends to a large extent on the hardware concept and on the vendor selected. A standard instrumentation system has the advantage of a proven and debugged design, but the possibilities of influencing the details of the implementation are smaller.

The selected instrumentation hardware will actually provide a kind of system design language by which the components of the system are combined. This design language will influence not only software modules but also the actual hardware produced. The design language provides several interfaces by which the different parts of the control and instrumentation are specified, ordered, tuned, produced, tested and installed.

In control and instrumentation there has been a shift from analog to digital systems (cf. Wahlström et al., 1983). The digital systems have several advantages, for instance:

- higher sophistication
- greater reliability
- easier installation and testing
- easier to use
- easier to modify

The digital systems have also made it possible to shorten the time between design freeze and plant start-up.

The clear distinction between the instrumentation and the process computer has been blurred in the development of microcomputer-based instrumentation systems. Most instrumentation systems today have features which earlier were considered typical of process computers. The efficient use of process computers is to a large extent dependent on the availability of a library of tested software modules.

Control and instrumentation design uses different computerized tools to a large extent during the whole production process from tenders to accepted delivery. Typical systems are:

- component and price lists for preparation of tenders
- production planning systems including inventory control
- production control and manufacturing systems
- cable routing systems
- automatic testing including dynamic simulation
- installation management and preventive maintenance systems
- word processing and computerized information retrieval

There is also a trend towards implementing the interfaces between the systems in such a way that information may be transferred in a machine readable format. This points towards a larger integration of future computer systems where design, manufacturing, installation and testing represent different facets of only one process.

Technical development has been introducing new generations of control and instrumentation systems at a rapid pace. When the projected lifetime for industrial plants is between 20 and 40 years it means that most plants will undergo a major revision of their control concepts at least once. This means that control and instrumentation design will also be a concern in plants that have been in operation for some time.

3.2 Elements of control room design

Control room design is carried out as a part of the control and instrumentation design (cf. Ranta, Wahlström, Westesson, 1981). The starting point for control room design is the preparation of the plant automation concept. The automation concept defines the roles of the operators and the automatic control system in different operational states of the plant. In defining the automation concept, consideration has to be given to the type of plant,

staffing policy, available infrastructure, etc. There is usually a minimum level of automation determined by the safety, availability and economy requirements for the plant.

There is also a maximum level of automation set by costs and available technology. In between there are possibilities either to select a high level or a low level of automation. There is, however, a clear trend towards increased automation, which has been facilitated by the improved cost-performance relation of new control and instrumentation systems.

The following general areas of control room design can be identified:

- general lay-out of control room
- control boards and panels
- computer displays
- alarms and other operator aids
- recording and reporting systems
- operational procedures
- maintenance interfaces

The design in the different areas has to be based on the specification of general principles for information coding and presentation which are used in all areas as applicable. The system used for the naming of different objects, e.g. signals, components, etc., of the plant forms an important part of the design where an efficient system has to be selected and applied in a logical way.

The general principles for information coding and presentation are then further concretized, where parts of the control boards and panels are assigned to different subsystems, functions and plant operational states. The detailed design is then carried out by selecting displays and controls for the signals and by giving them specific

locations in the control boards and panels. The final quality check of the design may be carried out using a mock-up of the control room. The general problem of validation of a man-machine interface has been touched on by Hollnagel (1981).

The design of the alarm system and other operational aids should also be carried out by specifying general principles before the systems are designed. Where unproven designs are proposed, it is necessary to build a demonstration system with which concepts are tried out before the final design is fixed. In the demonstration, care has to be paid to the transportability of results obtained to full-scale implementation.

The operational procedures are often written as a task separate from the control and instrumentation design. One possibility is to involve the personnel that will be responsible for the operation of the plant. The writing of the procedures is again divided into two tasks where the general lay-out of the procedures is established before the procedures are written. The general documentation of the plant and the control and instrumentation form another part of the written material included in the control room.

3.3. Quality measures in control room design

The quality of the control room design depends on the operability measures of the plant. The control room should support the operators in their tasks and it should also provide a work environment compatible with human needs. When considering the long term aspects of the control room, there is no contradiction between the system's view and the human view.

The following quality measure factors of control room design have been identified, Rasmussen (1981):

- compatibility and sensitivity,
- trustworthiness,
- clear division of responsibility,
- flexibility with respect to operators' demands,
- reversibility and error tolerance.

Compatibility and sensitivity imply that the control room supports the operator so that he can keep his expectations updated and select a proper cognitive level in carrying out his tasks. Trustworthiness means that the systems in the control room are reliable and that the information presented is dependable. In the control room there should be a clear border of responsibility between the automatic systems and the tasks where the operator is supposed to take action. The systems should also conform flexibly to the operator's preferences, give the operator a possibility to pace his own work, and provide support for all the different tasks required. The control room actions should also always be reversible when possible, and the systems should have a large amount of inbuilt error tolerance.

There are also other quality measures which the control room should satisfy, such as:

- flexibility with respect to changes
- maintainability
- expandability
- time and costs for design and construction

The requirements are occasionally incompatible and in this case a compromise has to be found as a part of the design effort.

During the design process there is a continuous assessment and optimization of the design with respect to the selected quality measures. In addition, there are regular

design reviews included where the quality is checked before a design is approved for construction. A final check on the operability of the control room can be obtained using simulation and a mock-up of the control room.

The final result can be assessed using a check-list (e.g. EPRI, 1984). It is, however, very difficult to get a valid assessment of the operability of a control room based only on details (cf. Norros, Ranta, Wahlström, 1983). An assessment of the design quality has to be made on the same level of abstraction where the designer has the freedom to select between two alternative designs. This implies that also goals, requirements and specifications should be included in the quality control of the design project.

3.4 Documenting the reasons in the design

The requirements and the quality measures of the design will govern the design process. The designer in his work translates the requirements and earlier design into intentions and reasons on which he will base his design decisions. This means that there is a step which is usually not documented in the specification or in the technical solutions. This missing step could on some occasions make it very difficult to trace the connection between the requirements and the technical solution.

The problem in the control and instrumentation design is seen in the difficulty of making changes in the original design because design changes have to be verified. This difficulty implies that modifications of the design often are built as additions to the original design and not as changes. This means that the plant gradually becomes more complex when new systems of exceptions are built onto the old design.

According to present design practices, there are no systematic procedures for documenting the reasons for the design solutions. The documentation of the design has to be done as a part of the design activity, otherwise it is very likely that there will be deficiencies in the documentation. The design has, on the other hand, to be completed before it is documented, as otherwise changes will cause additional documentation tasks. The solution to this problem is to combine the design and documentation effort to make the design self-documenting.

The documentation of the reasons of the design solutions makes it necessary to establish formal design routines by which that information is collected and stored. There should also be some agreed format, which is used to express the purpose of a technical solution because otherwise large variations between different designers are likely.

A simple system for documenting the reasons into the design can be obtained by assigning different goal and requirement identification codes. The designer can then give a formal reference for a technical solution to the corresponding goal or requirement. A computerized system can be built to generate such references automatically based on the inquiries the designer is making in his navigation in the design data base.

3.5 Display of design information in the control room

The designer is in the design process describing his mental model of the process using the terms of the design language. The operator will acquire a mental model of the process which is based on his training and his own operational experience. There will naturally be large similarities between the designer's and the operator's view of the process, but there could also be important differences, and views may differ from the actual design.

One possibility to cope with the problem is to give the operator more direct access to the design data base. The plant documentation will naturally provide such an interface, but it is impractical to use in the control room work. Another possibility, which is becoming feasible with the introduction of computer-aided design, is to give the operator a computerized interface to the design data base. The interface can be realized to provide a kind of computerized knowledge data base to the plant which may be used as a training aid during quiet operational periods.

The control and instrumentation systems in the future are likely to be realized as a network of independent data processing nodes. The flow of information between the nodes will then be governed by standard mechanisms for transmitting data packages from one computer to another. The networks are also likely to include the facility of using any other of the data processing nodes from each of the nodes as a virtual terminal. This means that the operator can be provided with direct access to the interface of the computer-aided design systems used. Indirect access to the design data base can be realized using different automatic production tools. The tools also provide additional flexibility in building different types of operator aids. The present methods for building computerized operator aids already often use the principle of establishing a computerized data base, which gives large flexibility in realizing the details of the operator interface.

The interface to the design data base from the control room has its largest potential in the disturbed situations where there seems to be confliciting requirements regarding the actions to be taken. A possibility to check the purpose of, e.g. an interlock, may give the operator confidence in bypassing it in a situation where it is not applicable. The designer's

intention documented as a part of a control loop may also help the operator to diagnose and correct failures in the controller. In another situation the chain of associations between two objects of the design data base may make the operator aware of an important connection between two subsystems which will help him to avoid an operational error.

4 COMPUTER-AIDED DESIGN

Computers are the obvious solution to many of the human short-comings in the handling of large amounts of information and so also in the field of design. Control and instrumentation design and control room design particularily, have many similarities but also important differences as compared to other areas of design. This means that a computer system intended to support control room design should be based on a careful assessment of the needs, combined with an evaluation of available functions of a commercial system. The optimal system for a specific environment will always be a combination of standardized parts and tailor-made solutions. The section contains the following parts:

* Computerized design methods
* Advantages of computer-aided design
* Handling lists of plant objects
* Formalization of the design process
* Association networks of plant concepts
* Retrofitting of plant design data bases

4.1 Computerized design methods

The introduction of powerful computers has made it possible to rationalize the design process. At the same time there has been pressure to increase the productivity of the designer in order to cope with increased design costs. It is therefore very natural that there has been an increased interest in computer aided design (CAD).

There are commercial CAD systems available, but they to a large extent concentrate on the generation and handling of two and three-dimensional drawings. The systems have been applied to design in different fields, such as:

- electronics design,
- building and construction work,
- manufacturing.

The development of the present CAD systems has meant that many data processing and algorithmical problems have been solved and has thus itself generated a research field The use of computers in the control and instrumentation design has been advancing from the most urgent needs, and different stand-alone systems have been utilized. In the field of nuclear power, computers have been used to support design e.g. in the following tasks:

- transient analysis using accident codes
- design and verification of control concepts using simulation models of plant processes and control systems
- generation and handling of lists of signals, components, displays, alarms, etc.
- reliability calculation and risk assessment
- management and production of drawings and other plant documentation.

The functional realization of a CAD system is illustrated by Figure 4.1. The designer has access to the design

data base through a user terminal. The design data base is stored in different data files and user access to the design data is either on a read-only or a read and write basis. Several designers can work with the system in a time shared mode. From his terminal the user can initiate the following general classes of dialogue:

- make inquiries on data stored in the design data base,
- ask for sorted listings and different reports,
- give new data to the system,
- modify old data in the system,
- generate new transformed data files from the data in the data base,
- check the validity and the quality of data in the design data base.

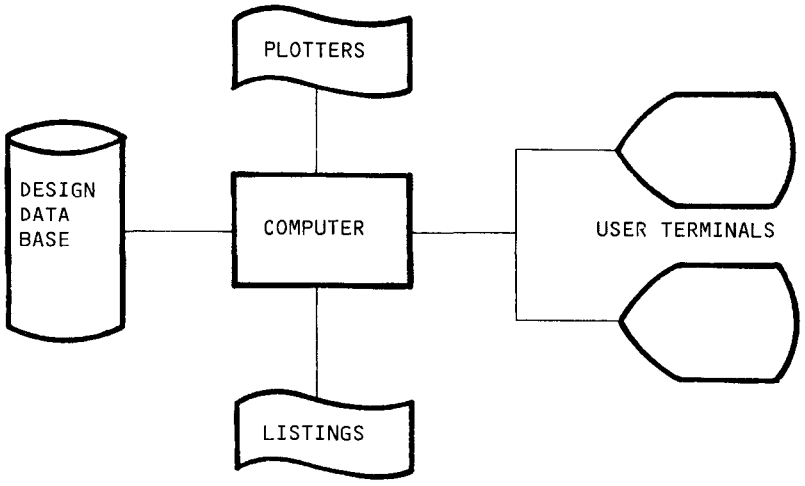


Figure 4.1 Functional realization of CAD system.

4.2 Benefits of computer-aided design

The use of computer-aided design has the potential to increase the productivity of the designer. The most

important effect is, however, when plant lifetime costs are considered, the increased quality of the design (cf. Lauber, 1983). The increased quality has several components, which may be attributed to the more efficient design process and the use of automatic design tools.

A computerized design data base gives the designer far more efficient access to the current state of the design than any manual system. This makes it possible for him to avoid the danger of working with outdated design. The computerized interface makes it also possible to build a system of associations between the objects of the design. This makes it easy for the designer to make cross-checks between requirements and technical solutions that have an influence on the design he is working on.

The management of changes is also far more easy in a computerized design data base, because the change will be introduced in only one place and is immediately effective. The computerized design data base can also keep track on parts of the design which will be influenced by a change.

A computer-aided design system can also be constructed to standardize the design by the use of default mechanisms. The system will then automatically present a suggestion for a design solution for the designer, which he can approve or disapprove. The standardization of the design can also be enhanced by building in different guidance and help mechanisms by which the designer can make inquiries.

Automatic verification of the design can be applied by building software modules which check the content of the design data base according to some accepted verification rule. The most simple check possible to carry out, is to check that all items of the design data base have been defined when the design project is completed. Other verification procedures could use a general design rule, which has to be satisfied for some subset of the

design data base. The verification modules may also be used to compare two alternative designs by calculating a performance measure of the proposed alternatives. One example of an algorithm for an automatic verification of proposed alphanumeric displays has been reported in Danchak (1985).

Automatic production can be used for the generation of software modules or to prepare parts of the documentation.

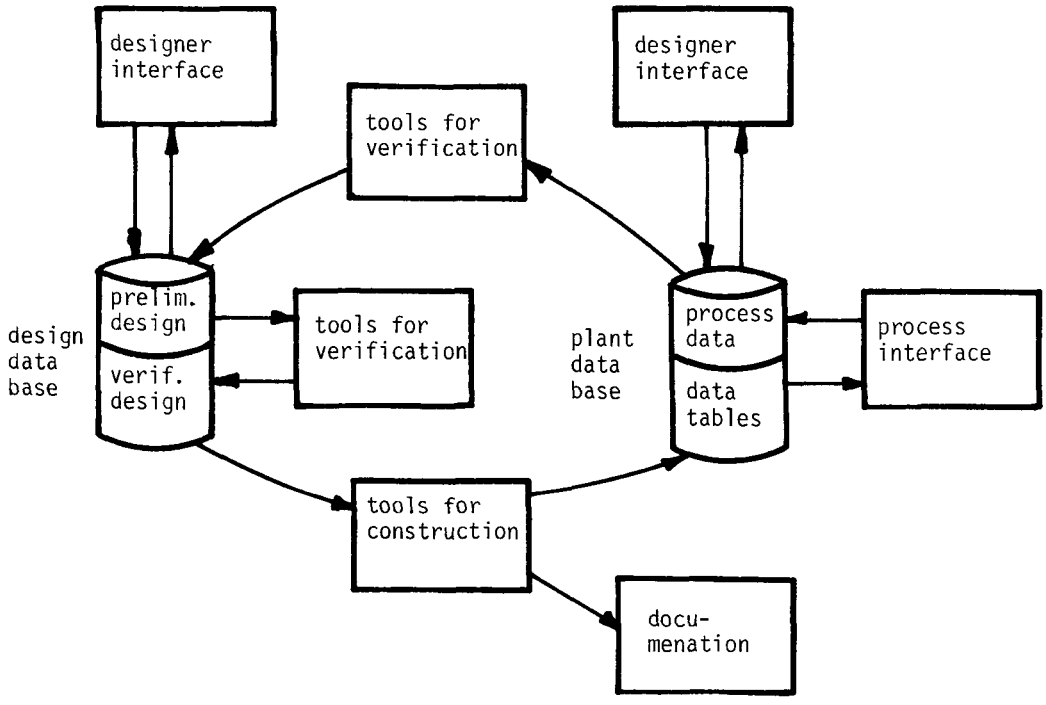


Figure 4.2 The computerized tools of the designer and the operator and their relations to the design data base and the process data base.

Computers used in the controls and instrumentation will require a large programming effort, which at least partly can be realized using program generators. Program generators may also be used for the production of other software, e.g. to be used in training simulators.

The preparation of plant documentation is an important part of the design process which is easy to automate. The generated documents can also be used as sheets for the collection of additional design information. The use of document generators makes it also possible to tailor the documentation for a specific user. Figure 4.2 is presents this general view of a design system.

4.3 Handling lists of plant objects

In the design process it is necessary to establish different lists of plant objects and their attributes. Typical plant objects appearing in the list are

- measuring points
- control elements
- control room displays
- alarms
- plant components

The attributes of the objects include identification, location, power supply, cabling, logics, associated drawing, data processing, etc.

The lists can be generated from a data base in which the plant objects have been stored in standardized records. The classification system by which plant components are given codes are then used as the primary search keys in the data base. Table 4.3 presents a typical compilation of listing keys and corresponding items listed.

The handling of ordered lists of plant objects provides many of the benefits of computer-aided design systems earlier indicated. The classification system itself

Listing key	Items listed
system	components
	measuring points
	control elements
	procedures
display page	components
	measuring points
	control elements
	procedures
procedure	components
	measuring points
	control elements
	displays
algorithm	analog signal
	binary signal
	control element
	parameters
measuring point	display
	procedure
	algorithm
	equipment

Table 4.3 Examples of ordered lists possible to generate on a specific listing key

provides paths of association which can be used to list all components or signals within a given subsystem. The naming conventions for panels, displays and control

process components
- valve
- pump
- pipe
- tank
- heat exchanger
- special components

measuring point
- pressure
- temperature
- flow
- level
- radiation
- concentration

control and instrumentation
- control loop
- sequence control
- protection logic
- interlock logic

display pages
- tables
- control displays
- drawings
- curves
- special displays

Table 4.4 Example of a compilation of objects with application to a power plant

schemes also provide a possibility of making the necessary associations in the data base for the listing of all signals addressed in a panel, a display or a control algorithm.

The lists generated by such a system are used in the design process as documents of reference. The lists may also be used to provide a simple check on the completeness of the design. In testing the installation at the plant the lists are important instruments, which are used to tick off the work as it progresses.

The lists of plant objects are also supplied with reference to actual hardware and the lists are included as enclosures with the tendering documents. A list of typical objects of a power plant is given in Table 4.4. Sometimes there are direct connections between the lists generated and the project management, e.g. when the lists are transferred in machine readable format between the computer systems used for project management and the design data base.

Simple record-oriented data base systems together with an efficient plant classification system can provide a design system including many of the advantages of computer-aided design systems earlier indicated. A record- oriented data base system does not, however, provide for flexibility in defining new concepts. Neither will a record- oriented data base system provide the rich structure of associations between different concepts used in the design. Natural language descriptions are also difficult to implement in the record oriented-structure.

Recent innovations in the theory of data base systems have made it possible to use efficient search strategies with different kinds of search keys. For power plant application the existence of an unambiguous component-naming convention using a classification system provides the most natural primary search key. The use of linked

random storage files makes it possible to use names, with redundancy without sacrificing fast access to items in the data base.

4.4 Formalization of the design process

Present design practices rely to a large extent on natural language descriptions. This means that requirements, goals and technical solutions are embedded in the system descriptions, serving as one of the inputs for control and instrumentation design. In order to carry out computer searches in the design data base, the information has to be content-coded in such a way that, e.g. a requirement is given an identification code and is stored among the requirements.

The need for content-coding of information on the design data base implies that a far-reaching formalization of the concepts has to be carried out. Formalization means that the semantics of the design specifications have to be written using an artificial design language with a well defined syntax. This means that the governing concepts i.e. goals, requirements, functions and technical solutions of the actual design process should be extracted, combined and defined as artificial concepts to be used in the CAD system.

To illustrate how the formalization of the natural descriptions may be worked out, one can consider e.g. the goal that the display system should be easy to operate. This goal is then given an identification code indicating that it is a terminal goal. This goal may be broken down e.g. as in the following two quantitative subgoals, which are given their own identification:

- the complexity index of each display should be less than a specified parameter
- the distance between two objects operated within one step of a procedure should be less than a specified parameter.

An automatic checking of the goal and two subgoals will then require that the technical solutions of the displays and the procedures can be written into the design data base. The algorithms for calculating the complexity index and the distance between two objects must be defined and programmed. When the displays base has been designed, then the designer can initiate the automatic checking, which calculates the complexity indices and the distances. If the goals are fulfilled, then the goals are given a reference to the displays and vice versa to indicate that the design has been proven. If not, then the designer gets information on which displays should be improved. If a proven design is changed then the system should remove the corresponding reference from the display and the goals and generate a reminder to the designer that the goal checking algorithms should be run anew.

The artificial design language cannot, however, deviate too much from the design practices used, because it should be possible for designers to adopt the new practices with a minimal amount of training. In the artificial design language there should be necessary support also for the use of natural language descriptions. The natural language descriptions could, however, be restricted to records of finite length, which are used to characterize some general concept, such as a description or a purpose.

In the formalization of a design language it is also important that persons with different educational backgrounds are able to use the language. The design language should also support all the different views of the design which are applied during the design. The language should have a clear and simple structure to make it easy to build different interfaces to it. The most important interfaces to the design data base are:

- interfaces between different designers and the design data base

- interfaces between the design data base and verification tools
- interfaces between the design data base and production tools
- interfaces between the design data base and control room operators

4.5 Association networks of plant concepts

The formalization of a design language means that a formal system should be built defining all the concepts, used in the design process. To be feasible such a formal system has to fit very well with the natural language models of the plant used by plant designers and operators. The formal description should include:

- the concepts
- the syntax of the language
- the semantics of the sentences
- the associative relations between the concepts
- a system of metaconcepts and metarelations

The concepts should include plant objects, their attributes, actions, parameters, logical delimiters etc. The syntax of the language is a description of how the defined concepts may be combined to form allowed sentences within the formal language. The semantics of the language is a natural language description of the meanings of the concepts and the allowed sentences. The relations between the concepts are actually embedded in the syntax and the semantics but are to be handled separately because they should provide the search mechamism within the data base. The metaconcepts and the metarelations form a kind of artificial core of the formal design language, which is used to define the concepts and the relations. The concepts, objects, attributes, actions etc. are thus actually metaconcepts because they are use to characterize very general types of concepts. Similarily the relations

being connected; being a member, having a property, etc. are examples of metarelations.

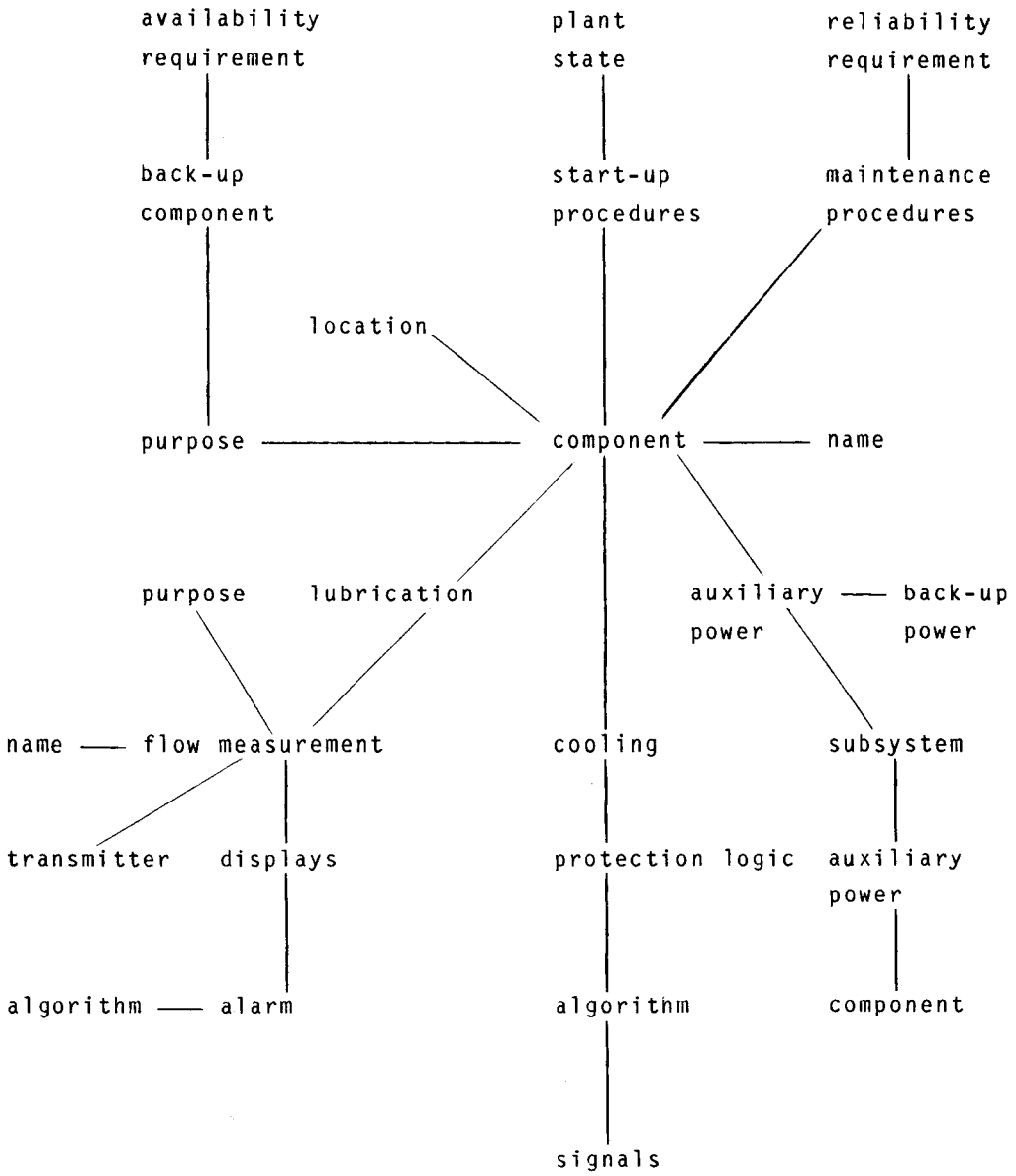


Figure 4.5 Example of possible associations starting from a component. In the design data base it should be possible to move from one object to another along the relations handled.

The construction of the design language could proceed by defining a minimal set of plant concepts in interviews with designers and operators. Table 4.4 compiles some typical plant objects, which will be included in the set of concepts. The associative relations between the concepts are then collected and documented within a number of prototype descriptions. A final formalized model of the plant concepts can then be found as a structure which encompasses most of the descriptions. The model of the plant concepts should be based on a system and task description and can be documented using a standard model for describing formal languages e.g. the Baccus-Naur formalism (cf. Tuominen, Wahlström, Timonen, 1978)

The associative links of the design process in terms of the two design dimensions, the aggregation dimension and the abstraction dimension, are one important part of the association network. The associations between concepts, subconcepts and superconcepts is another important mechanism. Figure 4.5, give some typical association networks between plant concepts.

Considering the general association network, one could separate the following meta-relations:

- types
- connections
- attributes

The type relation indicates a coupling from the object to another more general object and thus introduces a hierarchy of the objects. One example of the connection relation is the piping network connecting different components of the plant. The connection relations introduce a general coupling between two objects which is not hierarchical. The attributes of an object represents a set of different properties associated with the object. The connections and the attributes can be typed in the

same way as the objects. One example of possible relations is given in Table 4.6.

type	being part of
	being included in
connection	giving input to
	getting output from
	satisfies
	satisfied by
	to be applied in
	requires application of
attributes	name
	identification code
	purpose
	location

Table 4.6 Examples of relations between concepts. The relations will provide an associative branching from each concept considered in the design data base.

The use of a conceptual model as the basis for an understanding of the plant is illustrated in Hollnagel, Woods (1982). The formalization of such a conceptual model also has interesting implications on how the plant is operated.

4.6 Retrofitting of plant design data bases

There are several advantages of computer-aided design for new constructions and therefore one could ask if there are benefits with using computers for the design data base also in plants already in operation. If so, it may be advantageous to consider the computerization of the design data base as a retrofitting effort. Considering the

present problems with plants already in operation, one can identify the following

- handling of plant documentation
- management of changes in the plant,
- installation of new operator aids.

The plant documentation contains immense amounts of information and the filing system has to be very flexible if a document sought is to be found. The application of methods developed for modern libraries may provide remedies for many of the problems of cross-referencing and executing document searches. Where the documents do not contain key words, such can even be generated automatically by programs operating on the titles of the documents. Modern information storage techniques may also provide a possibility of storing most of the information in machine-readable format only.

The management of changes is the touchstone of any design data base, and in using manual systems there is always the risk that some important part of the plant or the plant documentation is not updated. One example is the operational procedures, which are very seldom kept up to date with respect to the annual revisions at a plant. Even the use of computers in the form of word processing systems makes it far more easy to make necessary updates as a result of a change.

The handling of the operational procedures, however, represents a task where the benefit of more advanced systems than simple word processing may be found. By the establishment of a standard vocabulary, one can ensure that a consistent naming convention is used in all operational procedures. One can also establish simple search routines for specific components, which means that a listing of routines effected by a change is obtainable. A standard coding of the procedures makes it also possible to include the procedures in computerized operator aids.

The installation of new operator aids always means that some design effort has to be carried out. In the construction of specific operator aids, there is the benefit of building tools by which the design is implemented. Such a design tool will always mean that some description of the plant is built into a design data base which is used by the operational aids. This again points to the benefit of having the plant data in machine-readable format, which gives the possibility to build the system using construction tools rather than to build the system manually.

Considering the lifetime projections of an operating plant with the prospect of going through at least one major revision of the control and instrumentation, there are many arguments for the computerization of the design data base. The revision is likely to be a major redesign which will heavily rely on the use of computers and computerized control and instrumentation systems.

A retrofit of an operating power plant in terms of a computerized design data base will naturally not cover all the aspects considered feasible in a new design.

5 AN APPROACH TO DESIGNER WORK BENCH

Computers have had a large impact on many human activities and it is likely that they will also have one in the field of design. What the final impact will be remains to be seen, but experience from other fields has shown that it is likely that work practices, organizations and even the concept of the design profession will change. If the changes are properly considered before introduced the chances are higher that the development will be a positive one. The designer work bench outlined below tries to identify some technological details of the development and their impact on control room design. The section contains the following parts.

* System modelling languages
* The construction of design tools
* Default and inheritance mechanisms
* Automatic verification tools
* Automatic production tools
* Technological changes in the design process

5.1 System modelling languages

The design of complex industrial plants is placing new demands on the integration of different views towards design. These demands together with the power and flexibility of modern computer systems make feasible new integrated approaches towards the design process. This means that not only the control and instrumentation design should be integrated into one approach, but also other phases of process design. This concept gives a more general view of a designer's work bench with a tool box of computerized design tools.

The designer's work bench means that a common framework for all aspects of the design should be established. This again implies that a common language for the modelling of the system to be designed should be constructed (cf. Lind, 1983). Such a system modelling language would provide a framework for the designers in their task of describing their visions of the plant to be constructed.

It is clear that a system modelling language will not provide only one, but rather many descriptions of the plant (cf. Rasmussen, 1979). One aspect of the modelling would certainly be a 3-dimensional computer model of the buildings and the piping of the plant. Another view in the modelling would be the main components, their characteristics and their connection by the piping network. A third view is provided by the cabling, both for the control and instrumentation and the auxiliary power system. A fourth view is provided by the fault trees and cause consequence diagrams of the risk assessments of the plant.

Within the control and instrumentation, one benefit of using a high-level system modelling language is that a common description can be used, regardless of the final implementation, for a control law. A common description

makes it possible to advance the functional design fairly far without making the final decision where to locate the control law. When the final decision has been made, program generators can then be used to convert the functional description to operational procedures, process computer software, or instrumentation system hardware descriptions.

Item	Content of design data base	Task of production program	Use of the model
plant structure	plant components and their interconnection	preparation of input data for standardized simulation models	calculation of plant responses during transients
control structure	controllers, protection logics	compilation of simulation models	verification of control schemes
control room description	procedures, locations of displays and controls	preparation of a time line model of operator tasks	assessment of operability of control room
structure of safety systems	engineered safeguards, protection systems	preparation of fault trees and cause consequence diagrams	reliability analysis and risk assessment
instrumentation structure	locations of flow, level and pressure measurements	search for aggregates in terms of mass and energy flow	assessment of completeness of instrumentation for diagnostics
plant concepts	concepts, concept types and their relations	preparation of a formal model of plant concepts	elimination of contradictions and verification of *completeness*
plant dynamics	simulation model for calculation of responses to control inputs	calculation of small signal model	construction of control algorithms

Table 5.1 Possibilities for automated production of models for the verification and validation of plant design.

The system modelling language would provide a computerized model of the plant, which can be used for different automated verification checks. The model is also available as a computerized blueprint of the plant for a more or

less automated production of the different parts of the real plant. The newly adopted concepts of computer-aided manufacturing make it also possible to proceed fairly far in pursuing a really automated production system. Some possibilities for the use of a computerized design data base are illustrated in Table 5.1.

5.2 The construction of design tools

The construction of design tools will follow similar paths as used in software design, where several hierarchical levels of tools are built. The implementation of a computer-aided design system should as far as possible rely on the use of standardized systems. With the present level of technology this means that an efficient computer together with a suitable operating system is selected for the final implementation. The computer and the operating system selected will influence the transportability of the design data base to other computers. The final selection will depend on the demands on facilities, and the capacity and efficiency of available data base systems. At present, there are many commercial systems available and a suitable relational data base system would seem to provide an efficient building platform. Other possibilities are the computers used in the field of artificial intelligence.

The next level in the construction process of the design data base is to use the primitives of the data base system to define a set of primitive concepts, by which the design concepts are implemented. The definition of the primitive concepts also includes the definition of typing mechanisms and the handling of connections and attributes. The primitive concepts in this connection are equivalent to the metaconcepts and the metarelations touched on earlier.

When the primitives of the design data base have been defined, the actual objects of the design process itself

can be defined. The definition of the objects involves the utilization of the typing mechanisms, the connections and the attributes to define an internal structure in the design data base. This general approach of building the system modelling language on several hierarchical levels is illustrated in Table 5.2.

Level	Design activity
4	design activity using concepts of actual plant
3	definition of plant concepts using general metaconcepts
2	realization of a set of metaconcepts using a relational data base
1	implementation of a relational data base using commands of the operating system
0	operating system of computer used

Table 5.2 The general principle of building the design data base from computer primitives

The construction of the design data base using several hierarchical levels, where primitives on one level are used to define the primitives on the next level, provides a simple interface with the design data base. With this constructional procedure it is possible to exchange parts of the hardware or software of the design data base without facing a complete rewriting of all the definitions. The interface between different data bases is also simple to build, because it is only necessary to find

a common interface level for the data transmission mechanisms. Even if the data bases have been using different construction principles, it is possible to duplicate the necessary transitions between the hierarchical level of the implementation.

5.3 Default and inheritance mechanisms

The need for the standardization of the design can easily be implemented in a computer-aided design system by different default mechanisms. A default mechanism provides an assumed value for the objects of design in the data base. The designer can then either accept or change the default value when an object is the target of a design task. The default mechanism is used to define typical constructions which the system will propose as a design solution whenever the designer attempts the design of an object which is of the defaulted type.

Another mechanism by which standardized solutions are built into the design is inheritance mechanisms, which go through the typing concept. The mechanism implies that each object will inherit the properties of its parent objects. The inheritance mechanism means e.g. that the same types of attributes that have been defined for an object type can be transferred to be defined for each of the objects of that type. Similar inheritance mechanisms may be defined for the connections and the attributes. The inheritance mechanisms may be used to verify the completeness and the consistency of the design by checking whether child objects have all the inherited properties defined and whether the child and parent properties defined are consistent.

The default and inheritance mechanisms will be defined by using the primitives of the design data base. A change in the default or the inheritance mechanisms will affect the design data base fairly extensively, but will be traceable

through the tree of definitions of the data base concepts.

The default mechanisms should also be extended to encompass larger entities in terms of definitions using the concepts of the design. Such a macro definition mechanism means that it is possible, e.g. to define a standard flow controller by a typing mechanism, which by its attributes supports the definition of a flow meter, control valve, algorithm, display and control element. A general controller can be a similar extension of all different controllers. This mechanism makes it easy to duplicate the same standard solutions throughout the plant. The generation of drawings, wiring diagrams and procedures can then be done automatically by a program making necessary modifications of applicable attributes.

The inheritance mechanisms makes it also possible to construct more complicated schemata, which are inherited through the typing mechanisms. They can be connection types and attribute types, which are supposed to be inherited by the use of object types i.e. use of an object type will not only transfer attributes but also attribute types from the parent to the child objects. Such mechanisms suppose that both object types are typed with consistent connection and attribute types.

The combination of production modules with the default mechanisms makes it possible to automate the design by letting the system propose design solutions. A production module will in this connection combine information in the data base according to some specified rules and present the result for the designer as a candidate design. Such a system will approach expert systems for design, where a number of design rules can be integrated in a knowledge base for the design process (cf. Barr, Cohen, Feigenbaum, 1981).

5.4 Automatic verification tools

The verification of the completeness and consistency of the design is an important task of the design process. It is also a very time-consuming and monotonous task, where the alertness of the designer plays an important role. It is therefore very natural to try to build automatic verification tools, which can search for deviations from accepted designs rules.

The most simple automatic verification system is search for data items not defined in the data base. By the system of inheritance, mechanisms such as verification become a comparatively efficient tool to ensure that everything has been considered. This verification procedure implies that the default values also should have a flag, indicating that the value has been accepted.

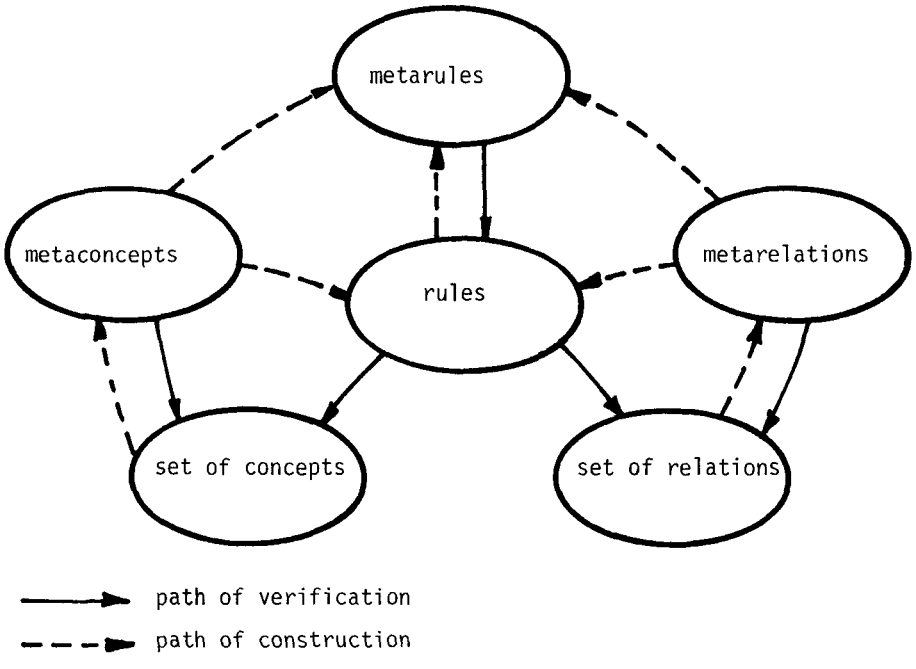


Figure 5.3 The procedure of construction and verification of items in the design data base

Another simple verification procedure is to check the input from the designer with respect to plausibility. This also means that each designer has some area of responsibility where he is able to work. All other areas of the design data base are then considered as read only areas, where the designer is allowed to look at the data but not to change it.

Automatic verification of the design data base implies that some general rule can be constructed which the data has to satisfy. Such rules will be defined on a higher level than the rules included in the default and inheritance mechanisms, because to be verifiable they should not be built into the design. Such rules will be either a part of the design requirements, or the result of a deliberate design effort. Figure 5.3 illustrates this general approach. Guide-lines represent typical examples of design rules for which automatic verification procedures may be built.

In considering control room design there are different possibilities to carry out automatic verification procedures. One possibility is to correlate the operational procedures with the control room description in terms of displays and controls, and verify that it is possible to carry out the required actions within the time frame allowed. Such a verification procedure makes it also possible to assess the control room lay-out in terms of operator movements in a specific transient.

Another possibility is to verify the consistency of the man-computer dialogues with respect to more general rules on how commands and messages should be constructed. Such systems have been proposed e.g. for the verification of command structures of text editors.

5.5 Automatic production tools

The most important production tools are directed towards

documentation. The use of document generators gives ample flexibility to tailor the documentation to specific needs. One example is the divergent needs for the operational procedures to be used in the training and in the control room. The documentation tools make it also possible to use the control room computers for paper-free documentation of plant design.

The use of computers in the control and instrumentation makes it possible to generate parts of the operational software by program generators. The description of the control concepts in terms of a functional language may be utilized to compile the control algorithm into the programming language of the target machine. This means that a compiler is written for an automatic compilation to the target machine rather than coding the control algorithms directly for the target machine. The implementation control and instrumentation will also need reference to different data base items, e.g. to set up tables for the computer inputs and outputs. It may also be possible to generate algorithms for the validation of process signals (cf. Lind, 1984).

The plant description in terms of components, their properties, and their interconnections, can be used to generate simulation models by setting up the data bases and the connections between models. The simulation models may then be used to verify the control concepts in different simulated transients.

The plant descriptions can also be used to automatically generate fault trees for use in safety studies. The generation of the fault trees will in this case assume that minifault trees have been defined for the different components under consideration (cf. Suokas, Karvonen 1985).

The naming of plant components is a task where a large

number of conventions are used in spite of the selected classification system. By building a naming system as a module into the design system, it is possible to make the rules explicit and thereby ensure a logical application of the classification system. A similar system could also be used for the writing of a procedure where a module for the handling of terms could propose standard words to be used. The module may also include syntax checking and modification to make all procedures conform to the same accepted standard.

5.6 Technological changes in the design process

It is very clear that computers will play an increasingly important role in the design process in the future. Some of the reasons have been touched on already, but perhaps the most important is the increasing complexity of modern industrial plants. The increasing complexity of the plants implies that the design effort is growing rapidly. The increased complexity also makes the design more vulnerable, because the management of the design process is becoming increasingly difficult. The increased size of plants also implies that the economic risks associated with poor design quality are increasing.

The increased complexity of industrial installations has an impact on the operation of the plants. It is getting considerably more difficult for one operator or even a team of operators to handle all possible combinations of events. Still the complexity of the plants makes it necessary to have human operators as the final line of defence with an unforeseen disaster. The only possibility to cope with the complexity of the operational tasks is to present all possible information the operators may need on the plant and its past, present and future states. The construction of such plant-wide information systems will require a considerable design effort together with a better understanding on what information is needed in different operational situations.

Another development trend which may be seen are the new components in the roles of the control room operators. They have already been participating in design projects, they have been given maintenance tasks and they have been writing operational procedures. One can expect that the new control room systems will provide more functions for building specialized displays, which the operators can learn to use. This may even lead to a situation where the operators will be responsible for the detailed display design and the designers only build the tools for the design. The same trend may apply also to other areas in the control and instrumentation design.

The development of more efficient concepts of high-level programming languages has been seen as a trend for a time in the field of software engineering. A similar development of high-level design languages can be expected. One could even argue that the object of design effort is slowly shifting from the plant itself towards the design tools. This means that future computer-aided design systems may contain only the tools by which new tools are built and not the design tools themselves.

Another tendency in the design process is that by use of the computerized design tools it is possible to make the technological decisions for the control and instrumentation late in the design. This has also the benefit of allowing the design to be completed to a far greater extent than at present before the restricting decisions have to be made. This also implies that it is possible to utilize the latest technology for the control and instrumentation, with the correspondingly longer projected lifetime.

The introduction of the computerized design tools will tend to make the borders between different areas of design less sharp. This means that there will be an integration of different design data bases. The integration will, however, not imply establishment of one

large monolithic design data base; a more likely development is the introduction of a network of independent CAD systems. The apparent integration of the design data bases is then realized using well-defined interfaces and mechanisms for data transmission. A conceptual model of the future design process is given in Figure 5.4.

The design process and the design organization will also be influenced by all the innovations made in the field of office automation. This means that electronic mail will be used between different members and parts of the design organization. Word processing systems will be a standard feature of the designer work bench and standard schemata will be used for minutes and messages. The project management routines, such as budgeting and schedules, will be integrated in the information system of the design organization.

The experience needed by the future designer will certainly be in the area of software engineering. He should be very familiar with the use of different conceptual models and he should be able to express such models in a formal programming language. He should also have experience in the use of available systems for office automation. This background does not, however, mean that the designer will manage without a thorough knowledge of the process he is designing for. Again it may be difficult to combine all the required abilities in just individual persons, which means that design probably will continue to be a team effort.

In many fields where computers have been introduced, a fear has risen that the computer may automate the man out of the system. One could ask if the same fear may be applicable also in the field of design. With automation there is always some tacit human knowledge which may be lost, and this will probably also be the case in the field

of design. In considering control and instrumentation design for complex industrial installations however, there seems to be no fear that the designer will be put out of a job by the new systems. Instead, it would seem to be necessary to mobilize all the inguenuity of man to design the industrial systems of the future, which must be safe and economic to operate.

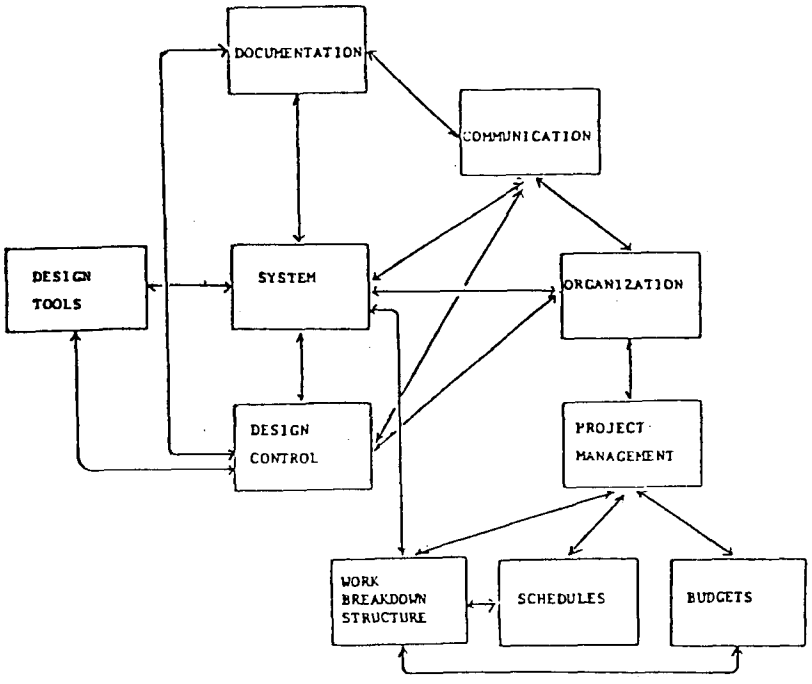


Figure 5.4 A model of the design system.

6. A DEMONSTRATION SYSTEM

The system modelling language elaborated below has been built on many of the ideas contained in the section on the designer's work bench. When the SML demonstration was built, these ideas were however not as clear as they were in writing this report. The actual demonstration was also far more modest than originally intended. In spite of this it is our hope that the reader should be able to grasp at least a part of the possibilities inherent in using computer-aided design in the field of control and instrumentation. The section contains the following parts:

* Goals and scope of demonstration
* SML concept
* Elements of SML
* Use of SML
* Handling of requirements in SML
* Connections between SML and the design process
* Features of the SML concept
* Implementation of the SML concept
* SML-user interface

6.1 Goals and scope of the demonstration

The motive for building a demonstration system as a part of the project was twofold:

- it is necessary to check the feasibility of the proposed concept before a full-scale implementation can be initiated,
- the demonstration system can serve to give rough estimates of the effort needed for the implementation in terms of cost and manpower.

It is immediately clear that the demonstration system has to be restricted both in the level of detail and with respect to the functions implemented.

In specifying the scope of the demonstration system all references to automatic drafting and 3-dimensional modelling were excluded. The reason was not that these parts would be unimportant in a CAD system, but that they could be implemented using commercially available modules. The functions intended solely for the management of component data such as code, type, location, manufacturer, state, etc., were also not included, because many utilities and vendors are already using such systems. The demonstration system was instead directed towards the problems of the design where additional benefits of a computerized data base could be seen.

The demonstration system especially addressed the problem of establishing a computer-traceable connection between the requirements and the technical solutions. An additional aim was to build the demonstration system to support the division of the design data base according to the two dimensions of design; the dimension of abstraction and the dimension of decomposition/aggregation.

The aim of the demonstration was also to gain experience in how the present design practices could be formalized. Formalization implies that the concepts relevant to the design process are extracted from the natural language descriptions and are integrated in to the formal design language. In this process the designer uses the concepts of the designed system modelling language, which provides him with a set of design tools. The target for the demonstration was the properties of the system modelling language not the design process itself. The construction of the design language will be a part of the design process itself and therefore the demonstration concentrated on the establishment of the necessary metaconcepts of the system modelling language.

The definition of the metaconcepts of the demonstration system was based on an information analysis of concepts from a nuclear power plant. The information analysis was used to convert sample system descriptions into a formalized models of metaconcepts, by which an equivalent plant description can be built. This formal model was then reduced to a minimal system of metaconcepts, which could be used to reconstruct all the concepts used. Appendix 2 illustrates the use of the present demonstration system. The demonstration has been built to allow also restricted natural language descriptions.

6.2 SML concept

The formal language used in the design may be seen as a tool for building a model of the plant. This model will then serve as the blueprint for a more or less automated construction process. Therefore the demonstration system in the LIT-3.1 project rapidly evolved into the concept of a system modelling language (SML).

In the definition of the SML concept, formalism from the field of software science was used because it is advantageous to use standardized constructions. This means that a language of metaconcepts is defined by which the concepts of the SML are defined and built. This approach has also the benefit of allowing the addition of new and the deletion of old concepts with time. The language of metaconcepts thus includes only the skeleton of the SML, with all the power plant-associated concepts to be defined by the application as a part of the design project. During the LIT-3.1 project, the SML concept went through several rounds of complete revision. The SML concept has been described in more detail by Heinonen, 1985.

The definition of the SML concept was similar to an actual design project going through the following phases

- collection of the concepts to be used during the design process,
- description of collected concepts using a standardized set of metaconcepts,
- establishment of requirements on the number and record lengths of attributes of the concepts used,
- feeding the concepts to a common data base,
- description of the design in terms of the agreed concepts,
- feeding in the design to the design data base.

The dimensions of the design are in the SML concept simply described by the standard relations between the metaconcepts. This means that one could use an arbitrary rich division with respect to both dimensions of the design, the abstraction and the aggregation dimension. The earlier division of the dimension of abstraction into the why, what and how classes, and the dimension of

aggregation into the system, subsystem and component classes, are thus only to be seen as a matter of convenience not as a restriction.

The definition of similar concepts is defined with the typing mechanism of the SML, which gives a method for defining sub- and superconcepts, their names, attributes and relations. The relations between the concepts are implemented by a general connection mechanism which makes it possible to implement a rich structure of connections between the data base items. The association structure will allow the user to navigate in the data base along the relations. This means e.g. that the following associations can be made:

- systems to subsystems to components and back
- requirements to functions to components and back
- objects to attributes and to attribute types and back
- object type to other objects of that type
- object to object outputs and object inputs to other objects.

The default and inheritance mechanisms described earlier have not been very extensively included in the present version of SML.

6.3 Elements of SML

Considering the elements of the SML, one should separate between the concepts of the metalanguage and the concepts of the design data base. In selecting the elements of the metalanguage there are two conflicting requirements; on one hand it should be possible to describe the richness of

concepts of a real power plant, and on the other hand one should select the most simple structure possible. This means that one should consider the following questions:

- which types of building blocks are necessary? (activities, processes, procedures, events, devices, data),
- which is the level of detail that has to be included?
- which type of description language should be used for the implementation? (relational, procedural).

In the present form the SML gives a loose formalism, by which the design could be described in terms of structure, requirements and functions. The basic building blocks selected for the metalanguage are

- objects,
- connections,
- requirements.

An object is a metaconcept, which has a name, a type and a number of attributes. A connection is a metaconcept operating on the set of objects and it establishes a binary relation between its operands. A requirement is a special type of entity which can be attached to objects and connections.

To establish the concept of classes to which sets of objects belong, the following constructs are used

- object types,
- connection types.

A hierarchy of associations can then be defined as an object type being a subtype of an other object type (cf. Figure 6.1).

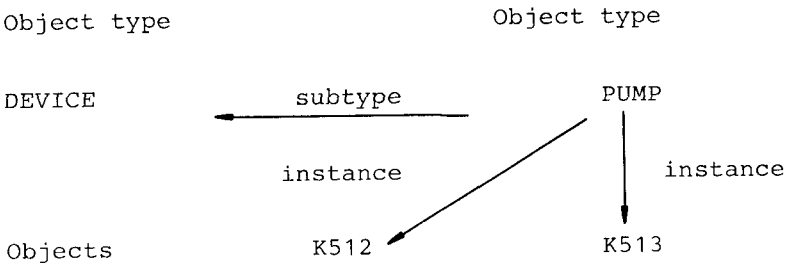


Figure 6.1 An example of objects and object types

The connection between the design data base and the design process is established with the concept procedure. A procedure is attached to an object or an attribute of an object to indicate that a design activity has to be carried out when the object or attribute is addressed. The concept of a procedure can then be used, e.g. to indicate that a consistency check or a verification procedure has to be carried out.

The metalanguage of the present SML can be described with an entity relationship model (cf. Figure 6.2), where an entity stands for an arbitrary concept of the metalanguage (indicated by arrows). The interfaces between the entities are conveyed with the following additional concepts:

- satisfies,
- attached,
- composition,
- reference.

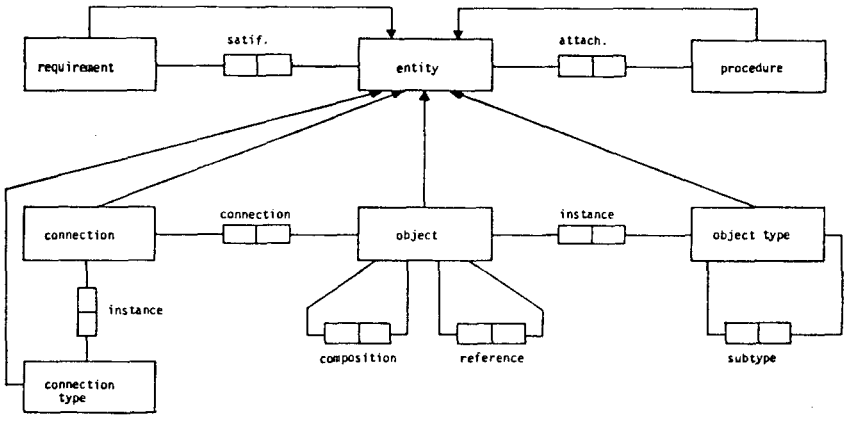


Figure 6.2 Entity-relationship model of the SML data base

6.4 Use of SML

Each of the SML entities uses a set of attributes depending on the specific entity. The following general types of attributes are used:

- name,
- purpose,
- parameter,
- ports,
- algorithm,
- status,
- verification.

The name is a unique alphanumerical string given to each entity and each object in the whole data base. The name serves as the basic search key for all data in the data base. In practical cases it is advantageous to construct a naming system, by which all objects are given names according to some general rules.

The purpose of the object is defined as an alphanumerical string giving a short explanation in natural language. The purpose may include references to entities and objects of the data base.

The parameters of the objects are numerical values and their corresponding units, which characterize the objects. One object may have several sets of parameters.

The ports are used to build a structure of connections between the objects in such a way that an object is defined to have a specified number of input ports and ouput ports. The ports are used, e.g. to describe the piping network of the plant.

Algorithms are used as a general term for computations which are made in some of the objects. Typical examples of algorithms are the automatic control algorithms and the logic for the plant protection. The details of the algorithm are described using a language that depends on the specific implementation.

A status is used to flag different conditions in the construction of the actual data base during the design. Typical examples are by whom and when the information for one object has been defined, updated and verified. The use of the status makes it possible to search for objects of the design that have not yet been finalized.

The verification attribute applies to the requirements which should have some procedure of verification. The verification could either be manual or automatic, and may be seen as an algorithmic rule which has to be satisfied for all the objects of a specific type in the data base.

6.5 Handling of requirements in SML

The requirements form the most important construct in the present SML demonstration. The systematic handling of requirements and the associated technical solution also provide an interesting application potential, both with respect to the control room operation and the problems of modifications in the design.

The requirements to be imposed on a design are usually documented in natural language descriptions of design rules, standards, guide-lines and accepted practices together with the design specifications. The formalization of the requirements will therefore be a considerable effort in itself.

Requirements can be classified with respect to the following general properties:

- structural requirements (dimensions, redundancies, material, etc.),
- functional requirements (capacity, operation, accuracy, etc.),
- positive and negative formulation (the object must have or must not have a property),
- static and dynamic dependence (the requirement is always applicable or only in a specified dynamic state),
- quantitative and qualitative requirements,
- requirements with a varying priority (e.g. demands, preferences and defaults).

The connections between the requirements and the technical solutions are built both top-down and bottom-up as the design process proceeds. In the top-down mode the designer defines deduced requirements from the given

terminal requirements and a network of requirements is gradually built. In the bottom up mode the designer gives the reasons for the technical solution chosen in terms of formal SML requirements. As a result of the design activity, all the requirements and the reasons should be formally defined and there should be connections linking them all together. The system of interconnected requirements makes it possible to verify the design in searching for initial and terminal requirements and the chains connecting them.

The demonstration data base was built using the technical specifications of the liquid waste system of a BWR plant. The technical specifications were written in natural language, and 16 formal requirements were identified and documented. The number of requirements for a real design project could easily be of the order of several hundred.

6.6 Connections between SML and the design process

The SML concept has been defined based on needs seen especially in control room design. The control room integrates all operational aspects of the plant, and that means that control room design has connections with all parts of the plant. The control room will also convey an operational model of the plant to the operator.

The plant as reflected in its documentation will form the most important part of the design data base. The control and instrumentation together with the power supplies for plant components is another large part of the design data base. The third part of the design data base is associated with the control room itself.

The process objects defined on a system and a subsystem level should include all the different components used in plants of different designs. This means that the following general types of objects should be defined:

- pumps,
- valves,
- pipes,
- heat exchangers,
- tanks,
- special components.

The control and instrumentation objects will be associated with the different systems and subsystems of the plant. The following general types of control and instrumentation objects should thus be defined:

- measuring points,
- control elements,
- cabling,
- power supplies,
- controllers,
- control sequences,
- protection logics.

The objects associated especially with the control room and the control room operators are the control room equipment, procedures, and personnel supporting the control room tasks. The control room equipment will contain the following types of objects:

- displays,
- controls,
- alarms,
- display pages,
- communication equipment,
- other types of control room equipment.

The separation of the specific and the general information is obtained by the separation of an object and an object type. This means, e.g. that a pump and a pump type differ in respect that the data for the pump will give its

specific name, its capacity, etc., and the pump type will have a name attribute, a parameter attribute giving its capacity, etc. Assigning specific names to object types makes it possible to execute searches for object types in the same way as for objects themselves.

6.7 Features of the SML concept

The SML concept provides a framework for the construction of a CAD system. The features of a CAD system built within the framework will thus depend on how the features of the SML concept are utilized. Some of the needs identified on a system modelling language have, however, been specifically addressed in the present SML concept.

Regardless of the need for a formalization of the design data base, it is clear that there have to be possibilities to use natural language descriptions. In the SML concept such descriptions are proposed for inclusion in the attributes, purpose and description.

Design verification is one of the most important tasks of the design process and can be done in two ways either operating directly on the data base or producing separate verification tools for the design data base. The checking that requirements are fulfilled is directly supported in SML using a list of requirements and a reference to requirements for each of the entities.

The decompositional structure of the SML concept makes it possible to proceed down to the descriptions at the implementational level. It is also possible to break the design up into sub-areas to be implemented on physically different CAD systems. Lower-level objects can be defined before higher-level objects are defined or vice versa, which makes it possible for the design to proceed either in a top-down or bottom-up fashion.

The SML concept also provides a more general framework for a knowledge-base of a plant. Such a knowledge-base can serve as a building platform for the construction of expert systems, which can be used to support the design and the operation of the plant.

The use of the concept of algorithms makes it possible to build automatic compilers and program generators for the control software. The structural information encoded in the map of connections between the objects of the data base will determine, e.g. the connection between simulation models. The algorithmic description of operational procedures may again be interpreted in terms of control action to be applied when the simulation model is used.

The design process can be guided by integrating check-lists and design guide-lines in the man-machine interf ce of the final CAD system. The SML concept also supports the definition and use of default values and macro-definitions in the design process.

6.8 Implementation of the SML concept

The implementation of the SML concept is based on a relational data base model. This means that a more complex structure than a simple record-oriented data base is required. In the demonstration system, the relational structure has been implemented by programming the standard mechanisms of the SML concept separately.

The SML concept has been implemented on a VAX11/750 computer within the VMS operating system. The programming language has been Pascal, and the file manipulation has been built using the RMS software. In writing the demonstration system, portability has been considered.

This means that it should be comparatively easy to transfer the software to any other computer having similar performance to the VAX11/750 computer.

The present SML implementation is composed of the following parts (cf. Figure 6.3):

- entity manipulation,
- application software,
- file handling interface.

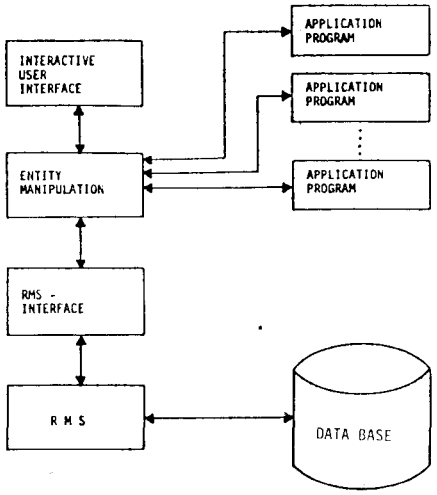


Figure 6.3 Implementation of the SML

The entity manipulation is directed to the metalanguage part of the system modelling language. The entity manipulation system is accessed from a user interface, which gives different users different possibilities to manipulate the design data base.

The application software is directed to different verification, construction and listing programs. In the present demonstration system, the application system is

almost non-existent. In real application, verification of the control room design may be carried out, e.g. using the following quality measures:

- availability of plant information as needed in operational procedures,
- distance between displays and controls as needed in plant transients,
- separation of alerting cues as determined by different plant transients,
- number of active cues in specific plant transients.

The file handling interface is computer-dependent and has to be rewritten if the present SML program is transferred to another computer. The file handling interface has been written to include the variable length data fields, which are used for natural language descriptions.

6.9 SML user interface

The user interface of a CAD system is very crucial when its efficiency is considered. The user interface must be interactive if the full benefit of the CAD system is to be utilized. It should also be a multi-user system, because several designers are supposed to work with the system concurrently.

In the demonstration system, a far more modest user interface has been written. The present version of the SML concept includes an interactive user interface consisting of a working area and a set of commands. The working area has buffers for each type of entitity (object, object type, requirement, procedure, connection, connection type). The following set of commands has been included:

- display help information,
- list commands,
- create an entity in the working area,
- clear an entity in the working area,
- delete an entity from the data base,
- modify an entity in the working area,
- list entities,
- store entity in the data base,
- find an entity in the data base,
- read an entity from the data base to the working area,
- read next/former entity from the data base to the working area,
- get entity (find and read),
- show the state of the user interface,
- set default parameters of the user interface,
- save and destroy an earlier saved state of the user interface,
- open a command file.

As the present SML version is intended for research and demonstration purposes it has been implemented only with a one user interface. The code has not been optimized for efficiency.

7. CONCLUSIONS

The benefit of computer aided design (CAD) systems in the design of control rooms is beyond doubt, and the question is only which features should be included in the CAD system and which not. The benefit of a computerized design data base may be so large that it could be feasible to transfer parts of the design data base to a computerized system as a retrofitting effort for old plants. For a new plant the benefit of the CAD system is seen in improved design quality and in increased designer productivity.

The construction of a CAD system should be based on a thorough analysis of information flow during the design process. The analysis should concentrate on when and by whom design information is used and produced. The CAD system should be constructed with a consideration of human errors in the design in order to support the designer in tasks where errors are likely to occur. The CAD system

should also pay attention to the organization of the design, but it is also important to note that the introduction of CAD is bound to change the requirements on the organization.

The most important feature of the CAD system is the management of design changes. By using computerized searches in the design data base, it is possible to trace the requirements and all the paths of influence that have to be considered before the a change can be implemented. Such a feature necessitates, in the case of control room design, management of a rich set of relations between the different data items in the design data base.

Features to be included in a CAD system to be used in a real project will always depend on the special needs in that project. This means that a CAD system will always need some tailoring before use in a specific project. Commercially available systems such as drafting packages, 3-dimensional modelling packages, and data base systems offer efficient buildings blocks in a CAD system to be used for control and instrumentation design.

The present design practices include a large number of different natural language descriptions. The introduction of CAD tools will imply that many of the descriptions have to be formalized to enable the design process to be broken down into a number of data processing activities. The requirements imposed on the design data base can then be assessed considering the data input and ouput of the data processing activities, the tools and resources that are required and how the activities are controlled.

Design can be seen as a modelling activity where the designer describes his vision in terms of a language which has the property of being able to be converted more or less automatically into the final construction. This view

suggests the use of a system modelling language as the basis for a CAD system. The vision of the designer is built into the system, but has also to be conveyed to the operator in his training. The computerized data base has also interesting applications in the presentation of information in the control room.

As a part of the LIT-3.1 project one approach for the system modelling language, the SML concept, has been constructed. Being a limited effort, the SML rather serves as a vehicle for the demonstration of a concept than a proposal for a CAD system. The demonstration has also served as a pilot for an assessment of the required resources and costs for a CAD system to be implemented for control room design.

On the basis of this limited experience it seems feasible to extend the demonstration to involve also direct efforts of industry. The preparation of such a follow-up project has been started. Experimentation with the system and projections of possible user dialogues give reason to believe that the proposed SML concept could serve as the basis for future CAD systems used for control room design.

The implementation of the ideas of the SML concept should proceed from an analysis of plant concepts, which are refined to form the metaconcepts to be used in the construction of the CAD system. The final reporting of the LIT 3.1 project is thus to be considered as giving a framework and a box of ideas for the construction of a CAD system for control room design. The realization of the CAD system will depend on when it is going to be used, because rapid technological development is introducing new computers and new software at a great rate. It is also very likely that technological development will change design project management considerably.

The design process itself involves many questions which could be interesting from the viewpoint of cognitive psychology. One of the questions is how the tools the designer has at his disposal will affect his understanding of the plant he is designing and how the tools will affect the likelihood of different design errors. Research in the design area seems well justified in terms of its importance and the scarcity of clear guidance on how design projects should be assembled.

8. REFERENCES

Barr, A., Cohen, P. R., Feigenbaum, E. A. (1981). The Handbook of Artificial Intelligence, vol I, II, III. Pitman, London 1981 and 1982.

Bishop, P. Ball, A., Barnes, M, Humphreys, G. Dahll, Lahti, J. Yosimura, S. (1985): Project on diverse Software - an experiment in software reliability, Safecomp '85 Italy.

Chalvy, L., Foisseau, J. Rosalie: (1983). A C.A.D. object-oriented and rule-based system. In proceedings of the IFIP 9th World Computer Congress, Paris, France, 1983. North Holland, Amsterdam. pp. 501-505.

Cotterman, W., et. al. (eds.) (1981). Systems Analysis and Design: Foundation for the 1980's. North Holland.

Danchak, M, M,. (1985): Quantitative methods for judging display design quality, IEEE third conference on human factors and power plants, Monterey, Ca., USA.

EPOS-80 - The development support system EPOS (1980). Institute for control engineering and process automation, University of Stuttgart. 44 pp.

EPRI (1984). Human factors guide for nuclear power plant control room development, Electric Power Research Institute, Palo Alto Ca, USA, EPRI NP-3659.

Haase, V. H., Koch, C. R. (eds.) (1982). A special issue on application oriented specifications. Computer, 15, nr 5, 10-59.

Hanes, L, O'Brien, J., Disalvo, R. (1982): Control room design; lessons from TMI; IEEE Spectrum, June, 1982.

Heinonen, R., Ranta, J., Wahlström, B. (eds.) (1983). Design methods and computer-aided design of process automation. Technical Research Centre of Finland, Symposium series 31/83.

Heinonen, R., Haarla, J. (1985): A data base for process automation design, 3rd IFAC/IfIP international symposium, CADCE '85 July 31-August 2, 1985 Lyngby, Denmark.

Heinonen, R., Ranta, J., Wahlström, B. (1984). A conceptual framework for the design of complex automation systems. 9th world congress of IFAC, 2-6 July 1984, Budapest, Hungary.

Heinonen, R. (1985). A high-level automation system design language. Technical Research Centre of Finland, Electrical Engineering Laboratory, Research Notes, Espoo 1985. Research Reports 518.

Hollnagel, E., Pedersen, O. M., Rasmussen, J. (1981): Notes on human performance analysis Risø National Laboratory, Denmark, Risø-M-2285.

Hollnagel, E. (1981): The methodology of man-machine systems; problems of verification and validation, Risø National Laboratory. Denmark, Risø-M-2313.

Hollnagel, E., Woods, D. (1982): Cognitive systems engineering; new wines in new bottles, Risø National Laboratory Denmark Risø-M-2330.

Jackson, M.A. (1982): System development, Prentice-Hall.

Lauber, R.J. (1983): Specification languages and computer aided development support techniques to achieve reliable and safe systems; present status and future directions, in Heinonen, Ranta, Wahlström (1983) pp. 13-30.

Lind, M. (1981). The use of flow models for automated plant diagnosis. In Rasmussen,J., Rouse,W.B. (eds.), Human detection and diagnosis of system failures. Plenum Press, New York.

Lind, M. (1983): A systems modelling framework for the design of integrated process control systems, Risø National Laboratory, Denmark, Risø-M-2409

Lind, M. (1984): Information interfaces for process plant diagnosis, Risø National Laboratory, Denmark, Risø-M-2417.

Lindqvist, J. Rydnert, B. Stene, B. (1984): Safety oriented organization and human reliability, First International symposium on human factors in organizational design and management 21.-24.8.1984 Honolulu.

Norros, L., Ranta, J. Wahlström, B. (1983): Assessment of control rooms of nuclear power plants, Technical Research Centre of Finland, Research Reports 184.

Ranta, J., Wahlström, B., Westesson, R. (1981): Guide-lines for man-machine interface design, Technical Research Centre of Finland, Research Reports 23/1981.

Ranta, J. (1983): Modelling and structuring of design process of process information systems, in Heinonen, Ranta, Wahlström (1983) pp. 47-85.

Ranta, J. (1985). Automation systems and control room design: problems and methods. Research reports 517.

Rasmussen, J., Lind, M. (1981). Coping with complexity. Risø National Laboratory, Denmark, Risø No. M-2293.

Rasmussen, J., Pedersen, O., Carnino, A., Griffon, M., Mancini, G., Garnolet, A; (1981). Classification system for reporting events involving human malfunction. Risø National Laboratory, Denmark Risø-M-2240.

Rasmussen, J., Pedersen, O. M. (1982): Formalized Search Strategies for human risk contributions: a framework for further development, Risø National Laboratory Denmark, Risø - M - 2351

Rasmussen, J. (1976): Outlines of a hybrid model of the process plant operator, in Sheridan, Johannsen (1976).

Rasmussen, J. (1979): On the structure of knowledge; a morphology of mental models in a man-machine context. Risø -M-2192.

Rasmussen, J. (1981): Human systems design criteria; state of the art and future perspectives, in Computers in Industry 2 (1981) pp. 297-309 North Holland.

Rasmussen, J. (1978): Notes on human error analysis and prediction, Risø National Laboratory, Denmark, Risø-M-2139.

Rasmussen, J. (1981): Human errors. A taxonomy for describing human malfunctious in industrial installations, Risø National Laboratory Denmark, Risø-M-2304.

Rubinstein, E., Mason, J. (1979): TMI; The accident that shouldn't have happened and a technical blow by blow, IEEE Spectrum Nov. 1979 pp. 32 - 42.

Sheridan, T. B., Johannsen, G. (1976) (eds.): Monotoring behavior and supervisory control, Plenum Press, 1979.

Suokas, J., Karvonen, I. (1985): A comparison of automatic fault-free construction with hazard and operability study, Technical Research Centre of Finland, Research Reports 330.

Tuominen, L., Wahlström, B., Timonen, J. (1978): A system and task description for the operating personnel of the Loviisa nuclear power station, Technical Research Centre of Finland VTT/SÄH Report 39, September 1978.

Wahlström, B., Rasmussen, J. (1983): Nordic co-operation in the field of human factors in nuclear power plants, in Nuclear Power Experience, IAEA Vienna, 1983, IAEA-CN-42/247, pp281-290.

Wahlström, B., Juusela, A., Ollus, M., Närväinen, P., Lehmus, I., Lönnqvist, P., (1983): A distributed control system and its application to a board mill, Automatica, vol 19 No 1 pp 1-14.

Wahlström, B., Heinonen, R.: Presentation of excerpts from the design data base for control room operators, Enlarged Halden Programme Group Meeting on Computerized Man-Machine Communication, Gothenburg 3.-7.6.1985.

The reports from the Risø National Laboratory may be ordered by mail from the address

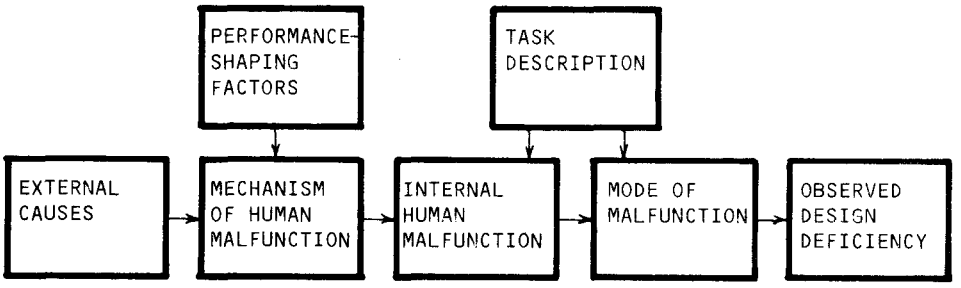
Risø National Laboratory
The Library
P.O. Box 49, DK-4000 Roskilde
Denmark

The reports from the Technical Research Centre of Finland may be ordered by mail from the address

Government Printing Centre
Marketing Department
P.O Box 516, SF-00101 Helsinki
Finland

A classification system for design errors

The classification system is based on the following causal explanation (c.f. Rasmussen et al. 1981).



The different boxes in the chain represent different facets of the design errors and can be correlated with each other for an investigation of the most important deficiencies in the design process. The following subdivision of the different boxes is suggested:

Observed design deficiency (ODD)

ODD 1 Design requirement not considered
- 1.1 Incompatible design
- 1.2 Ambiguities in design
- 1.3 Incompleteness in design

ODD 2 Unsatisfactory compromise between design requirements

ODD 3 Bug in design

Mode of malfunction (MOM)

MOM1	Specified task not performed	
	1.1	Omission of a part of the task
	1.2	Extra act done in task
	1.3	Unsatisfactory performance in task
	1.4	Delays in task
MOM2	Commission of erroneous act	
	2.1	Wrong subject selected
	2.2	Wrong act selected
MOM3	Sneak path, unlucky coincidence	

Task description (TD) described in task (T) and phase (P)

T	Task within design activity	
	1.	Setting goals and priorities
	2.	Interpreting requirements
	3.	Planning of activity
	4.	Selecting design alternatives
	5.	Managing changes in design
	6.	Documentation of design
	7.	Quality assurance
P	Phase within design project	
	1.	Preproject
	2.	Specification
	3.	Functional design
	4.	Detailed design and implementation
	5.	Construction
	6.	Test operation
	7.	Commercial operation

Internal human malfunction (IHM)

- IHM1 Need for decision not detected
  - 1.1 Need for decision not presented
  - 1.2 Need for decision not observed
  - 1.3 Signal rejected
- IHM2 Error in collection of information
  - 2.1 Information not asked for
  - 2.2 Information not checked
  - 2.3 Information not understood
- IHM3 Error in generation of design alternatives
  - 3.1 Alternative not observed
  - 3.2 Insufficient alternatives generated
  - 3.3 Alternative not considered
- IHM4 Error in evaluation of design alternatives
  - 4.1 Consequences of design decision not evaluated correctly
  - 4.2 Evaluation not carried out far enough
  - 4.3 Personal preferences dominating
- IHM5 Error in selection and carrying out alternative
  - 5.1 Wrong alternative selected
  - 5.2 Error in carrying out selected alternative

IHM6 Execution documentation and information dissemination

- 6.1 Delay in execution
- 6.2 Documentation not carried out
- 6.3 Insufficient documentation
- 6.4 Information not distributed

Mechanism of human malfunction (MHM)

MHM1 Discrimination

- 1.1 Stereotype fixation
- 1.2 Familiar short cut

MHM2 Observation

- 2.1 Information not observed
- 2.2 Misinterpretation
- 2.3 Pieces of information not combined

MHM3 Recall

- 3.1 Forget isolated part of task
- 3.2 Mistake between alternatives
- 3.3 Agreed solution forgotten
- 3.4 Other slip of memory

MHM4 Interference

- 4.1 Side-effects not considered
- 4.2 Intermixing of two or more tasks

MHM5 Physical coordination

- 5.1 Motorskill variability
- 5.2 Spatial misorientation

Performance shaping factors (PSF)

- PSF1 Stressing environment
  - 1.1 Time pressure in project
  - 1.2 Cost pressure in project
  - 1.3 Personal disagreement between project members
  - 1.4 Other kinds of stress
- PSF2 Organizational atmosphere
  - 2.1 Management style
  - 2.2 Goal identification
  - 2.3 Style of communication
  - 2.4 Performance feed back
- PSF3 Task characteristics
  - 3.1 Structured vs. unstructured
  - 3.2 Content of challenge
- PSF4 Inadequate tools
  - 4.1 Information system
  - 4.2 Communication system
  - 4.3 Design tools
- PSF5 Inadequate training
  - 5.1 Lack of understanding
  - 5.2 Lack of experience

External causes (EC)

- EC1 Excessive task demand with respect to
  - 1.1 Knowledge needed
  - 1.2 Time for execution
  - 1.3 Memory required

EC2 External events

2.1 Distraction

2.2 Change in project plans

EC3 Designer incapacity

3.1 Sickness

3.2 Fatigue

3.3 Lack of motivation

EC4 Latent system conditions

4.1 Errors in design data base

4.2 Lates changes making design obsolete

EC5 Inherent human variability

EXAMPLES OF THE USE OF THE SML SYSTEM

The comments are given in boxes lined by '*'.

1 INTERACTIVE USER INTERFACE

```
**************************************************************
*                                                            *
* In the following "$" is the prompt of the VAX11/750 operating *
* system and "." is the prompt of the SML system.            *
*                                                            *
**************************************************************
```

```
$ run sml

This is System modelling language (SML) version
number 1.1 interactive user interface, welcome.
The language has been developed at the Technical Research
Centre of Finland, Electrical Engineering Laboratory.

With this program you can create, delete and modify
objects, object-types, connections etc. The program
is informative and it's going to ask you what you
want to do during the execution of this program.
If you answer with a '?', you'll get some more
information about the possibilities that you
can choose from.

The current time is 30-NOV-1984  15:30

First, give the name of the database you want to
handle (max. 6 characters).
THE NAME OF THE DATABASE ?  demo
GIVE YOUR INITIALS:  jh
```

```
**********************************************************************
*                                                                    *
* If the user types '?' more information on the use of the           *
* SML system is shown on the screen.                                 *
*                                                                    *
**********************************************************************

.  ?

With this program you can add, delete, list and modify:
                        -object-types
                        -objects
                        -requirements
                        -connection-types
                        -connections
                        -procedures

which exist each in its own file in the same database.
Furthermore you can 'navigate' in the database and then
you can do different kinds of pick ups and listings. If
you want more information about an instruction, you can
type '?<instruction>' (for example '?delete'), and then
the program will give you information about the specific
instruction. By typing two question-marks ('??') you'll
get more information about all the instructions available.
If you type 'bye', the program execution will end.

**********************************************************************
*                                                                    *
* List of the commands of the user interface are shown if            *
* '??' is typed on the terminal.                                     *
*                                                                    *
**********************************************************************

.  ??

The possible instructions are:
     ?
        (for help information)
     ??
        (for listing the instructions)
     CREATE ENTITY
        (creates a new entity to the working area)
     CREATE-KEY ENTITY
        (creates a new key value for an entity)
     CLEAR ENTITY
        (clears the entity from the working area)
     CLEAR STATE
        (clears the whole user state)
     CLEAR-KEY ENTITY
        (clears the key value of an entity)
     DELETE ENTITY ÄPARAMETER1Å
```

```
   (deletes the entity from database)
MODIFY ENTITY
   (modifies the entity that exists
    in the working area)
LIST ENTITY ÄLISTING-DEVICEÅ ÄLISTING-TYPEÅ
   (lists the entity or entities the
    user wants listed)
TYPE ENTITY
   (lists the entity that exists in
    the working area to the terminal)
STORE ENTITY
   (stores the entity in the database)
FIND ENTITY
   (finds the entity from the database)
READ ENTITY
   (reads the entity from the database)
READ-NEXT ENTITY
   (reads the next entity from the
    database)
READ-PRIOR ENTITY
   (reads the prior entity from the
    database)
GET ENTITY
   (the find and read operations together)
GET-TYPE ENTITY
   (reads the type of the entity from
    the database)
GET-OWNER ENTITY
   (reads the owner of an object
    from the database)
RETURN-ORIGINAL ENTITY
   (reads the original object from the
    database, can be used after
    the get-owner statement)
LIST-OWNERS ENTITY
   (lists all the owners of one object)
GET-COMPONENT ENTITY
   (reads the component of an object
    from the database)
GET-ORIGINAL ENTITY
   (reads the original object
    from the database, can be used afte
    the get-component statement)
LIST-COMPONENTS ENTITY
   (lists all the components of one object
LIST-CONNECTIONS ENTITY
   (lists all the connections of one object)
SHOW STATE
   (lists some state information on the
    terminal)
SHOW ENTITY
   (shows all the entity names of an entity-
```

```
        type that exist in the database)
     SHOW-KEY ENTITY
        (shows the key value of an entity)
     SET-DEFAULT PARAMETER
        (sets a default value on the parameter)
     SHOW-DEFAULT PARAMETER
        (shows the default value of the parameter)
     SAVE STATE,UNSAVE STATE
        (saves/unsaves the user state
         to/from a temporary file)
     E<COMMAND-FILE NAME>
        (see text below)
     LOG-ON
        (puts the log keeping on)
     LOG-OFF
        (puts the log keeping off)
     BYE
        (terminates the program execution)

An entity corresponds to one of the following: object-type,
object, requirement, connection-type, connection,
procedure.
A parameter corresponds to one of the following:
entity-name, listing-device

If you write a '?'-mark before the instruction you'll
get some more information about the specific instruction.
About the command-files you'll get more information by
typing '?E'.

You can also use shortened forms of the instructions and
the entities, but be sure to make them unique
(for example st o-t is unique but s o-t is not).

The instructions can be written in either capital letters or
small letters.

*****************************************************************
*                                                               *
* By typing 'show state' the user gets                          *
* the current state of the user interface                       *
*                                                               *
*****************************************************************

.show state

                  DATABASE NAME:  demo
               THE CURRENT TIME:  30-NOV-1984
                       THE USER:  jh
       DEFAULT VALUE FOR ENTITY:  object
DEFAULT VALUE FOR LISTING DEVICE:  terminal
```

```
                KEY NAME:    CONTENTS OF BUFFER:  FILE POINTER:

OBJECT-TYPE: valve-display   valve-display    valve-display

OBJECT:      v127-display    v127-display     v127-display

REQUIREMENT: r018            r018             r018

PROCEDURE:   p001

CONN. -TYPE: redundancy

CONNECTION:  redu-connection1



2  EXAMPLES OF ENTITIES


*********************************************************************
*                                                                   *
* The definition of a quite general object type                     *
* with which all kinds of subprocesses can be defined.              *
*                                                                   *
*********************************************************************


subprocess  OBJECT-TYPE
  PURPOSE
    Subprocess definition

  AUTHOR:  jh  26-NOV-1984
END  subprocess

*********************************************************************
*                                                                   *
* A certain object 's342' which has many requirements               *
* is of type subprocess.                                            *
* The requirements are numbered r001,... .  If a requirement        *
* is defined elsewhere, only the number must be given when the      *
* subprocess is defined.  The text is then taken automatically      *
* from the requirement file.  Otherwise the text of the             *
* requirement can be given here.                                    *
*                                                                   *
*********************************************************************


s342  OBJECT OF TYPE  subprocess
  PURPOSE
    System 342: Liquid waste system.
  REQUIREMENT
```

```
    r001:
      System 342 must serve F3.
    r002:
      Earthquakes need not be taken into account.
    r003:
      The reuse of water must be maximized.
    r004:
      There must be so much redundancy in components and
      flow routes that there is 24 hours time for
      repair.
    r005:
      If a passive component or an active large component
      is damaged, the functions must be restored in
      one week.
    r007:
      Electric supply is taken from the 10kV net of F3.
    r008:
      Redundant components are supplied from different
      electric nets (C or D).
    r013:
      Operators work 40 hrs/week, at other times the system
      342 is controlled from the control room of F3.
    r014:
      System 342 must have so much automation that the
      shifts of F3 are not disturbed too much.
    r015:
      There is only local control if the function is
      needed only a few times a week.
    r016:
      There must be displays in the control room if
      continuous supervision is needed
      (measurements, alarms,...).

  AUTHOR:  jh  26-NOV-1984
END  s342

*********************************************************************
*                                                                   *
* The requirement r008 which the previously defined subprocess      *
* 's342' must satisfy is presented here in more detail.             *
*                                                                   *
*********************************************************************


r008 REQUIREMENT
  TEXT
    Redundant components are supplied from different
    electric nets (C or D).
  PARAMETER
    type                 :  mandatory
    verifying_method     :  automatic
```

```
    status                : verified
  SATISFIED BY
    s342

  AUTHOR:  jh  26-NOV-1984
END     r008


*********************************************************************
*                                                                   *
* a-control is an object type with which analog controllers and     *
* measurements can be defined.                                      *
* m211 is a pressure difference measurement.                        *
*                                                                   *
*********************************************************************


m211  OBJECT OF TYPE  a-control
  PURPOSE
    Pressure difference in filter f11.
  PARAMETER
    name                : m211
    fcode               : PdA

  AUTHOR:  jh  26-NOV-1984
END  m211


*********************************************************************
*                                                                   *
* p11 is an object of type pump.  It has several parameters and     *
* it has also two ports by which it can be connected,               *
* for example, to some other parts of the plant description         *
*                                                                   *
*********************************************************************


p11  OBJECT OF TYPE  pump
  PURPOSE
    Pumps water from the tank t11 through the
    clearings chain.
  PARAMETER
    name                : p11
    type                : centrifugal
    capacity            : 20kg/s
    location            :
    power               : 27kW
    calculationdata     : 1.0MPa/60C
  PORT
    port1               :
    port2               :
  ALGORITHM
```

```
    Continuous running. No run-time measurement.

  AUTHOR:  jh  26-NOV-1984
END  p11

**********************************************************************
*                                                                    *
* Tank t11 has two requirements which it must satisfy.               *
*                                                                    *
**********************************************************************

t11  OBJECT OF TYPE  tank
  PURPOSE
    Collects the system drainage
  REQUIREMENT
    r025:
      The collection tanks of subsystems 1, 2 and 3 are
      emptied through the lowest points. However,
      there must also be an outlet at a height of 0.5 m
      from the bottom.
    r026:
      The outlet of the collection tanks is selected
      manually (lowest or 0.5m point).
  PARAMETER
    name                 :  t11
    type                 :  cylinder
    volume               :  225m3
    location             :
    calculationdata      :  water filling/60C

  AUTHOR:  jh  26-NOV-1984
END  t11


3  NAVIGATION IN THE SML DATABASE


**********************************************************************
*                                                                    *
* The keys can be specified by giving a key name, a range of key     *
* names or '*' which means that all the entities of corresponding    *
* type are searched for.                                             *
*                                                                    *
**********************************************************************


.  create-key object
THE OBJECT KEY NAME ?  v113
.  get object
.  list object
```

```
v113  OBJECT OF TYPE  valve
  PURPOSE
    The filter f11 outlet valve.
  PARAMETER
    name                :  v113
    type                :  valve
    location            :

  AUTHOR:  jh  26-NOV-1984
END  v113


**********************************************************************
*                                                                    *
* The owners of an object are the objects that define the object     *
* as their component.  The object can have several owners.           *
*                                                                    *
**********************************************************************

.  list-owners object
THE OWNERS OF THE OBJECT v113 :
    f11-connection : filter-connection
    f11-function : filter-function
    parallel-connection : filter-connection
    serial-connection : filter-connection
    v113-display : valve-display


**********************************************************************
*                                                                    *
* The objects can also be used to represent also quite abstract      *
* features of the plant description.  f11-connection represents      *
* a certain plant state.                                             *
*                                                                    *
**********************************************************************

.  get-owner object
.  list object

f11-connection  OBJECT OF TYPE  filter-connection
  PURPOSE
    This object shows which valves must be open
    when only the f11 filter is operating.
    (The filter f11 is redundant with the filter
    f13 so it is possible for only one of
    them to be operating at a time.)
  PARAMETER
    name                :  f11-connection
    type                :  filter-connection
    location            :
  COMPONENTS
```

```
    valve1= v111
    valve2= v112
    valve3= v113

  AUTHOR:  jh  28-NOV-1984
END  f11-connection


- - - - - - -


.  list object

v113-display  OBJECT OF TYPE  valve-display
  PURPOSE
    Shows the state and location of the valve v113.
  PARAMETER
    name                :  v113
    type                :  x-valve
    location            :  y
    status              :  y
  COMPONENTS
    signal= v113

  AUTHOR:  jh  26-NOV-1984
END  v113-display
.  list-components object
THE COMPONENTS OF THE OBJECT v113-display :
    v113 : valve
.  list-owners object
THE OWNERS OF THE OBJECT v113-display :
    f11-display : PI-display
.  get-owner object
.  list object

**********************************************************************
*                                                                    *
* f11-display shows many plant components and uses several other     *
* subdisplays.                                                       *
*                                                                    *
**********************************************************************

f11-display  OBJECT OF TYPE  PI-display
  PURPOSE
    The display shows the pressure difference in the
    filter f11.
    Also the flow after the filter and the conductivity
    before it are shown.
    The display also shows if the valves that control
    backspooling flow to the filter f11 are open or not.
  PARAMETER
    ident               :
```

```
    name                : f11-display
    type                : PI-display
  COMPONENTS
    component1= m211
    component2= m314
    component3= m711
    component4= v113-display
    component5= v112-display
    component6= v116-display
    component7= v114-display
    component8= v115-display

  AUTHOR:  jh  26-NOV-1984
END  f11-display

*************************************************************************
*                                                                       *
* The components of a display can be read into the user                 *
* interface buffer.                                                     *
*                                                                       *
*************************************************************************

.  get-component object
.  get-component object next
.  get-component object next
.  get-component object next
.  get-component object next
.  list object

v112-display  OBJECT OF TYPE  valve-display
  PURPOSE
    Shows the state and situation of the valve v112.
  PARAMETER
    name                : v112
    type                : valve
    location            : y
    status              : y
  COMPONENTS
    signal= v112

  AUTHOR:  jh  26-NOV-1984
END  v112-display

.create-key object
THE OBJECT KEY NAME ?  p11
.  get object
.  list object

p11  OBJECT OF TYPE  pump
  PURPOSE
    Pumps water from the tank t11 through the
    clearings chain.
```

```
  PARAMETER
    name                :  p11
    type                :  centrifugal
    capacity            :  20kg/s
    location            :
    power               :  27kW
    calculationdata     :  1.0MPa/60C
  PORT
    port1               :
    port2               :
  ALGORITHM
    Continuous running. No runtime measurement.

  AUTHOR:  jh  26-NOV-1984
END  p11


**********************************************************************
*                                                                    *
* The entity connection can be used in many ways.  In principle it   *
* is only a binary relation.  Here a connection type 'redundancy'    *
* is used to connect two redundant pumps.                            *
*                                                                    *
**********************************************************************

.  list-connections object
    THE CONNECTIONS OF THE OBJECT  p11 :

 FROM: port1                    TO: p13.port1 TYPE OF pump
 CONNECTION_NAME: redu-connection1
 CONNECTION_TYPE: redundancy

 NO CONNECTIONS DEFINED TO PORT  port2


4  LISTING FORMS


**********************************************************************
*                                                                    *
* The entities can be listed in many ways.  Parts of the entity      *
* descriptions can be left off.                                      *
*                                                                    *
**********************************************************************

.  list object author

p13  OBJECT OF TYPE  pump
  PURPOSE
    jh  26-NOV-1984
  PARAMETER
```

```
    name                 :  p13           jh  26-NOV-1984
    type                 :  centrifugal   jh  26-NOV-1984
    capacity             :  10kg/s        jh  26-NOV-1984
    location             :                jh  26-NOV-1984
    power                :  1.5kW         jh  26-NOV-1984
    calculationdata      :  1.0MPa/60C    jh  26-NOV-1984
  PORT
    port1                :                jh  26-NOV-1984
    port2                :                jh  26-NOV-1984
  ALGORITHM
    jh  26-NOV-1984

  AUTHOR:  jh  26-NOV-1984
END  p13

.list requirement author


r016 REQUIREMENT
  TEXT
    There must be displays in the control room if
    continuous supervision is needed
    (measurements, alarms,...).
  PARAMETER
    type                 :  mandatory      jh  26-NOV-1984
    verifying_method     :  automatic      jh  26-NOV-1984
    status               :  verified       jh  26-NOV-1984
  SATISFIED BY
    s342                  jh  26-NOV-1984

  AUTHOR:  jh  26-NOV-1984
END     r016
```

LIT final reports:

LIT(85)1	The human component in the safety of complex systems.
LIT(85)2	Human errors in test and maintenance of nuclear power plants - Nordic project work.
LIT(85)3	Organization for safety.
LIT(85)4	The design process and the use of computerized tools in control room design.
LIT(85)5	Computer aided operation of complex systems.
LIT(85)6	Training in diagnostic skills for nuclear power plants.

These reports are available at the following organizations:

Technical Research Center of Finland, VTT/INF Vuorimiehentie 5 SF-02150 Espoo 15	LIT(85)1 & 4
Studsvik Energiteknik AB S-611 82 Nyköping	LIT(85)2
Statens Vattenfallsverk Fack S-162 87 Vällingby	LIT(85)3
Risø National Laboratory Postbox 49 DK-4000 Roskilde	LIT(85)5 & 6

Handling charge USD 10,- per report to be forwarded with order.

Norstedts Tryckeri, Stockholm 1985