Design Approaches: A Comparative Study of Information System Design and Architectural Design

A. OLERUP
Lund University, Dept of Information and Computer Science, Solvegatan 14a, S-223 62 Lund, Sweden

In this paper two approaches to design are compared, namely those proposed by Langefors and Alexander. The Langeforsian approach aims to introduce effective methods for information system design and to utilise the potential of computers and information technology in controlling organisations and improving their performance. The aim of Alexander's approach is to improve architectural design by bridging the gap between designer and users. Both approaches have given important contributions to design in their fields. In both approaches graphs and matrices are important tools. Also, procedures for finding subsystems and components are put forward in both approaches, and the process of design is subdivided. The two approaches are different with regard to requirements definitions, in the Langeforsian approach goals are crucial, while misfits are focussed in Alexander's approach. The notions of goals and misfits are examined. Finally, systems-fit approaches to information system design are discussed.

Recieved June 1989, revised November 1989

1. INTRODUCTION

Design and design products are ubiquitous. In addition processes of design receive increased attention. Resources and work are devoted towards improving planned design process. Methods and techniques for design are introduced in many design fields, into a young field such as information system design as well as into an established field such as architecture.

The interest in processes and methods for information system design (ISD) has been stimulated both by developments in information and computer technology as well as by the increasing importance of formal information systems in organisations, which are created through planned and conscious design efforts. The scope of design has expanded, including not only technical issues but also matters such as modelling the user system and involving users in design. A broad range of methods and techniques for designing information systems and managing the ISD process has been introduced. The aim is to improve ISD and to increase the quality and usefulness of information systems.

Similarly, considerable efforts have been made to introduce formal methods into architectural design in order to assist in clarifying the design problem and bridge the widening gap between designers and users.

In this paper I wish to compare the Langeforsian approach[17] for information system design and Alexander's[2] approach for architectural design. Both approaches have had considerable impacts and have given decisive contributions, though of very different kinds, to the study and practice in design. The Langeforsian approach proposes a framework for information systems and ISD, and it has generated a large number of detailed design models. The contribution of Alexander is quite different, since it generated an extensive debate questioning the need, usefulness and role of formal methods in architectural design.

In the second half of this paper, I intend to discuss some alternative presuppositions for information system design, drawing on the comparison between the Langeforsian approach and Alexander's approach. In general, existing approaches for ISD are based on models of goal-seeking behaviour. Alexander's approach suggests another possibility.

2. THE LANGEFORSIAN APPROACH TO INFORMATION SYSTEM DESIGN

The intentions of the Langeforsian approach[16, 17] are to introduce effective methods for ISD, and to utilise the potential of computers and information technology in controlling organisations and improving their performance. Early efforts to design information systems (for business use) often used unsystematic and random methods, and in general they often meant mechanising existing procedures and routines. Large amounts of resources were consumed in designing an information system, yet the information system failed to provide the expected benefits to the organisation. Recognising this Langefors[17] argued for using formal, i.e. mathematical, methods for ISD. In order to provide a systematic, scientific and solid base for ISD, the foundations for a theory of information systems were put forward.

A theory of information systems needs to consider the role of information and information systems in organisations. In the theory proposed by Langefors[17] it is a fundamental assumption or choice that organisations can be considered as simple cybernetic systems (of the first order) similar to mechanical systems and machines. Accordingly an information system is viewed as an instrument for controlling and managing another system (referred to as an operating system) or activity, with the aim of achieving optimal utilisation of resources, and maximal economic returns for the organisation as a whole. ISD then refers to efforts to design a rational and effective information system for controlling and managing an organisation towards optimal results.

It is a basic idea in the theory of information systems[17] that organisations can be described and controlled as technical systems, and therefore they require technical, or instrumental, rationality. This means that there are no differences between controlling a machine or controlling human activities. The fact that some components in the latter case are human beings is not considered to be of any consequence.

On this basis an abstract model is developed outlining how a computer-based information system can be used for controlling an operating system.[17] The assumptions are: first, it must be possible to define goals, precisely and operationally, for the operation of the system; further, the system must be controllable, and able to react in predictable and deterministic ways; and finally, it must be possible, and feasible, to formalise controlling a system.

Langefors[19] recognised that it is never possible to satisfy these assumptions completely. Still he believed that in most cases it will be possible to obtain a good enough approximation to these assumptions, and therefore it will be possible to design information systems according to the principles of the theory of information systems. In addition he believed that further research on formulating goals and on human behaviour will contribute towards removing the remaining obstacles on building 'rational' systems.

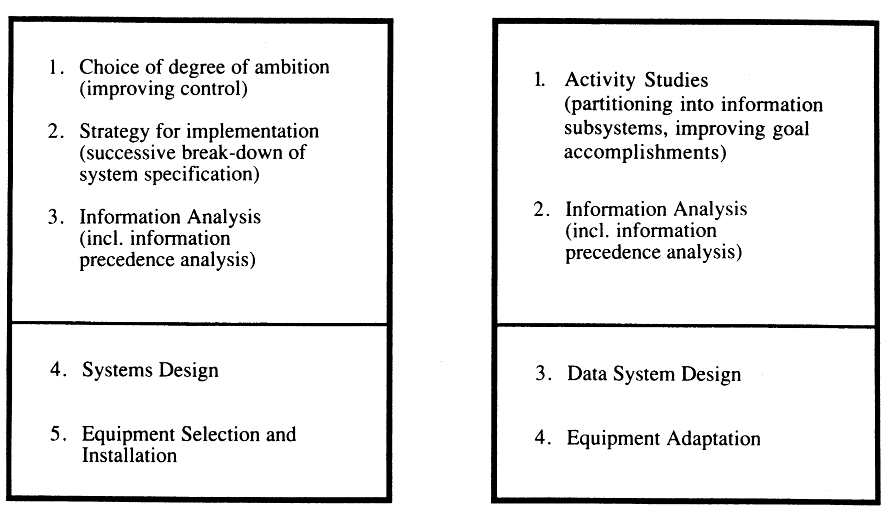
Designing information systems does not only require a theory of how information systems can be used to control and manage an operating system, it also requires a methodology or a strategy for designing an information system. This is often taken to imply a subdivision of the design-work or design process. A simple differentiation can be made into *what* an information system must do and *how* this can be accomplished and realised.[17] Langefors[18] proposed that information system work be divided systematically into a limited number of method categories. Each category includes distinct tasks requiring different skills and therefore making it possible to obtain economics of specialisation.[18]

As previously noted the intentions of the Langeforsian approach[17] are to improve design methods and so improve management control. The aim is to design technically rational and effective information systems for management control and optimal organisational performance. It is proposed to accomplish this by using formal, mathematical models and methods, including graphs. The approach presupposes that goals are fundamental for ISD problems, since management control must be oriented towards organisational goals.[19] Furthermore, operating activities in an organisation are identified by functional subdivision, and each function assigned a goal, and finally goals form a hierarchy.

The core of the Langeforsian approach[17] is a formal description and representation of the detailed design problem using directed graphs or corresponding matrices for analysing and depicting the structure of information-sets and relationships between information sets (i.e. the information requirements an information system must satisfy). The procedure concentrates on information sets (what), ignoring processing (how), it starts from a number of crude information sets, needed for controlling an operating activity towards some goal. The analysis continues until only elementary information sets are identified. Directed graphs mapping the connections between information sets and graphs depicting the structure of information sets are constructed. The digraphs, and corresponding matrices, are next used to find combinations of information sets which are efficient from a processing and storing point of view.

The work of Langefors[17,19] has had decisive impact on ISD by consistently insisting on theoretical work as a fundamental background for ISD, thus establishing ISD as a separate and important area within information and computer science. The theory of information systems has also provided the basis for the ISAC methodology.[26] Langefors[18] suggests a simple differentiation of ISD into five method areas, which are systematic but not chronologic. The division into four method areas in the ISAC methodology[26] obviously builds on this (Fig. 1). Furthermore, the method of documenting information precedence analysis employed in the ISAC methodology was originally proposed[25] as a way of documenting precedence analysis according to Langefors.[17] The ISAC methodology is unique among ISD methodologies since it is not only based on an explicit theoretical framework, but it has also been used in many practical applications.

The major shortcomings of the Langeforsian approach can be attributed to the choice of mechanical system models for organisations. As a consequence human beings and machines are considered to be equivalent components, and technical rationality is emphasised. This was, however, far from unique during the 1960s, which was characterised by a belief in the advantages of technical progress. It was firmly believed that progress in computer technology would make it possible to manage and control organisations rationally and scientifically.



Langeforsian approach ISAC methodology

Figure 1. Method areas in analysis and design of information systems, according to the Langeforsian approach[18] and the ISAC methodology.[26]

So, it was quite common to view organisations as mechanical systems. Such a rational and mechanistic approach to organisations has many points in common with scientific management and classical organisational theory, which have both been thoroughly criticised.

The major impacts and greatest merits of Langefors's work[17,19] are due to both theoretical and practical factors. First, it is based on an explicit statement and examination of a number of fundamental and explicitly chosen assumptions involved in research within the framework of a theory of information systems. Second, it has been practically implemented as an ISD methodology, the ISAC methodology.[26]

3. ALEXANDER'S APPROACH TO ARCHITECTURAL DESIGN

The primary rationale behind Alexander's[2] approach is the increasing distance between designer and user in architectural design. The user and the designer are rarely the same person. Instead they are roles assumed by separate individuals. Alexander suggests bridging this gap by using formal models, diagrams or graphs; such representations make mental pictures communicable, discussable and open to criticism. The intention of the approach is to overcome the fundamental difficulty in design work: predicting and perceiving the pattern of relationships which the use of a new artifact brings into existence.

The approach[2] proposes a way of stating design problems which is based on the idea that every design problem begins with an effort to achieve fit between two entities: the form in question and its context. The form is the solution to the problem, while the context defines the problem (i.e. the requirements to be met). Typically, a design problem has requirements which need to be met, and there are interactions between the requirements. A formal statement of the design problem refers to a set M (or misfits): 'The problem is defined by a set of requirements called M. The solution to this problem will be a form which successfully satisfies all of these requirements' (ref. 2, p. 93).

The core of Alexander's approach is a formal description and representation of the detailed design problem in terms of non-directed graphs and corresponding matrices. In order to arrive at a graph (and a matrix), the approach calls for first identifying all the requirements of a particular design problem. The procedure starts from a number of broad classes of requirements. By successive division and partition a tree is derived, uncovering in detail the sets of requirements. The requirements at the most detailed level comprise a full specification of the design problem. The list of requirements needs to be as exhaustive, comprehensive and complete as possible; further the requirements must be of equal scope and of great detail.[2]

In a well-known example of an Indian village, Alexander[2] identifies all the misfits in the life and work of the village, i.e. all potential problems and weaknesses imaginable in farming, cattle-raising, housing, water supply, etc. A total of 141 misfits are identified. The second phase involves looking for interactions between the listed requirements. Next, the interactions between each pair of requirements are labelled as positive, negative or neutral dependent on the impact of requirements on each other. This labelling is necessary to be able to find strongly connected components of requirements, thus identifying subproblems of the original design problem.

When the graph of requirements and interactions has been constructed, it is used to find suitable components, each of which will include requirements from several classes. The procedure involves successively and from the bottom bringing together requirements into components, and these into larger components. Each component is made up of requirements richly or strongly connected, while connections with other components are weak. In this way the design problem is defined in terms of a number of subproblems which can be tackled independently. The sheer size of the graph, and matrix, makes processing by computer necessary.

In the example of the Indian village interactions amongst misfits are identified and a matrix constructed. Next components are formed from requirements which are closely interrelated, components may then be handled as separate design problems dealing with various facets of village life.

The outcome is a tree of the detailed bottom-up composition of the design problem, conversely the tree also presents top-down decomposition of the design problem. Thus, based on a graph of requirements and interactions, a tree showing the decomposition of the design problem is derived (Fig. 2).

It is important to note that different types of diagrams are used in Alexander's approach. There are graphs which represent the requirements and their interactions. There are trees describing the subdivision of requirement classes into detailed requirements, which are worked out before the graphs are constructed. Trees are derived from graphs to describe the clustering of requirements into components, i.e. the composition of the design problem. The second type of tree does not show how requirements interact with each other, but instead how requirements are clustered into and belong to components or subproblems. (A graph shows that two elements A and B

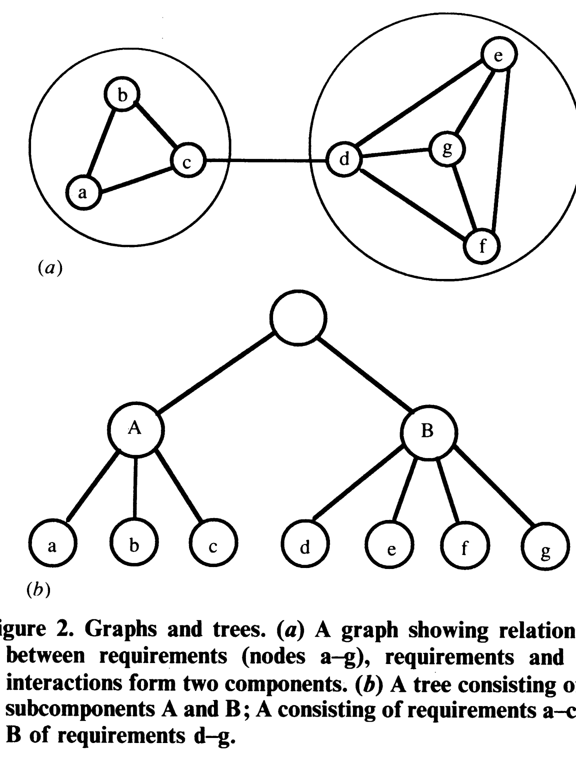


Figure 2. Graphs and trees. (*a*) A graph showing relationships between requirements (nodes a–g), requirements and their interactions form two components. (*b*) A tree consisting of two subcomponents A and B; A consisting of requirements a–c, and B of requirements d–g.

interact, while a tree shows that the elements belong to some component alpha.)

The work of Alexander[2] has had an extraordinarily lasting impact on thinking about design method, not only in architecture, since it introduced some new concepts and ideas. Apart from some illustrations of the approach[2, 8] there is only one published documentation of an attempt to use the method, and this did not result in any noticeable practical success.[20] Criticisms have brought attention to a number of dubious assumptions and failings in Alexander's approach. These are summarised below.

Some of the shortcomings of Alexander's approach can be attributed to a view of design problems as having a mechanistic character and requiring technical rationality; this is obvious from the formal definition of the design problem (cf. the previous quotation).

First, the approach presupposes that it is possible to make an exhaustive and complete list of requirements (a complete specification) at the start of the design process; however requirements are likely to be discovered, asserted or changed at any point during the process, as Lawson[20] and many others have noted. Further, it is assumed that a designer's personal observations and thoughts are likely to provide an adequate means of representing all of the things that affect, or are affected by, a new design.[15]

Second, it is assumed that the listed requirements are of equal value and that the interactions between them are all equally strong.[20] It is, however, likely that some requirements are more important than others. All requirements need not be considered at the same time; some may advantageously be postponed until later stages of the process.

Third, the approach fails to appreciate that the implications for the form of the solution are much more profound from some requirements and interactions than from others.[20] This implies an assumption that an interaction is independent of the solution that is eventually arrived at, and that additions and changes made in the future will not themselves alter the pattern of interacting requirements and the decomposition of the pattern.[15]

Alexander's description of the history of design is, however, acknowledged.[20] Briefly, Alexander[2] describes design as historically having moved through two phases and into a third one, at the time he wrote. In the first phase, the *unselfconscious culture*, form-making and using are closely integrated and cannot be separated; the problem remains static and stable; novices learn from gradual exposure the craft in question, by imitating and correcting mistakes. There is a lack of understanding of theoretical background. In the second phase, the *selfconscious culture*, form-making and using have become distinct; the problem may suddenly change; the novice learns on the basis of general principles. In the unselfconscious culture, form is shaped by the interaction between the actual context's demands and the actual inadequacies of the form, while in the selfconscious culture, form is shaped by a conceptual interaction between contextual pictures and form-diagrams, by using fairly concrete pictures, diagrams, and drawings. In the third phase, Alexander[2] argues, the selfconscious culture can be improved by taking the use of pictures a further step from concrete drawings towards abstract diagrams, which retain only the abstract structural features of the form. The approach proposed by Alexander[2] belongs to the third place.

The major contributions and greatest impacts of Alexander's work, despite the aims stated in his book[2] come neither from the possible practical usefulness nor from the automatic procedure for decomposing a pattern of requirements. The importance of Alexander's work lies in the debate he caused by proposing a view of design which was an antithesis to the prevailing one at that time. Alexander[2] argued strongly that design (process as well as product) can be improved by introducing formal methods and techniques, while according to a generally held view, design cannot profit from formal procedures, since it is an intuitive and individual activity. Finally, the introduction of a new notion: misfit, is an important contribution, but largely underrated in the design literature.

4. GRAPHS AND MATRICES IN DESIGN PROBLEMS

Graphs and their corresponding matrices have been proposed as useful tools in several fields of design as well as in other disciplines. They provide succinct ways of representing elements and their connections. If properly organised they can aid in identifying essential aspects of a problem. Graphs are of two types: non-directed graphs and digraphs (directed graphs), in the latter case links between nodes (i.e. elements) are arrows, but not in the former. For each graph or digraph, a unique matrix can be constructed; such a matrix is known as adjacency matrix in graph theory,[13] but by various other labels in other fields.

In architectural design, Alexander[2] argues for the benefits of representing design problems, particularly requirements and interactions, as graphs and matrices. With regard to information systems, precedence graphs and precedence matrices for describing information sets and their relationships have been introduced.[17, 21] Further, graphs and adjacency matrices have been found useful in formulating and solving problems in operations research.[10] Finally, Jones[15] briefly discusses interaction matrices and interaction nets as methods of exploring problem structure.

Alexander[2] develops elaborate procedures for automatically arriving at a (de)composition of a design problem, once a graph has been constructed. Similarly Langefors[17] develops matrix procedures in order to find information systems satisfying constraints in external and internal memories, which today have, however, largely been removed by developments in computer technology. Both Alexander[2] and Langefors attempt to formulate some basic graph and matrix theory crucial for their development of applied procedures.

The first efforts to apply graphs to practical problems were directed towards specific problems in each field, and each field proceeded independently of all others. The need for applying such useful tools as graphs was felt to be more urgent than the formulation of general theories of graphs. Independently and in parallel with the applied work on graphs, mathematical theories of graphs and digraphs were formulated and developed.[13]

In recent years, graphs or digraphs and adjacency matrices have had some renaissance with regard to information systems[12] and system–environment relationships, for example organisations and their environments.[6] This revival of the use of graphs draws on formal mathematical theories of graphs and concentrates on practical applications of graphs.

Summarising, graphs have been put forward as tools in many fields of design. They may be useful in conjunction with other design tools, as complements, supplements or alternatives; and they may also be included as parts of methods.

5. FINDING SUBSYSTEMS OR COMPONENTS

Analysis is the first phase in Alexander's[2] approach, when all requirements and their interactions are identified and listed, and a graph constructed. The second phase involves *synthesis*, when requirements are clustered into larger and larger components and a tree of problems, subproblems and subsubproblems, etc. is constructed giving a decomposition of the design problem. A decomposition need not result in disjoint components; a small number of requirements may belong to more than one component (as is illustrated by ref. 8). The purpose is to find a set of design problems which can be solved independently, and where the impact of future external changes is constrained. In addition components can be exchanged, altered and replaced without affecting other components.[2, 15, 20]

With regard to imperceivable systems, Langefors[17] formulates a design theory which suggests a procedure or strategy for implementation. The original system is partitioned into a set of subsystems, which are disjoint and exhaustive. A check is made that the partition corresponds to the original system. Each of the subsystems is then partitioned, the partition checked, etc. When no further levels of subsystems are defined, final checks are made all the way to the original system. In this way, there is *downward specification* corresponding to Alexander's analysis, and *upward construction*, corresponding to Alexander's synthesis. The purpose is to identify subsystems which can be tackled and implemented independently, as well as exchanged independently, due to (for example) external changes, without affecting other subsystems.

The approaches proposed by Alexander[2] and Langefors[17] are very similar, though there are some discrepancies in terminology. They have similar purposes; the phases correspond; they aim at a complete and detailed problem definition before tackling the problem. There are few points of difference. Components or subsystems are specified top-down and checked bottom-up according to Langefors, while a decomposition is arrived at through a bottom-up procedure only based on the detailed top-down specification of requirements according to Alexander. Further, according to Langefors the subsystems need to be disjoint; not so for Alexander. These differences need hardly have any practical implications.

6. SUBDIVISION OF THE DESIGN PROCESS

The formal design approaches proposed by Alexander[2] and Langefors[17] are based on conceptualisations of the design process as a rational and scientific process. They are both only superficially interested in subdividing the process apart from the distinctions between analysis and synthesis, as well as between the what and the how. Langefors[18] suggests a very simple differentiation into method areas (Fig. 1).

Detailed design models, intended for practical use, have been proposed in both ISD and architectural design as well as in other fields of design. Detailed design models are arrived at by applying the principles for finding subsystems to the design process. In general, a detailed design model attempts to formalise the design process by specifying a number of well-defined activities or stages. The activities are ordered, the basic order is a linear sequence, though other types of sequences are quite common in engineering and architecture[15] as well as in ISD.[4] Detailed design models assume that a design process is initiated and terminated at specific points of time. In addition, they prescribe that each stage is to be completed before moving on to the next one, thereby reducing the need for iterations.

There are numerous detailed models for ISD, e.g. the ISAC methodology,[26] each model introduces a number of stages, similarly detailed design models can be found in engineering and architecture.[15, 20] All these models are remarkably alike, irrespective of design field, and despite differences in terminology. This suggests clearly that they are based on similar ideas, reflecting a predominant view that design processes ought to be rational, scientific and systematic. Accordingly, design processes starts with *analysing* requirements, then goes on to *designing* a solution, and finally *implementing* it. An examination of detailed design models discloses that they do not provide descriptions of the process. They do not tell us how designers work, but rather what designers are supposed to produce. In particular, it is the most important purpose of detailed design models to serve as bases for contracts and payments. This is the case in architectural design.[20, 31] A similar notion of contract is implicit in many models for ISD (e.g. SIS 1975).

7. REQUIREMENTS DEFINITION

Formal approaches to design start with a list of requirements which need to be satisfied. A majority of design approaches are based on a *goal-centred view* focusing on stakeholders and goals. The approach to ISD proposed by Langefors[17, 19] is a typical goal-centred one. A few design approaches are based on a *systems-fit view* focusing on the need for matching the target system (to be designed) and the larger system. The approach for architectural design proposed by Alexander[2] is a good example of the systems-fit view.

7.1. A goal-centred view

Many design approaches assume that goals are crucial for design. It is argued that goals are prerequisites for identifying requirements, in particular ISD approaches assume that goals of information systems need to be found. Langefors[19] argues cogently for deriving information requirements and other characteristics of information systems from goals. In a goal-centred approach for ISD goals refer to desirable behaviours and to desirable future states of an information system, perceived by different stakeholders.

Developing and using information systems affects goals of several stakeholders, both internal and external to an organisation. Goals of stakeholders are generally different and may often be in conflict. Processes for resolving conflicts among goals include sequential attention to goals, but they are also games comprising moves, countermoves and outcomes followed by renewed rounds of moves, etc. The processes are influenced by several factors, e.g. alternatives available to a stakeholder, the scope and intensity of the effects of an information system on a stakeholder, and the power a stakeholder can exercise based on available power bases.

Obviously, there are considerable practical difficulties in determining stakeholders and goals. Recognising the existence of stakeholders and stakeholder-goals many formal approaches to ISD attempt to include analysis of stakeholders and goals as integral activities, in order to provide bases for determining information requirements and desired characteristics of an information system. The difficulties inherent in such analyses have often resulted in simplified conceptualisations of goals, giving preference to aspects, which are common to the stakeholders and quantifiable, such as simple measures of profitability. Further, such analyses often assume that goals can be nicely organised into a hierarchy.

Defining the requirements of a design problem according to Langefors[17] is closely related to the procedure for defining subsystems. When partitioning a system into subsystems, the external properties of the subsystems are defined in such a way that, together, they satisfy the external properties of the immediately superior system from which the external properties of the subsystems are derived. Clearly, these external properties are goals, since they prescribe desirable behaviours, and indicate desirable future states of affairs. Langefors[19] argues that business firms *per se* have goals which are distinct and separate from individual objectives of organisational members as well as from objectives of stakeholders, and further this is obvious since different goals of several persons cannot be used for managing a business firm.

7.2. A systems-fit view

Alexander[2] does not relate requirements to goals. He does not even mention the word goal; instead he puts forward the notion of misfit. Many discussions of Alexander's work (e.g. refs 15, 20) do not mention the term misfit, yet the notion of misfit is probably Alexander's most crucial contribution to thinking about design.

When initially conceptualising design problems, Alexander[2] states as a basic idea that every design problem involves an effort to achieve fitness between two entities: the form in question and its context. The object of design is not form alone, but the ensemble of form plus context as a whole. A desirable property of an ensemble is good fit, relating to some particular division into form and context of an ensemble. In order to achieve good fit, Alexander[2] recommends that the process for achieving good fit between two entities be seen as a negative process of neutralising the incongruities, irritants or forces which produce misfit.

The reasons for focusing on misfit, instead of good fit, are quite simple and straight-forward. A good fit depends on the context; unfortunately, it is generally hard to describe the context of a design problem. To make matters worse, there are no directly practical ways of identifying good fit, though bad fit will be recognised when it occurs.[2] Thus in practice, good fit is seen from a negative point only, i.e. good fit means an absence of bad fit. A bad fit is easily recognisable, since it refers to a single identifiable property of an ensemble which is immediate in experience and describable; therefore, when an instance of misfit occurs, it is possible to point specifically at what fails and to describe it.[2]

There is a positive way of describing good fit as the simultaneous satisfaction of a number of requirements, and a negative way, i.e. good fit as the simultaneous nonoccurrence of the same number of corresponding misfits. In practice, Alexander[2] maintains that the positive description will never be as natural as the negative one. In every-day life we do not pay much attention to good fit, until something occurs to produce a bad fit which brings the absence of good fit to our attention.[2]

Thus, Alexander[2] proposes that requirements of a design problem should be considered from a negative point of view, in terms of potential misfits, i.e. disturbances of the fit between form and context. Only those relations in an ensemble between form and context need to be considered which obtrude most strongly, which most clearly demand attention, and which seem most likely to create problems. The process of neutralising incongruities is based on a fundamental principle for identifying misfits, i.e. 'Any state of affairs in the ensemble which derives from the interaction between form and context, and causes stress in the ensemble, is a misfit' (ref. 2, p. 101).

The concept of misfit, introduced by Alexander[2] is a primitive one, in the sense that it is not explicitly defined. *Misfit* is only given meaning intuitively and implicitly, as anything which must be removed in order to achieve good fit. Conversely, *good fit* is the absence of bad fit.

7.3. Good fits, misfits and goals

In a systems-fit view, fit is a central concept, matching or consonance may also be used as synonomous or equivalent terms. Fit is a relationship between at least two distinct systems, i.e. a division of an ensemble into form and context.[2] When two systems are related and dependent, they may be matching or mismatching. Thus the concept of fit may simply be dichotomised into good fit and misfit (i.e. bad fit). As a rule, good fit is a desirable characteristic of the relationship between a form (e.g. an information system) and its context (e.g. the human system, the internal organisational environment, or the external organisational environment).

Though *good fit* is desirable it is hard to describe and there are no practical ways of identifying good fit.[2] Good fit is rarely consciously noticed, it is simply taken for granted, as long as it exists.

On the other hand, *bad fit* is easily recognised when it occurs; it refers to a single identifiable relationship, between two systems, and it is immediate in experience and describable; further, it is possible to indicate specifically what fails and to describe it.[2] A misfit attracts attention and demands correction, it brings the absence of good fit to our attention.

Good fit is obtained by removing misfits. In practice,

	ought to be	ought not to be
exists	good fit 1 11	12 misfit
does not exist	21 goal	22 good fit 2

Figure 3. Goals and misfits.

therefore, good fit means an absence of bad fit. Misfit occurs when the interaction between form and context causes stress. The process of achieving good fit between two entities can be seen as a negative process of neutralising the incongruities, irritants or forces which produce misfit.[2]

In summary, good fit is desired and indicates the absence of misfit. Misfit may exist but it is not wanted, since it causes stress, and therefore, it needs to be removed. This indicates a crucial characteristic of fit, which may be further clarified by contrasting misfit and goal.

It was previously suggested that goals are desirable states of affairs, in the near or distant future, for some stakeholder. A goal is something wanted but which does not exist, it indicates a state to move towards. On the other hand a misfit creates problems and, therefore needs to be removed. A misfit exists, but it is not desirable, it indicates a state to move away from. The differences can briefly be summarised: *a misfit exists but ought not to be*, while *a goal ought to be but does not exist*.[24]

Based on these distinctions, it is proposed that there are two dimensions involved as shown in a 2×2 matrix (Fig. 3). The previous discussion makes labelling two of the cells obvious. Goal belongs in cell 21, since it is something desirable that does not exist. Similarly, misfit belongs in cell 12, since it indicates that something exists which needs to be removed.

The remaining two cells have been labelled good fit 1 (cell 12) and good fit 2 (cell 22), respectively. These two types of good fit correspond with two ways of describing good fit.[2] Good fit may be described positively in terms of simultaneously satisfying a number of requirements, this is good fit 1 (cell 11). On the other hand, good fit may be described negatively as the simultaneous non-occurrence of the same number of corresponding misfits, this is good fit 2 (cell 22). Thus good fit 1 is the existence of property A, while good fit 2 is the non-existence of property not-A.

The major conclusions suggested by the previous discussion and the matrix are: Misfits do not exclude goals, and neither do goals exclude misfits. Goals, misfits and good fits are not interchangeable, instead they are distinct concepts. They may, however, sometimes be used as supplementary notions referring to qualitatively different modes of action, and also to provide a comprehensive picture.

8. CONTRASTING THE APPROACHES

The end products of the approaches are quite different, and they are realised in as different kinds of materials as is possible. In the Langeforsian approach an end-product is realised as a computerised information system, while it is realised as a building, or an urban layout, in Alexander's approach. Nevertheless, there are both similarities and differences between the two approaches; further, the similarities may be attributed to the introduction of a scientific approach to design, while the differences may be attributed to the different histories of design with regard to architecture and information systems.

8.1. Some similarities

Crucial in both approaches is a systematic and elaborate subdivision of a design problem, requiring a detailed and exhaustive problem statement before doing synthesising design work. Specifically, subsystems are found through top-down analysis followed by bottom-up synthesis in Alexander's approach, while they are found and defined through downward specification and upward construction in the Langeforsian approach. The aim in both cases is to find a structure for a design problem making it possible to design and exchange components separately and independently. Detailed design models referring to the two approaches are derived by applying similar principles to design processes.

Formal and mathematical methods are introduced in both approaches in order to improve the effectiveness and efficiency of design. In the Langeforsian approach graphs and matrices are used to find clusterings of information requirements. In Alexander's approach they are used to find clusterings of requirements for an architectural problem. It is the presupposition in both approaches that requirements have been identified completely and exhaustively in advance.

Both approaches are heavily prescriptive. Thus they reflect a common view of how design problems ought to be solved, and how designers ought to work. They do not reflect what a process of design is like in practice. Further, both approaches imply a view of design as a combinatorial problem. They are also mechanistic in their approaches.

Clearly, these similarities can be attributed to the fact that both approaches advocate a scientific approach to design. Their ideal appears to be the exact and physical sciences, in addition to a datadriven approach to formulating theories and hypotheses (i.e. extensive data collection before formulating hypotheses). Design, according to both approaches, is a rational process, requiring complete knowledge and unlimited information processing resources.

Advocating science as an ideal for the practice of design may easily lead to a serious epistemological mistake, namely scientism. This is the idea that the practice of design could and should be converted into a science, i.e. design practice as a scientific discipline. Even though practice of design is not a scientific discipline, it is still both possible and desirable to use scientific methods in practical design work. Furthermore, there is strong need for proper scientific theories both about design and for design. However, neither Alexander's approach (according to ref. 27) nor the Langeforsian approach succeed in completely avoiding the trap of scientism.

8.2. ... and a difference

It is crucial in both approaches to identify requirements and make a complete and exhaustive list of requirements in advance. According to the Langeforsian approach requirements are related to goals. In Alexander's approach requirements are found by examining potential misfits. This difference can be attributed to the different traditions and history of information system design and architecture.

The lack of tradition and history in ISD has meant that designers, who often had a background in engineering, have looked towards engineering, and design of computer hardware, or the physical sciences for models which could be applied to the design of information systems. In engineering, it is assumed that a design problem is well defined and the goal is known. Problem solving becomes close-ended, and the task is to satisfy a set of physical constraints and performance specifications.[31] This means putting off generating a form until a specification is available.

Architecture, on the other hand, has a considerable history and a long tradtion. Traditionally, architectural designers began by proposing a solution or conjecture to a design problem, then testing it against requirements, and next the solution and constraints were modified by revision and refinement.[31] Requirements were not identified and analysed in detail in advance. Instead it was an ongoing process of iteratively putting forward solutions, testing them against requirements, and finally modifying solutions and requirements until no further inadequacies were found. In this iterative process actual misfits are gradually removed making partial solutions increasingly satisfactory.[31]

Unlike traditional architectural design, Alexander's approach attempts to list potential misfits exhaustively and completely in advance. A procedure which has not met with success.[15, 20] Still, the notion of misfit appears to be a crucial one in architectural design.

9. CONCEPTUALISATIONS OF FITS AND MISFITS

The notion of misfit puts the focus on relationships between an object (or form) and its context. In general, there are a large number of relationships between a form and its context. Conceptualising these are clearly crucial in a misfit approach. In principle, there are two alternatives. It may be assumed that the relationships are *independent*, thus making it possible to treat a relationship between a form characteristic and a contextual factor, while disregarding other such relationships. Alternatively it may be assumed that the relationships are *interdependent* and interacting, thus making it necessary to consider relationships jointly and concurrently. When relationships are assumed independent, it is possible to make a distinction between selection and interaction approaches to fit. System approaches to fit are indicated, when relationships are assumed interdependent.

In a *selection* approach to fit it is hypothesised that a single form characteristic is related to a single specific contextual factor, but without examining implications of fit on performance.[9] It is simply taken for granted that a good fit will mean improved performance. Fit may be the outcome of evolutionary or ecological processes or it may be the outcome of intentional or management processes to influence constraints.

In an *interaction* approach, it is hypothesised that a single form characteristic and a specific contextual factor interact, their joint interaction affecting performance.[9] Variations in performance are assumed to depend on the interaction of form characteristics and contextual factors. Good fit is present when the interaction improves performance.

In the selection and interaction approaches, it is assumed that relationships are independent, hence form may be decomposed into parts or components, that can be examined independently and separately, and finally the parts and their performances are aggregated.

In contrast, a *system* approach to fit[9] implies a much more complex picture. Relationships between form and context are complex and interdependent. Contingencies, alternative forms and contexts as well as performance interact, counteract and coact in complex ways; contingencies imply systemic choice and constraint. Performance is both an independent and a dependent variable, it is both an outcome of a certain form and context ensemble and an input to decisions about making changes in the ensemble. There is a need not only to match form to one or two contextual factors, instead there are a large number of contextual factors to consider, which often have contradictory impacts. Further, a need for consistency means that matching form and context will be a question of maintaining a working mixture of relatively good fits (but not necessarily perfect fits) and relatively bad fits, which jointly contribute towards improved performance.

In a systems-fit approach relationships are interdependent. A form cannot be decomposed into parts which are independent and mutually exclusive. Fit is a quality resulting from the pattern of form characteristics that matches the contextual configuration and that also is internally consistent. Fit provides the conditions for improved performance, but does not contribute directly to performance.

A systems-fit approach implies a holistic view, it is the role of fits, their positional value,[1] in the whole system and for the performance of the whole system which is the key. Thus parts cannot be aggregated. Finally, in a systems-fit approach not only are fits and misfits considered but also implications for performance.

In the selection approach, fit is a simple congruence between a contextual factor and a form characteristic. The interaction approach is slightly more sophisticated, since the interaction of a contextual factor and a form characteristic has effects on performance. The systems approach is still more sophisticated, since systems-fit involves complex and interdependent interactions of context and form. The appropriate approach to fit depends, however, on the characteristics of design problems.

10. SYSTEMS-FIT APPROACHES AND INFORMATION SYSTEM DESIGN

The core of a systems-fit approach is finding and removing misfits, which sometimes may be easy to identify but which mostly are not. Clearly, presuppositions about the design problem have important implications. On the whole, three design situations may be identified: well understood, fairly well understood, and finally not-well understood.

Some design situations are *well understood*, then goals are clear, and the end-product of design is well known as well, thus design problems are clearly defined. A design approach means using well-known design methods and tools in well-understood ways, for example decomposing a design problem and using formal methods. Requirements are completely listed in advance. Technical rationality is emphasised and mechanistic assumptions are common. Thus these approaches may be labelled *mechanistic*. When the design situation is well understood, a systems-fit approach to design means finding and removing a complete set of misfits in advance. This might be accomplished by introducing a simple form of fit into, for example, the Langeforsian approach to ISD.

When design situations are only *fairly well understood*, design problems are at first not well defined. It is, however, generally assumed that iterative procedures can be used to clarify the definition. In information system design participation and prototyping have been supplemented to mechanistic approaches, in order to remove some of their deficiencies. In a participative, but still mechanistic, approach analogue models are used to gradually find the information requirements. A distinguished example is the ISAC methodology,[26] which both insistently advocates user participation and is thoroughly grounded in the Langeforsian approach. In prototyping, usually including participation, a first draft of an information system is revised and improved on, i.e. iconic models are used as vehicles for finding requirements. Thus both types of approaches involves a procedure for finding all requirements in a finite number of steps. The outcome of this procedure is a complete specification of requirements. The design process then proceeds exactly as when the design situation is well understood at the outset.

Supplementing mechanistic approaches with participation is not without risks for users. Users run the risk of becoming hostages, when participation is used as a means of ensuring design experts access to information and knowledge necessary for building an information system. In this way design experts are provided with complete knowledge, thus an approximation to the well understood design situation is obtained. Participation has often been naive and innocent in conception, particularly with regard to power and influence, furthermore distant and future users have been disregarded.

A systems-fit approach, when the design situation is fairly well understood, includes procedures for finding and removing misfits until, after a finite number of steps, there are no misfits left, or only inconsequential ones. The process has a definite end, and a final outcome, just as when the design situation is well understood.

There is an increasing number of design situations which are *not-well understood*. They are increasingly becoming ambiguous,[28] muddled[22] or wicked,[30] suggesting a lack of clarity of goals as well as increased uncertainty and equivocality in cause and effect relations. Goals are unknown and unstable, they appear, disappear and reappear. Understanding and interpreting a problem or a situation is not straightforward, problems change, they can only be resolved, and they mean using new design methods and tools in new, not-well-understood ways.

Under these circumstances a goal-centred approach is not adequate and will simply fail, since it is not possible to identify goals that will remain stable for long enough periods of time. Further, an information system is evolving and subject to changes, and so it is never designed but only redesigned. Neither is it possible to find procedures for identifying requirements or misfits until a final set is found, instead the set of requirements is constantly changing and evolving. Clearly, a systems-fit approach needs to incorporate three basic principles. A design process is not finite, instead it is an *ongoing process* without end, only varying in degrees of activity. Second, the process needs to have *procedural rationality*, since it is the working of the process that is crucial. Technical rationality (i.e. rationality of end-products) is not feasible, since the process is constantly changing direction. Finally, a systems-fit approach also suggests a need for reconceptualising *participation*, allowing it to play an active, crucial and determining role, and also giving it adequate support.

Similarly, Checkland[7] argues that 'hard' systems thinking, beginning from an explicit statement of goals, is not appropriate for soft problems, where goals are often obscure and problems are fuzzy and unstructured. Instead a *soft systems methodology* is proposed that uses action research to intervene into ongoing processes of change. The soft systems methodology involves formulating root definitions based on a rich and comprehensive understanding of a problem situation, then building conceptual models and comparing these with the problem situation, and finally suggesting actions to improve the problem situation.

Approaches to ISD based on principles of design as an ongoing process and having procedural rationality are rare. Proposals involving evolutionary design[23] and semi-confusing information systems[14] have been put forward. Evolutionary design does not assume that it is possible to arrive at a final set of requirements as does prototyping. The notion of semi-confusing information system is based on the idea that an information system is a hypothesis which is constantly subject to testing and rejected when it is no longer valid.

In ISD much work has gone into mechanistic and goal-centred approaches appropriate when design situations are well understood or fairly well understood. Much less attention has been paid to systems-fit approaches, which are particularly relevant when design situations are not well understood.

11. CONCLUSION

Ever since the first information systems were designed some decades ago there have been an interest in formal methods in order to facilitate the job of designers. Increasingly, information systems are designed for application areas and introduced into organisational areas, which are complex, and changing, and which also lack goals. There are growing demands for methods, not only in order to facilitate design but also to improve the efficiency and effectiveness of design. The number of formal methods grows and continues to grow. On the reverse side, these formal methods are, however, rarely as successful as expected, since they are based on rational assumptions about design situations, as well known with clear goals. The lack of success of formal methods often leads to the introduction of further formal methods, which unfortunately repeats the mistakes.

Obviously, improvements are urgently needed in ISD. There is a need for methods adapted to increasingly changing and ill-structured design situations. This requires improved understanding of both formal methods and design situations, which may be obtained in several ways. Existing ISD methodologies need to be critically examined with particular attention to their structure, assumptions and characteristics,[29] and comparing these to the characteristics and requirements of design situations. An alternative approach is to draw on design and design methodologies in other fields. This has the advantage that problems and pitfalls in a design method may be identified at an early stage. Methodologies from other design fields needs to be carefully vetted, in terms of basic assumptions and similarities, in order to prevent or forestall difficulties and problems.

Adopting the second alternative, this paper examined the Langeforsian approach to information systems design and Alexander's approach to architectural design. The Langeforsian approach is goal-directed while Alexander's approach is oriented towards removing misfits. Despite differences in subject-matter, the two approaches are similar in many respects, which suggests that they have both been influenced by a movement towards rationality in design. More importantly, these similaritis also lends support to proposing systems-fit approaches in ISD. Since the essential feature of a systems-fit approach is design as an ongoing process, having procedural rationality and involving active participation, such an approach is appropriate when design problems are not-well understood. Goal-centred approaches have been successful in well-understood situations but not when design situations have been not-well understood. Still, there remains much work to be done on systems-fit approaches, which by necessity must be ongoing since the issue is far from well understood. It would be a contradiction in terms to suggest that systems-fit approaches are well understood, when they are oriented towards not well-understood design situations.

Finally, when developing systems-fit approaches it is vital to recognise the various forms of fit and goal. It has been argued, in this paper, that goals and misfits are not mutually exclusive, instead they have quite different connotations. Goals are something desired, while misfits are not but need to be removed. Further, two kinds of good fit have been suggested. The concepts of good fit and misfit are, however, still more complex, since at least three forms of fit can be identified.

REFERENCES

1. A. Angyal, A logic of systems. Reprinted in vol. 1 of ref. 11 (1941).
2. C. Alexander, *Notes on the Synthesis of Form*. Harvard University Press, Cambridge (1964).
3. H. Ansoff (ed.) *Business Strategy*. Penguin (1969).
4. L. Bally, J. Brittan and K. H. Wagner, A Prototype Approach to Information System Design and Development. *Information and Management*, **1** (1977).
5. J. Bubenko Jr. *et al.*, *Systemering 70*. Studentlitteratur, Lund (1970).
6. D. Cartwright and F. Harary, A graph theoretic approach to the investigation of system–environment relationships. Reprinted in ref. 11. Originally published (1977).
7. P. Checkland, *Systems Thinking, Systems Practice*. Wiley, Chichester (1981).
8. S. Chermayeff and C. Alexander, *Community and Privacy. Toward A New Architecture of Humanism*. Penguin (1966).
9. R. Drazin and H. Van de Ven, Alternative forms of fit in contingency theory. *Administrative Science Quarterly*, **30** (1985).
10. S. E. Elmaghraby, *The Design of Production Systems*. Reinhold, New York (1966).
11. F. E. Emery, *Systems Thinking*. Vols 1, 2, revised edn, Penguin (1981).
12. K. Ewusi-Mensah, Computer-Aided Modeling and Analysis Techniques for Determining Business Information Systems Requirements. *Information and Management*, **5** (1982).
13. F. Harary, R. Norman and D. Cartwright, *Structural Models: An Introduction to the Theory of Directed Graphs*. Wiley, New York (1965).
14. B. Hedberg and S. Jönsson, Designing Semi-Confusing Information Systems for Organizations in Changing Environments. *Accounting, Organizations and Society*, **3** (1978).
15. C. J. Jones, *Design Methods. Seeds of Human Futures*. Wiley, New York (1980).
16. B. Langefors, Some Approaches to the Theory of Information Systems. BIT, **3** (1963).
17. B. Langefors, *Theoretical Analysis of Information Systems*, 4th edn (1st edn 1966). Studentlitteratur, Lund (1973).
18. B. Langefors, Management information system design. *IAG Journal (Amsterdam)*, **2** (4) (1969).
19. B. Langefors, *System för Företagsstyrning* [Systems for management control], 2nd edn. Studentlitteratur, Lund (1970).
20. B. Lawson, *How Designers Think*. The Architectural Press, London (1983).
21. I. J. Lieberman, *A Mathematical Model for Integrated Business Systems. Management Science*, **2** (1956).
22. C. E. Lindblom, The science of 'muddling through'. Reprinted in ref. 3 (1959).
23. H. C. Lucas, Jr. The evolution of information systems: from key-man to every person. *Sloan Management Review*, **19** (1978).
24. H. Lundbergh, Misfitanalys – ett alternativ till målanalys? [Misfit analysis – an alternative to goal analysis?] Undergraduate degree paper, Lund University, Informationsbehandling (1972).
25. M. Lundeberg, Information subsystem for setting of sale goals. Example of alternative method of documentation for the information analysis. In ref. 5 (1970).
26. M. Lundeberg, G. Goldkuhl and A. Nilsson, A Systematic Approach to Information Systems Development. I*: Introduction. II: Problem and Data Oriented Methodology. Information Systems*, **4** (1979).
27. J. Lundequist, *Norm och Modell samt ytterligare några begrepp inom designteorien* [Norm and Model plus some further concepts from design theory]. Dissertation. School of Architecture, Royal Institute of Technology, Stockholm (1982).
28. J. G. March and J. P. Olsen, *Ambiguity and Choice in Organizations*. Universitetsforlaget, Bergen (1976).
29. T. W. Olle *et al.*, *Information Systems Methodologies. A Framework for Understanding*. Addison-Wesley, Wokingham (1988).
30. H. W. J. Rittel and M. M. Webber, Dilemmas in a general theory of planning. Reprinted in vol. 2 of ref. 11 (1974).
31. J. A. Wise, Decisions in Design. Analyzing and aiding the art of synthesis. In ref. 32 (1985).
32. G. Wright (ed.), *Behavioural Decision Making*. Plenum, New York (1985).