

Collage of Uffe Lantz.

# Information systems for architectural design

– experiences from the BAS-CAAD and ACTIVITY projects

Anders Ekholm

Architectural design is a process of creative problem solving, moreover architects manage not only information about the building but also about the user organisation. Therefore, information systems for architectural design must support the development of new ideas and concepts, and must be able to handle both building and user data.

The paper presents two recently developed prototype applications, built on a conceptual model for the architectural design process, BAS-CAAD with the objective to support creative design, and ACTIVITY with the objective to support user activity modelling as an integrated part of building design.

## The revolution of architectural design practise

During the last 10–15 years the personal computer together with different software for visualisation, calculation and simulation have revolutionised architectural design practise. For most tasks the drawing-table has been abandoned in favour of the computer platform. Currently, a second revolution takes place where building CAD programs are shifting focus from drawing to

modelling.

Model based CAD means that the design, i.e. the conceptual model of the designed object, is represented externally, outside the mind of the designer. This requires that both the process of design and the objects of design must be problemised: What is design, and what is designed?

The general idea is that model based CAD-programs enable the designer to develop an object based “product” model of the building. Building product models enable computer integrated construction and facility management processes, CIC/FM (Björk 1995). They also enable new ways of managing and structuring design information (Eastman 1999).

However, building design is a complex process involving knowledge not only of the building or the built environment but also the user organisation activities (Ekholm 1987). The building design process starts with describing the user organization and its requirements on the building, then, gradually the focus is shifted towards the building, its construction and maintenance.

The design process is a process of search and gradual development (Simon 1981). It is less a matter of choice among ready-made alternatives than a matter of creative problem solving. A requirement on computer support for design is that both functional requirements and technical solutions can be developed and creatively combined.

The use of model based design and information management systems in the early stages during the programming and proposal stages, as well as in the facility management stages, put new requirements on these systems. Model based CAD-systems must allow an incremental determination of properties, and not prescribe a specific design procedure. It must also be able to represent user activities as model objects independently from the building or its spaces.

This article presents results from current research into these areas carried out at the division of Architectural and Building Design Methodology at Lund Institute of Technology. Results are presented from the BAS.CAAD project and the Activity project.

### **Aspects on building design**

Theoretical foundation for this research

The theoretical foundation for this research is based on Mario Bunge's Treatise on Basic Philosophy (Bunge 1977, 1979, 1983 and 1985). The concepts used in this paper have been presented in earlier writings by the author, the interested reader is recommended e.g. (Ekholm 1987, 1994 and 2001, and Ekholm and Fridqvist 1996, 1998 and 2000).

General aspects on design

The knowledge needed in different fields of design is so diverse, e.g. concerning functions and materials, that it is impossible to learn to design apart from specific knowledge of the artefacts to be designed Bunge (1985:227). However, there are generic theories of artefacts and of intellectual work that could be applied in every design field.

The following sections outline some generic aspects of the substantive and methodological knowledge that is relevant for architects as well as other designers. The first section presents some basic concepts that are

needed to describe the real world and conceptual representations, and the second section presents a generic model for the design process.

The development of methods and systems for architectural design must be based on knowledge of the architectural design process. In order to discuss the subject of architectural design some general aspects of design are necessary to clarify. The verb design means "to conceive and plan out in the mind", and the noun design means "a mental project or scheme in which means to an end are laid out" (Webster's 1999). Accordingly, a design (verb) process results in a design (noun).

### **Design as conceptual modelling**

Conceptual representations

A design is a conceptual representation of the intended artefact and its realisation. A conceptual representation is made up of concepts that refer to the intended artefact and represent its properties. This conceptual representation is developed during the design process. The building blocks of conceptual representations are two kinds of concepts, classes and attributes.

The process of discriminating between objects, concrete or abstract, results in the formation of kinds or classes, e.g. the class of buildings or the class of ideas (Bunge 1979:165). Similarities and differences between objects are based on differences in properties. The concept of a property is called attribute. Class concepts refer to the object as a whole, while attributes represent the object's properties (ibid 1979).

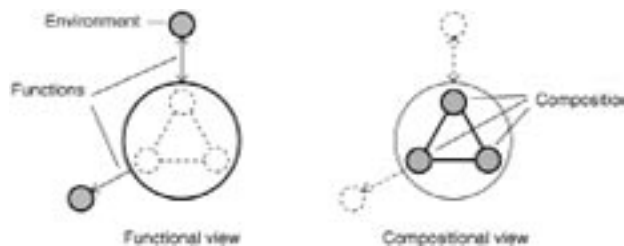
The object-property dichotomy is purely conceptual. A property has no separate existence from the objects that have them. It may be argued that the concept of property is questionable and could be regarded unnecessary; and that it would be sufficient to state that there are different kinds of objects. However, it is epistemologically useful to conceptually separate the object from its properties. During a process of investigation we attribute properties to objects and try out our hypothesis by testing whether the objects have the property or not. For example, design is characterized by conceptualising properties and attributing these to the designed object (Bunge 1983:165).

## Systems and their properties

A system is an extremely generic kind of thing; its properties are common to every complex thing. A system's composition is a set of parts with bonding relations, its environment is a set of things that interact with the system, its structure is the set of all relations, intrinsic and extrinsic, and its mechanism is a set of internal processes. A process is a sequence of events; an activity is a goal-directed process. An artefact is a man-made or man-controlled system; it is made with a purpose to make certain activities possible.

A human activity system that involves the use of artefacts is also called a sociotechnical system. Work is a specific kind of activity, it is a useful activity. A sociotechnical system engaged in some work activity is in management science called an "organisation" (Child 1984), "human activity system" (Checkland 1981), or "enterprise" (Bubenko 1993).

Properties of systems are either intrinsic or mutual. Examples of intrinsic properties are mass and material. A mutual, or relational, property depends on a relation between things. The relation may be functional, i.e. affecting the related systems, e.g. connected-to, sitting-



on, or controlling, or they may be comparative like spatio-temporal properties, e.g. longer-than, to-the-left-of, or during. In a relatively stable environment functions and spatio-temporal properties can be seen as intrinsic properties.

To adopt a view, or aspect, on a system is to observe a specific set of properties. Of specific interest to design are the functional and compositional views. A functional view focuses on the system's extrinsic properties, its relations to the environment, while a compositional view focuses on intrinsic properties, i.e. the properties

of the compositional parts and their structure. See Figure 1.

Figure 1. Functional and compositional views on systems

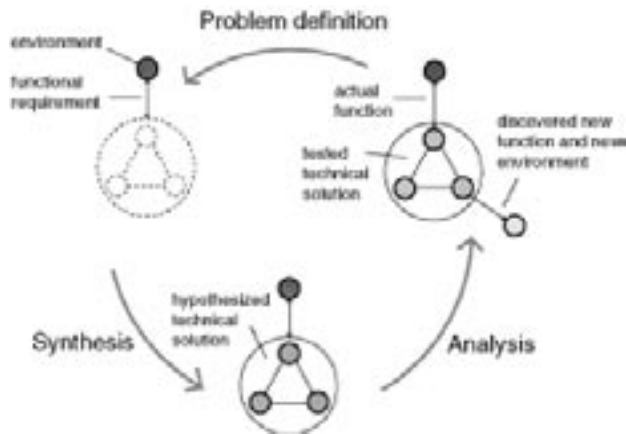
## Information systems, static and dynamic

An information system is a computer-based system, which makes it possible to store and retrieve information of relevance to the information needs of a user. It consists of a conceptual schema, an information base and an information processor (ISO 1985:15). The conceptual schema is made up of entities and relations that refer to and represent the objects that the system handles information about. The information base holds the entity values, e.g. measurements or other data that the user determines. The information processor is a software tool that makes it possible to query and update the conceptual schema and the information base.

The conceptual schema can be understood as an external conceptual representation of the object that the system handles information about. The entities of the conceptual schema together with the attribute values in the information base make up a model of a specific object. Information systems assist the user in managing information about an object. However, statements are restricted to those, which are made possible by the schema entities and their value spaces. This is problematic to an information system for design since it must assist in the development of a conceptual representation, i.e. in the development of the conceptual schema.

Information systems can be characterised as dynamic or static. The user cannot create new entities in a static system, which must be possible in a dynamic system. In a static system, the model objects have to retain their class and attributes once instantiated into the model. In a dynamic system the user would be able to reclassify and exchange attributes of the model objects during modelling.

An information system based on a fixed conceptual schema, or without possibilities for reclassification



of model objects, is not suitable for creative design, e.g. the earliest, most dynamic phases, since a fixed schema and static objects are at odds with the evolving semantics of design (Eastman and Siabiris 1995). The BAS-CAAD system is an example of a fully dynamic system, while the ACTIVITY system is static concerning definition of new attributes, but to some extent dynamic concerning reclassification of model objects.

## Design as problem solving

### Problem solving

Design may be seen as a problem solving process similar to problem solving in everyday life or in science. Problem solving in general is a process of purposeful exploration, "a problem arises when a living creature has a goal and does not know how this goal is to be reached" (Duncker 1945).

Therefore, problem recognition requires that a person has both knowledge and a goal. A goal can be defined as an intended state of a system. The system may be abstract, e.g. a scientific theory, or concrete, e.g. an industry product. The required knowledge is a background knowledge that allows the person to define the goal, and to recognise that certain solution knowledge is missing. A problem can be defined as lack of solution knowledge in relation to background knowledge and goal. A thorough discussion of the concept

of problem may be found in (Bunge 1983).

A tentative problem solution, also called hypothesis, describes a goal, and a testable cause of action that results in this goal. The test may be theoretical, relating the solution to existing knowledge, or empirical, involving an experiment or construction and test of the artefact or a model of it. The specific character of a design problem is its goal, a satisfactory state of a concrete system or process, and a cause of action that results in this. During the design process, hypotheses and tests are made alternately, and the intended artefact and its properties are determined incrementally.

### The problem solving cycle

Every design process is initiated by a problem. The problem definition is followed first by synthesis, leading to a tentative solution, and then by analysis, investigating the proposed solution. The synthesis question is: Which object has these properties? And the analysis question is the inverse: Which properties does this object have? Synthesis may be regarded as starting from a functional view on the object, proceeding from the outside inwards, or top-down, while analysis starts from a compositional view, proceeding from the inside outwards, or bottom-up. The result of analysis is added to the background knowledge. The design cycle, by Simon (1981:149) called the Generator-Test Cycle, proceeds until a satisfactory solution has been developed. The cycle is illustrated in Figure 2.

A design problem may be characterised as open or closed, and the process of problem solving routine or innovative. To a closed problem the determining factors and their combinations are well known; it may be solved by a routine that consists in selecting a prototype solution and determining the values of its attributes. To an open problem, neither the determining factors nor their combinations are known. A prototype solution cannot be applied since new kinds of things or processes must be explored or invented. Open problems have been called "wicked" (Rittel 1984).

## Prototype systems for building design

### The BAS-CAAD project

BAS-CAAD is an acronym for Building and User Activity

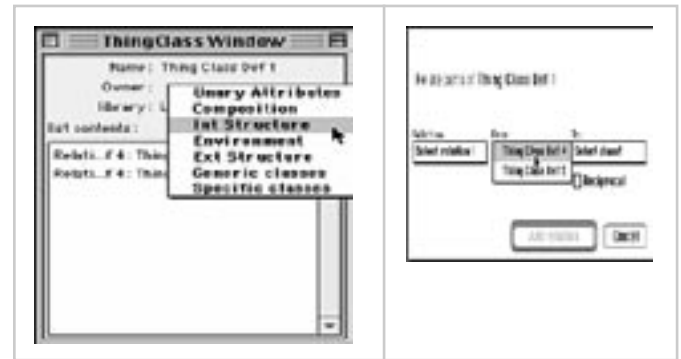
Systems Modelling for Computer Aided Architectural Design. The overall aim of the BAS-CAAD project has been to contribute to the development of CAAD tools for the early stages of the design process including brief development. One of the most important results of the BAS-CAAD project is its conceptual schema, based on the generic concept of system, from which different domain specific frameworks and product models can be built (Ekholm and Fridqvist 1998).

Design statements consist of attributes that characterise the designed artefact. The BAS-CAAD system is designed to collect, analyse and communicate such statements. Design statements may be complex, involving several kinds of attributes, e.g. “red, clay, chimney brick”, or they may be elementary, involving only one attribute, e.g. “chimney brick”. The statements need not be verbal, also 2D drawings or 3D models can be treated as design statements. Design statements can be stored as library objects, generic or project specific. Examples of such objects are everything from buildings and organisations to construction products, persons and equipment.

The entities in the latest implementation of the BAS-CAAD schema are ThingClass (T), Relation (R) and UnaryAttribute (A) (ibid). ThingClass refers to concrete systems, UnaryAttribute represents intrinsic properties of concrete systems, and Relation represents relations between, or mutual properties of, systems. The term UnaryAttribute designates an intrinsic attribute and not a mutual property, it also serves to distinguish these intrinsic attributes from the general concept of attribute used in the text.

A ThingClass consists of a 6-tuple of sets of attributes,  $T = (T_G, T_C, R_I, T_E, R_E, A_U)$ .  $T_G$  stands for “generic ThingClass”,  $T_C$  stands for “composition (or part) ThingClass”,  $R_I$  stands for “intrinsic Relation”,  $T_E$  stands for “environment ThingClass”,  $R_E$  stands for “extrinsic Relation”, and  $A_U$  stands for “UnaryAttribute”. The attributes reflect that a ThingClass can represent a system with composition, environment and structure.

Only seven different types of elementary statements are possible. The first is the statement “there is a thing called T”, which corresponds to creating and identifying a ThingClass. The other six elementary statements

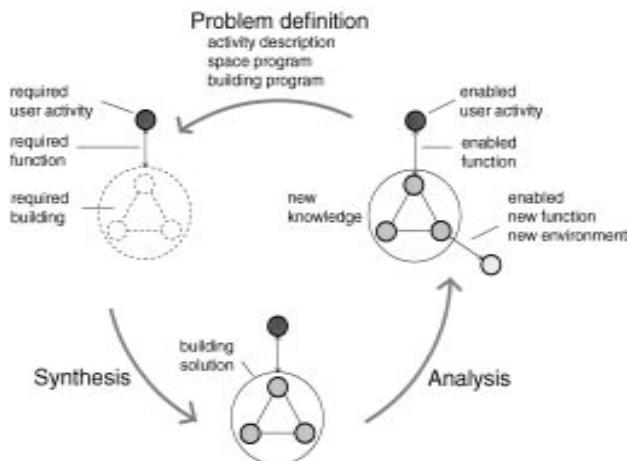


are:

- $T_G$ : T is a kind of Y (= kind T is a specialisation of Y)
- $T_C$ : T is composed of a P part
- $R_I$ : T is internally structured, so that a part of kind  $P_1$  has a  $R_I$ -relation to a part of kind  $P_2$
- $T_E$ : T has an environment, which includes an E thing
- $R_E$ : T is related by an  $R_E$ -relation to an E in its environment
- $A_U$ : T has the intrinsic property Q.

The ThingClass instance with no other attributes defined, refers to an object with only existence determined. The separation of object existence from object attributes allows the designer to add or remove attributes as the design process proceeds. This is a basic requirement on a dynamic information system for creative design, but diverges from the mainstream product modelling practice, which is based on the idea of static information systems and instantiation of predefined classes.

The BAS-CAAD system supports generic design operations like generalising, specialising, aggregating, decomposing, and adding and removing attributes. Design statements are stored in instances of ThingClass. Instance attributes are specified by reference to library objects. In order for the BAS-CAAD system to function in the construction context, it is necessary that its libraries contain entities belonging to established building classification and attribute systems. However, today these systems are poorly developed for use in the early design stage (Ekholm 2001b).



- Inclusion of meta data in the data model, e.g. originator, creation time, and official state
- Separation of function from technical solution in the data model or the library structure.
- Development of libraries to provide model objects adapted for the needs of the construction and facility management sector
- Incorporation of functions supported by different BAS-CAAD prototypes into a new prototype system
- Investigation of user interface for dynamic information systems for design.
- Further study of dynamic design systems, e.g. concerning automatic classification, and mapping between classification systems.

The BAS-CAAD system has been implemented in Smalltalk, an object oriented programming environment. A sequence of implementations are presented in Sverker Fridqvist's doctoral thesis (Fridqvist 2000). Figure 3 shows an example of a user interface.

Further prototype development may include the following aspects:

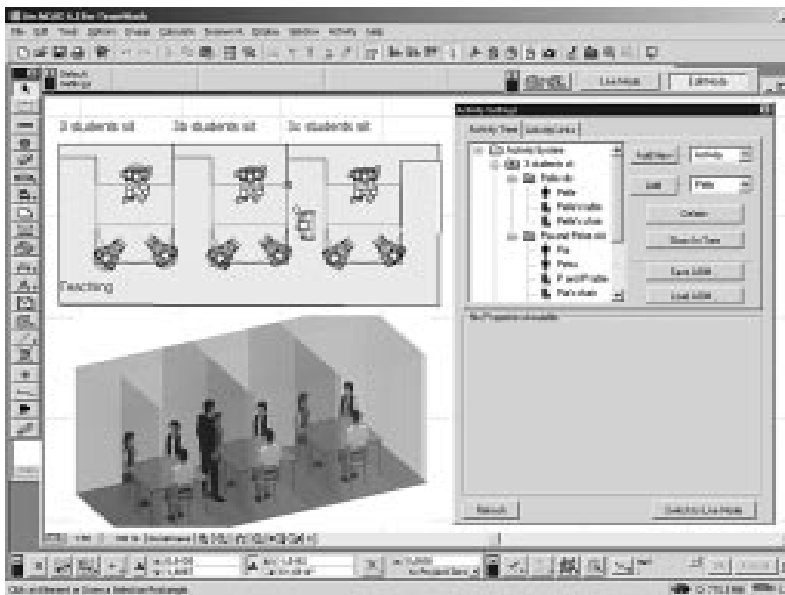
### The ACTIVITY prototype

User activity information in architectural design

When an organisation is formed or changed, it may require a new or renewed building. The process of acquiring a suitable building starts with a description of the organisation and its activities. The activity description is used as a basis for developing a space function program, which defines requirements on the building's spaces. The following step includes development of a

building program, which defines additional requirements on the building. The building program together with the activity description and space program are used as a background for building design, but can also be used for building performance analysis during the facility management stage. See Figure 4.

Model based CAD-systems for building design may support the fundamental design steps of problem definition, synthesis and analysis. An application that allows the development of a user activity description assists the problem definition work, while the building design application assists the synthesis work and allows the designer to document the building property decisions. Building design analysis includes tech-

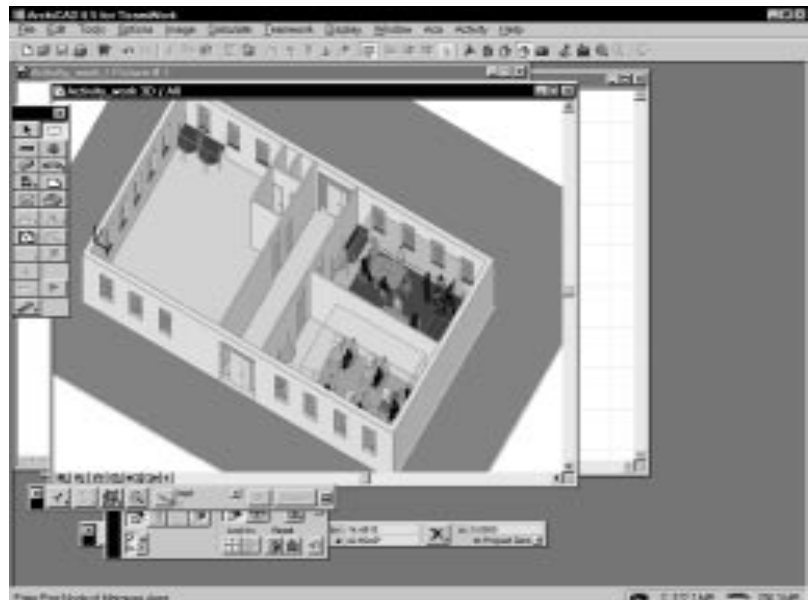
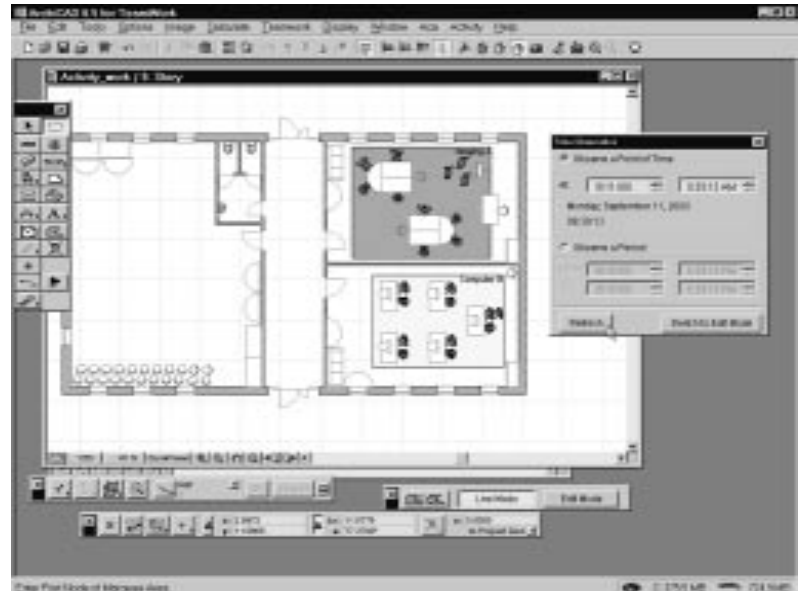


nical performance, usability and cost calculations. Certain geometrical properties can be visually evaluated, while other properties can be evaluated through calculation, e.g. fire escape routes, ventilation, and lighting levels. The test results are used as background knowledge for subsequent design process cycles.

### Implementation of ACTIVITY

The idea of modelling user activities as an integral part of the building design process is implemented in the ACTIVITY project. ACTIVITY was developed as an add-on to the established architectural design software ArchiCAD. An add-on is a separate program that expands the functionality of another program, and can only be run within this. ACTIVITY has its own user interface accessible from the interface of ArchiCAD, e.g. new menus, dialogue boxes, floating palettes, etc.

The basic entity is the Activity. It is based on a functional view on an organisation or part of an organisation. An Activity may have other activities as functional parts, or be a functional part itself of other activities. An Activity is composed of Person and Equipment. The constituent Person and Equipment may be determined for an Activity at any level in the "hierarchy". An Activity may have Name, Description, Duration, and Relations. There are four Relations that can be specifically shown: Visibility, Sound, Distance, and Adjacency. These may have values which however, can only be described, functionality is not implemented. A Person can only exist within an Activity. Equipment may be composed of other Equipment, and can







Anders Ekholm, Professor of Architectural and Building Design Methodology, Dept of Construction and Architecture, Lund Institute of Technology, Lund University  
<http://www.caad.lth.se/> Anders.  
[Ekholm@caad.lth.se](mailto:Ekholm@caad.lth.se)

## References

- Björk B.-C. 1995, Requirements and information structures for building product data models. VTT Publications 245 (Technical research centre of Finland, Espoo)
- Eastman C. 1999, Building Product Models: Computer Environments Supporting Design and Construction. CRC Pr, London.
- Bubenko jr J. A. 1993, "Extending the scope of information modelling", Report No. DSV 93-034. Department of Computer and Systems Science, KTH, Stockholm.
- Bunge M. 1977, Ontology I: The Furniture of the World, Vol. 3 of Treatise on Basic Philosophy. Reidel, Dordrecht and Boston.
- Bunge M. 1979, Ontology II: A World of Systems, Vol. 4 of Treatise on Basic Philosophy. Reidel, Dordrecht and Boston.
- Bunge M. 1983, Epistemology and Methodology I: Exploring the World, Vol. 5 of Treatise on Basic Philosophy. Reidel, Dordrecht and Boston.
- Bunge M. 1985, Epistemology and Methodology III: Philosophy of Science and Technology, Vol. 7 of Treatise on Basic Philosophy. Reidel, Dordrecht and Boston.
- Checkland P. 1981, Systems Thinking, Systems Practice. John Wiley & Sons, Chichester.
- Child J. 1984, Organization: A guide to problems and practice. Paul Chapman Publ. London.
- Duncker K. 1945, "On problem-solving Psychological Monograph 58, No. 270". Quoted from (Bunge 1983:235).
- Eastman C. M. and A. Siabiris 1995, "A generic building product model incorporating building type information". Automation in Construction, 3(4), p. 283-304.
- Ekholm A. 2001a, "Activity objects in CAD programs for building design". Forthcoming in Proceedings from CAADFutures 2001. TUE, Eindhoven.
- Ekholm A. 2001b, "BSAB och klassifikation för produktmodellering och design". Slutrapport förstudie 2001-04-10. AB Svensk Byggtjänst, Stockholm.
- Ekholm A. 1994, "A systemic approach to building modelling - analysis of some object-oriented building product models". In: Björk, B.-C. (ed.) CIB W78 Workshop, Aug. 22-24 1994, Esbo, Finland.
- Ekholm A. 1987, Systemet Människa-Byggnadsverk - Ett ontologiskt perspektiv. Statens råd för byggnadsforskning R22:1987, Stockholm.
- Ekholm A. and S. Fridqvist 2000, "A concept of space for building classification, product modelling, and design". Automation in Construction, 9(3), p. 315-328.
- Ekholm A. and S. Fridqvist 1998, "A dynamic information system for design applied to the construction context". In: Björk B.-C. and A. Jägbeck (eds) The Life-Cycle of IT Innovations: Proceedings of the CIB W78 Conference, June 3-5, 1998. Royal Institute of Technology, Stockholm.
- Ekholm A. and S. Fridqvist 1996, "Modelling of user organisations, buildings and spaces for the design process". In Turk Z. (ed.) Construction on the Information Highway. Proceedings from the CIB W78 Workshop, 10-12 June 1996, Bled, Slovenia.
- Fridqvist S. 2000, Property-oriented information systems for design. Prototypes for the BAS-CAAD project. (Diss.) Lund Institute of Technology, Lund University, Lund

- ISO 1985, Concepts and terminology for the conceptual schema and the information base ISO/DTR 9007 (TC97), also SIS teknisk rapport 311. SIS, Stockholm.
- Rittel H. 1984, In N. Cross Developments in Design Methodology. John Wiley and Sons, London.
- Simon H. 1981, The Sciences of the Artificial. MIT Press, Cambridge.
- Websters 1999, New Collegiate Dictionary. Electronic edition. G&C Merriam Co., Springfield, Massachusetts.