DESIGN AS PROBLEM HANDLING – OUTLINE OF A FRAMEWORK

ANDERS EKHOLM

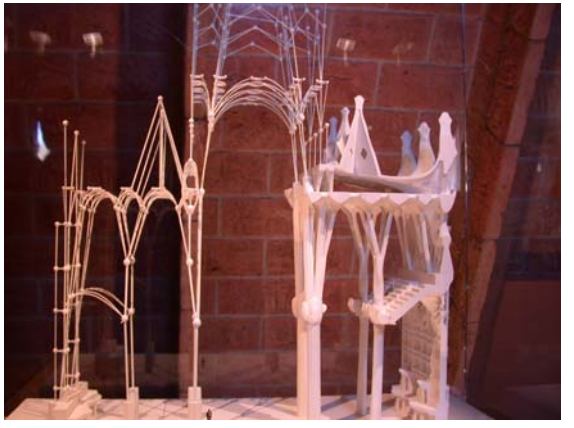
LUNDS TEKNISKA HÖGSKOLA
Lunds universitet

Disposition

- Background: BASCAAD, ACTIVITY
- A need for a theory of design, specifically arch design
- Design as problem handling, i.e. as idea and knowledge development
- Information systems for design

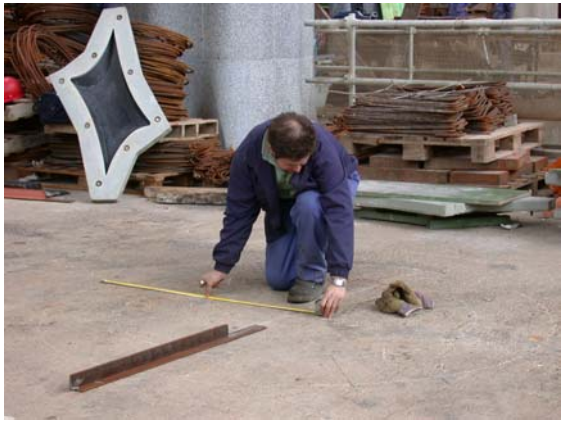
Design and Science

- Science tells us how things are; design tells us how things ought to be (Simon 1981)
- Technology is the science of design of the artificial
- *A design* "a mental project or scheme in which means to an end are laid out",
- *To design* "to conceive and plan out in the mind"
- Theories of design, or technology, are not as developed as theories of science



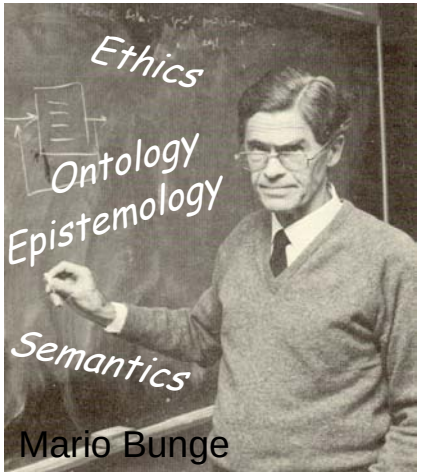
The need for a design theory

- Design requires both substantive and methodological knowledge
- Design requires domain specific knowledge, compare the design of social systems and motors
- Design skill requires comprehen-sive knowledge about the product
- A ”common ground”, a generic design theory, is required for the development of design methods, e.g. CAD, and for scientifically based design education

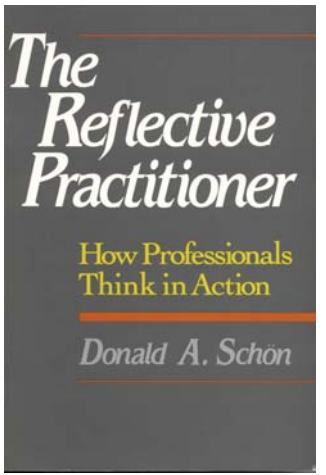


A framework for design

- Aim: a framework for design seen as technology
- Consequences for IT-support for design of the built environment
- Bunge's "Treatise on Basic Philosophy" meets Schön's "Reflective Practitioner" and this authors idea of the built environment as a socio-technical system with material and cultural properties



Mario Bunge



Technology and Architecture

- Architecture is not an art but a design discipline, a technology
- Architecture is:

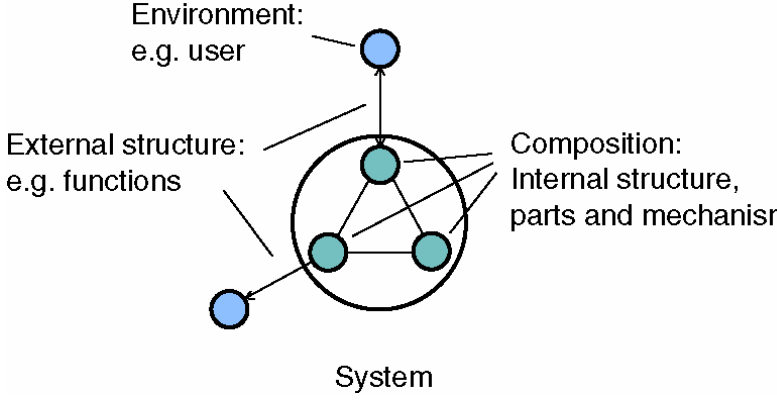
  the design of buildings and human activities that depend on use and experience of buildings
- A technological approach to architectural design theory as opposed to the traditional art theory approach

Philosophical basis for design

- A Design Theory must be based on general theories of artifacts and intellectual work
- Ontologi, among others Systems Theory
- Epistemology, the concepts of ”knowledge” and ”problem”
- Semantics, the relation concept-reality-symbol
- Ethics, the rightful action

Ontology for design

- Buildings are artifacts
- Artifacts are concrete systems made or controlled by man for a specific purpose
- Basic properties are composition, environment and structure, including state and history
- Properties can be divided into material and cultural



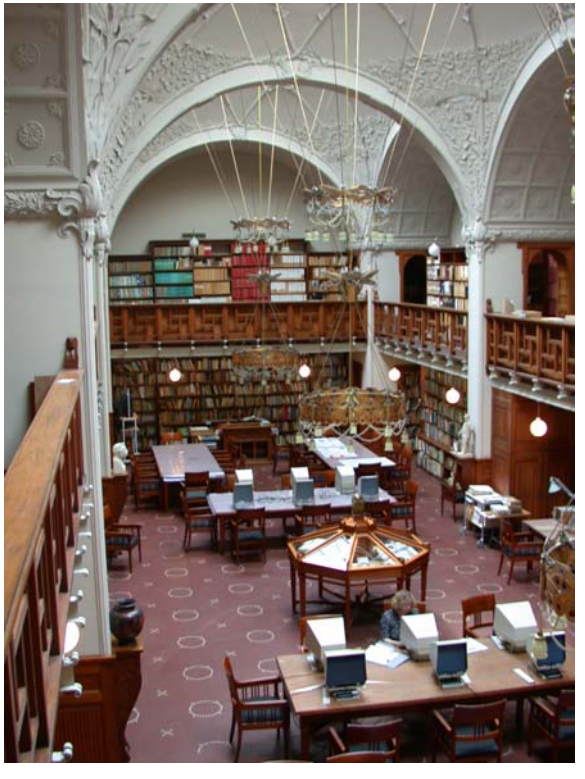
Epistemology for design

Knowledge

- *sensory-motor* – e.g. knowing how to throw a ball, or how to ride a bicycle;
- *perceptual* – *e.g.* seeing the size of a space, or identifying a false note in a song;
- *conceptual* – e.g. knowing the composition of a building, or the rules of chess.

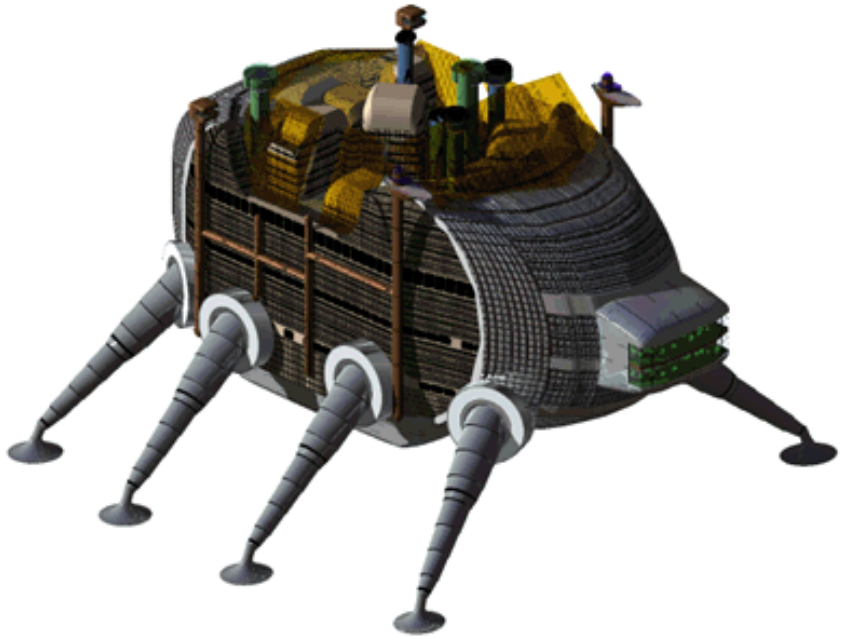
Field of knowledge

- Material framework
  - A group of people in a society and a domain of investigated objects
- Conceptual framework
  - Philosophical outlook
  - Formal knowledge: logic and mathematical theories
  - Knowledge from other fields
  - Problematic
  - Established knowledge
  - Aims and goals
  - Methods



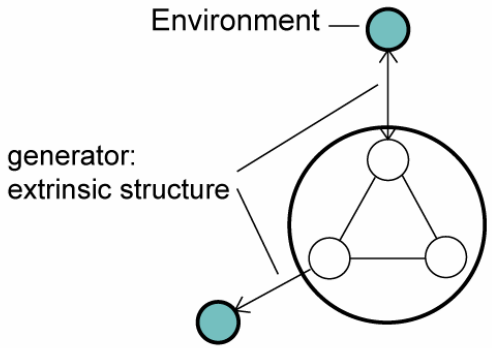
Problem

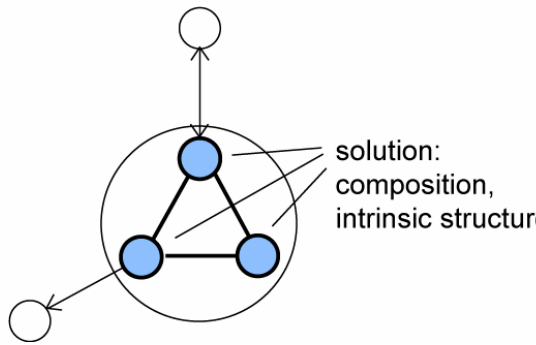
- Problems are discovered through curiosity
- A problem could be described as ”the difference between what is known and what one wishes to or needs to know”
- A problem is a conceptual object with three main constituents :
  - Presupposition(s)
  - Generator
  - Solution



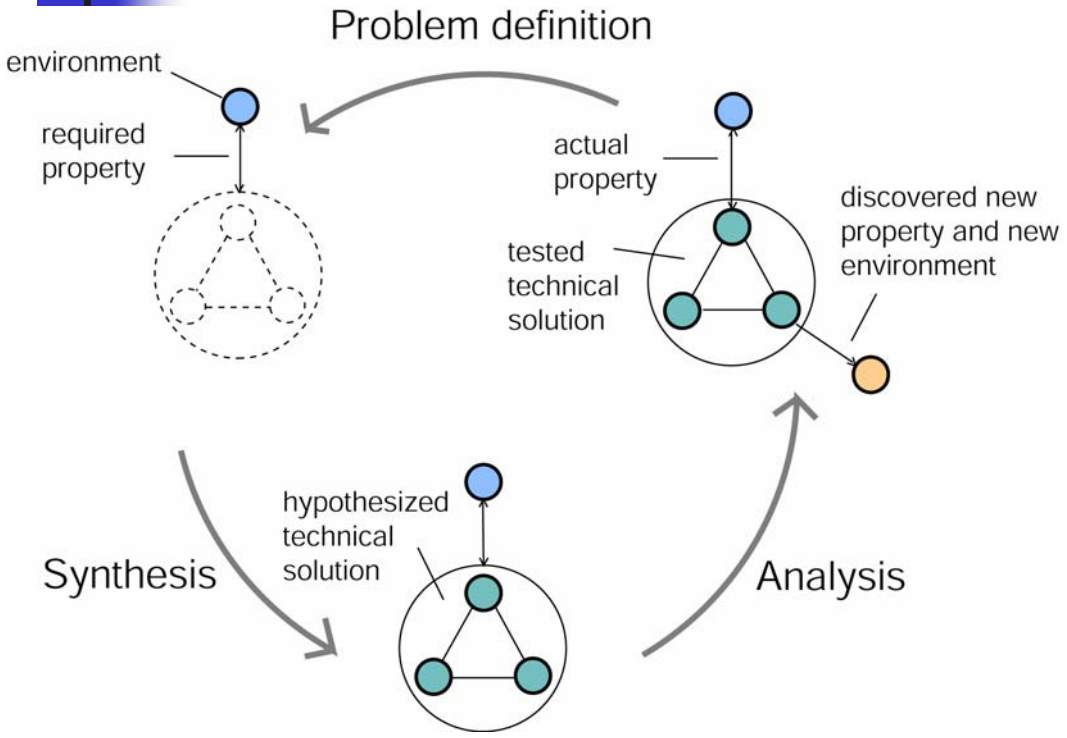
Aspects on systems

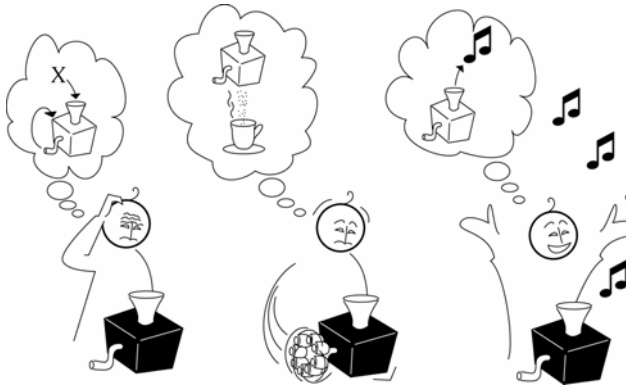
- Generator and solution are different aspects on the system
- A generator is an extrinsic aspect and regards the system's mutual properties
- The solution is an intrinsic aspect and regards the system's compositional properties
- The relation generator - solution is many to many





The problem handling cycle



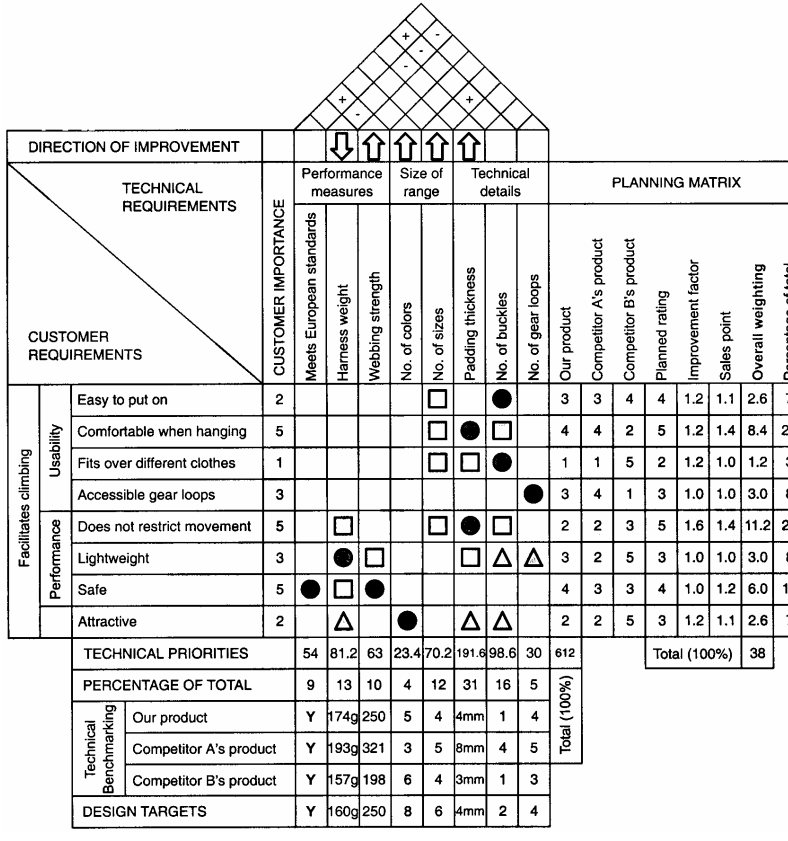


The ”Generator – Test Cycle”

- Synthesis problems – Analysis problems
- Routine problems – Research problems
- Cognitive problems – Practical problems

Problem definition

- Successive crystallization and clarification
- Problem setting - approach
- Schön: ”problem setting is a process in which, interatively, we name the things to which we will attend and frame the context in which we will attend to them”
- ”House of Quality”

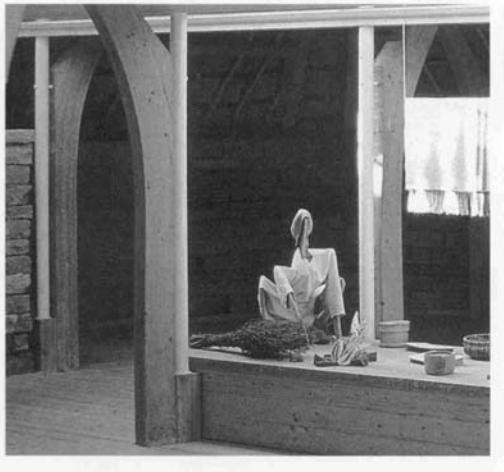


Synthesis

Creation of a “technical solution” through:

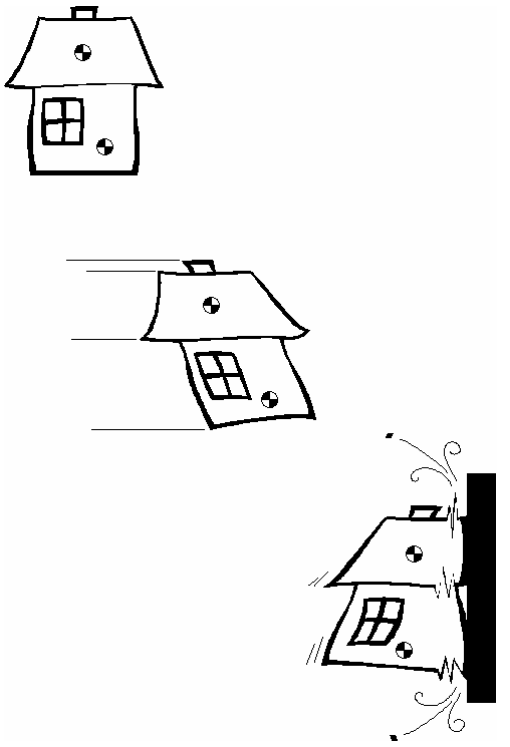
- Induction from observed cases (follow precedents)
- Association of facts in time or place (similarity of idea)
- Similarities of facts, e.g. analogies (model studies)
- Application of general principles (deduction from theory)
- Imagination and invention

Ovan pilastrar på plintar mot pelare med murbågar. (Gården av Bramante vid S. Maria della Pace i Rom.) Nedan stålpelare på klackar och mot limträbågarna i Eketorps museum.



Analysis

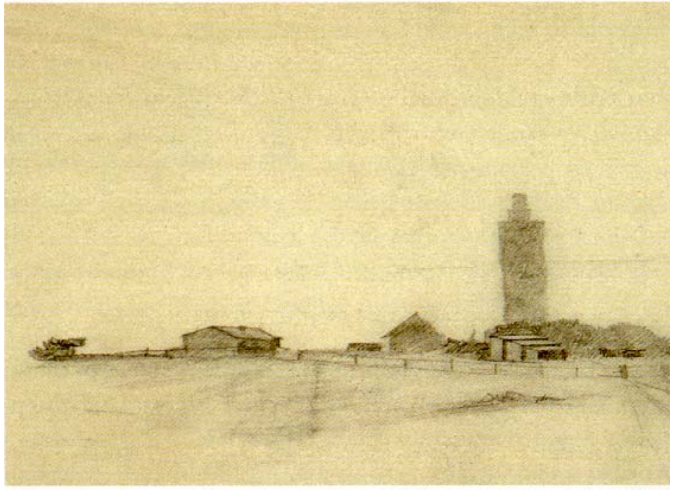
- A solution must be efficient and acceptable; it is evaluated through concretizing, model studies and simulation
- Reflection-in-practice: the problem is continuously evaluated
- Reflection-on-practice: critical method evaluation



Seeing-as in design

From an early generator:

In the natural environment carefully placed building

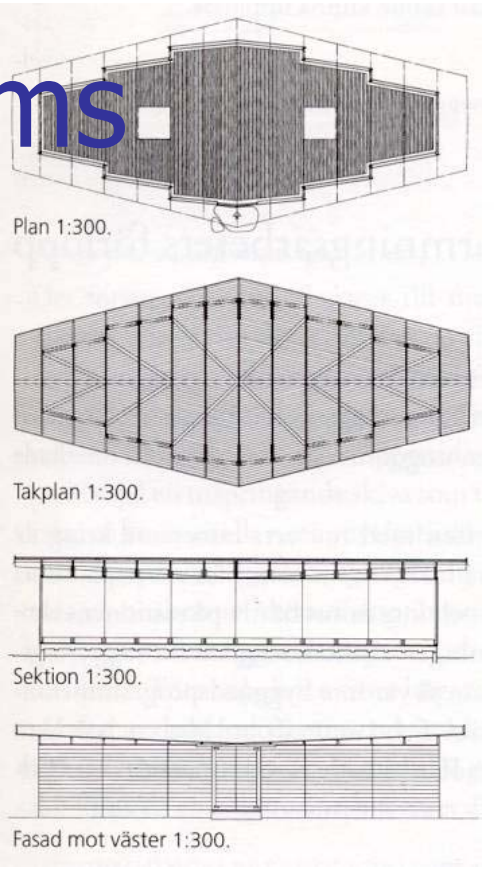


To a new generator:

A small public building

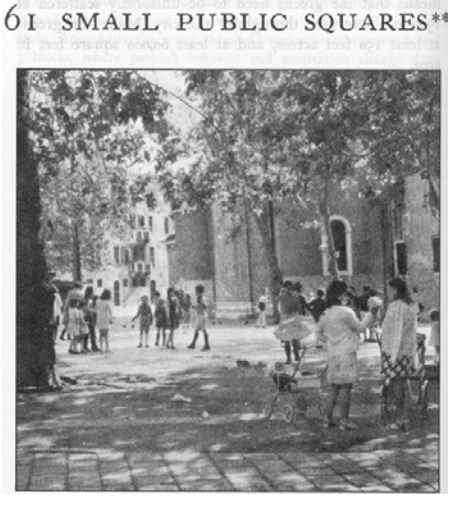
Architectural problems

- Background knowledge: the architect's conceptual framework, and architectural design skills
- Generators are often cultural properties, e.g. a user's experience of homeliness, efficiency, beauty and status
- Generators must be transformed into material and cultural properties of the building: e.g. spatial layout, material, function, geometry, colour etc.



Architectural design

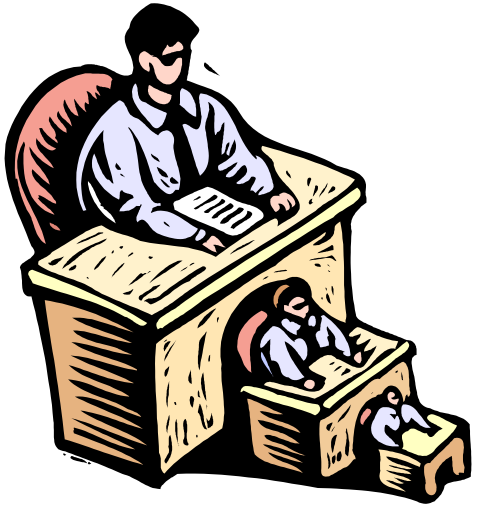
- Behavior settings (Barker)
- Patterns (Alexander)
- Situations (everyday language) man and environment in interplay
- Material and cultural properties must support the activity
- Lawson: ”the architect as always trying something new, but having low predictive capability and poorly equipped to learn from mistakes” (The Language of Space 2001).



Information systems

The three intertwined levels of software design (Fischer)

- Conceptual framework level
  - Domain independent schema
- Domain level
  - Domain specific schema
- Individual artifact level
  - Instantiates a domain level schema

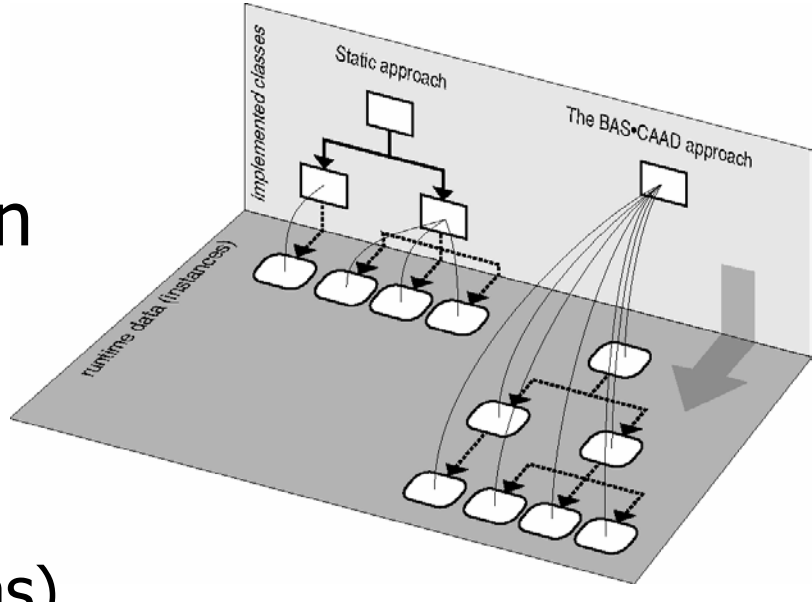


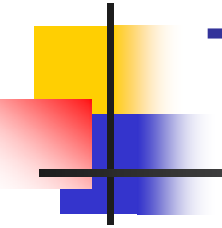
Development praxis

- Seeding
  - Generic software, e.g. object-oriented, and application software, e.g. CAD
- Evolutionary growth
  - Application in reality by domain experts, feed-back to application and system developers
- Re-seeding
  - Software adaptation by system developers and domain experts

Requir on synthesis software

- Instantiate a ”neutral” object, ID only
- Incremental specification of object attributes
  - Specialization
  - Multiple inheritance
  - Reclassification (seeing-as)
- Evolvable domain object library





To be continued