

The fundamental principles of Service Systems construction

Summary of PA116



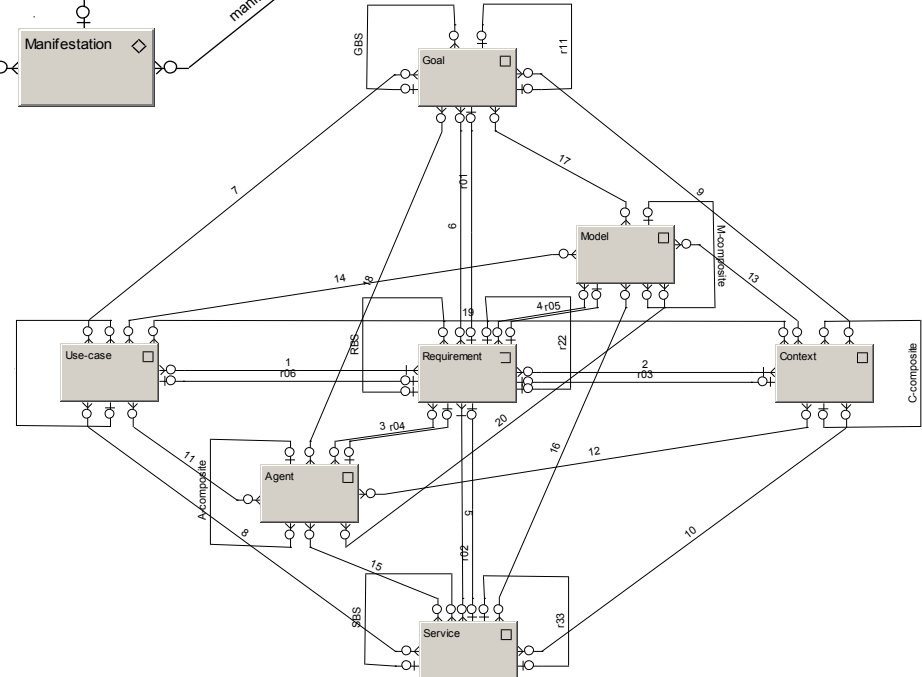
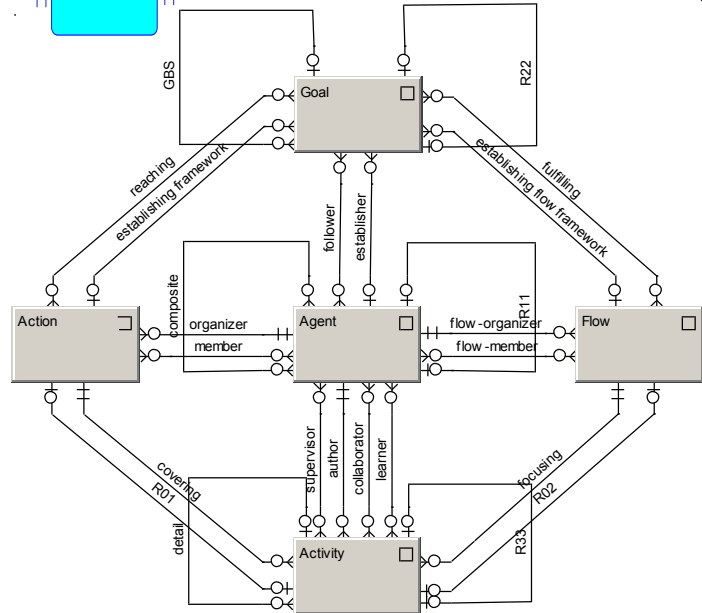
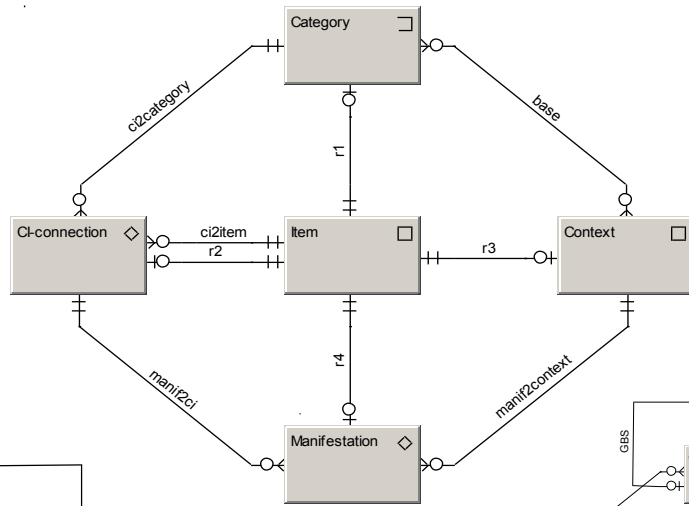
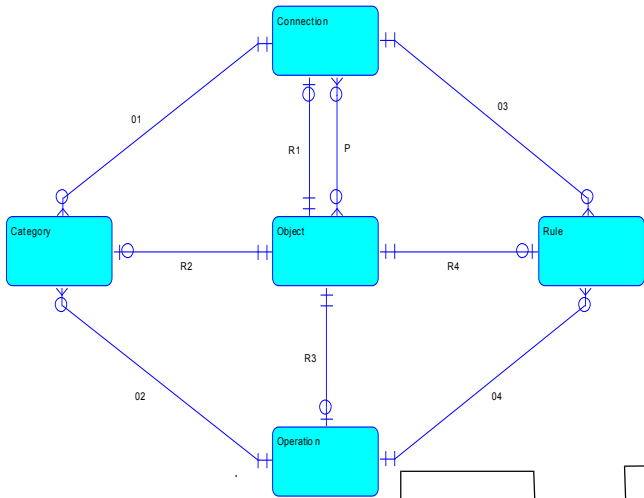
INVESTMENTS IN EDUCATION DEVELOPMENT

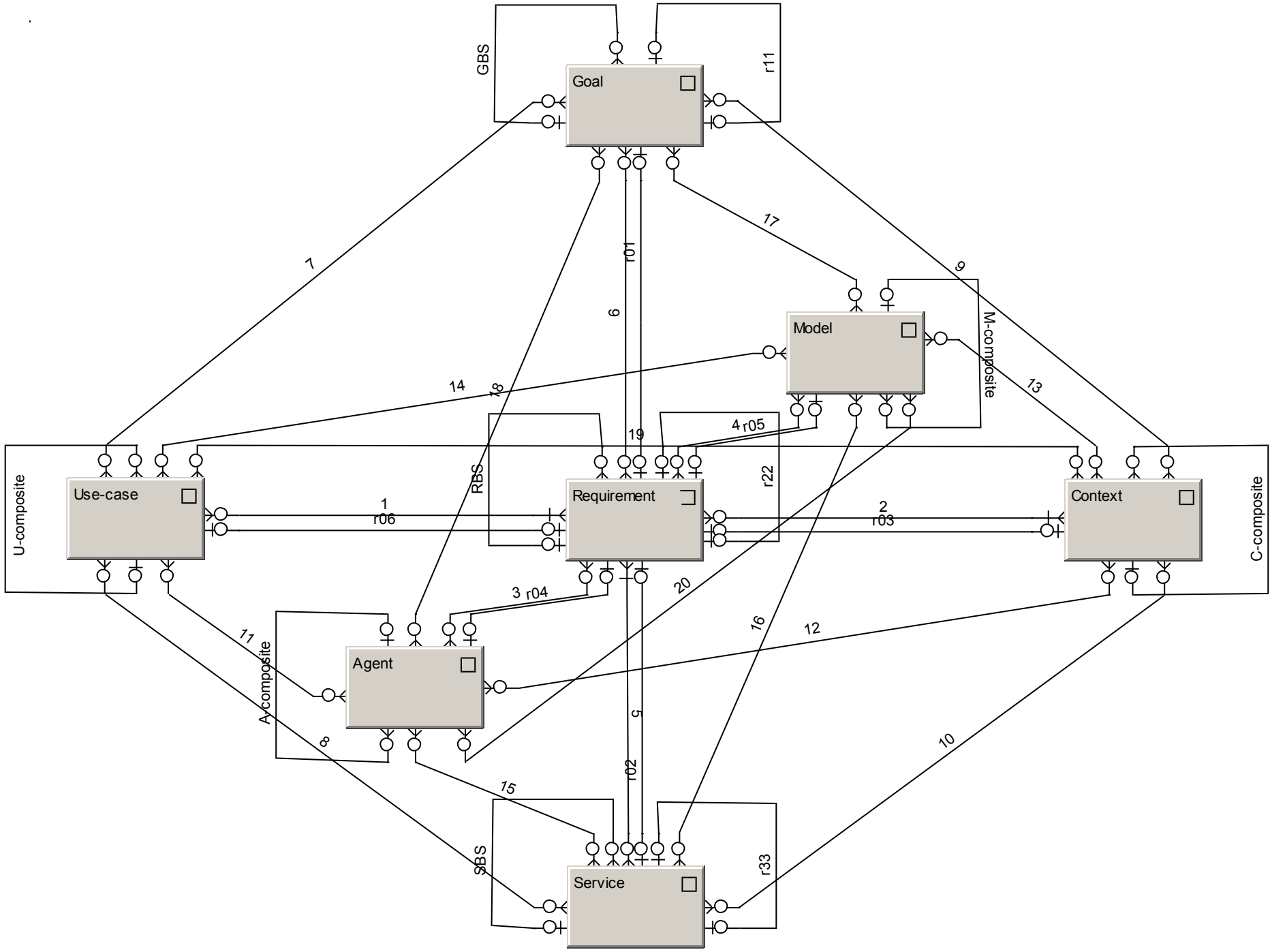
S

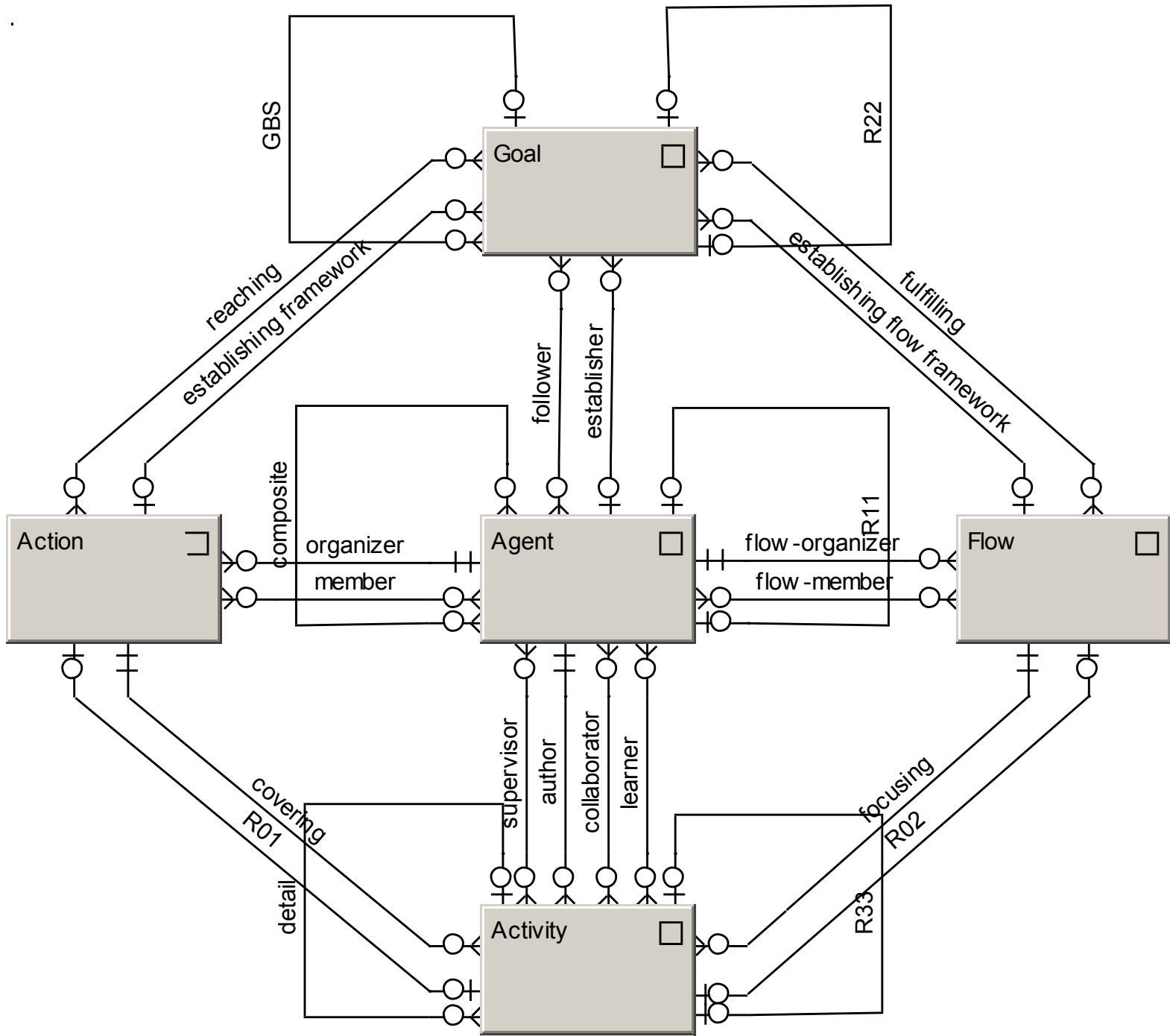
R

O

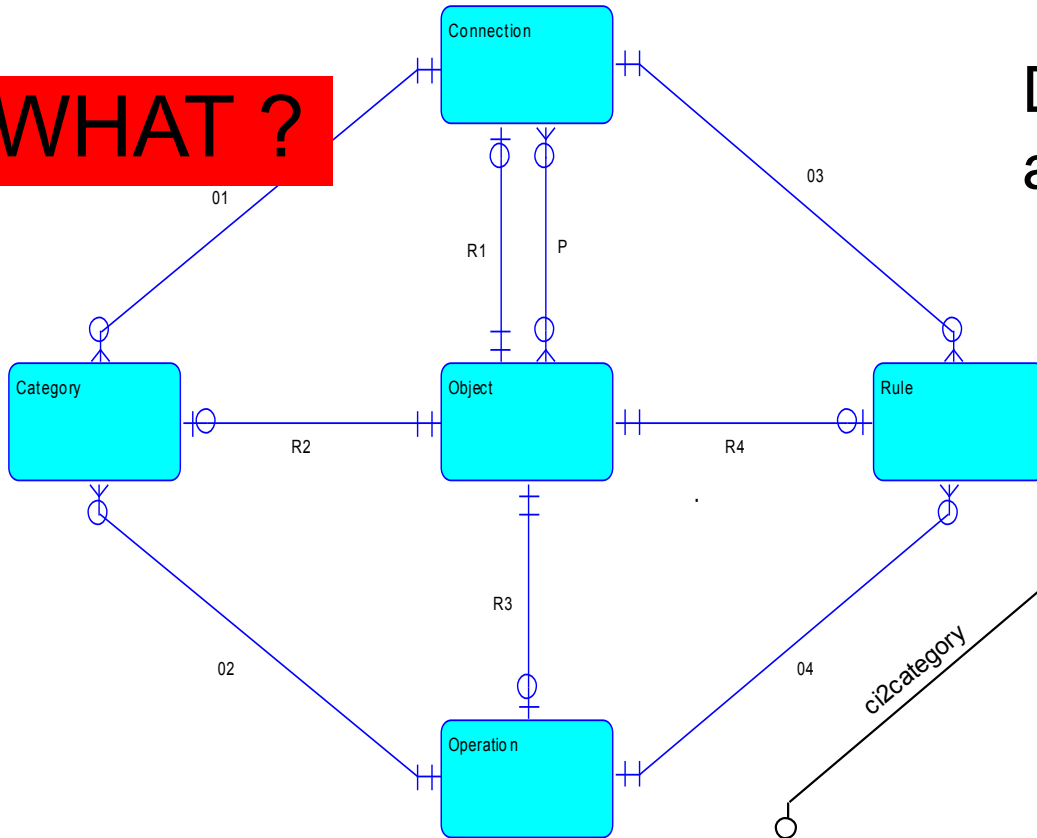
D







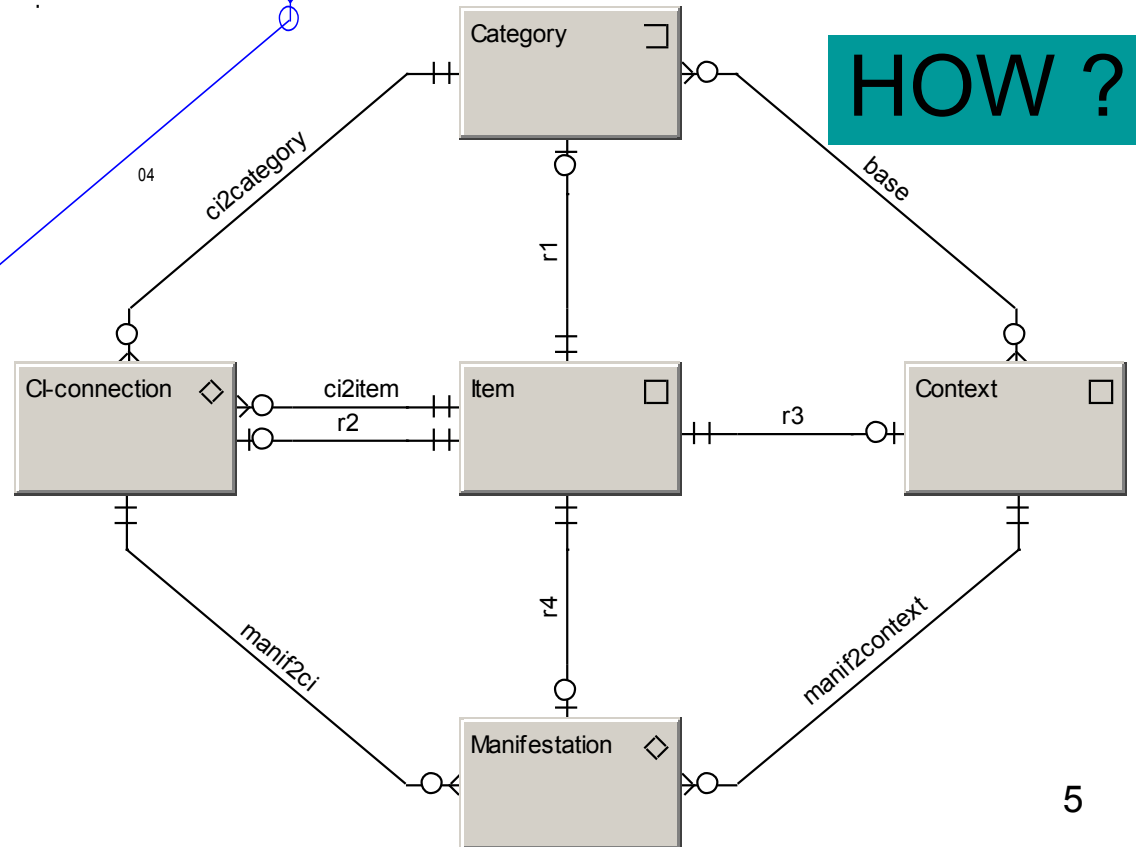
WHAT ?



Duality of what to focus on and how to manage it

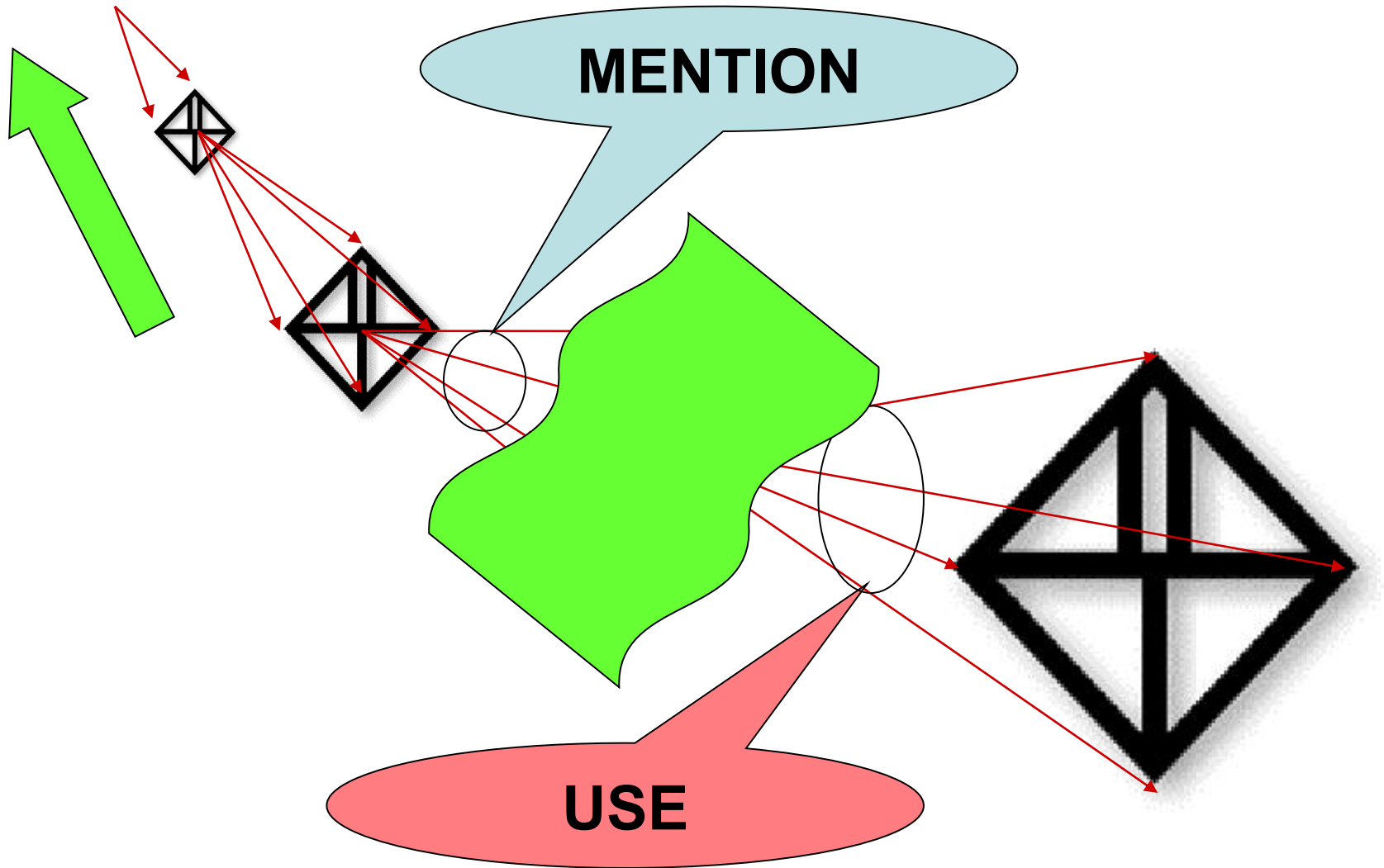
CONCEPTUAL MODELS

HOW ?

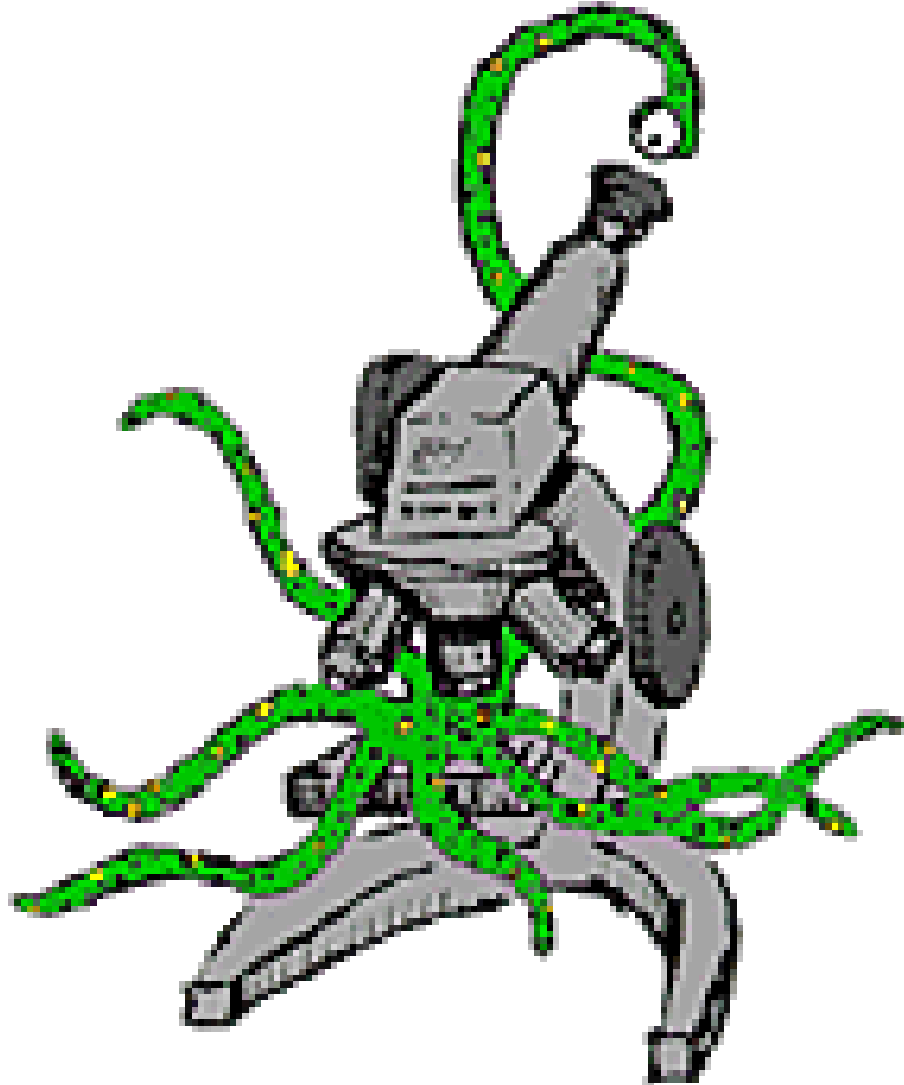


Two fundamental Diamonds

Where does it end / begin ???

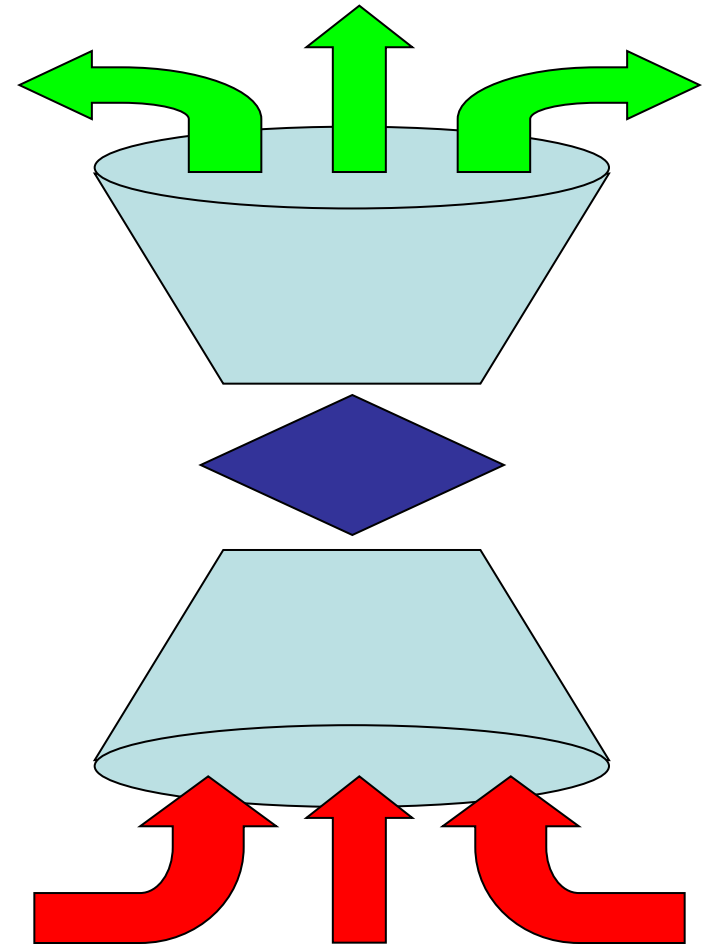


Self-reference

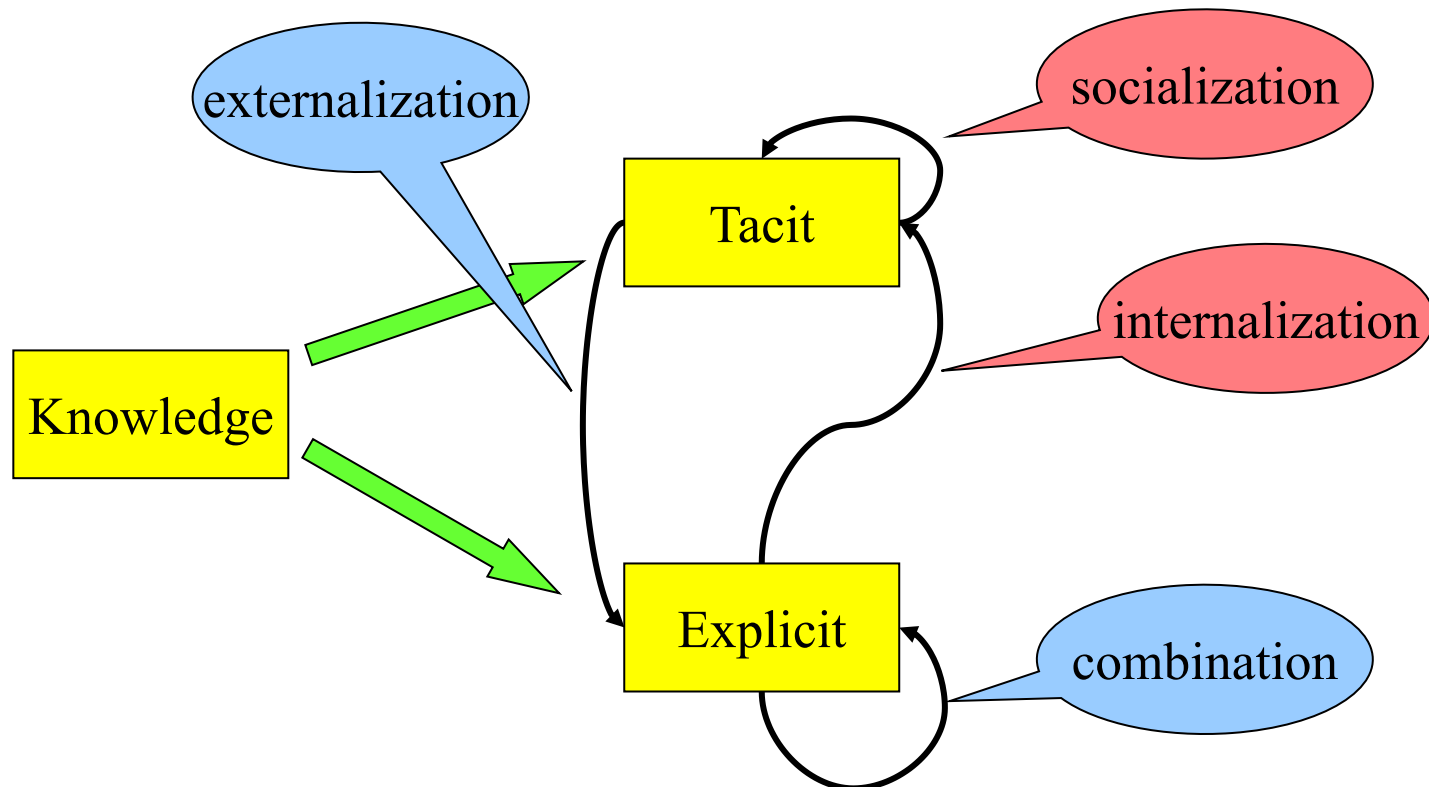


The double-funnel adaptive principle

- **Synthesis of information** from heterogeneous net of data sources
- **Visualization of the synthesis result** directly supporting decisions of an expert “**now and in a given situation**”
- **Support of momentary knowledge utilization**, not only the pre-prepared knowledge utilization
- **ADAPTIVITY !!!**



Knowledge Management means to keep the Nonaka-Takeushi cycle in a run !



Essence of communication, understanding, and modeling

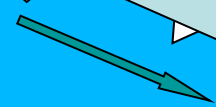
linguistic
EXPRESSION

represents



CONCEPT (closed
construction)

denotes



identifies



CYBERSPACE

OBJECT

“which is focused on”

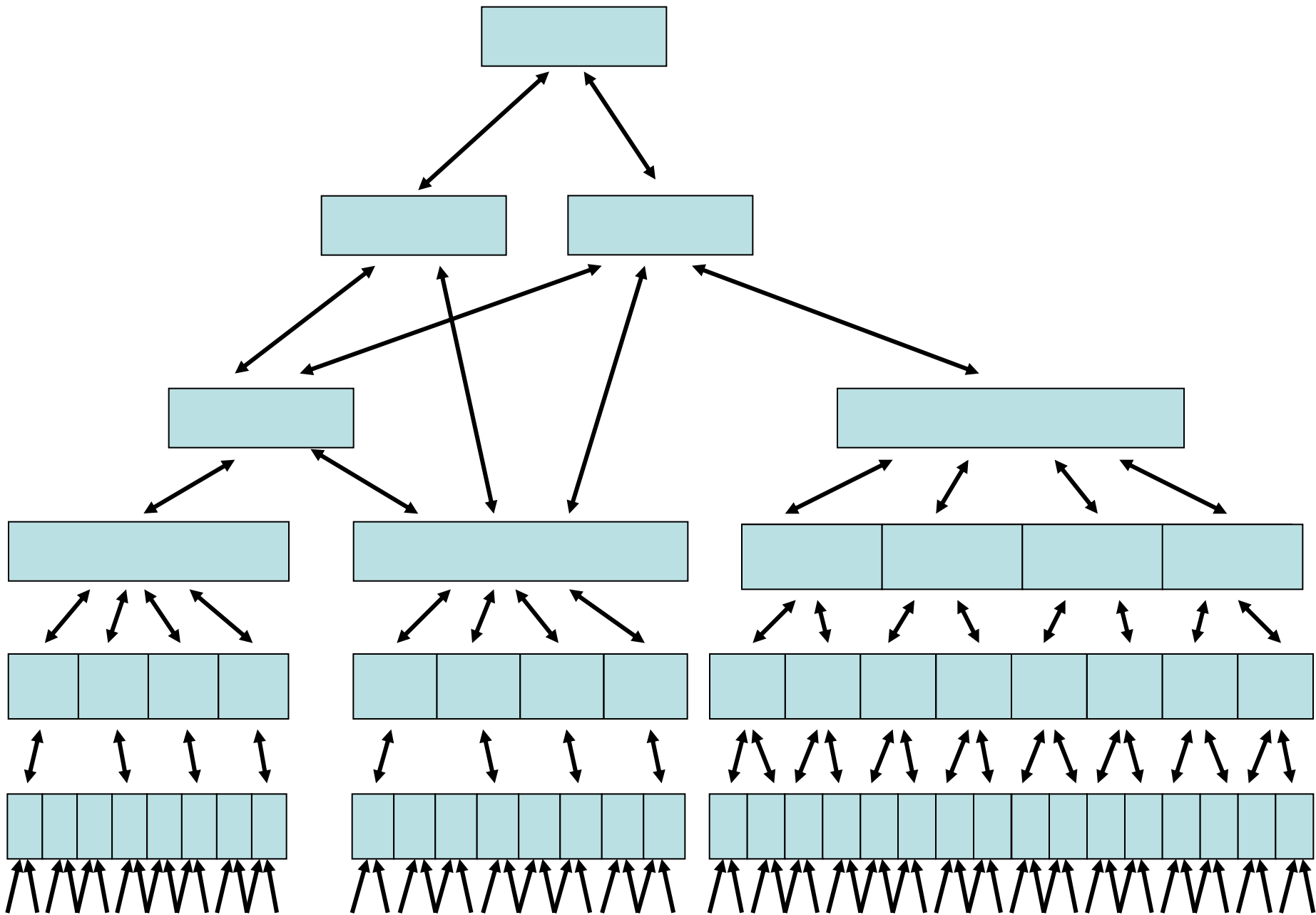
Strictly
empty

Extension/Intension

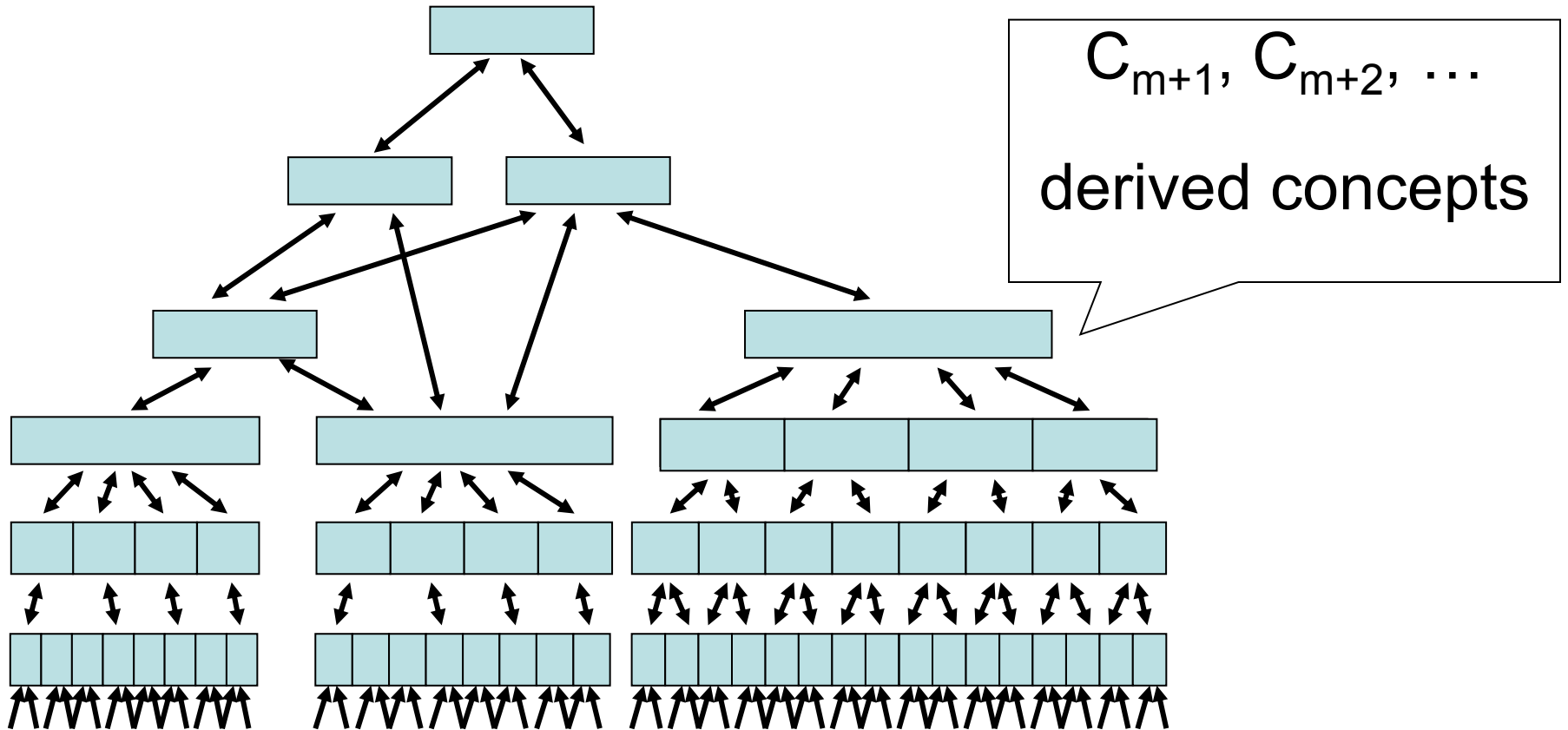
Concept (closed
construction)
higher order object

reference in the real world at a specific time

PHYSICAL WORLD

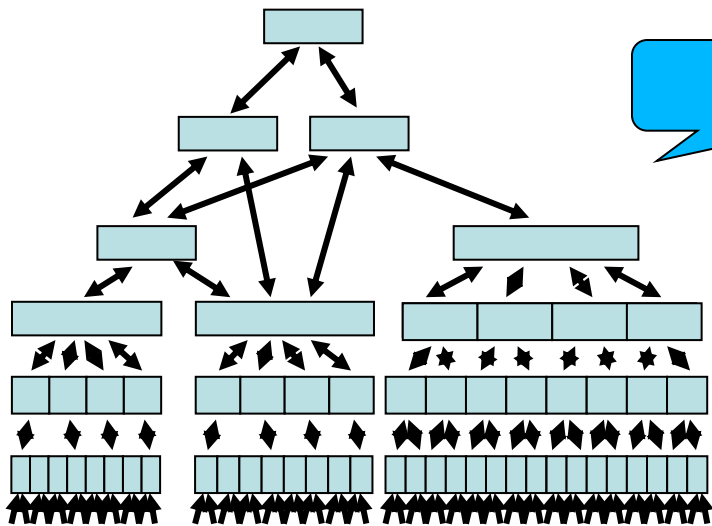


Conceptual system



C_1, \dots, C_m --- simple concepts (primitive concepts)

Relativity of primitive/derived



DOMAIN 1

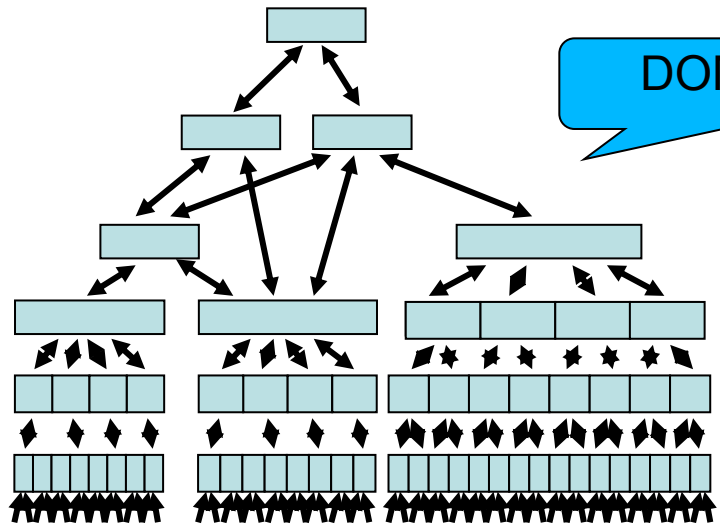
What is primitive within one domain can be derived in other domain.

What is primitive for an expert can be derived for a beginner.

primitive concepts

If we see in a way a similarity, we use the same, already known, words.

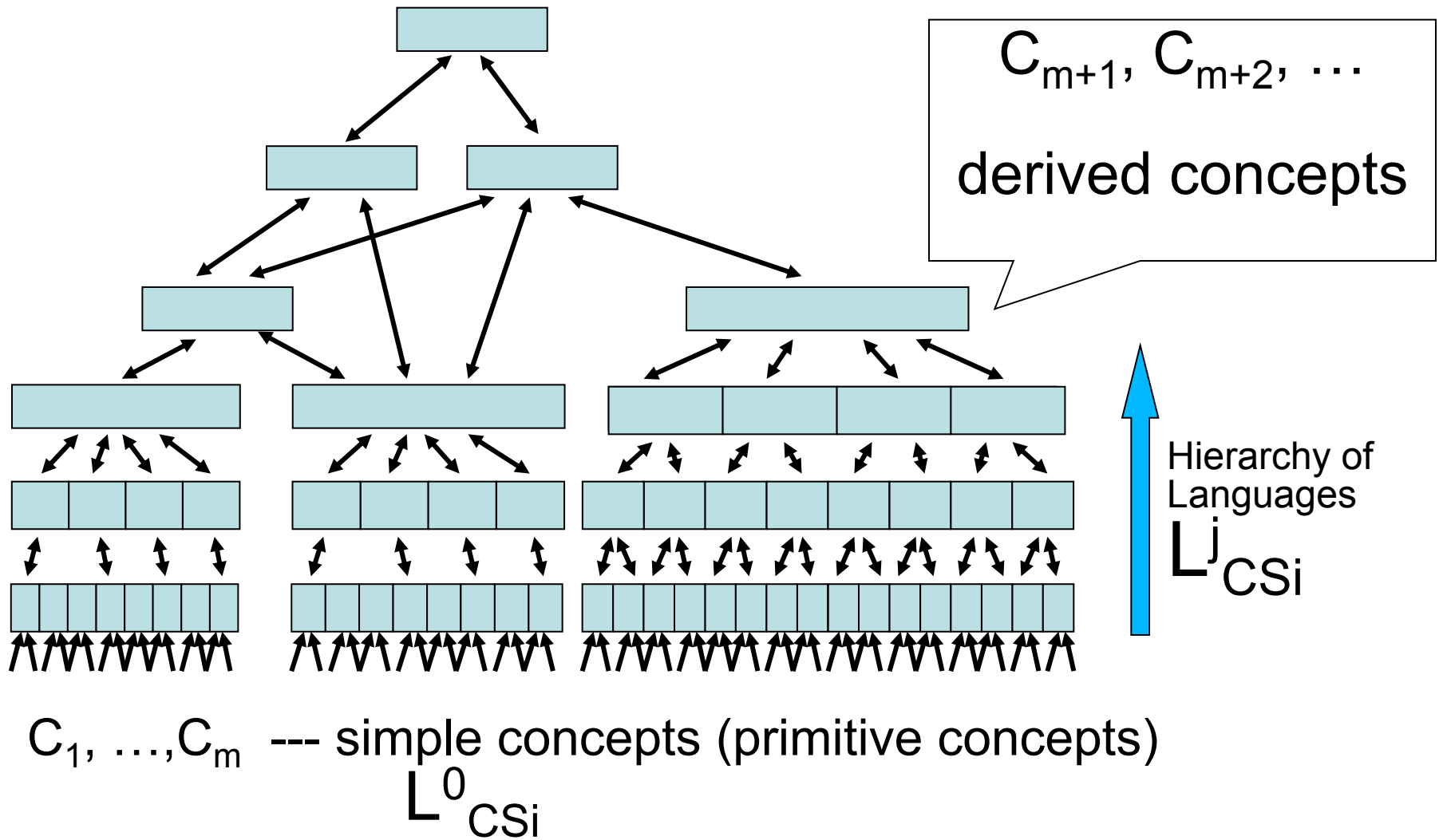
The **nouns are variables** “of type α ” from the ramified hierarchy.



DOMAIN 2

primitive concepts

Conceptual system and its hierarchy of languages



- NL is not something completed with fixed meanings of particular expressions.
- NL is a pool of particular hierarchies of languages over conceptual systems defining particular domains.
- NL is a dynamic phenomenon which develops continually.

Semantics of NL expressions is not something which exist as a static phenomenon;

Semantics is a mapping:

(NL expressions, Domain) \rightarrow Concepts

Semantics is created within the domain cognition process !

Basic level in hierarchical taxonomy (once more)

Superior level	animal	furniture
Basic level	dog	chair
Subordinate level	Labrador	rocking chair

- Our perception is anchored by our human dimension
- The reason of this can be explained by:
- Enactive Perception: “a dialogue” of the perceiving person and the environment/neighbourhood

Prototype Categorizing - result

- Select the ideal representative item
- (This is the “invariant representation” of what we call “category”)
- Express the fact that item I belongs to category \mathbf{C} by connection $(I, \mathbf{C}) \rightarrow \mathbf{c}$
 - where \mathbf{c} is measure of certainty
- Category is no more seen as a **container**, but it is seen as a **hook (central member = prototype)** to which we hang up other members

The "T"

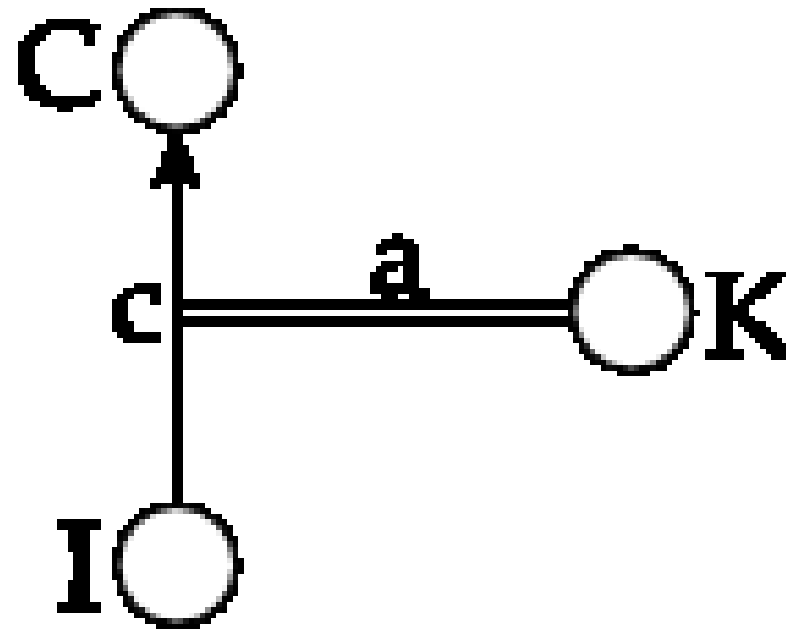
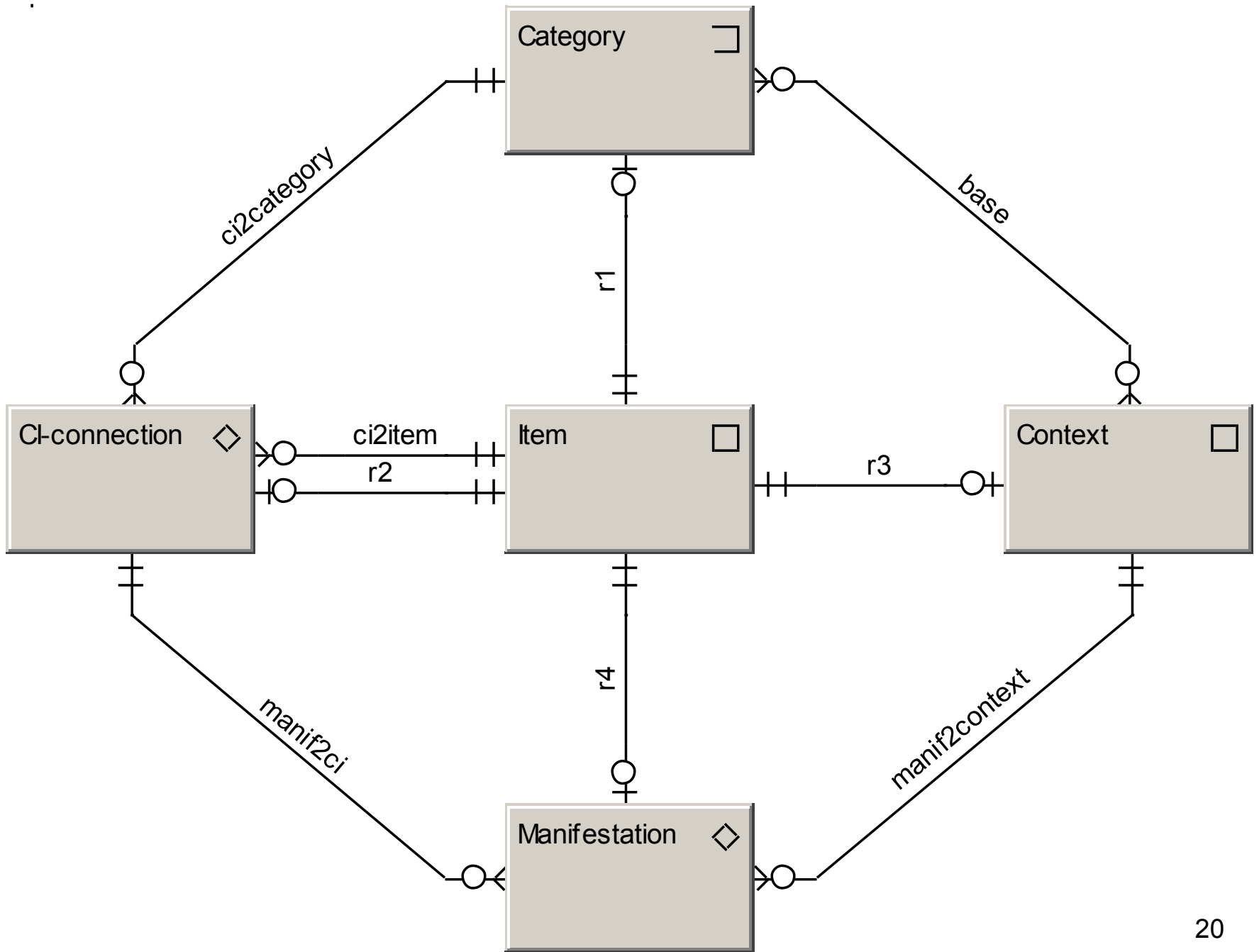
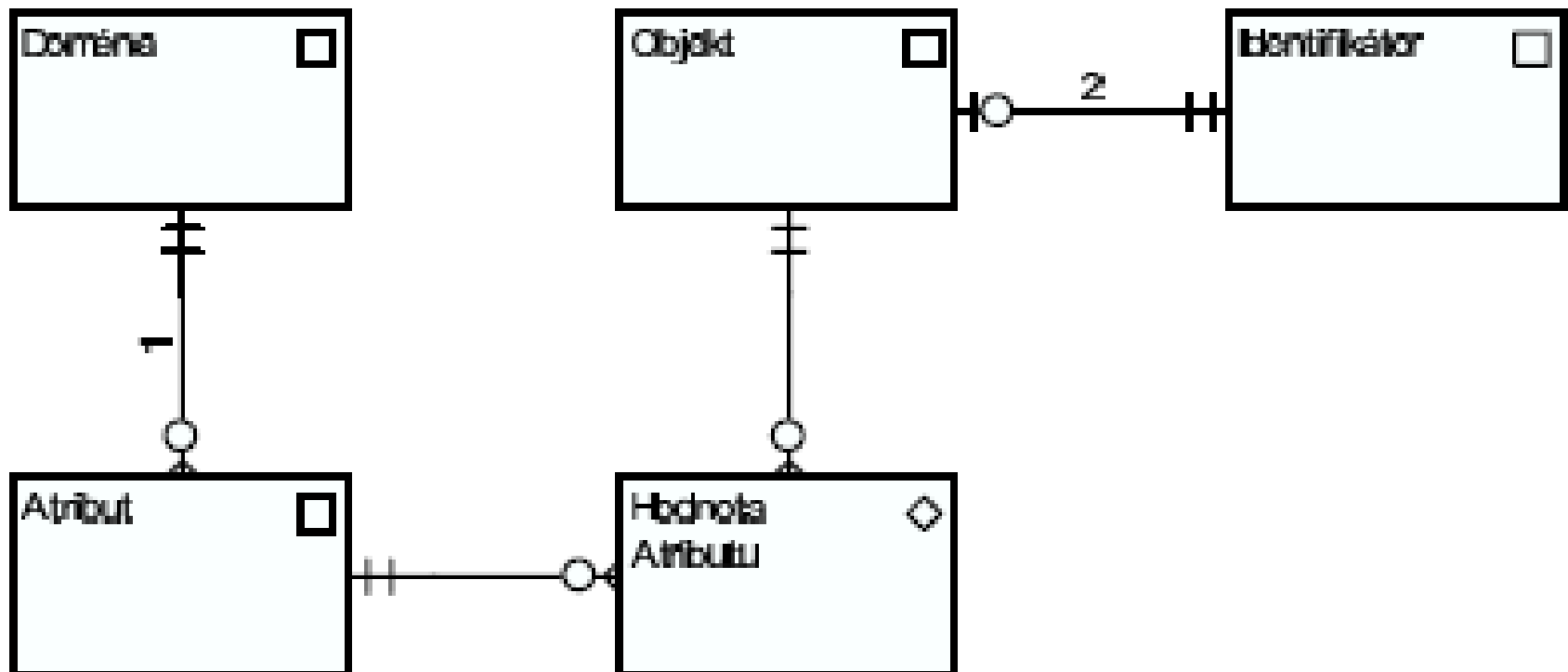


Fig. 1: T (elementary belief)



Data model of the Molten Objects data scheme



Connection oriented approach

- The basics is: we store instances of relationships not instances of previously determined complexes in a form of tables (from the beginning fixed)
- **Principle of connection based perception of models**
*Seeing a model of anything in a form of graph and thinking on this model we focus **primarily on relationships (graph edges)** not on objects (graph nodes).*
- Let's compare this with HIT method !!!

Specification of ADT (1)

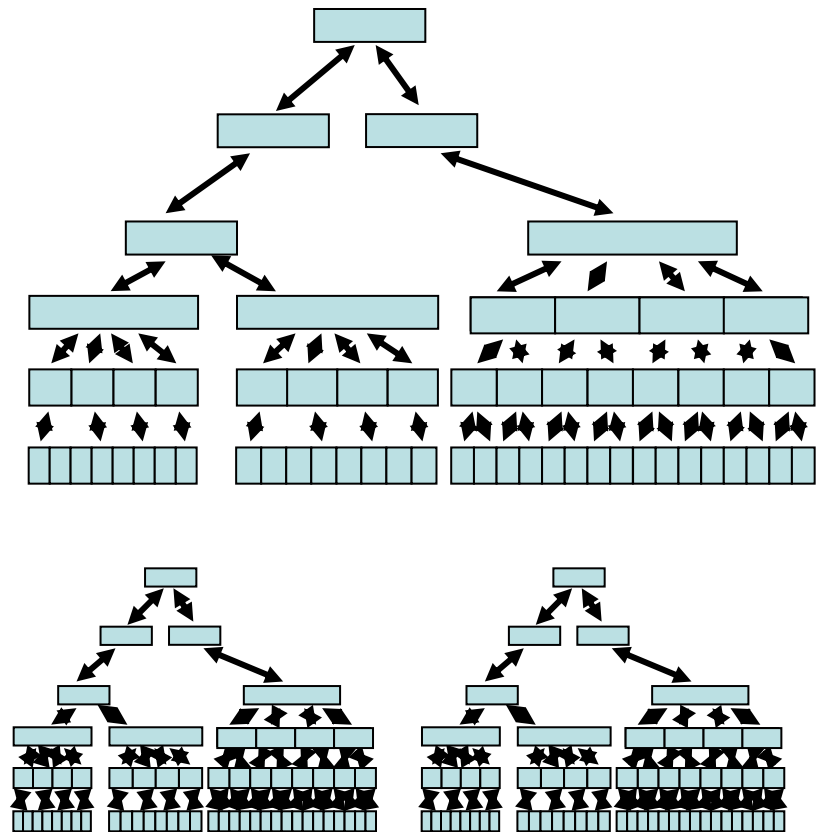
- Types
 - G
 - $STACK[G]$
- Functions
 - $put: STACK[G] \times G \rightarrow STACK[G]$
 - $remove: STACK[G] \rightarrow STACK[G]$
 - $item: STACK[G] \rightarrow G$
 - $empty: STACK[G] \rightarrow BOOL$
 - $new: _ \rightarrow STACK[G]$
- Axioms
- Conditions

Specification of ADT (2)

- Types
 - G
 - STACK [G]
- Functions
- Axioms
 - For any $x::G$, $s::\text{STACK}[G]$
 - A1. $\text{item}(\text{put}(s,x))=x$
 - A2. $\text{remove}(\text{put}(s,x))=s$
 - A3. $\text{empty}(\text{new})$
 - A4. **not** $\text{empty}(\text{put}(s,x))$
- Conditions

OOP = one of the best inventions

- ...an intelligent use of the “Fundamental hierarchy”
- ... a reverse process to “Breakdown structures”
- ... in a way a simulation of the natural process of “cognition by creation”



The Class in an OO analysis

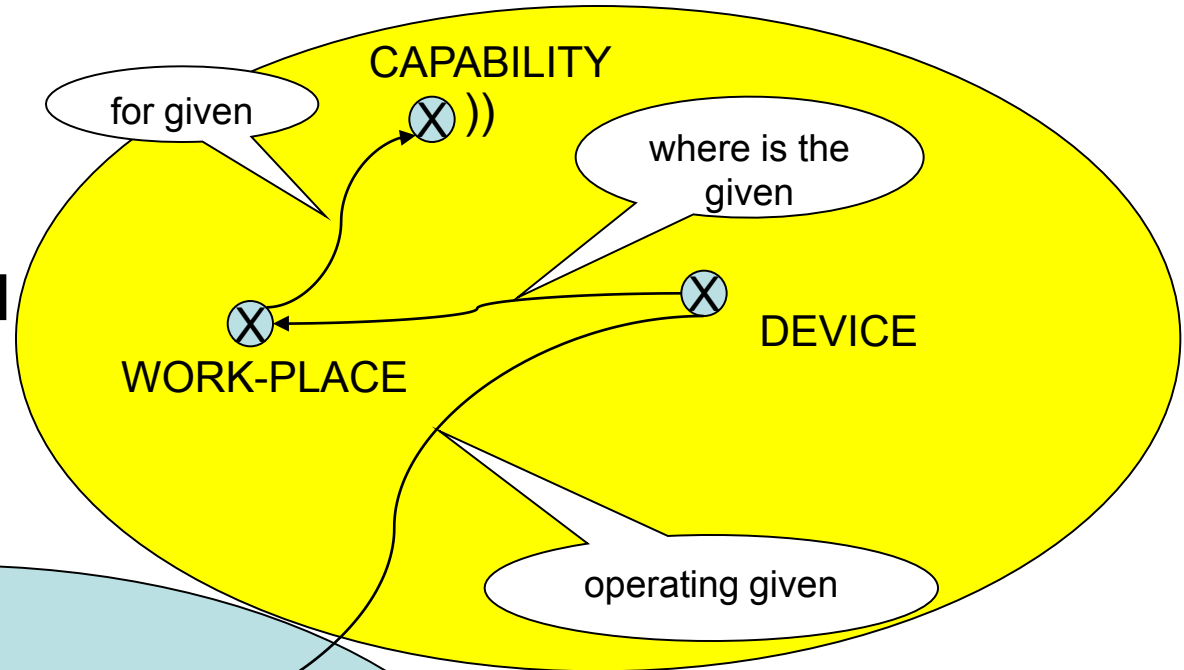
- **Class** defines “the shape” of its instances
- Each **object**, which is worth focusing attention, must be assigned to a **class**.
- Improper **class** selection “today” can cause big problems “tomorrow”!
- Situation change in real world causes necessity to change the assignment of some objects to classes, or to change the class structure design.

Issues of OOP (1)

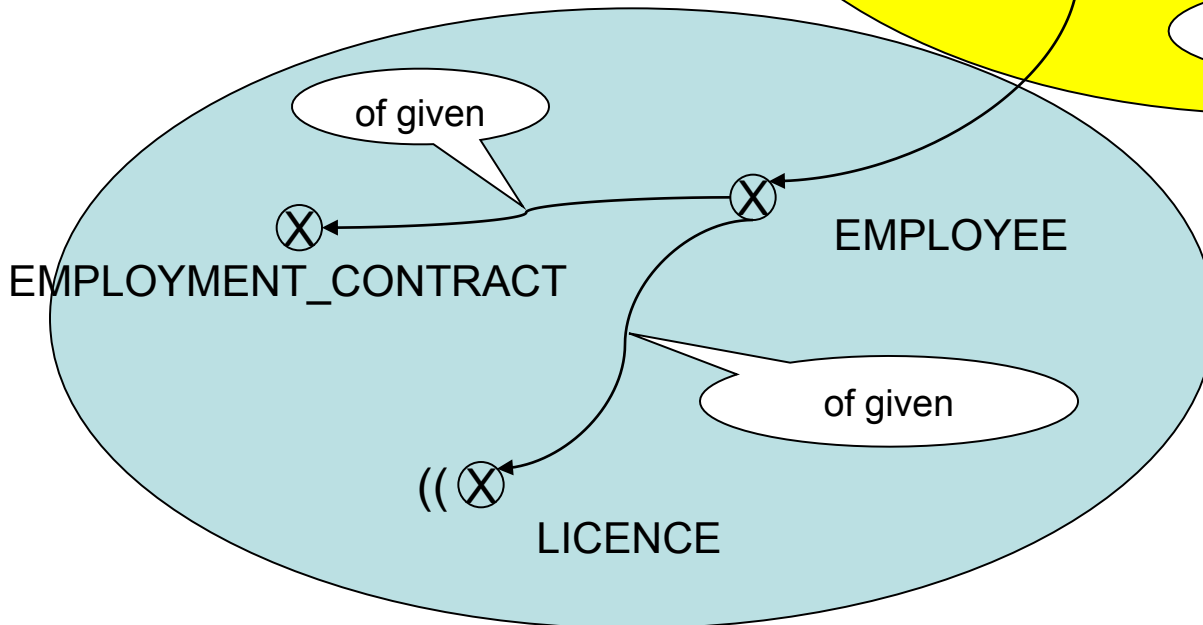
- ***Object Oriented Paradigm*** works well in the realm for which it was originally developed.
- This is **Programming**.
- **The realm of “artifacts” creating.**
- To mirror a realm of continual changes, improvements and developments doesn't fit to OOP very well.
- This is **Business Systems analysis and specification.**

Communication through interface

PRODUCTION

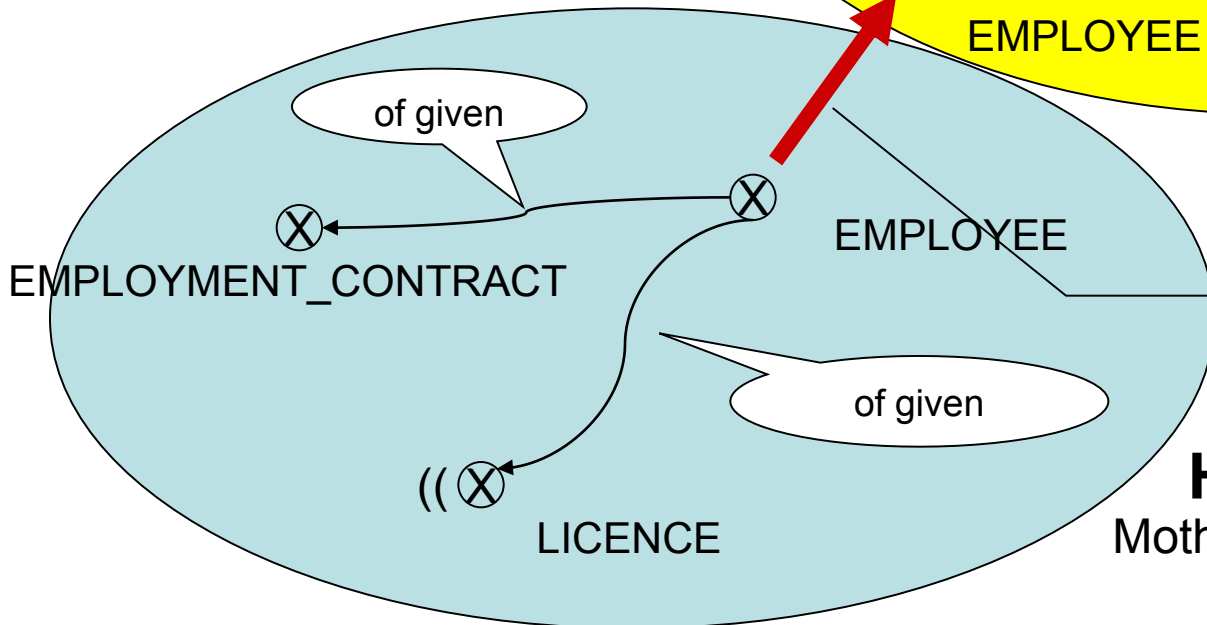
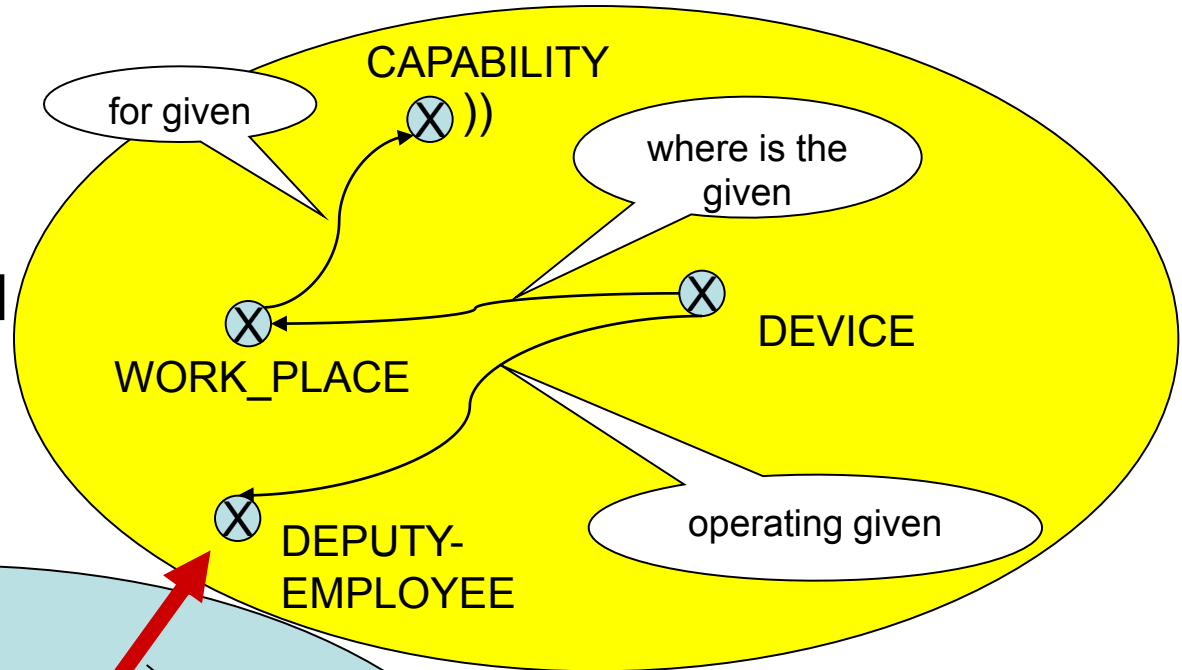


HRM



Communication via "Deputies"

PRODUCTION

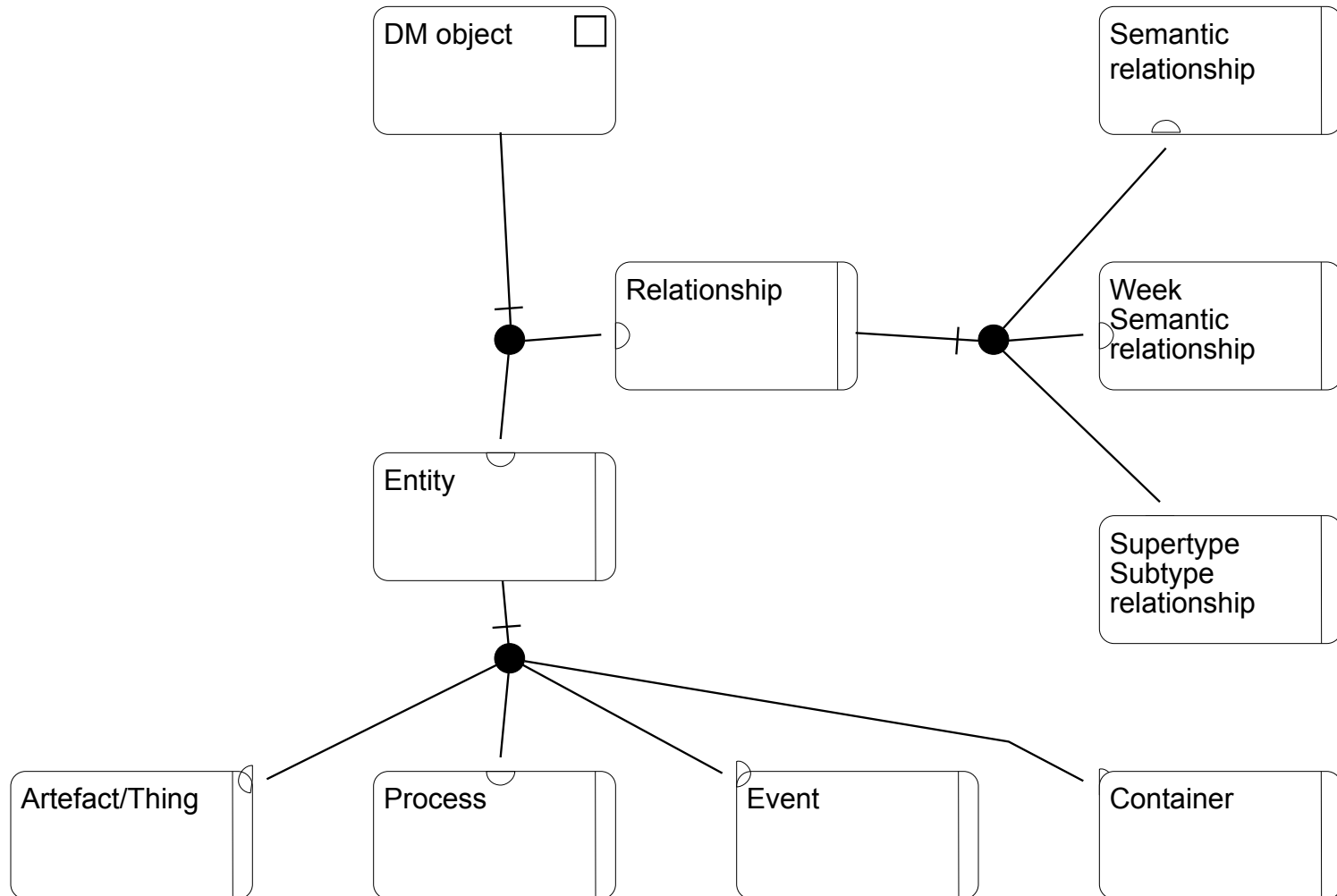


Usually asynchronous update

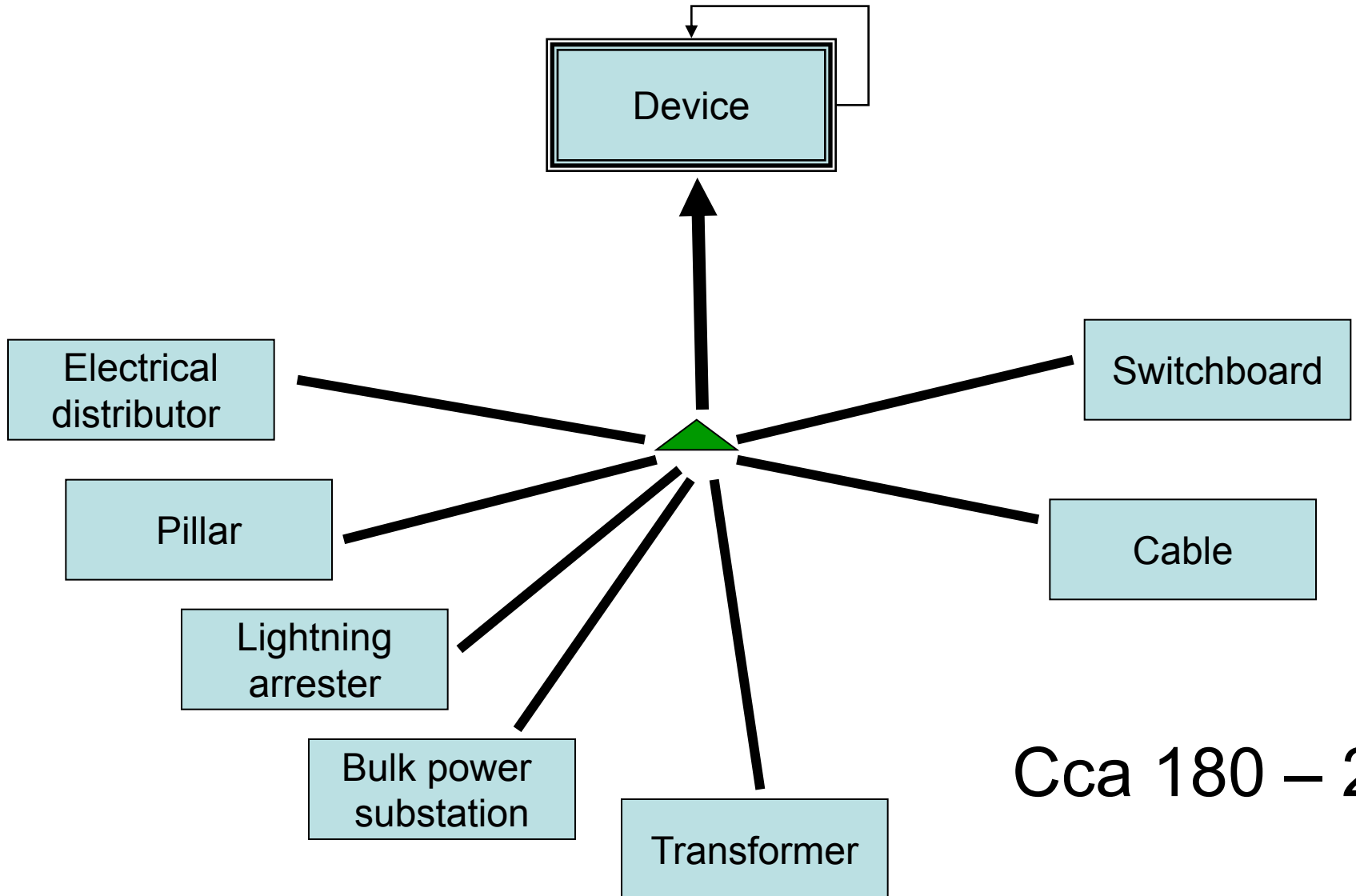
HRM
Mother component for EMPLOYEE

Categorization of DM objects

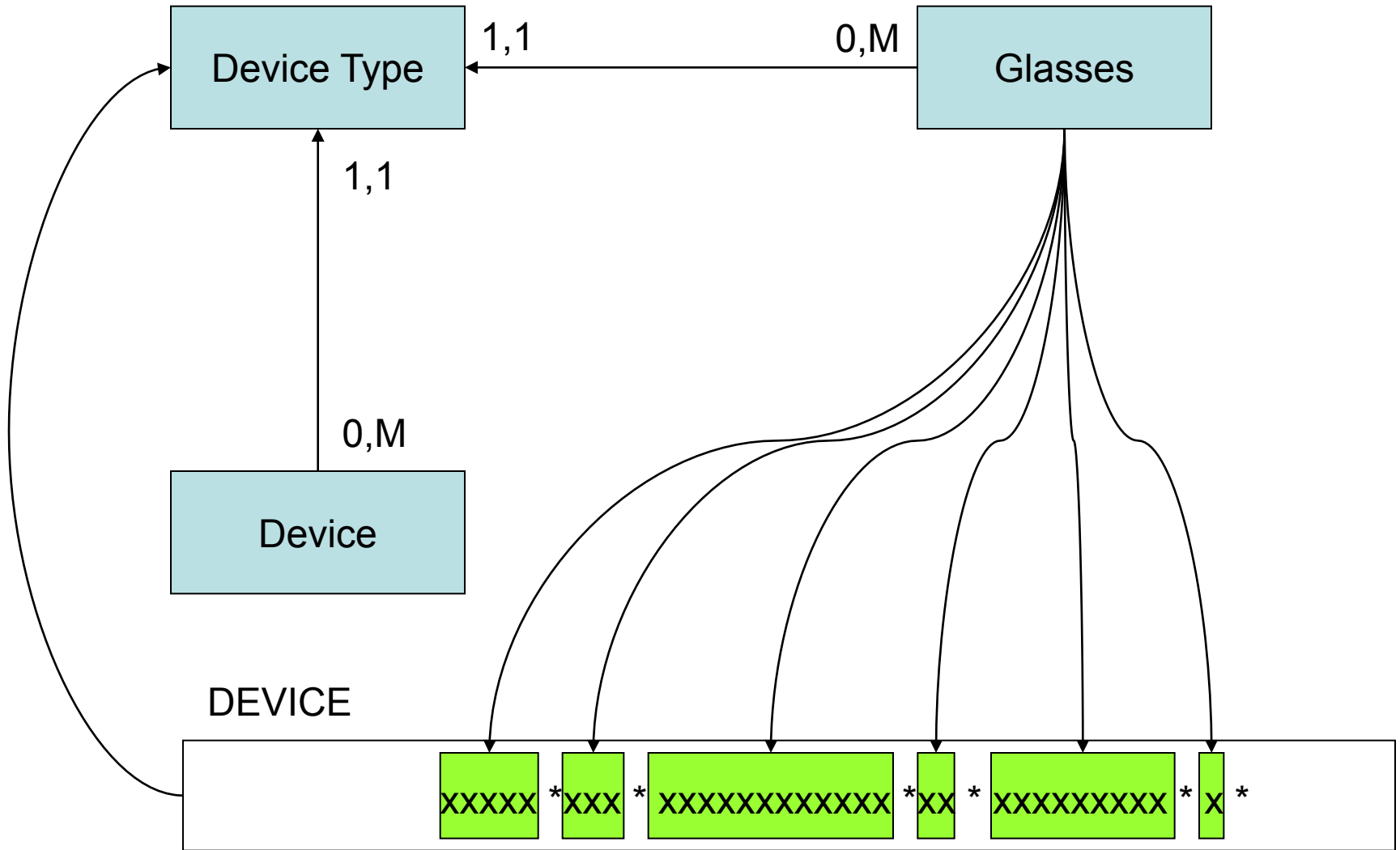
The root of the Tree



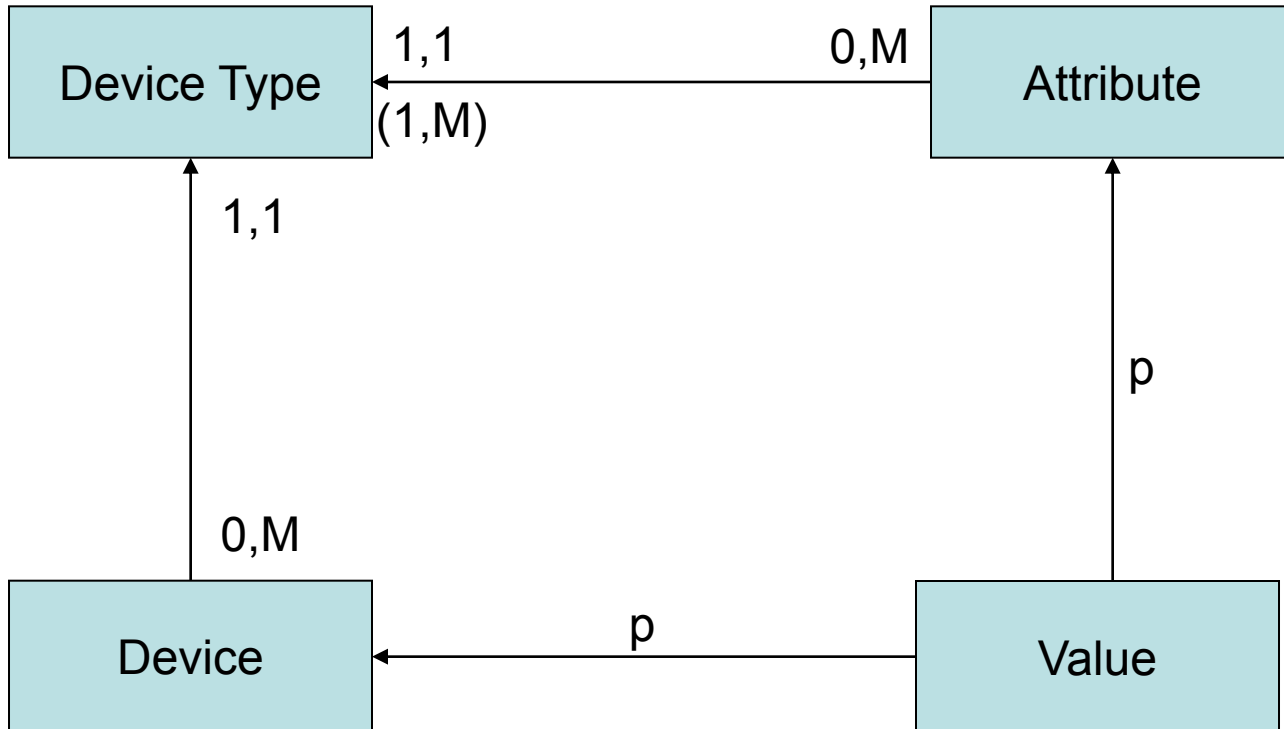
Connections between Supertypes



Using *glasses*

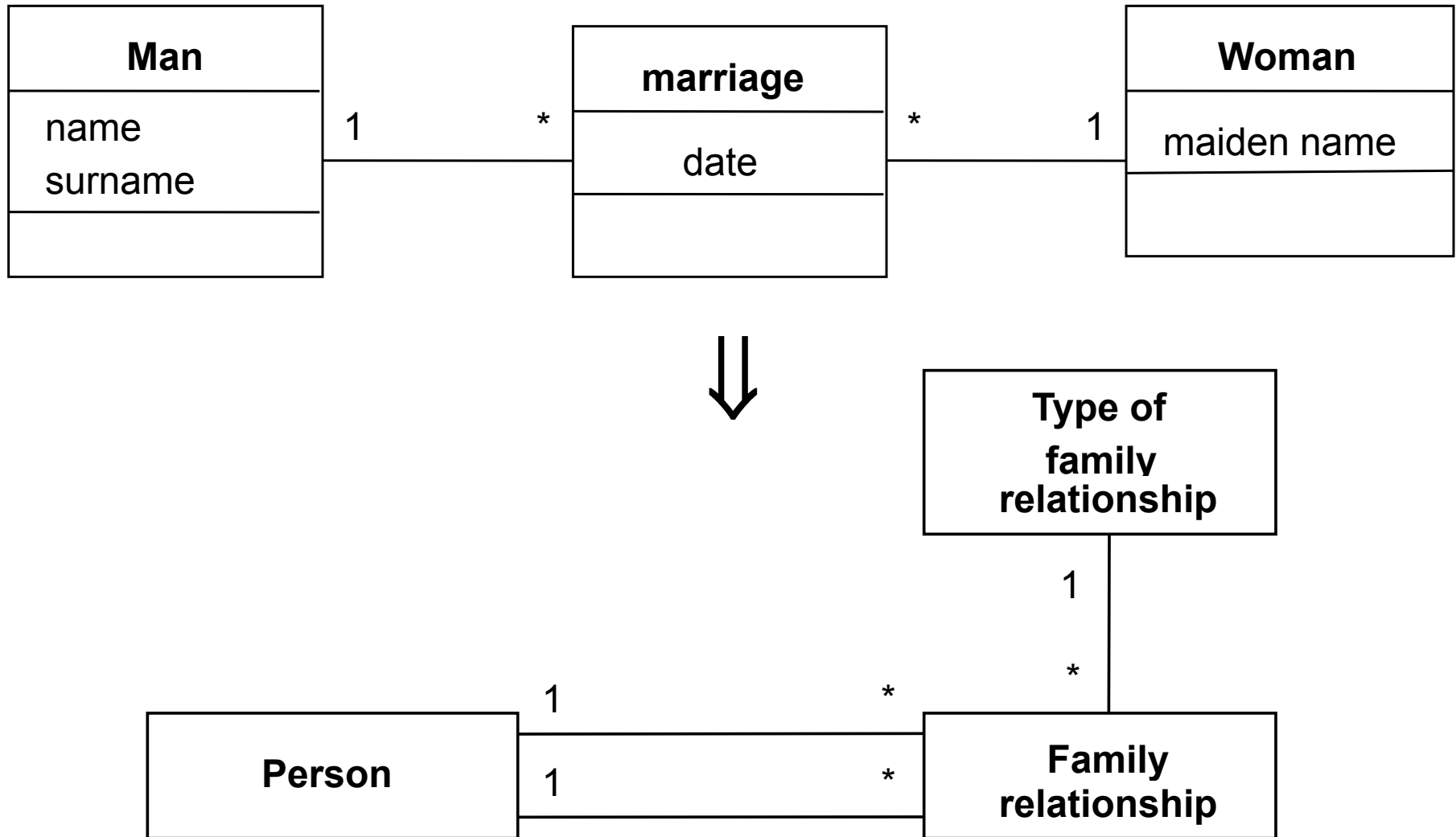


Using attribute as isolated entity



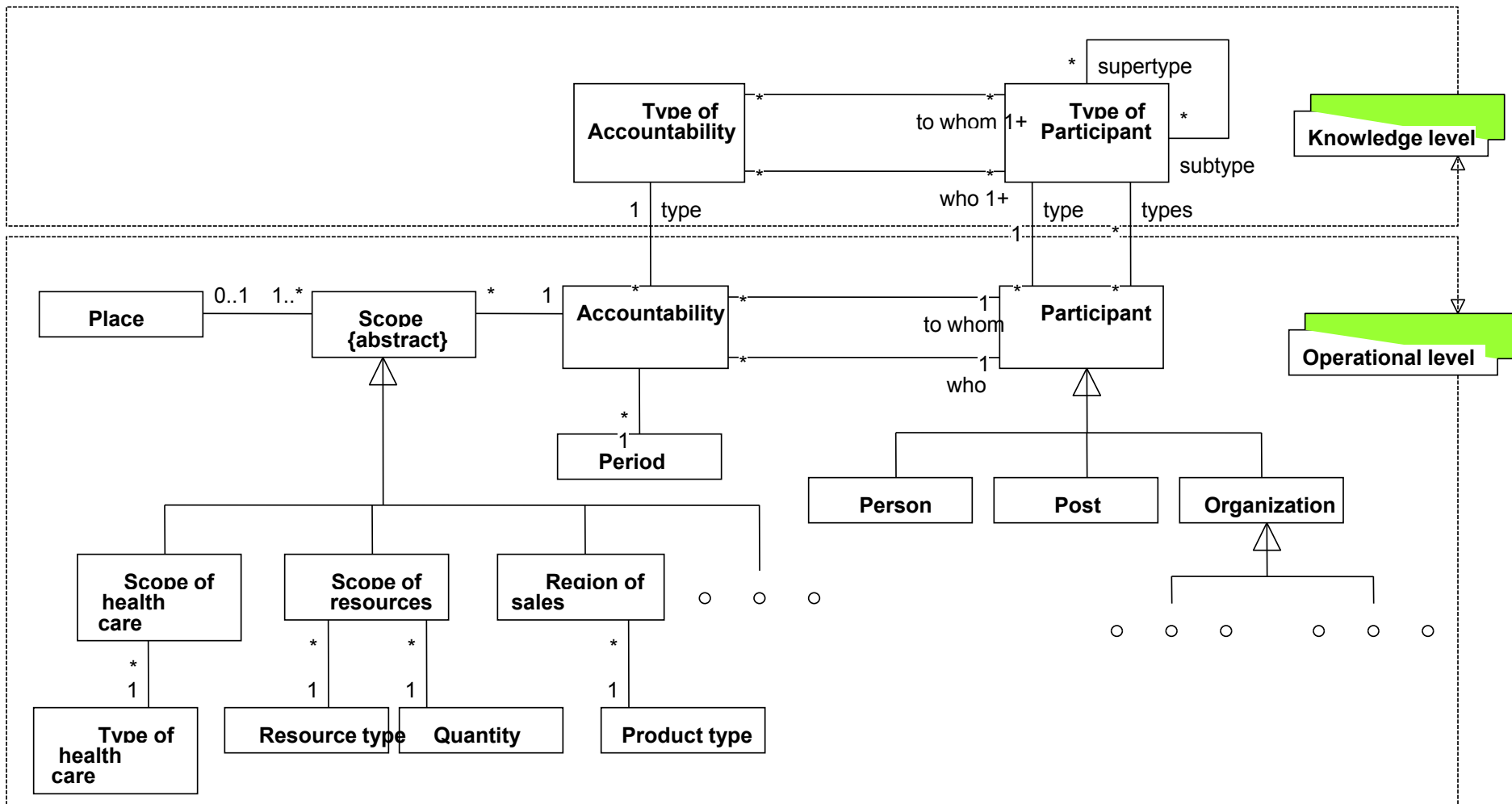
(Value) of given (#Attribute) for given (#Device) / 0,1:0,M

Abstraction of relationships

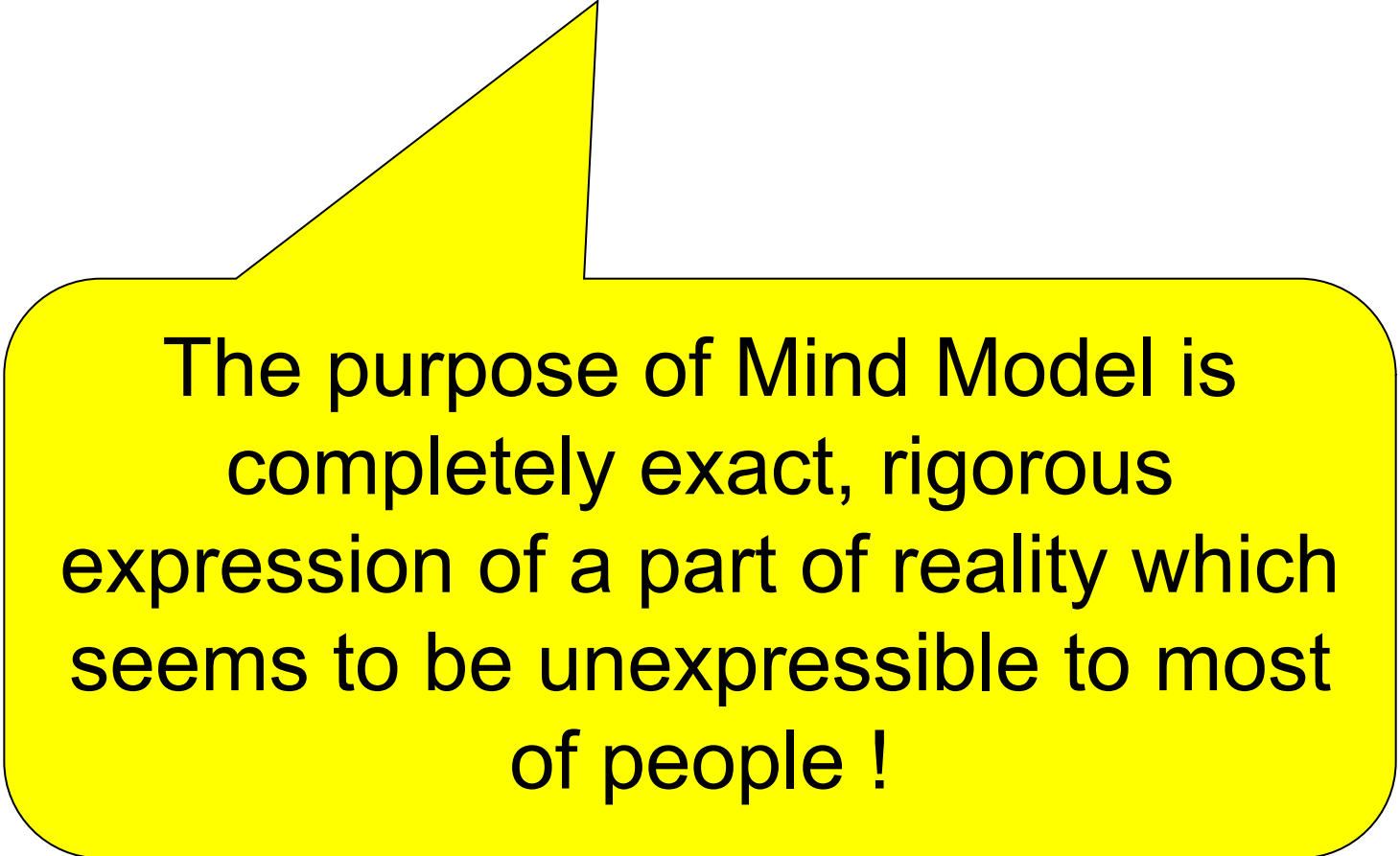


Analytic pattern Accountability

by Lubor Šešera, DATASEM'99



What is the content of the model
or
what the model expresses



The purpose of Mind Model is
completely exact, rigorous
expression of a part of reality which
seems to be unexpressible to most
of people !