

# Concepts, Conceptual Systems, Definitions.

PA116 – L11

(c) Zdenko Staníček, Sept 2010, Nov 2012



INVESTMENTS IN EDUCATION DEVELOPMENT

# Topics

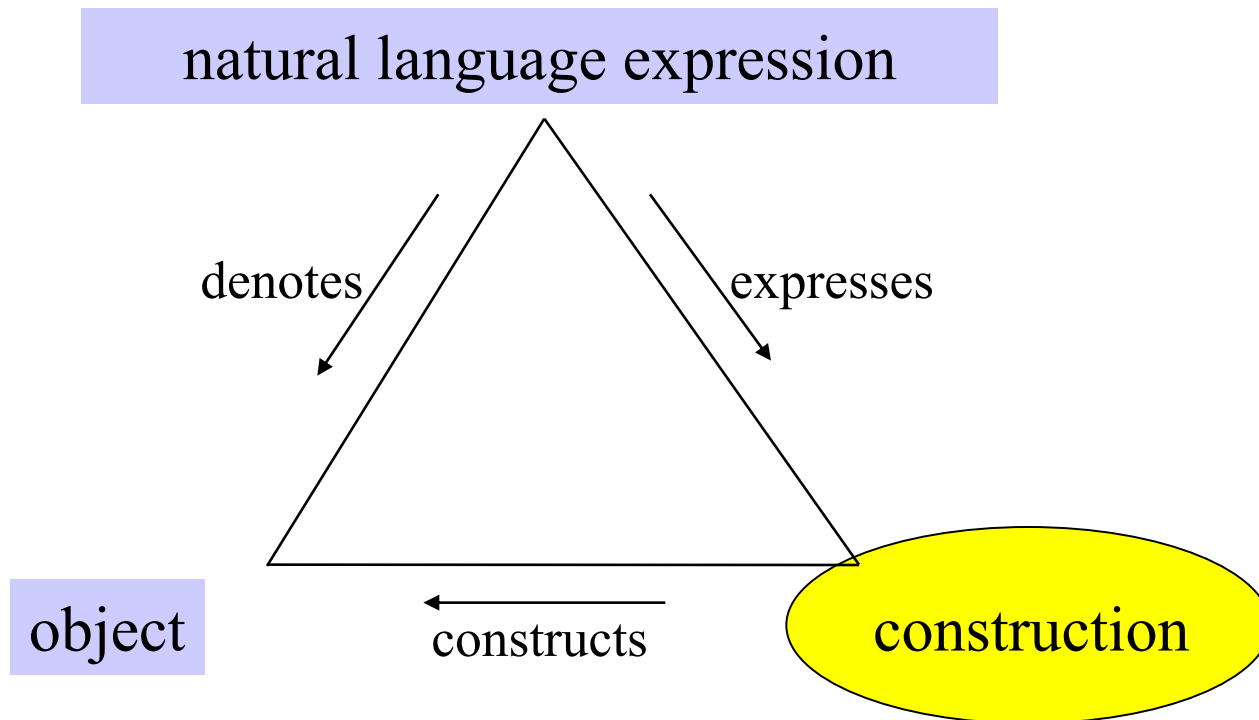
- Motivation
- Concepts
- Higher order objects
- Definition – what actually is it?
- Linguistic definition
- Conceptual systems
- What means DUM

# Motivation

- What is “concept”?
- problems of communications among various professional cultures
- chaos in the concept of “concept”
- concept is a class ... (most computer scientists)
- concept is a set of properties ...
- **concept** is an independently identifiable structured construct composed of knowledge primitives and/or other concepts ... (H. Kangassalo, The Finish School)

# Corrected Triangle of Denotation

correction of Frege-Church's "expression denotes the reference" and "expression expresses the sense"



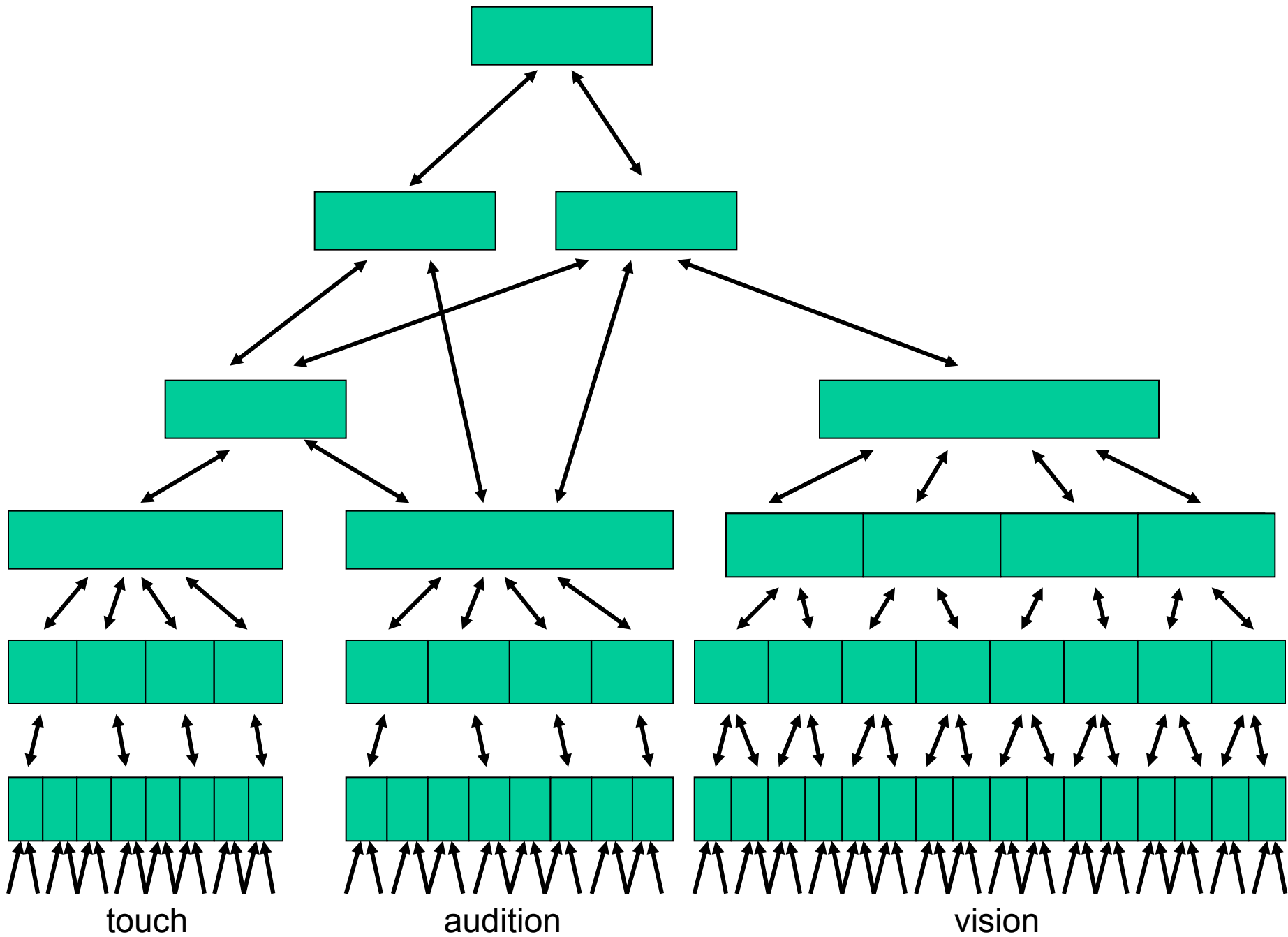
# Basic (commonly shared) intuition

(Platon, Bolzano, Frege, Church, ...)

- Concepts are “***identification procedures***” allowing common shared understanding of what we actually talk about
- Concepts are ***mental procedures*** our mind has to execute to identify an object which is focused on

# Jeff Hawkins' Hypotheses (in his book: "On Intelligence" )

- Neocortex stores sequences of patterns
- Neocortex stores patterns in an invariant form (invariant representation)
- A sequence of pattern is a pattern again
- Neocortex stores patterns in a Hierarchy
- Neocortex doesn't distinguish between external inputs and internal inputs
- Thus:  
**Concepts are invariant representations of sequences of (elementary) patterns.**



touch

audition

vision

# Example of “concepts”/ expressions pairs

- (1) Prime number  
(2) Natural number which has exactly two divisors
- (1) equilateral triangle  
(2) equiangular triangle
- (1) sons of the heir to the throne  
(2) men whose father is the heir to the throne
- (1) and (2) identify the same object anyway
- (1) and (2) are always different identification procedures  
(we have to invest intellectual endeavour to recognize  
that the given expressions identify the same object)



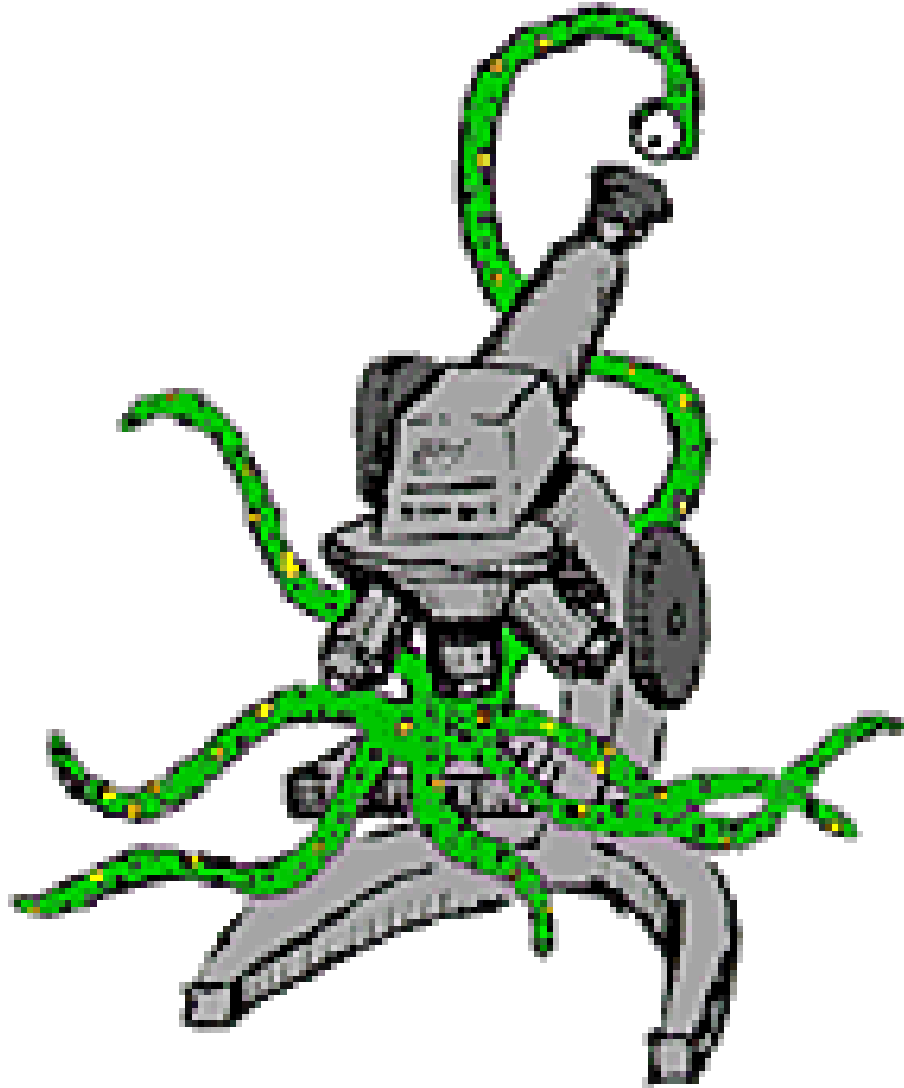
# Are there Objects in constructions?

- Let's have  
 $\lambda x(\sin x)$ ,  $x$  real number,  $\sin: \mathbb{R} \rightarrow \mathbb{R}$   
 $\lambda x(\text{Salary } x)$ ,  $x::\text{EMPL}$ ,  $\text{Salary}: \text{EMPL} \rightarrow \mathbb{R}$
- is  $\sin$ , or  $\text{Salary}$ , object or construction?
- according to the definition they are subconstructions
- $\sin$ ,  $\text{Salary}$  ... are atomic constructions
- It is necessary to separate precisely the world of objects from the world of their constructions, as we have no objects in our Memory – therefore we need **trivialisation**:
- **Let  $X$  be any object (or construction), then trivialisation  ${}^0X$  constructs just  $X$  without any change.**
- it is about elaboration and expansion of the definition of atomic construction
- Trivialisation constructs objects simply by pointing them.

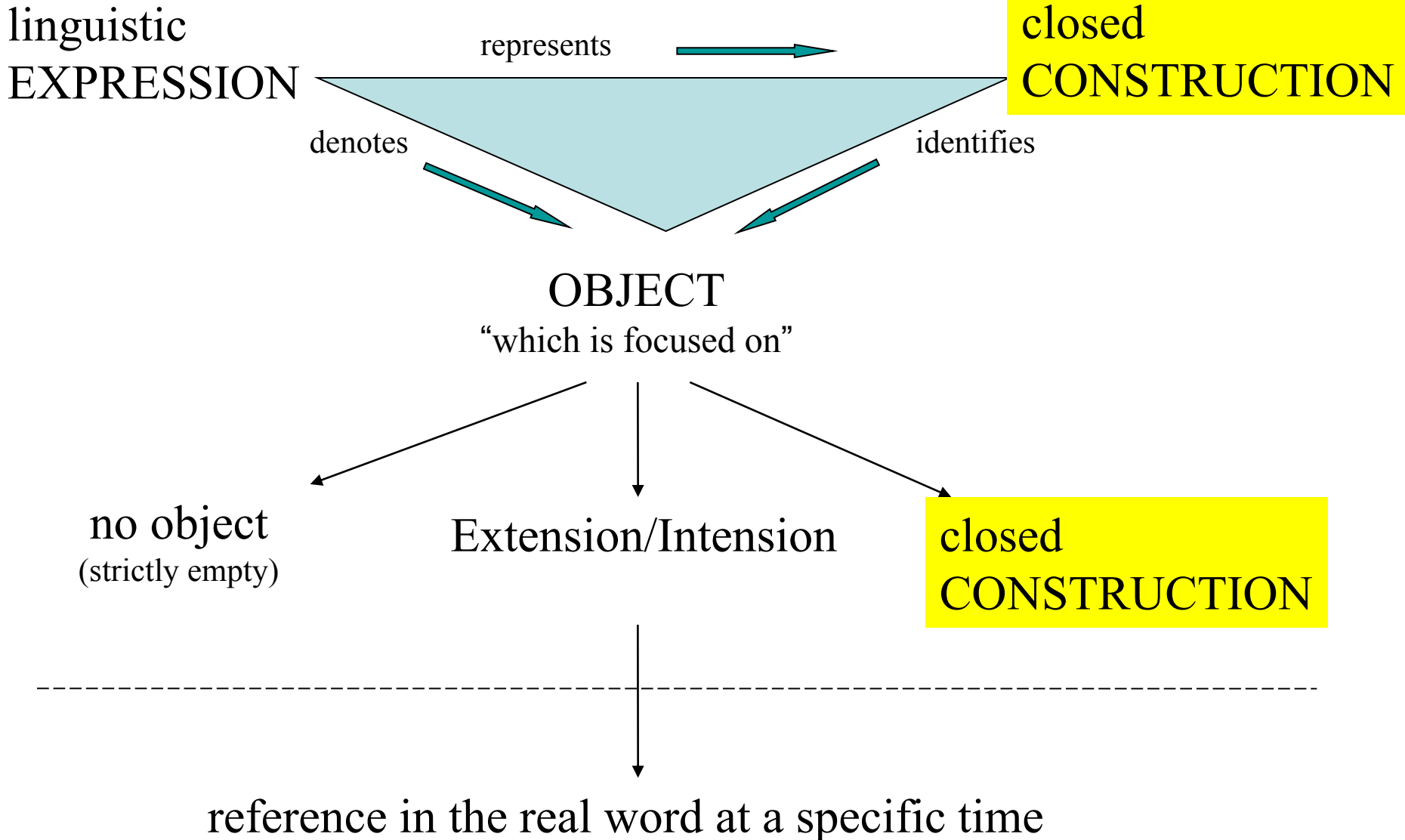
# Self-reference “on the stage”

- TIL with simple theory of types:
  - constructions are out of the focus of our research
  - constructions are only tools for a research of the World-of-objects
- What if we focus on a construction as an object of interest?
- Remember: Neocortex doesn't distinguish between external inputs and internal inputs

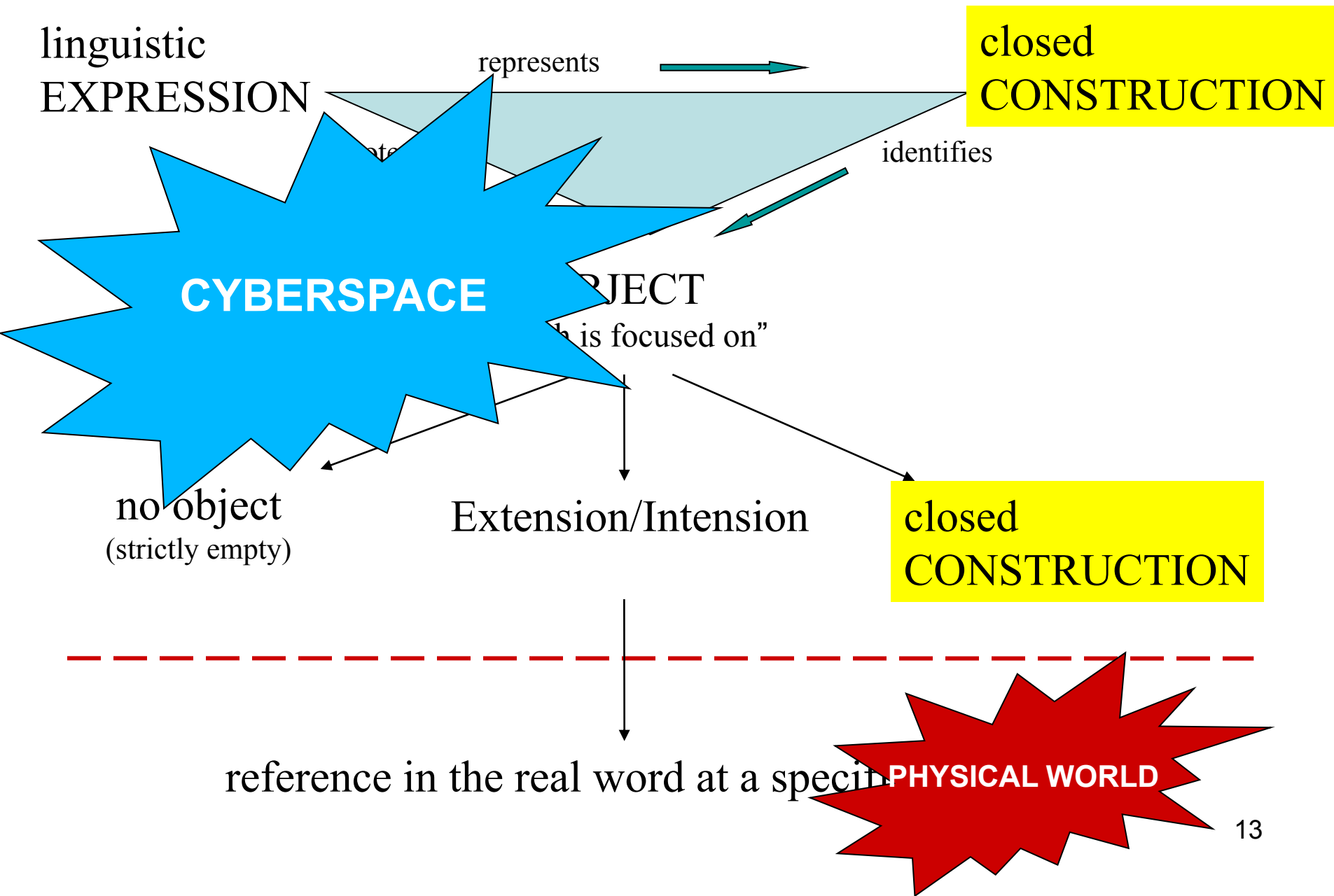
# Self-reference



# Self-reference in a scheme



# Self-reference in a scheme



# Consequences

- Bertrand Russell:  
*Ramified Theory of Types*
- Pavel Tichý:  
“... consistent logic theory is not possible to build on simple theory of types ...”
- **Constructions need to be included into observed objects of interest ...**
- ... and what about constructions of constructions ...?

# Constructions: an overview and **extensions** (1)

- **Trivial construction** – trivialisation  ${}^0X$   
object  $X$  is identified by itself (is constructed by pointing on it);  
external input pattern to our neocortex (or to **Memory**)
- **Atomic construction** – **variable**  
it constructs an object of a given type depending on a total  
function from variables to objects (called valuation)
- **Execution construction**  ${}^1X$   
Using  $X$  as construction, execution constructs what  $X$   
constructs. If  $X$  is not a construction, using  $X$  does not  
construct anything – we say that the construction is  $v$ -  
improper for any valuation  $v$ .

# Constructions – overview (2)

- **Double-execution construction  ${}^2X$**   
Using result of the execution of  $X$  as construction, the double-execution constructs what result of the execution of  $X$  constructs. If  $X$  is not a construction or  $X$  does not construct construction, double-execution of  $X$  does not construct anything – we say that the construction is  $v$ -improper for any valuation  $v$ .



# Constructions – overview (3)

- N-tuple or sequence construction  
 $(X_1, \dots, X_n)$   
... see DM1 or referred study materials
- Projection construction  
 $X_{(i)}$   
... see DM1 or referred study materials

# Constructions – overview (4)

- Abstraction construction – closure

$\lambda x_1 \dots x_n (X)$

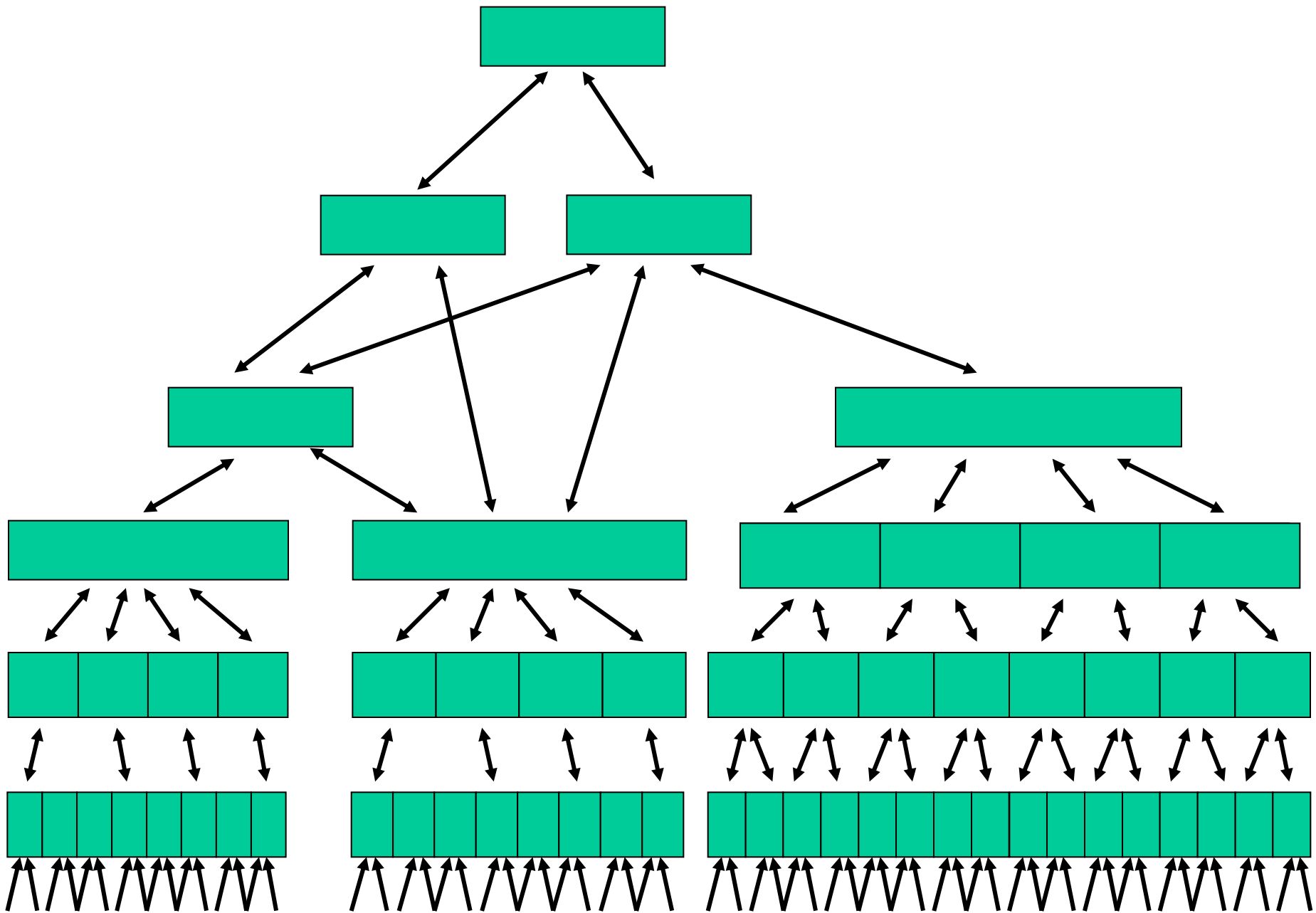
... see DM1 or referred study materials

- Application construction – composition

$[X X_1 \dots X_n]$

... see DM1 or referred study materials

- This is a complete apparatus of constructions which allows to describe also a self-reference (and what happens in the Fundamental Hierarchy)
- Construction is unable to be mentioned as object of interest in Simple Theory of Types; Ramified Theory of Types is needed !



# Definition of types above EB

Ramified theory of types is built above used cognitive base, i.e. above **EB** introduced in Simple Theory of Types. Following definition contains double nested inductive sub-definition.

$T_1$  (induction precondition):

every simple type above **EB** is type of order 1

$C_n$  (constructions of order  $n$ ):

describes constructions of order  $n$ ; they constructs objects of order  $n$

$T_{n+1}$  (inductive step – types of order  $n+1$ ):

inductively describes what types of order  $n+1$  are

# $C_n$ (constructions of order $n$ ):

Let  $\alpha$  be a type of order  $n$ . Then

1. every variable which  $v$ -constructs  $\alpha$ -objects is a construction of order  $n$
2. if  $X$  is  $\alpha$ -object, then  ${}^0X$  is a construction of order  $n$
3. Let  $X_1, \dots, X_n$  be constructions of order  $n$ . Then  $(X_1, \dots, X_n)$  is construction of order  $n$ .
4. Let  $X$  be construction of order  $n$  which constructs objects of the type  $(T_1, \dots, T_n)$ . Then for all  $i = 1, \dots, n$ ,  $X_{(i)}$  is a construction of order  $n$ .
5. Let  $x_1, \dots, x_n, X$  be constructions of order  $n$ , let  $x_i$  be variables. Then  $\lambda x_1 \dots x_n (X)$  is construction of order  $n$ .
6. Let  $X, X_1, \dots, X_n$  be constructions of order  $n$ . Then  $[XX_1 \dots X_n]$  is construction of order  $n$

# $T_{n+1}$ (types of order $n+1$ ):

Let  $\text{Cons}^n$  be a class of all constructions of order  $n$ .

- $\text{Cons}^n$  and every type of order  $n$  is a type of order  $n+1$ .
- Let  $\beta_1, \dots, \beta_m$  be types of order  $n+1$ . Then Cartesian product  $\beta_1 \times \dots \times \beta_m$ , denoted  $(\beta_1, \dots, \beta_m)$ , is a type of order  $n+1$ .
- Let  $\alpha, \beta_1, \dots, \beta_m$  be types of order  $n+1$ . Then  $((\beta_1, \dots, \beta_m) \rightarrow \alpha)$ , i.e. a class of all partial functions from  $\beta_1 \times \dots \times \beta_m$  to  $\alpha$ , is a type of order  $n+1$ .
- Nothing but what is constructed according to steps 1 through 3 is a type of order  $n+1$ .

(see definition of type within Simple Theory of Types in the study materials)

# Higher order objects

- If  $\alpha$  is a type of order  $n$ , then every  $X \in \alpha$  is an object of order  $n$ .
- Objects of order  $n$ , where  $n > 1$ , are called ***higher order objects***.
- ***The class of all constructions of objects of order  $n$  composes a type of order  $n+1$ .***
- Every ***construction of order  $n$***  is an ***object of order  $n+1$ .***

# ... And for what is it useful?

- Higher order objects and their constructions, especially trivialisation, provide the inspiration for MENTION and USE of connections, categories, operations and rules which are fundamental constructs for modeling.
- Knowledge is principally recursive !
- To understand (or to model, to execute) a knowledge-intensive Service System



# Higher order objects

- Order of an object depends on what the object constructs.
- Constructions of order 1 construct objects of types of order 1, so they belong to types of order 2 and they are objects of order 2.
- Constructions of order 2 construct objects of types of order 2, so they belong to types of order 3 and they are objects of order 3.
- etc.
  
- No construction can be found among objects of order 1.
- Simple Theory of Types deals only with objects of order 1 and works with type hierarchy which is infinite but still in limits of order 1.
- Ramified Theory of Types includes constructions as objects, thus it adds a new (infinite) hierarchy of orders.
- **Only a modern physics knows what elementary objects of the physical world are in itself. We have in our brains only constructions !!! What is interesting is: under influence of the same environment and situation these constructions are almost certainly the same !!!**

# Concepts – an approximation

- Based on the use of Ramified Theory of Types:
- **Concepts are closed constructions**

# Rules of $\lambda$ -calculus

- constructions  $\lambda x[{}^0\text{sin } x]$  and  $\lambda y[{}^0\text{sin } y]$   
are considered to be  
(quasi)identical

( $\alpha$  - rule)

- constructions  $[(\lambda x({}^0\text{sin } x)) {}^0\alpha]$  and  ${}^0\text{sin } {}^0\alpha$   
dttto

( $\beta$  - rule)

- constructions  ${}^0\text{Prime}$  and  $\lambda x[{}^0\text{Prime } x]$   
dttto

( $\beta$  - rule)

- constructions  ${}^02$  and  $\lambda x[{}^0= x{}^02]$   
dttto

( $\gamma$  - rule)

# Why this (quasi)identity ?

- When any two (quasi)identical closed constructions are USE-d as Constructions they give the same result or they both are improper.
- There is no reason to distinguish between two (quasi)identical close constructions when they are MENTION-ed as Objects.

# Quasi-identity

- Closed constructions  $c_1$  and  $c_2$  are quasi-identical if they are alpha- or beta- or gamma-equivalent. We say that  $c_1$  and  $c_2$  are in relation Quid.
- If (  $c_1$  Quid  $c_2$  ), then  $c_1$  constructs the same object as  $c_2$  , or both  $c_1$  and  $c_2$  are improper.
- **Lemma:** Quid is an equivalence relation.
- >>Prove it!

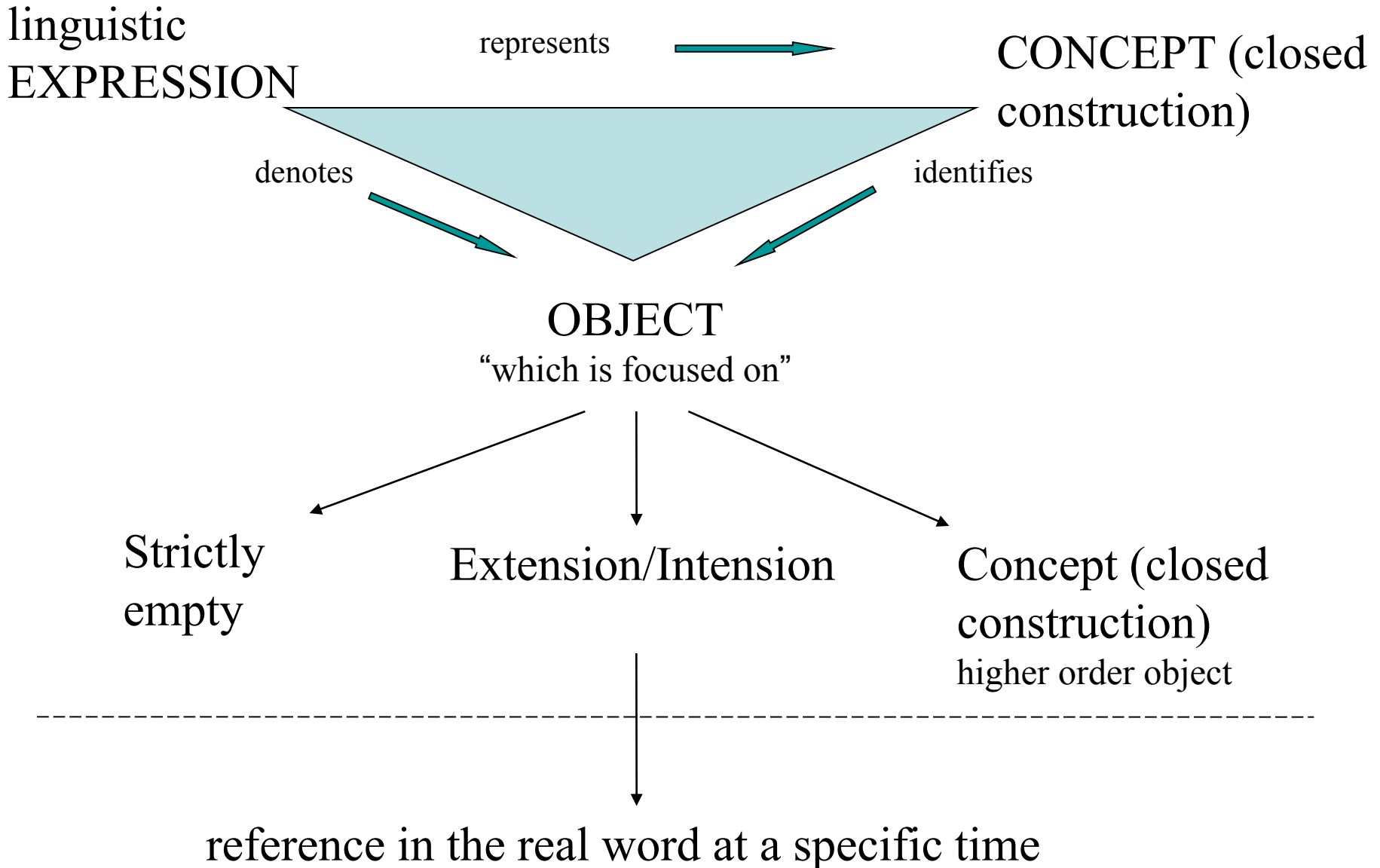
# Concepts

- **Concept is a closed construction belonging to a class of quasi-identical closed constructions.**

As constructions from a given class are not distinguished in the USE mode, we will not distinguish them in the MENTION mode.

- In Aleš Horák's dissertation, there is more precise definition based on canonical forms of constructions and sequences of quasi-identical constructions which are gradually derived from these forms. However, our definition is sufficient for our purpose.

# Essence of communication, understanding, and modeling



# Essence of communication, understanding, and modeling

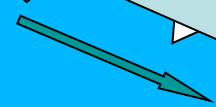
linguistic  
EXPRESSION

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**



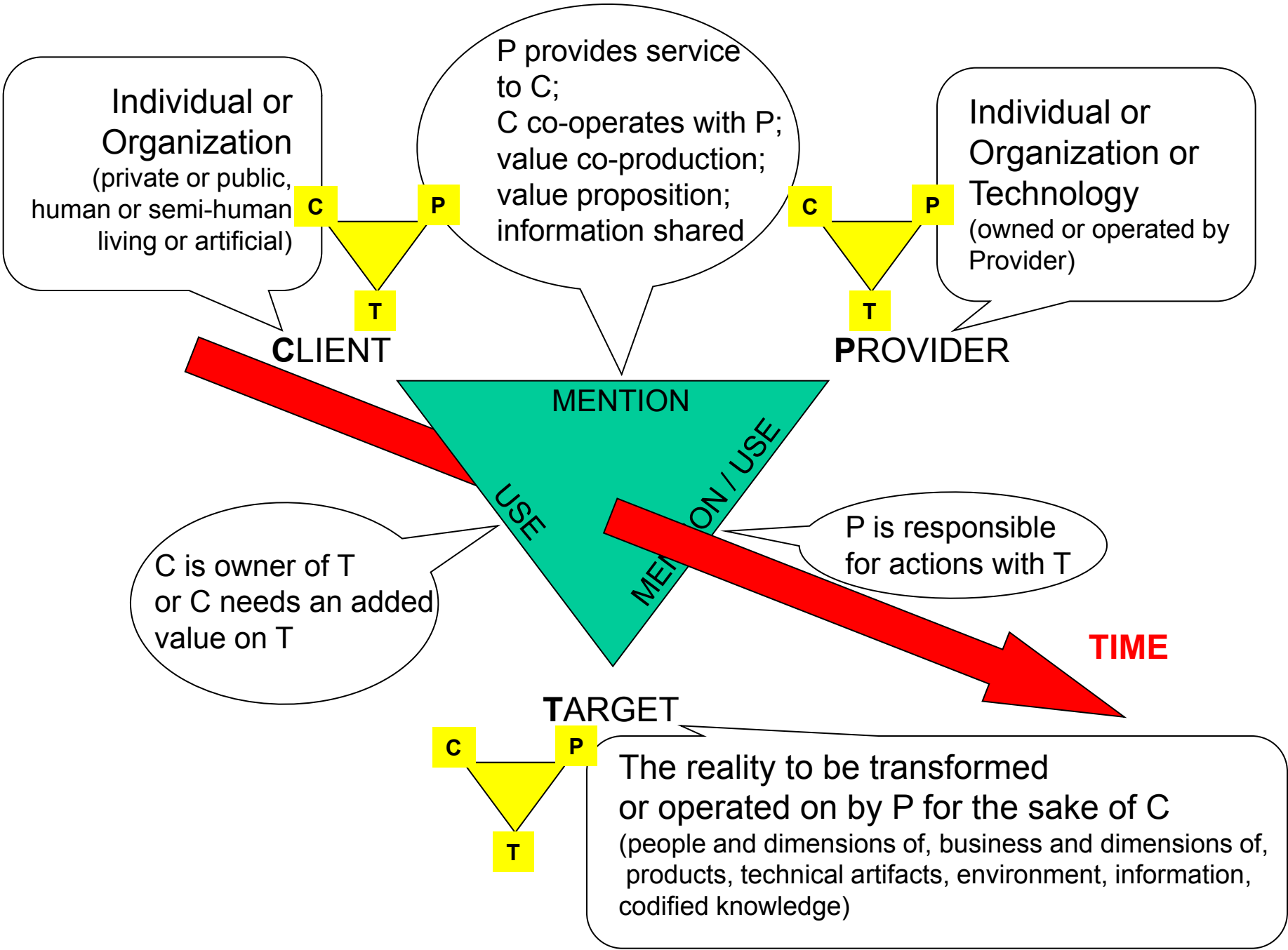
# Role of objects and constructions in conceptual (data) modeling

- OBJECTS – what is focused on
- CONSTRUCTIONS – what identifies focused objects (for all participants in a communication so that they have the same objects in the mind during this common communication)
- Conceptual (Data) Model contains constructions of objects (constructions which identify objects) corresponding especially to non-trivial intensions
- Database contains linguistic EXPRESSIONS (of a special language) which denote some objects (corresponding to extensions)

that was sufficient until recently  
while (primary) information systems were constructed only

# Service Systems—Modeling and Execution (SSME\*)

- Service System is a ***flexible*** and ***adaptable*** composite of people, technology, other service systems, wrapped together with specific knowledge and information, which can be seen at least in one context as a game of Client, Provider and Target, for the benefit of the Client.
- To model or to execute a Service System properly, higher-order-objects are unavoidable !!!



Individual or Organization (private or public, human or semi-human living or artificial)

C

P

T

CLIENT

P provides service to C; C co-operates with P; value co-production; value proposition; information shared

C

P

T

PROVIDER

Individual or Organization or Technology (owned or operated by Provider)

MENTION

USE

MENTION / USE

C is owner of T or C needs an added value on T

P is responsible for actions with T

TIME

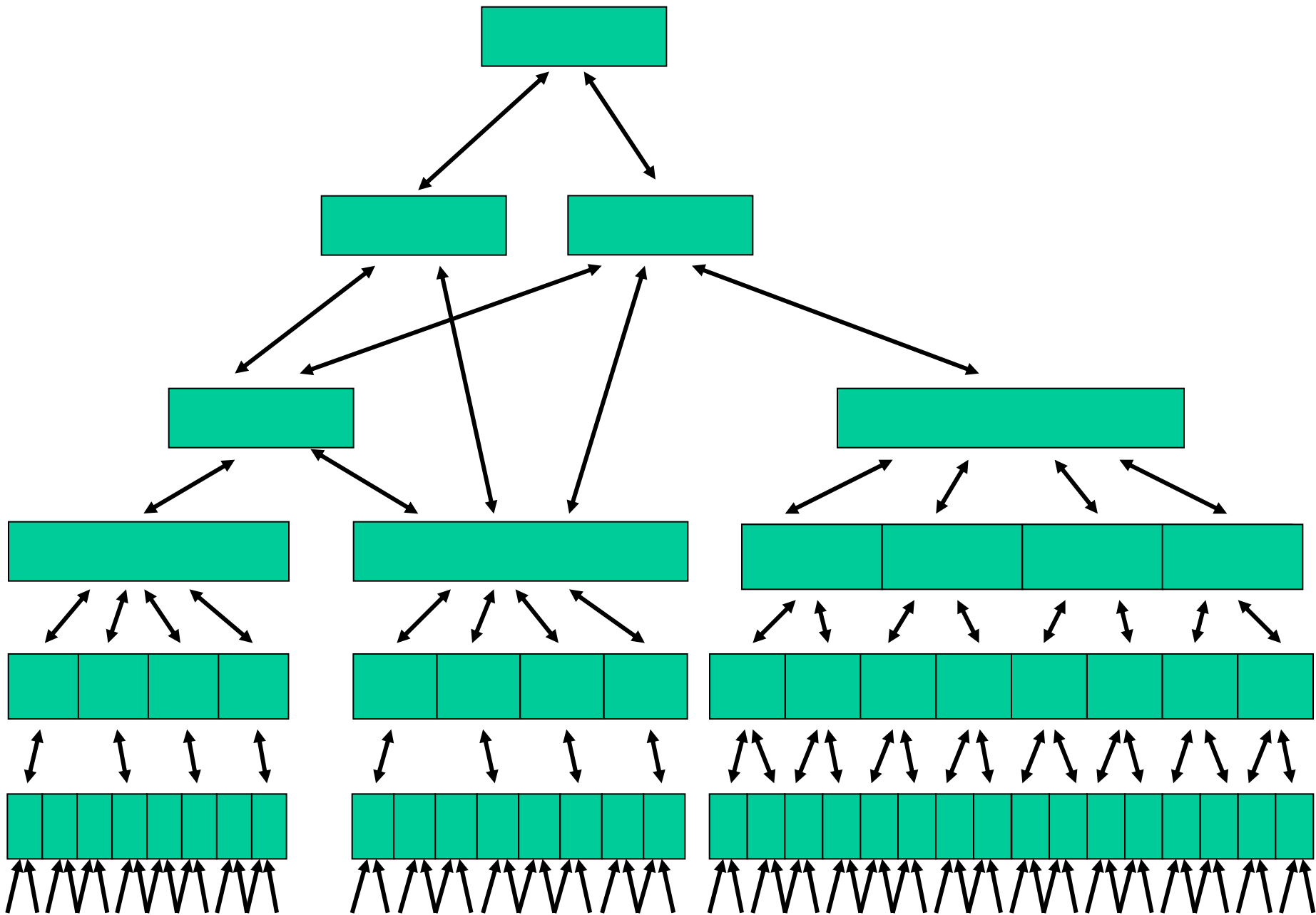
TARGET

C

P

T

The reality to be transformed or operated on by P for the sake of C (people and dimensions of, business and dimensions of, products, technical artifacts, environment, information, codified knowledge)



# What a definition is

What is defined,  
and by what it is defined

# What is defined here?

- “Employee of an organization is a human which has a contract of employment with the organization.”
- ... it is obviously not a definition of the expression “employee” !!!
- define = delimit, demarcate, set aside, identify
- does it identify a concept EMPLOYEE ???  
(= closed construction of a class of Quid constructions where each of them constructs an intension (property) “to be an employee”, i.e. constructs an object)

# So what is defined?

- “Employee of an organization is a human which has a contract of employment with the organization.”
- ! It identifies an object with a specific property.
- Conclusion: **Objects are defined and they are defined by concepts.**

# Examples of definitions

- Entity sort #EMPLOYEE

Object of the type #Employee is every person which has concluded a contract of employment with an organization according to the Labour Code of the Czech Republic.

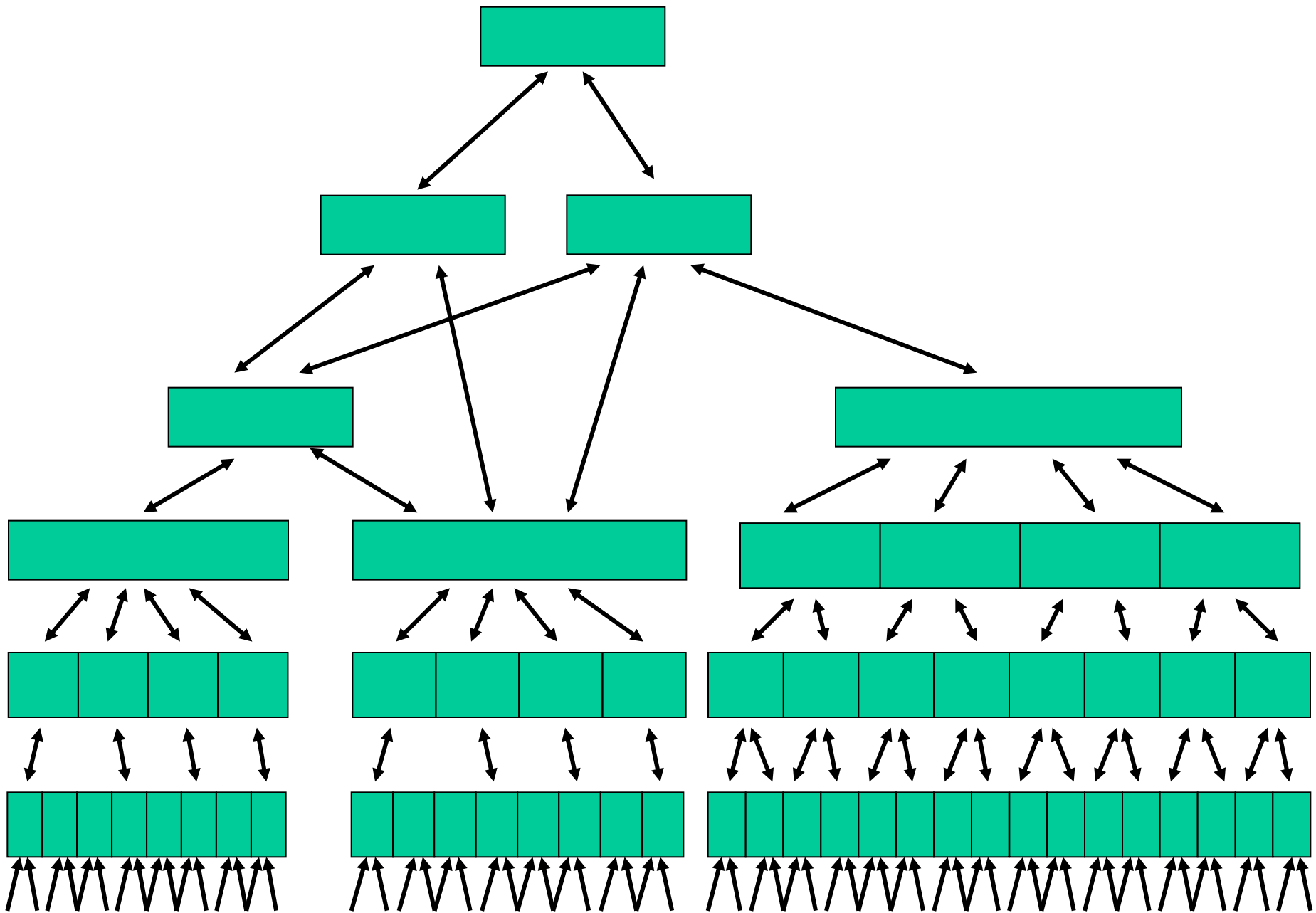
- Entity sort #ARTICLE

Object of the type #Article is every product or service or right which can be a subject of a sale including products, services and rights that did not exist yet but can be made for business activities.



# Is it all? And, what about:

- A number having exactly two divisors we will call prime number.
  - Is this a Definition? Isn't it an Abbreviation?
- What is a “Conceptual System”?
  - Is it just a simple collection of concepts?
  - Or, is it something more ... ?
- What is the role of Language (natural or artificial one) in this game with Concepts, and in wrapping them into Conceptual Systems ?



# Conceptual systems and conceptual models

A deeper insight into Cyberspace  
(what happens in the brain-engine when the  
inner inputs 10-times exceed the external  
ones...)

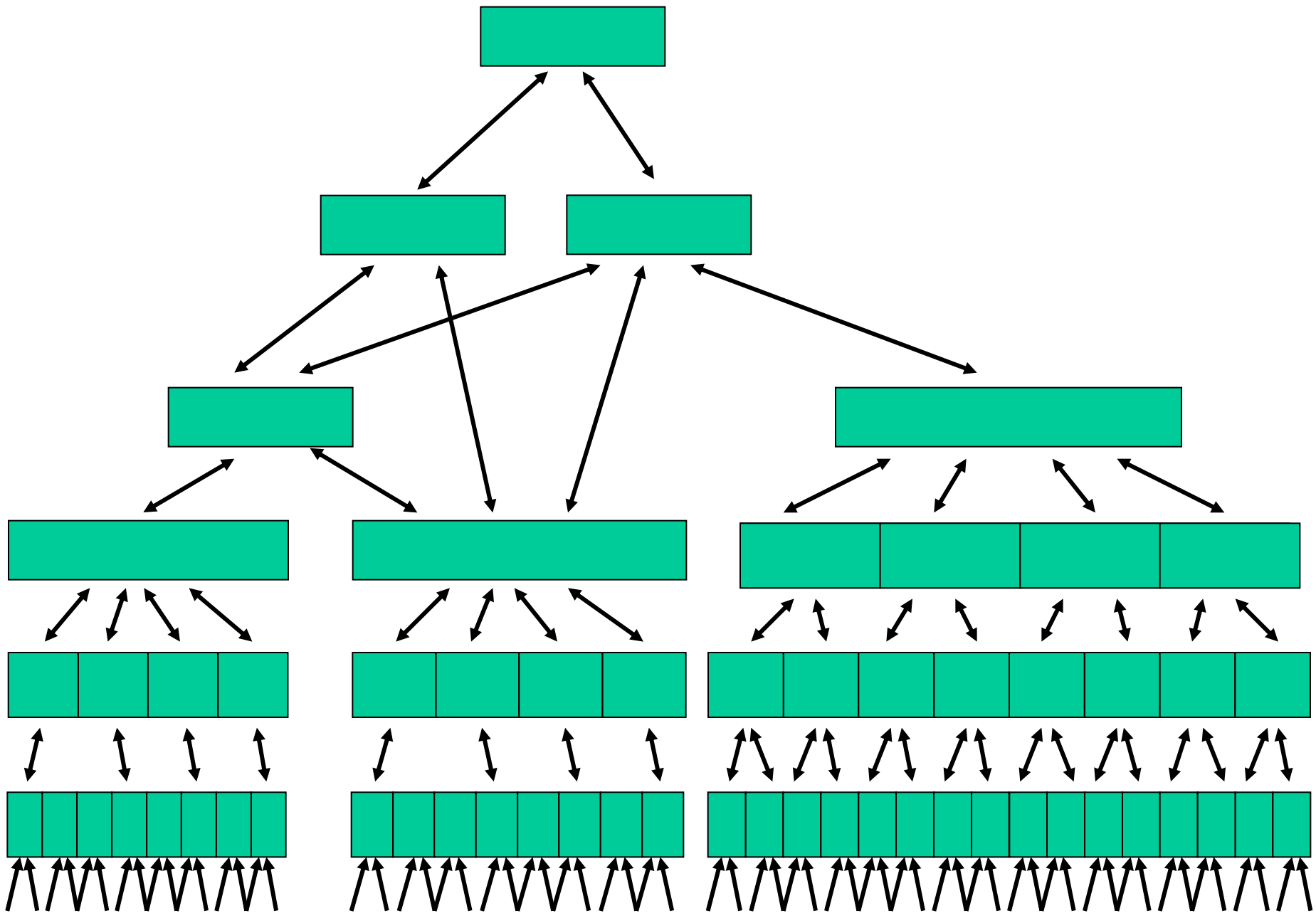
What can help us to understand Service  
Systems and their agents communication and  
thinking

# Conceptual systems and conceptual models

- Primitive/Derived Concepts
- Relativity of primitive/derived with respect to a chosen conceptual system (domain)
- Language of conceptual system (domain)
- Linguistic definition
- Hierarchy of languages over a conceptual system
- The nature of natural language
- What is a conceptual modeling in fact?

# Simple Concepts

- A simple concept is a trivial construction of an object (which is not a construction) or of an variable
- Where to find it within the Fundamental hierarchy?



# Simple Concepts

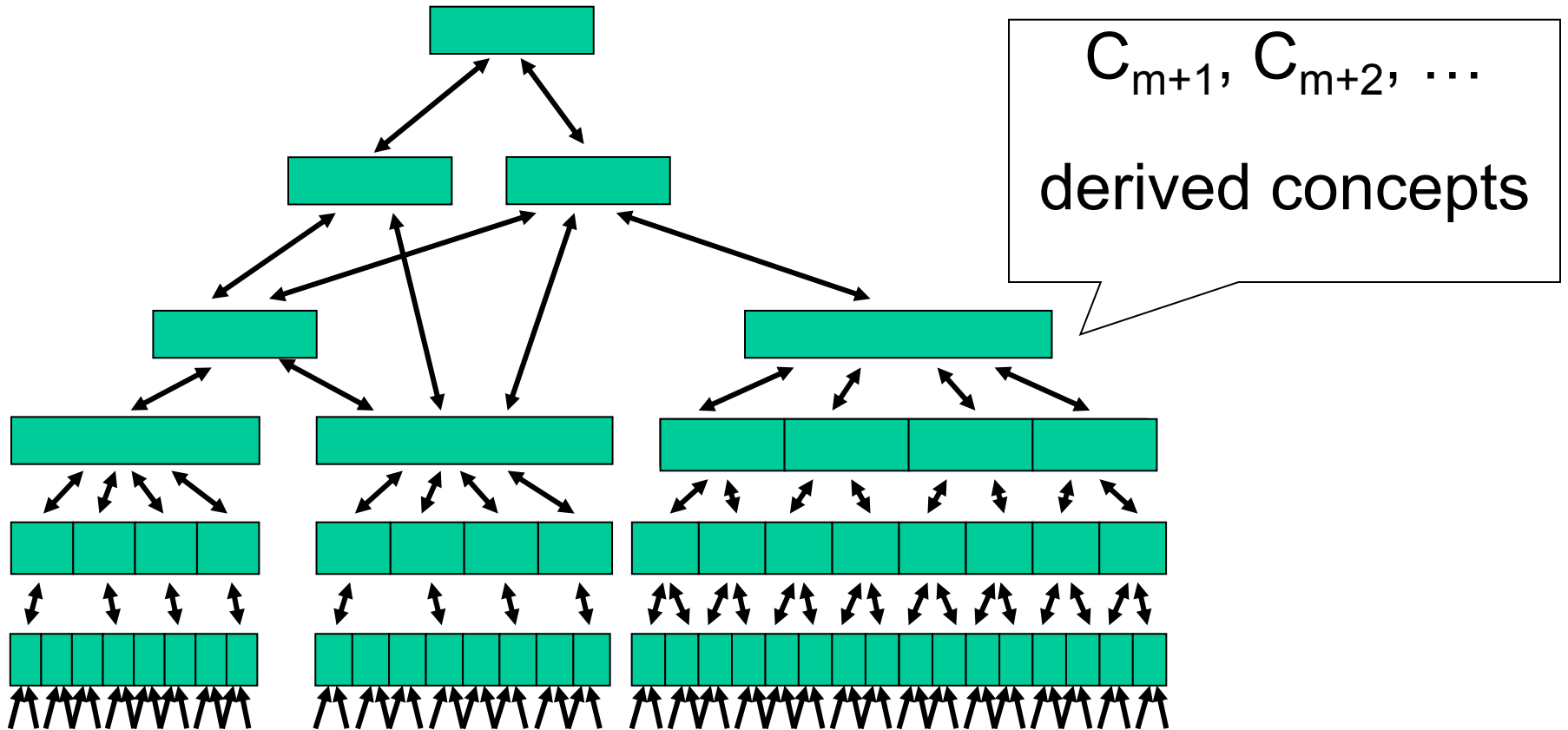
- A simple concept is a trivial construction of an object which is not a construction or of an variable
- Where to find it within the Fundamental hierarchy?
- A simple concept is a construction  ${}^0X$ , where  $X$  is a variable of any type  $\alpha$  of the ramified hierarchy or an  $\alpha$ -object which is not a construction.

# Concept's extension and content

- *Simple concepts point objects and this is clear without any additional explication. Not simple concepts point objects, too, but additional explication is needed to understand what the matter is.*
- The extension of a concept C is the object constructed by C.
- The content of a concept C is the set of all simple concepts that are subconstructions of C.



# Conceptual system

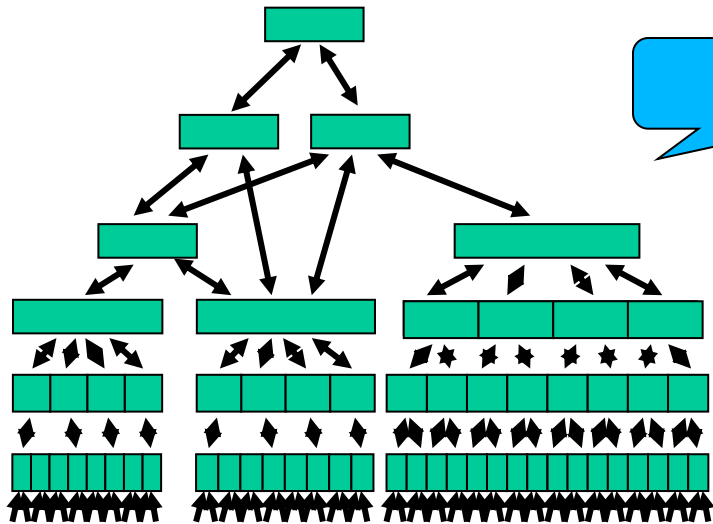


$C_1, \dots, C_m$  --- simple concepts (primitive concepts)

# Conceptual system CS

- $C_1, \dots, C_m$  --- simple concepts
- $C_{m+1}, C_{m+2}, \dots$  be concepts, all of them distinct from any of  $C_1, \dots, C_m$ , such that the subconstructions of  $C_{m+i}$ ,  $i > 0$ , are only
  - members of the set  $C_1, \dots, C_m$ , and
  - variables ranging over those types that are composed of types given by  $C_1, \dots, C_m$ .
- $PCS = \{C_1, \dots, C_m\}$  primitive concepts
- $DCS = \{C_{m+1}, C_{m+2}, \dots\}$  derived concepts
- $CS = PCS \cup DCS$

# Relativity of primitive/derived



DOMAIN 1

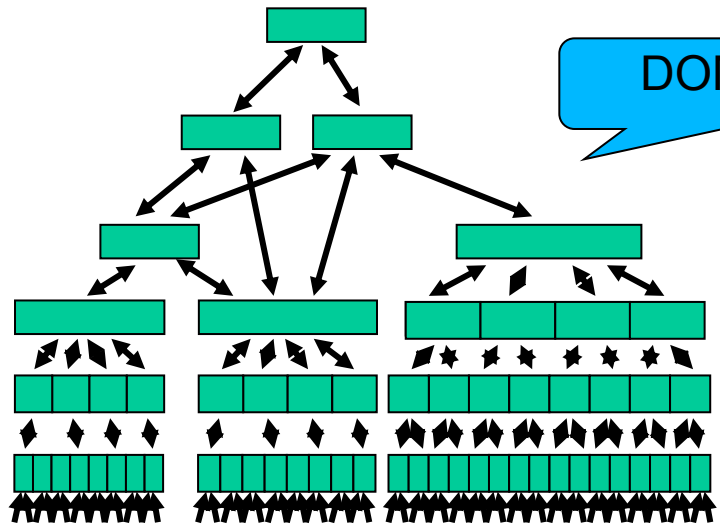
What is primitive within one domain can be derived in other domain.

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  $\alpha$ ” from the ramified hierarchy.



DOMAIN 2

primitive concepts

How we evaluate the situation depends on a **Context**.

The semantics is given by the couple:  
(Domain, Conceptual System).

# Concepts and Context relationship

- There is always lot of **contexts** from which a domain in a situation could be evaluated and/or studied.
- There is no one context which could be called “**a master context**” !
- Hence, there is no one conceptual system which could be said “a master one”.
- **Cn(...)** : **Concepts are context independent; selection of Conceptual System is context dependent !**

# What is an impact on Service Systems

- To work in a domain (to provide services), namely to provide innovations of services in a domain means to develop the **conceptual system of this domain.**
- Remember the three categories of information and knowledge which must be shared by agents of a Service System:
  - Language\*)
  - Laws (or Rules)
  - Measures

# Linguistic definition

- The standard explication of what a definition is speaks about “definiendum” (i.e., what is defined) and “definiens” (i.e., by which it is defined).
- The standard approach is a linguistic one, as it is based on NL (natural language) expressions and natural language semantics.
- These standard definitions have the form
- 'Definiendum = Definiens'.
- This is a short-cut! It introduces a name for something.

# examples

- We will call Prime Number each natural number having exactly two divisors.
- By the term Employee we will mean each person having valid employment contract with a company or an institution.

## **On the contrary:**

- *An object of a sort (#Emp) is each such person that has a valid employment contract with a company or an institution.*
- ... this sentence describes a construction of an object (entity sort) using given properties. At the same time it introduces a name of the entity sort.
- This is a combination of a pure definition in the basic sense and of a linguistic definition.



# \*) Language and CS

- Let CS be a conceptual system based on PCS =  $\{C_1, \dots, C_m\}$
- A language  $L_{CS}$  of conceptual system CS is a language satisfying the following conditions:
  - There are simple expressions in  $L_{CS}$  that represent  $C_1, \dots, C_m$ .
  - If E is expression of  $L_{CS}$  that represents construction X of  $\alpha$ -objects, then there are grammatical rules of  $L_{CS}$  that make it possible to create expressions  $E_{cons}$  of  $L_{CS}$  that represent any construction of any one of the 8 construction modes.
  - No other expressions are in  $L_{CS}$ .

This is not enough for a  
convenient communication and  
thinking !

We need a more comfortable  
tool !

# Hierarchy of languages with respect to a conceptual system (1)

- Let  $SE_j$  denotes a simple language expression, i.e. not structured expression (not composed of other language expressions).
- It is usual to take such expression as a name of this item which is denoted by this simple expression.
- Let  $CE_j$  denotes a complex language expression, a structured expression composed of other language expressions.

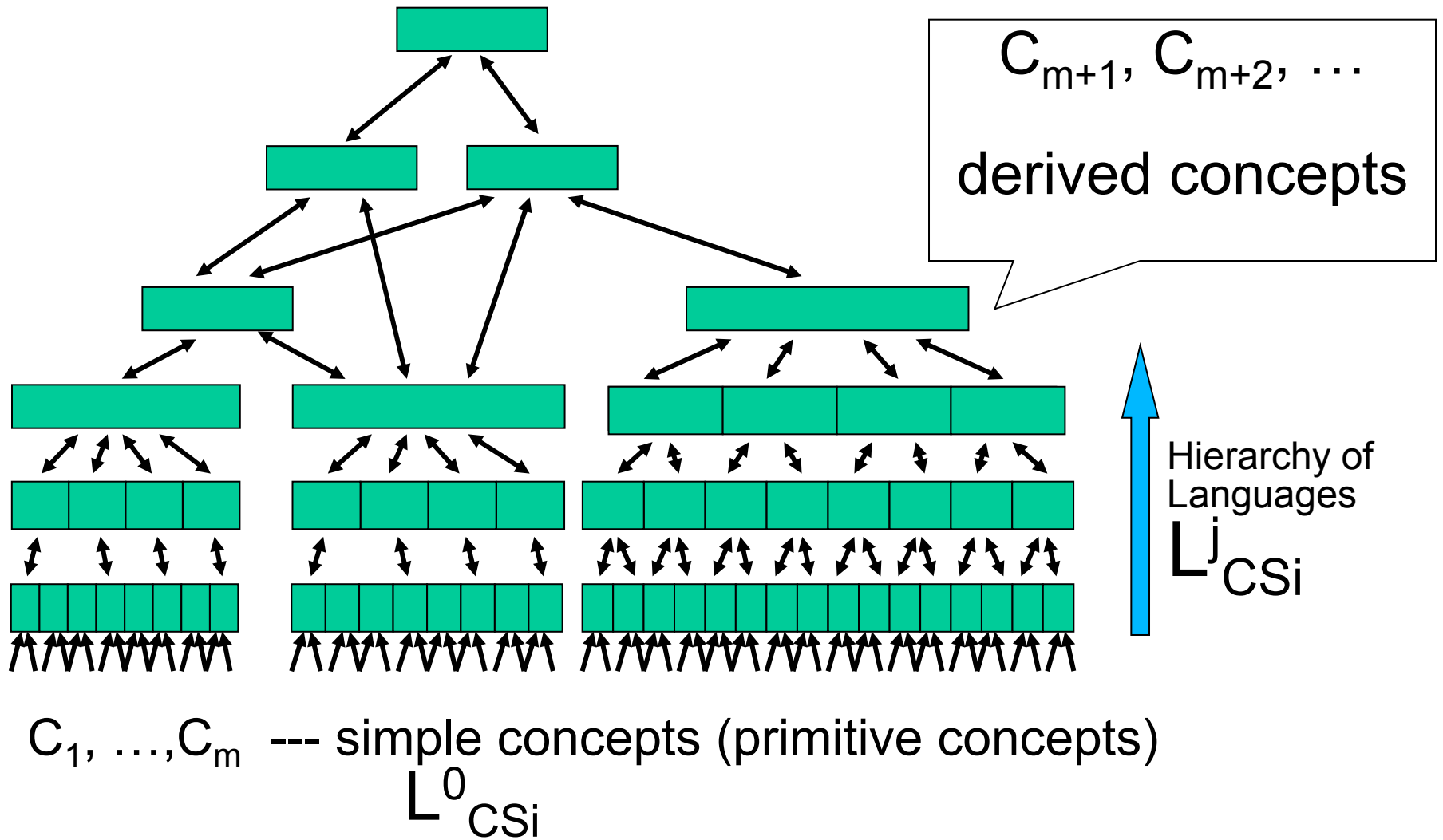
# Hierarchy of languages with respect to a conceptual system (2)

- Linguistic definitions in a language of  $i$ -th level with respect to a conceptual system CS are introduced inductively:
  - Let  $L_{CS}^0$  be  $L_{CS}$ .
  - Let  $L_{CS}^i$ ,  $i > 0$ , results from  $L_{CS}^{i-1}$  by adding a set of simple expressions  $SE_1^i, \dots, SE_k^i$ ,  $k > 0$ , together with expressions interpreted as true sentences:  
 $SE_1^i = CE_1^i$   
 $SE_k^i = CE_k^i$   
where  $CE_1^i, \dots, CE_k^i$  are complex expressions that contain only expressions occurring in  $L_{CS}^{i-1}$ .
  - The expressions  $SE_j^i = CE_j^i$ ,  $i > 0$ ,  $j = 1..k$ , are called linguistic definitions expressed by the language  $L_{CS}^i$ .

# Hierarchy of languages with respect to a conceptual system (3)

- Simple expressions  $SE^i_1, \dots, SE^i_k$ , are definienda, complex expressions  $CE^i_1, \dots, CE^i_k$  are definiens expressed by the language  $L^i_{CS}$ .

# Conceptual system and its hierarchy of languages



# Hierarchy of languages with respect to a conceptual system

- NL contains a wide set of languages  $L_{CS_i}^j$  of various conceptual systems  $CS_i$ .
- The hierarchy grows up to higher levels during the time in accordance with the step-by-step growing cognition of any given domain.

- 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.



(the last question!)

What is a conceptual  
modeling in fact ?

# What a conceptual modeling is in fact

- Discovering of suitable concepts identifying objects in the Domain under Discussion (DuD).
- Creating pragmatically a proper conceptual system.
- Step-by-step designing of a hierarchy of languages with respect to the conceptual system.
- Real understanding to DuD through previous steps, i.e. by creating of semantics of words connected to the domain.

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 !

# Conclusion

Thus the Conceptual Modeling is  
the way to understand

- domains
- environment
- systems

exactly!

And, moreover, it is possible!

... and aiming to create viable  
Service Systems  
it is necessary ...