Implementing a Knowledge-driven Hierarchical Context Model in a Medical Laboratory Information System

Jaan Pruulmann and Jan Willemson

Tartu, Estonia

ICGGI 2008

July 27, 2008 - Athens, Greece

Contents

Background objectives

- **In result of** our
  - Soviet past
  - Very intensive evolution
  - Today's open society
- We have **big variety in**
  - Doctors' background
  - Laboratory equipment
  - Laboratory processes
  - Computer systems

Need to cope with **coexistence** of **old** and **new** <u>understandings, equipment, technology</u> etc.

Small country – the amount of everything is tractable

Background (some numbers)

Clients	**470** medical **organizations**
	10 different types of software, including 4 with capability for some electronic communication
	1250 doctors (clinicians, surgeons, anaesthetists, family doctors etc.)
Tech-nology	About **800** individually described **tests**
	55 different, partially overlapped, **code sets**
	54 different, partially overlapped **technologies**
Struc-ture	**Three 24x7 laboratories:** one universal (in maternity ward), two specialised (clinical chemistry and haematology)
	Nine business-hours laboratories are specialized to specific testing technology (e.g. immunology) or to specific profile of local hospital (e.g. oncology)
	All have evolved in own **local society** and **context**

Requirements for LIS

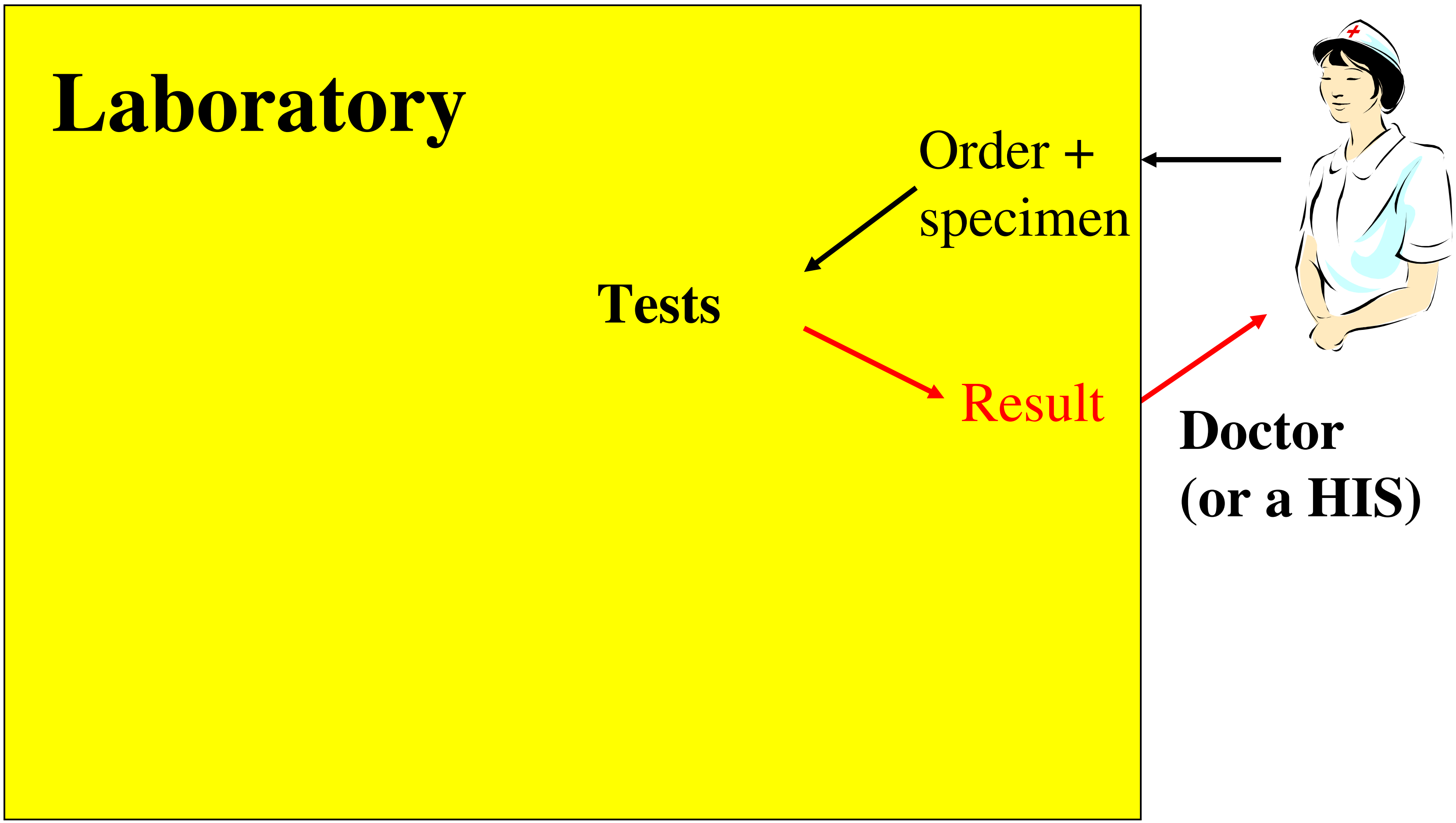
- LIS has to serve the common laboratory functioning 24x7 without breaks .
- LIS has to support
  - Structural **reorganization** of **laboratory itself**
  - **Renewing** testing **technologies** and **equipment**
  - **Evolution** of client **computer systems** and software
  - **Cooperation** with external laboratories

while being continuously in use

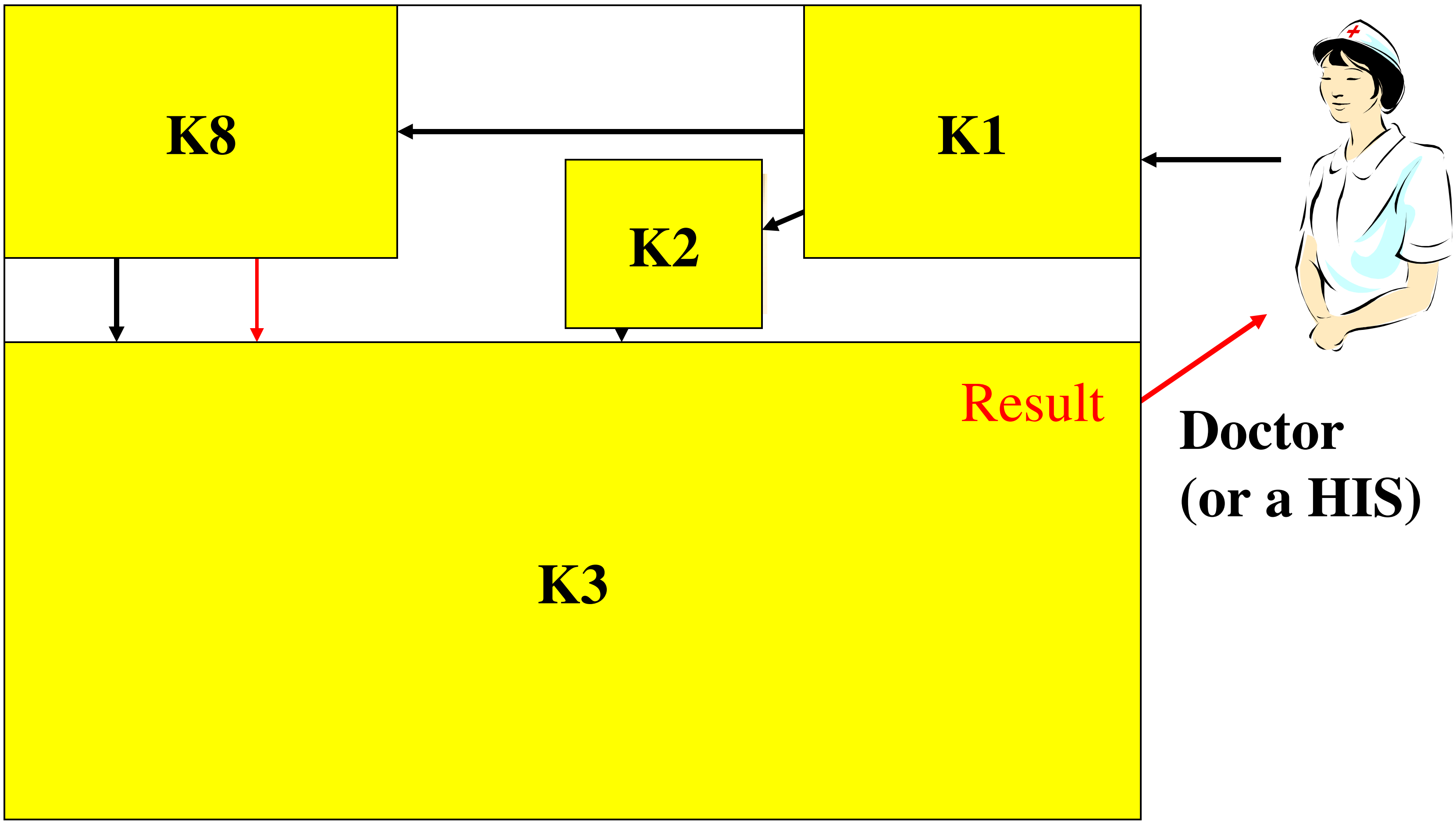
A dream about solution

- LIS should be an **intelligent system,** being capable to adapt oneself to expected changes
- It has to have **detailed, dynamic knowledge** about
  - Nature of laboratory processes
  - Individual laboratory technologies and equipment
  - Individual laboratory workers (co-workers)
  - Individual clients and their needs to laboratory
  - External systems it communicate with

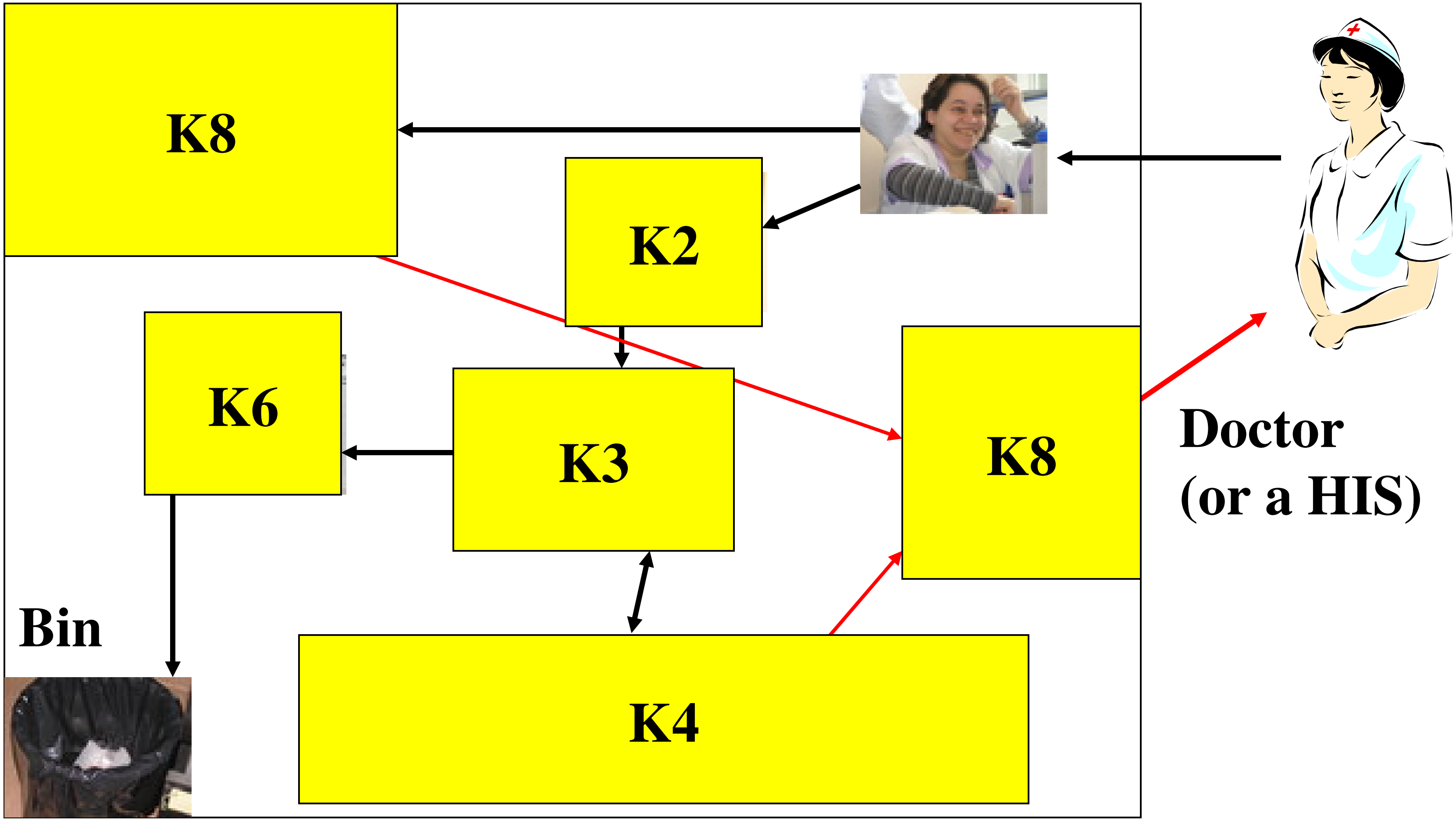
What is a medical laboratory?



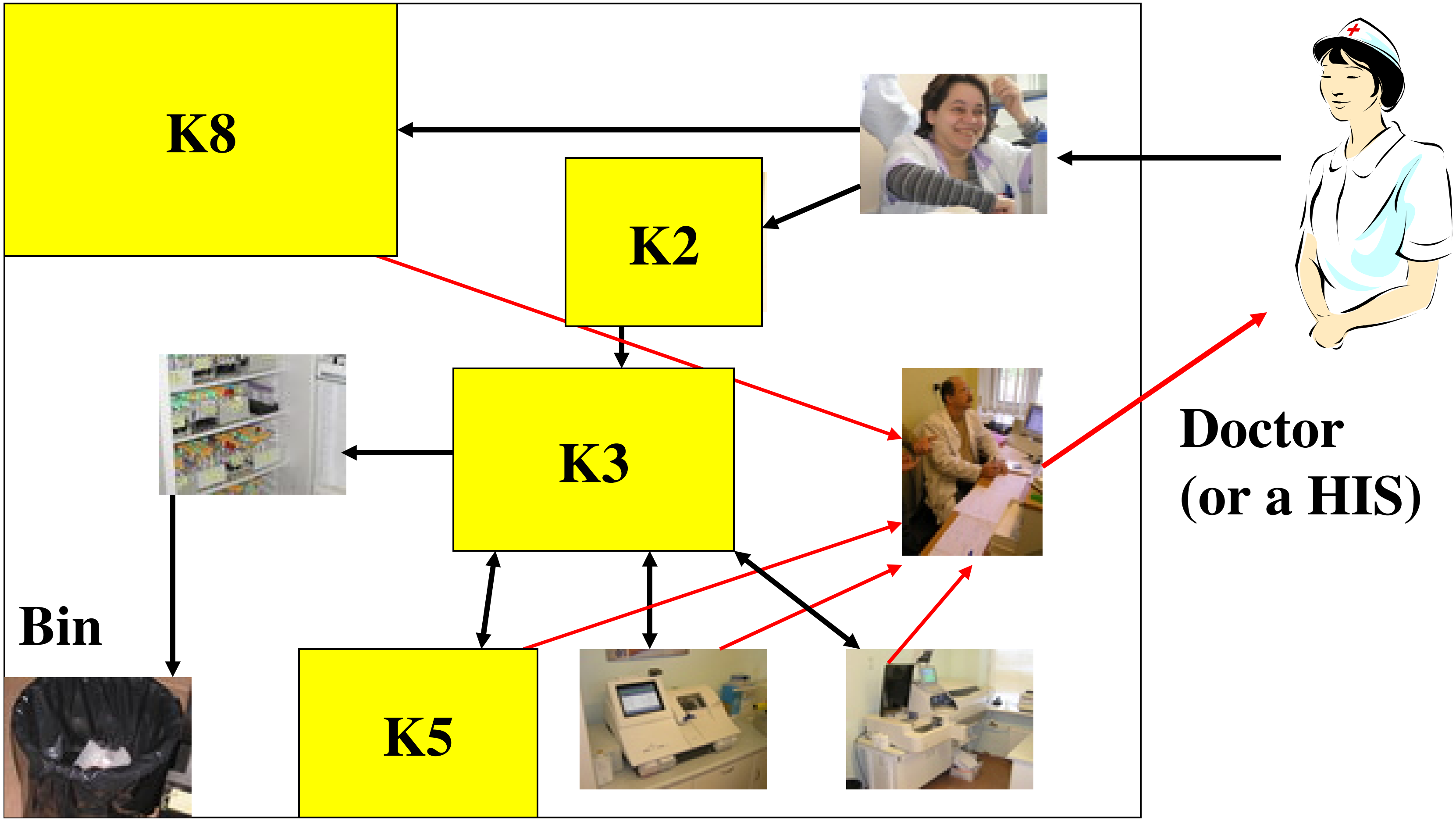
Hierarchical model of laboratory



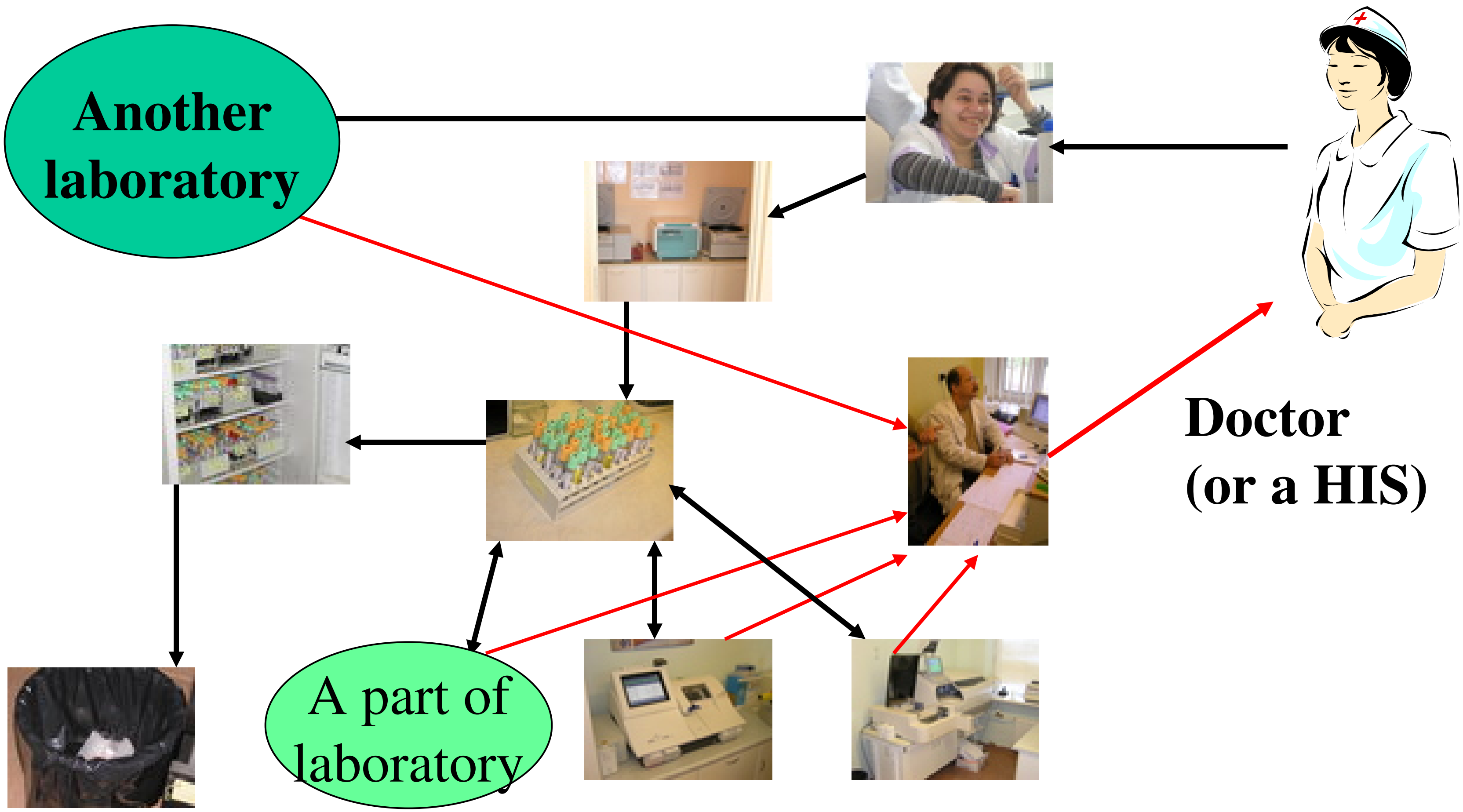
Hierarchical model of laboratory



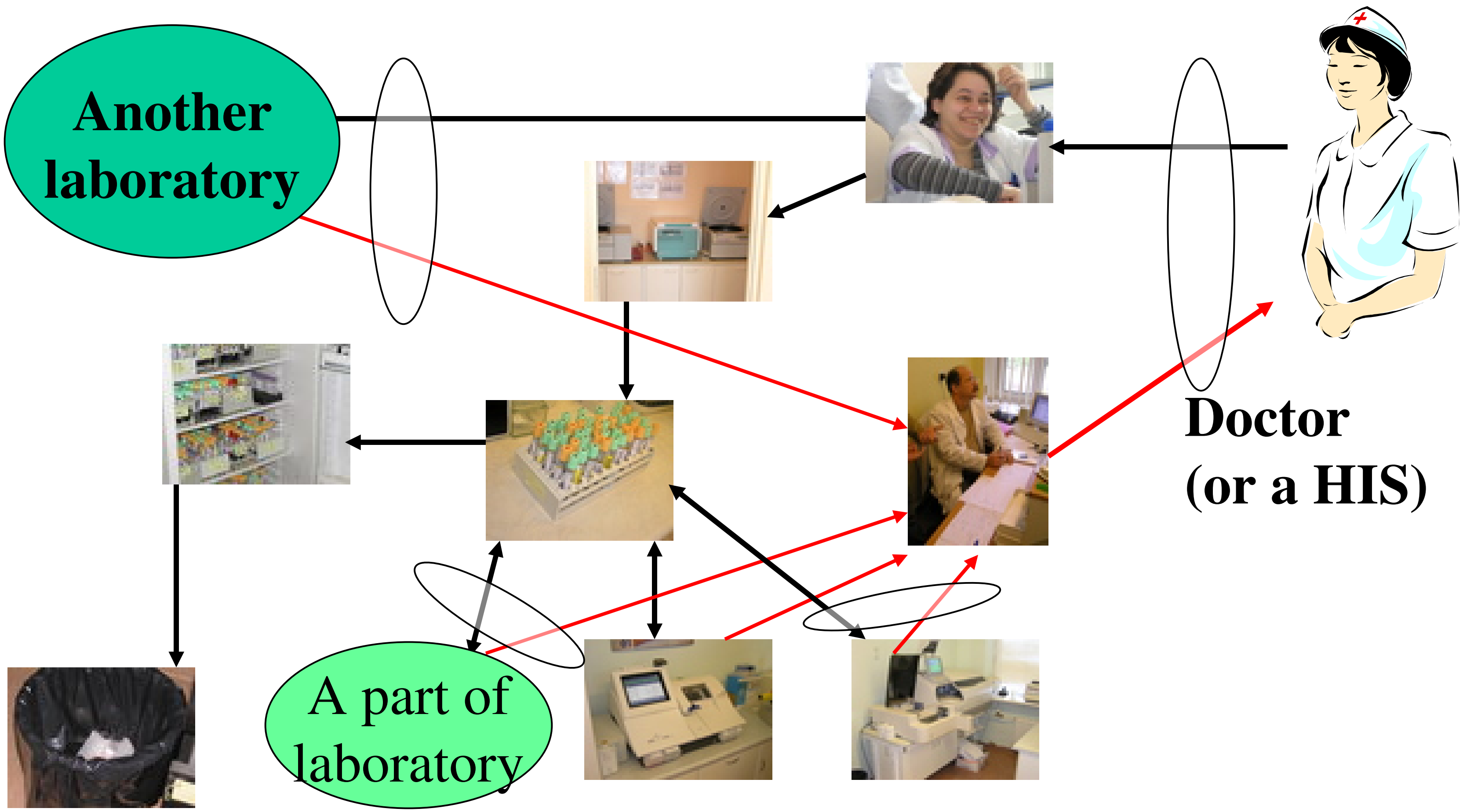
Hierarchical model of laboratory



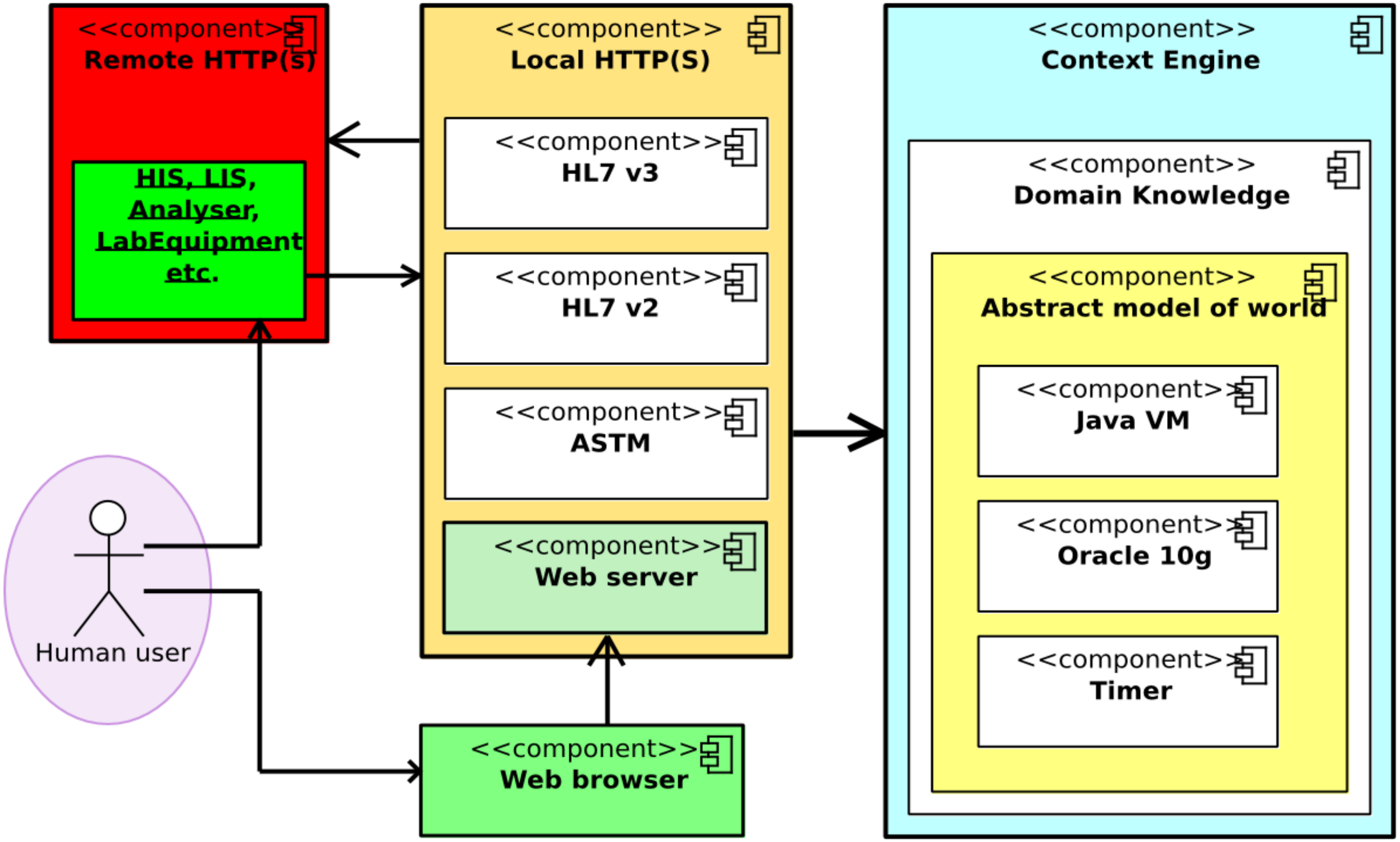
Hierarchical model of laboratory



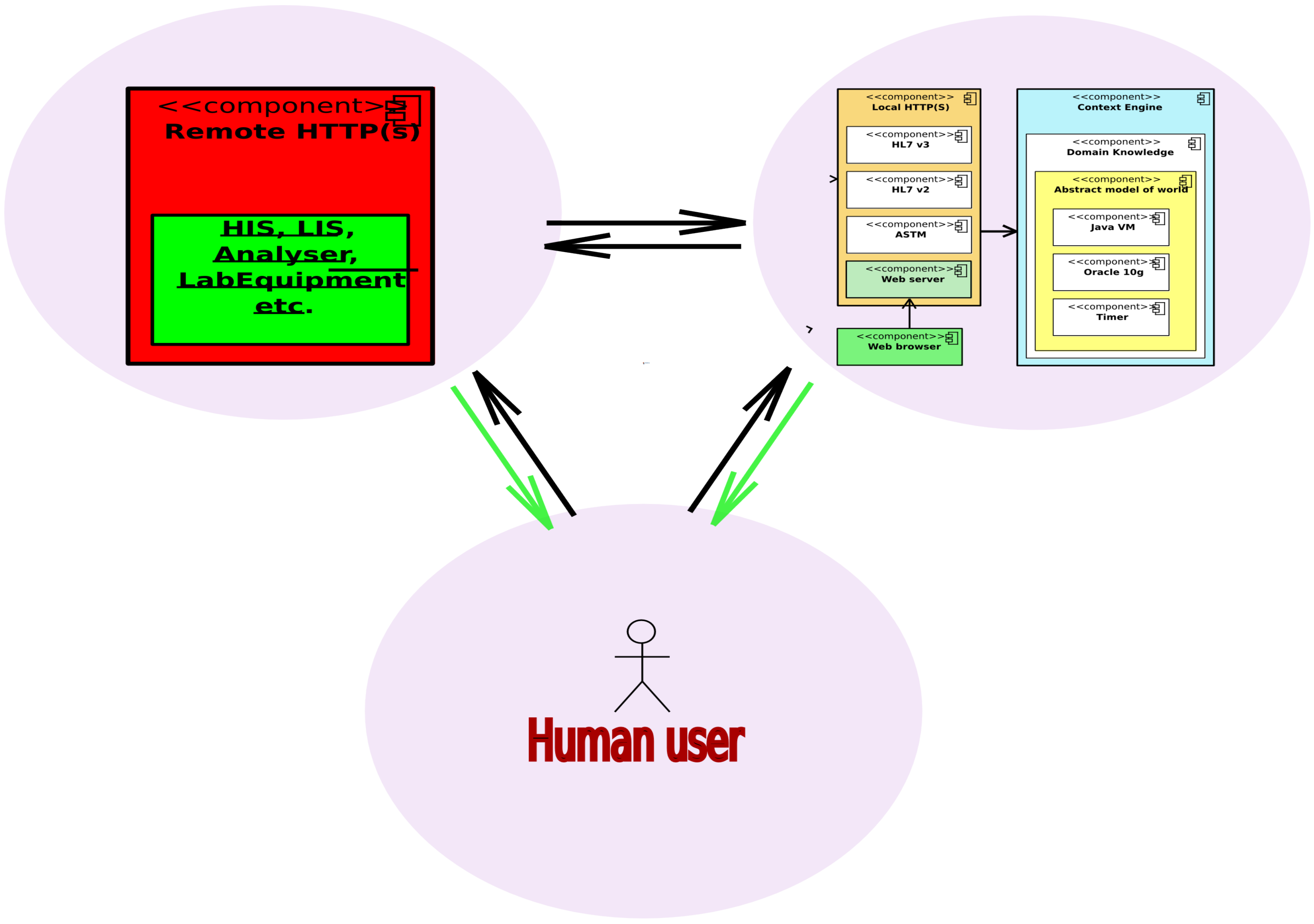
Hierarchical model of laboratory



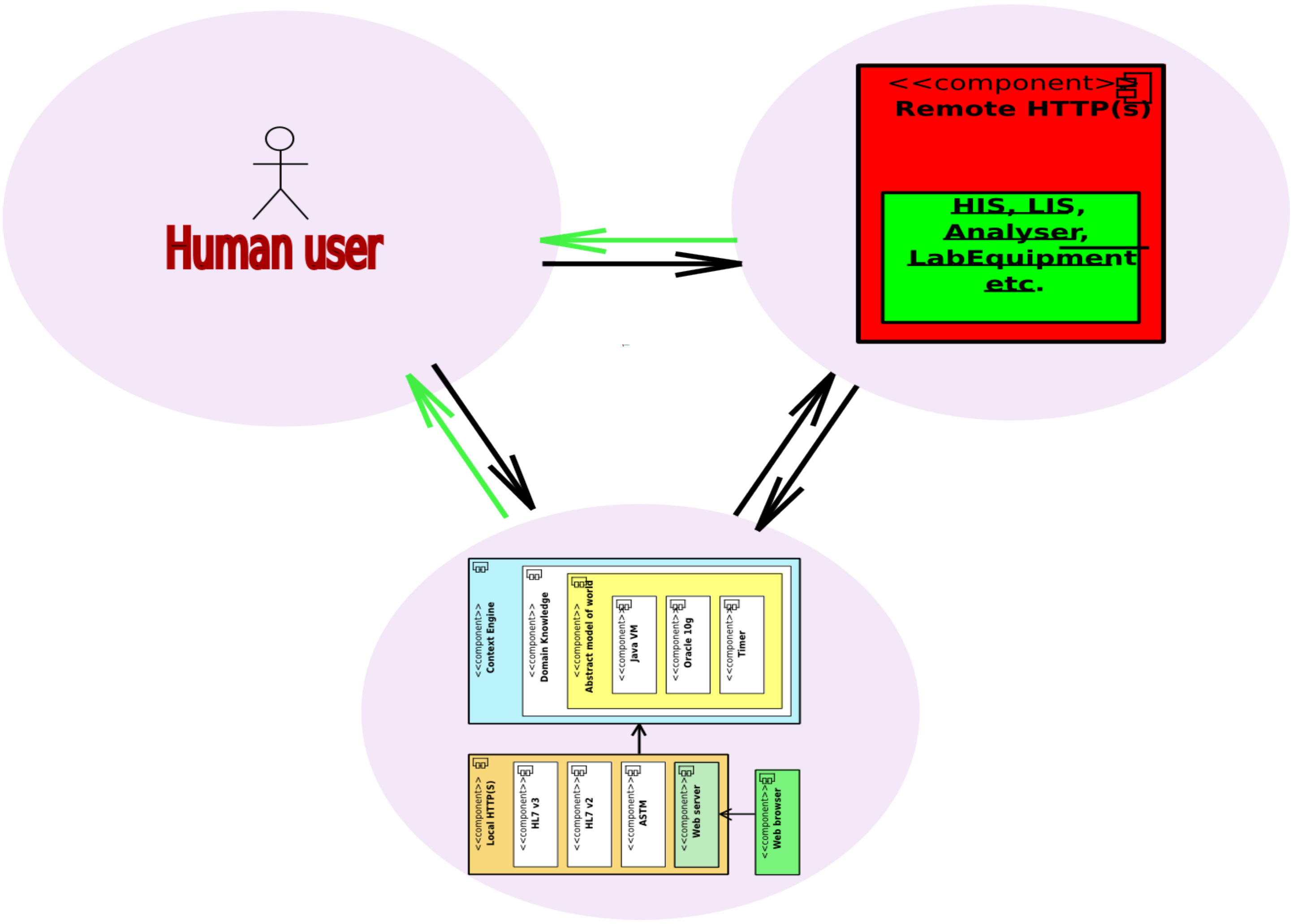
Structure of LIS



Structure of LIS



Structure of LIS



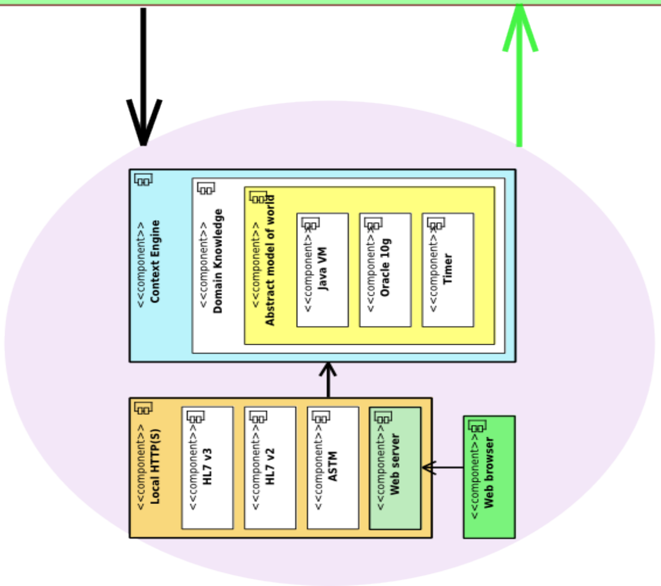
Concept of LIS

In real world, the LIS communicates with

Organization
Specimen
Client
Agreement
Doctor
Order
Workplace
Location
Test
Container
Technician

It sees directly

HIS
Equipment
PC
LIS



The real solution concept

- LIS has its **own conceptual model** about real world around it including **sub-models** about
  - Laboratory **medicine**
  - Laboratory **technologies**
  - Laboratory **organisation**, including LIS **users**
  - **Clients** and their organisation and users
- LIS acts as **an actor** who simultaneously **communicates with** other actors
  - **Human users**
  - **External systems**

The model bases on hierarchies

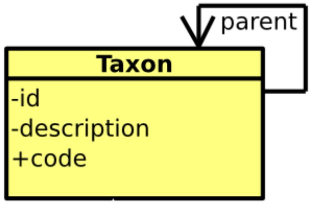
- Taxonomical hierarchy
  - A natural way to describe **abstract concepts.** A property of a taxon is effective for all its sub-taxa
- Organisational hierarchy
  - **Rights** are partially delegated (or dispersed) from a level to some taxon on next lower level. **Responsibilities** corresponding to rights are delegated, too
- Compositional hierarchy
  - Describes how the **things** are composed from **parts**

Why hierarchies?

- They base on common tree abstraction that is **simple in implementation**
  - A tree **can be evolved** easily **'on the fly'** when a new detail is reported by an external system or user
- Unlimited detailing is possible
- They are simple to understand, too

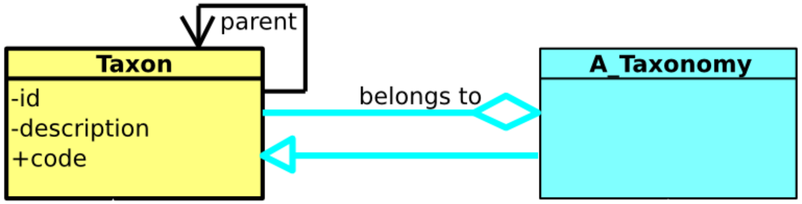
But the world is is not so simple. Is it?

Class model of context engine



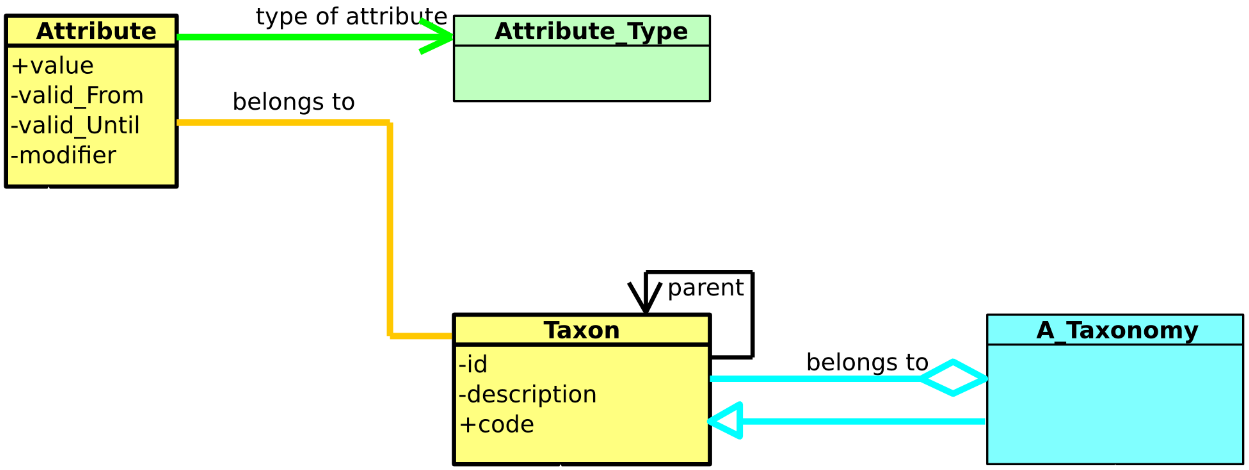
- A new abstraction can be created only on basis of some taxonomical difference according to abstractions existing before
- So all our abstractions can be in a common tree
- A taxon has its own id, code, description and parent

Class model of context engine



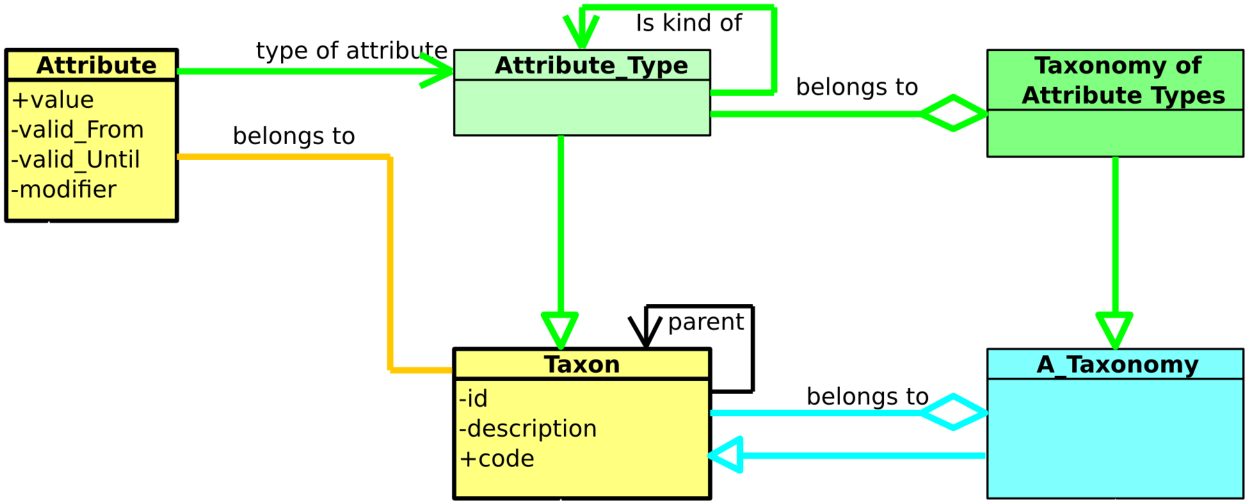
- A subtree of taxa constitutes a whole taxonomy corresponding to its root taxon
- E.g. any code set is a simple taxonomy

Class model of context engine



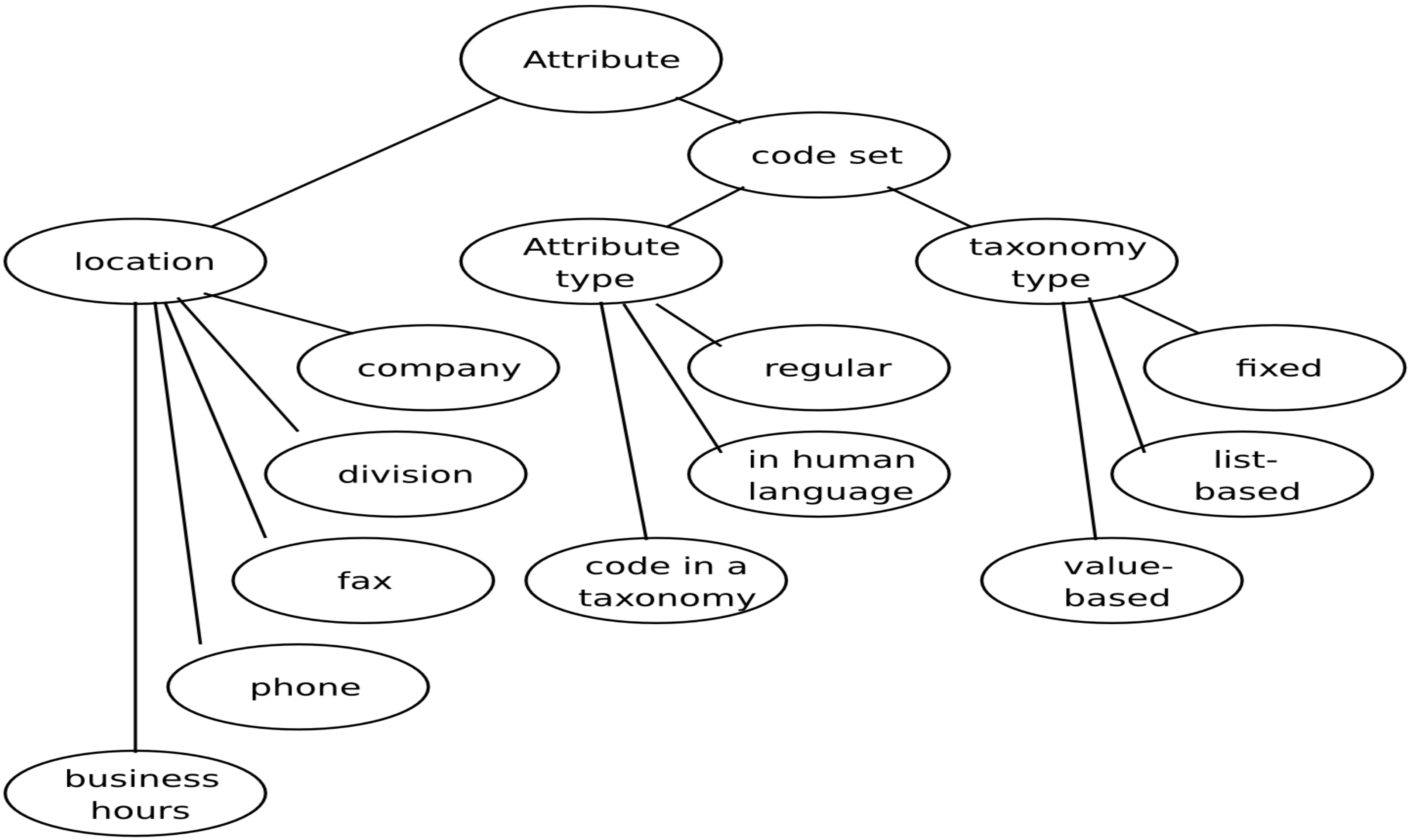
- Abstractions (taxa) may be described by a number of valued attributes of particular types

Class model of context engine

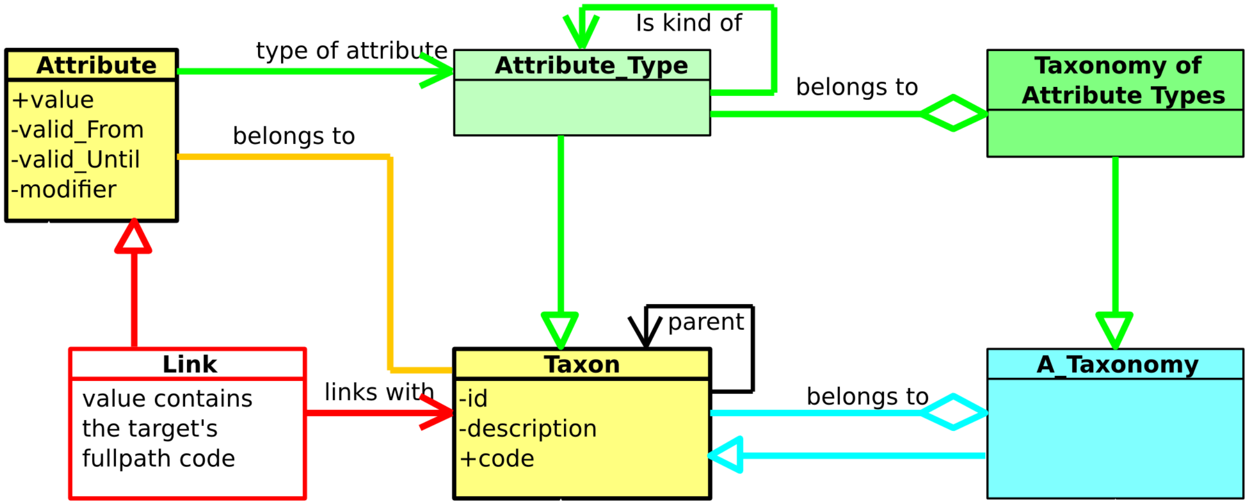


- An attribute type is an abstraction, too, and hence a taxon in the taxonomy of attribute types
- Attribute types may be hierarchically detailed

Attribute types (example)

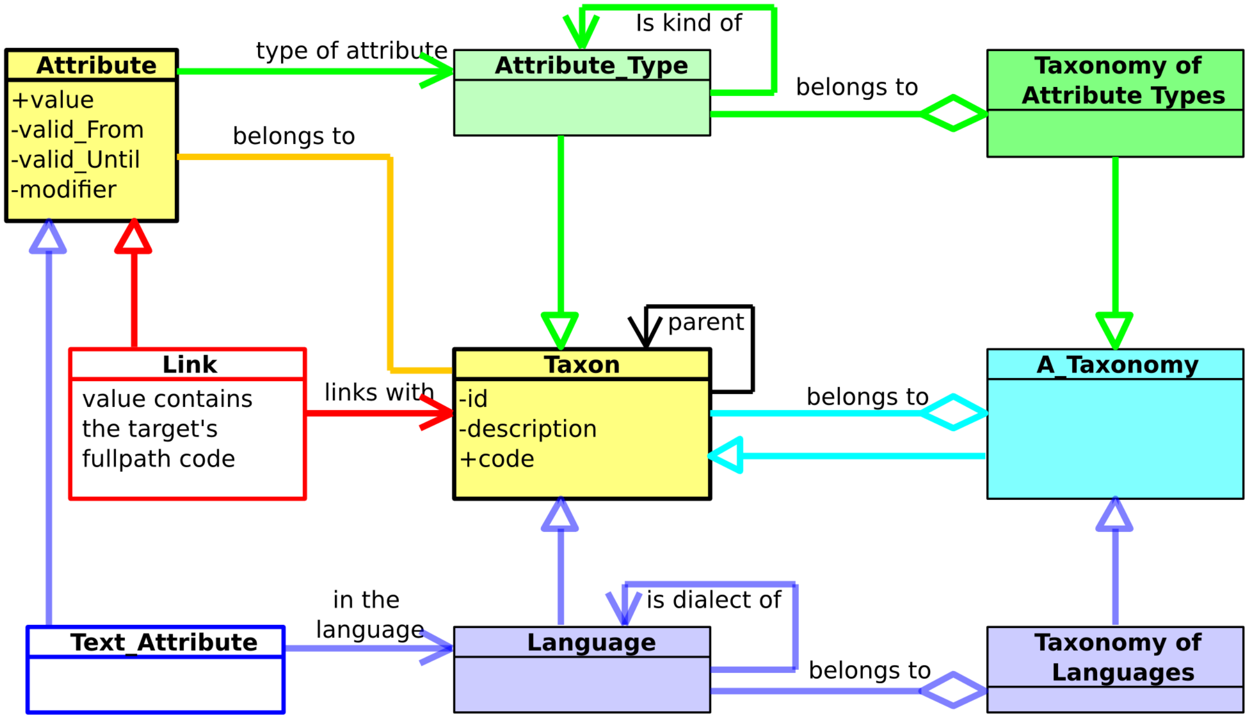


Class model of context engine



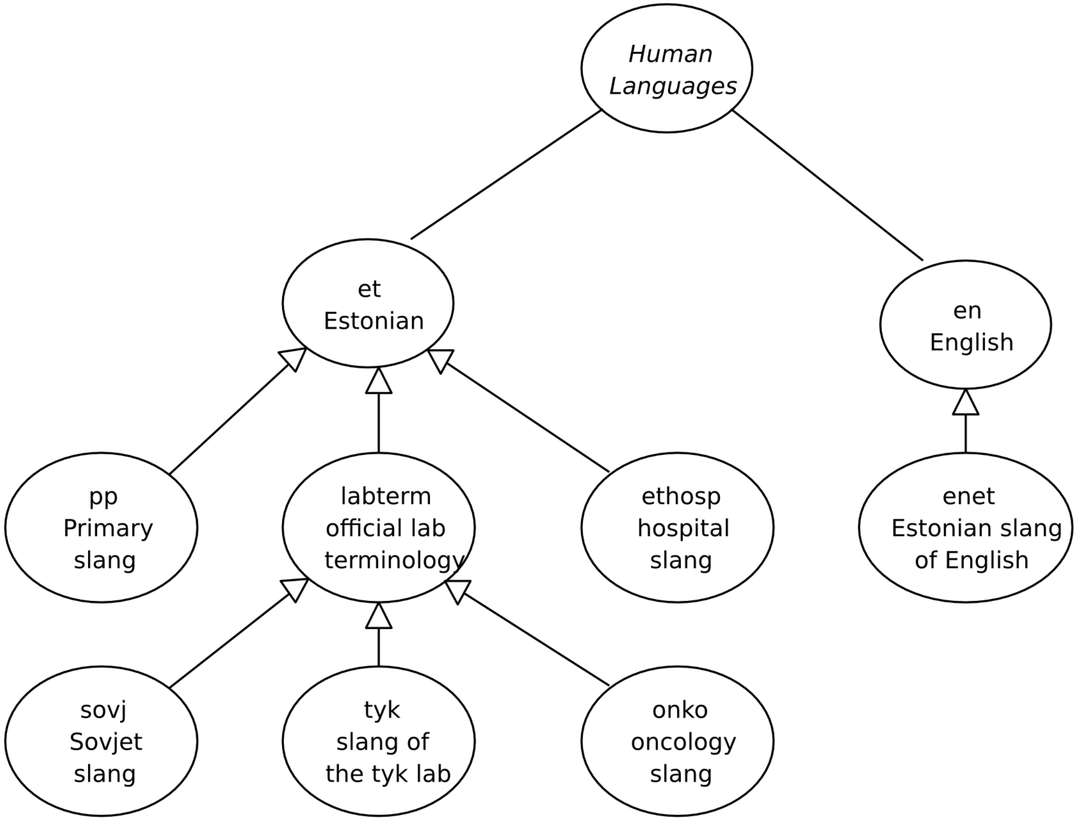
- Coded attributes are directly linked with the taxon that corresponds to the attribute value
- The organisational and compositional hierarchies are described by links with corresponding types

Class model of context engine



- A taxon can have a number of different type names in a human language or language dialect

Taxonomy languages (example)



Context algebra of codes

- Every taxon has its own code.
- Code is unique in the set of sibling taxa
- FPC – Full Path Code – list of codes from root taxon to the taxon itself.
- We can split a FPC into two parts at any tree arc

$$FPC = FPC_{context} + CODE_{in\_context}$$

  - Where $CODE_{in\_context}$ is code of the taxon when the taxonomy $FPC_{context}$ is determined by context
- A taxon has different codes in different contexts

Context algebra of codes (example)

[Root] – addr – Estonia – Tartu – Tähe – 4 – B – 201

Full Path Code: it can be split and combined

(context)	(address in context)
addr-Estonia	Tartu-Tähe-4-B-2001
addr-Estonia-Tartu	Tähe-4-B-201
addr-Estonia-Tartu-Tähe-4	B-201

Context algebra of codes (example)

Root – addr – Estonia – Tartu – Tähe – 4 – B – 201

Root – addr – Estonia – Tartu – Aleksandri – 9 – 316

Common location (context)	Destination address (address in context)
addr-Estonia	Tartu-Tähe-4-B-2001
addr-Estonia-Tartu	**Tähe-4-B-201**
addr-Estonia-Tartu-Tähe-4	B-201

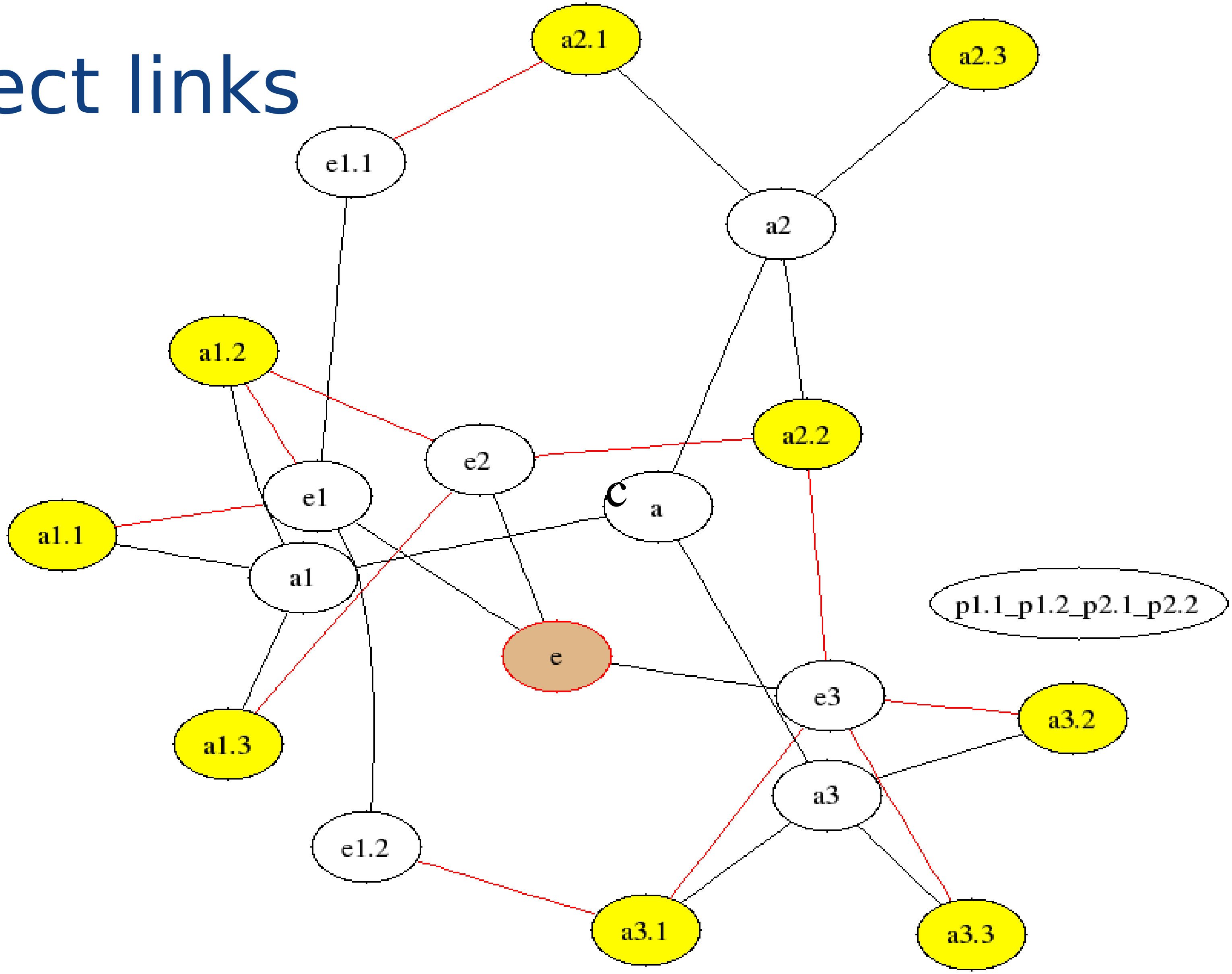
Context algebra of codes (example)

Root – addr – Estonia – Tartu – Tähe – 4 – B – 201

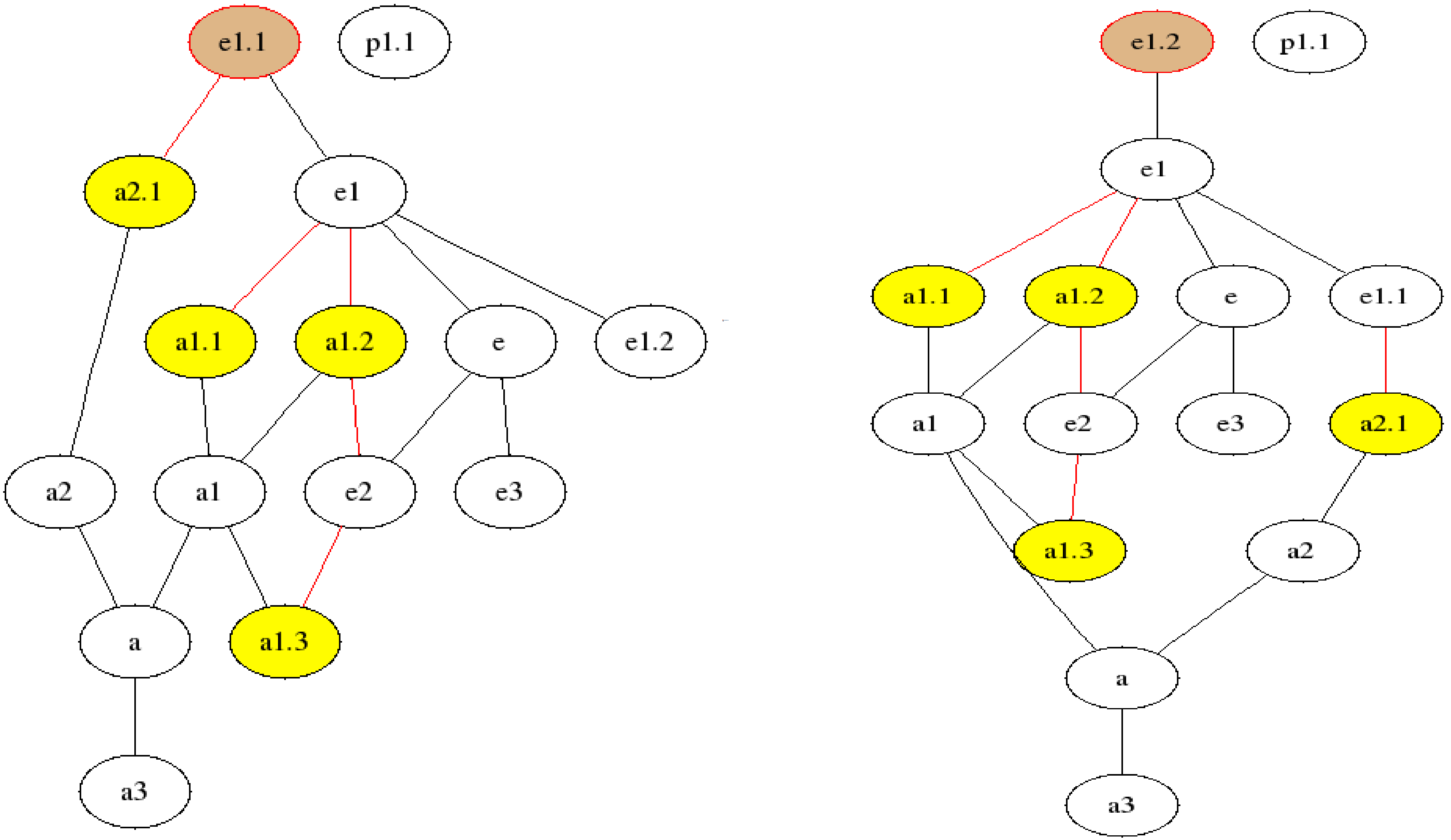
Root – addr – Estonia – Tallinn - Akadeemia – 21

Common location (context)	Destination address (address in context)
addr-Estonia	Tartu-Tähe-4-B-2001
addr-Estonia-Tartu	Tähe-4-B-201
addr-Estonia-Tartu-Tähe-4	B-201

Direct links



Context in work



Some preliminary results

Category	Release-time	After 12 months	Increase %
Taxons	3377	6038	78.8
Non-text attributes	20010	28050	40.2
Text attributes	4928	7844	59.2
Links	29586	46866	58.4

1. Evolution of the taxonomic hierarchy

Category	Release-time	After 12 months	Increase %
Containers	86	63	**-26.7**
Materials	231	317	37.2
Cont.×Mat.	19866	19971	0.5
C-M Links	755	312	**-58.7**

2. Self-optimization of the hierarchy

Results/Summary

- To construct a knowledge driven IS with AI, the following have proven to be useful for it:

1. The **regular** knowledge **mode cross-linked hierarchies** and **taxonomies**.
   - The context is focused by data in communication
   - The abstraction level is dynamically adequate
2. The concept of **IS** where it **is an actor** who just communicates with other actors
   - No hard-coded descriptions of process but reacting to input events in inputs' context
   - Figures out the role of observer

Thank You.