Educational modeling languages

ICOOL 2007 Preconference Workshop

Penang, June 11, 2007

VERSION 1

**Disclaimer: This is the first time I give this tutorial.
There may be typos and things to be improved ...**

Daniel K. Schneider

http://tecfa.unige.ch/tecfa-people/schneider.html

TECFA

**Faculté de Psychologie et des Sciences de l'Education
Université de Genève**

Code: ICOOL07-WS

Menu of the Workshop

Detailed TOC

1. Objectives and Plan

1.1. Objectives

- Familiarization with the concept « educational modeling language »
- Modeling languages in the context of educational design

1.2. Plan

- Definition of educational modeling language »
- Identification of the « problem space »
- Instructional design models
- Instructional design methods
- Overview of modeling languages
  - Content aggregation (IMS / SCORM)
  - Learning Design
  - Other
- Discussion

Important: Some slides will not be used for presentation.
They are included for (re) reading ....

1.3.What are educational modeling languages

Educational modeling languages (EML):

- define pedagogical scenarios
- are design languages.
- some incarnate norms and data standards

Objectives of EMLs (there is no general agreement):

- To formally model learner activities
- To exchange learning units (objects, scenarios)
- To execute learning units in a platform/player
- To sketch, design, plan, discuss learning scenarios

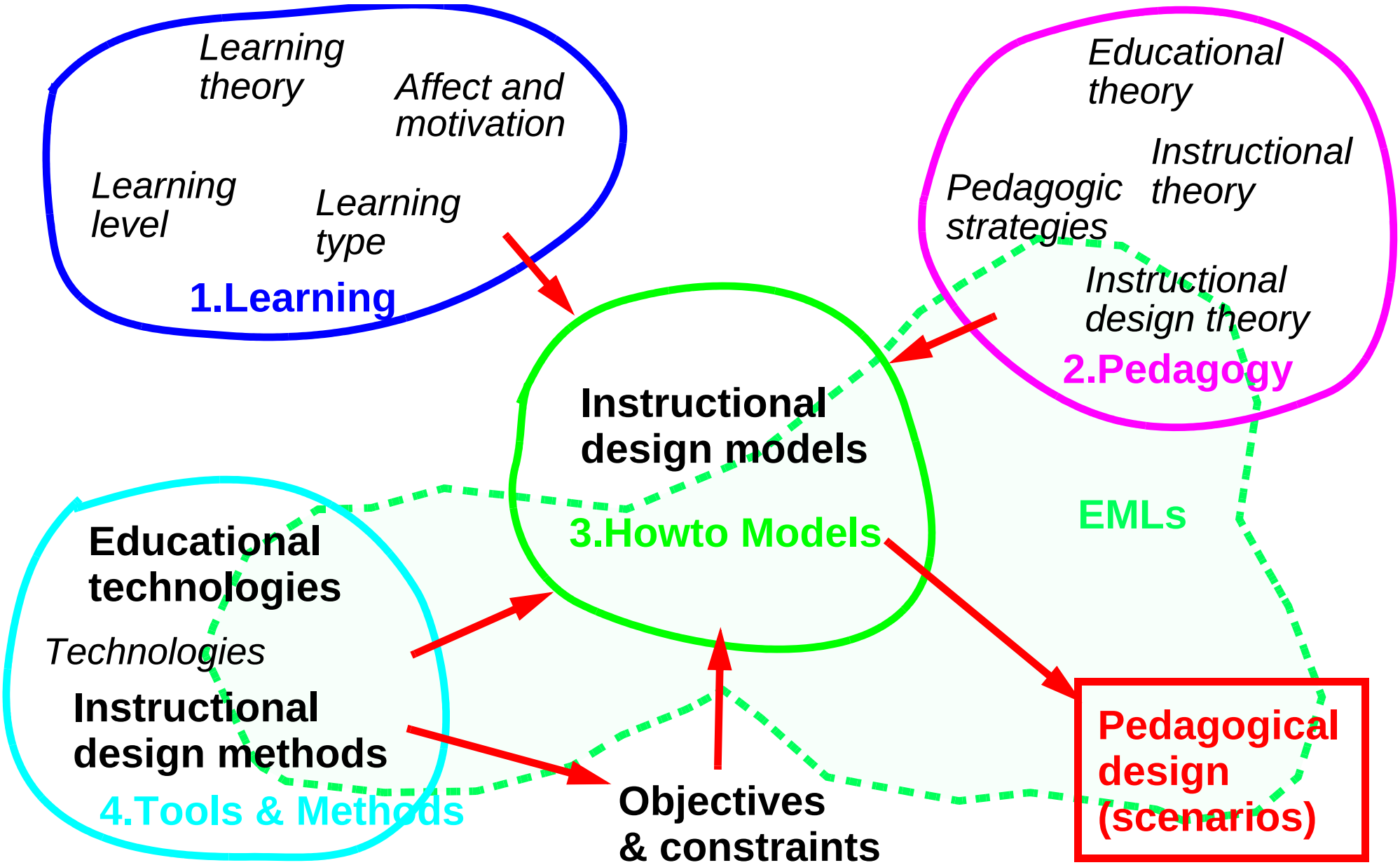
Several types of formal EMLs, for example :

- Menu-driven content aggregation: (Standalone) IMS Content Packaging
- Instructional design : Learning Design (LD) …
- Mastery learning : Simple Sequencing (SS) …

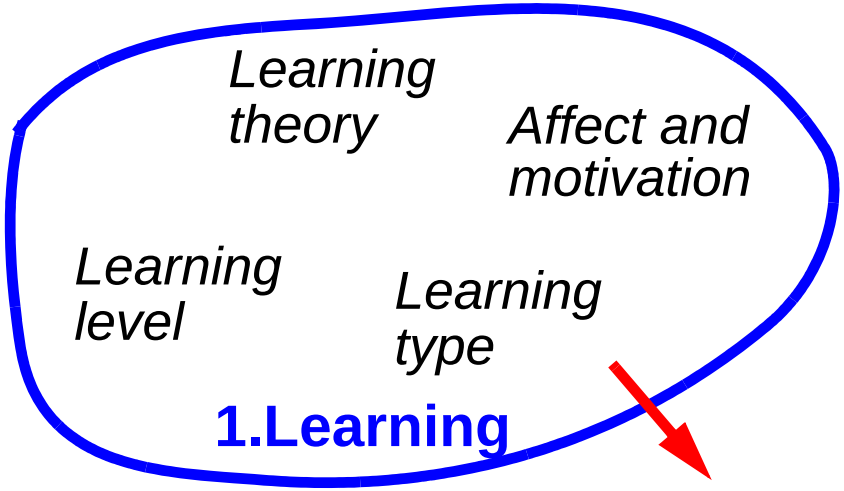
Informal EMLs

- Any kind of consistent drawing, lesson planning language, etc.

2. The problem space



3. Learning



Why bother ?

- "Learning" is a complex multi-dimensional phenomenon
- There are different learning types, learning levels etc.
- It’s not obvious to identify learning goals

  ... learning theory can help a bit, in particular its taxonomies

- Learning theory strongly influences pedagogical theory & practice

3.1.Types of Learning

1. **Attitudes:**
   - Disposition or tendency to respond positively or negatively ....
2. **Factual Information (Memorization):**
   - Processing of factual information and remembering .....
3. **Concepts (Discrimination):**
   - ... how to discriminate and categorize things. It is not related to simple recall and must be constructed.
4. **Reasoning (Inference, Deduction):**
   - thinking activities that involve making or testing inferences
5. **Procedure Learning:**
   - .... being able to solve a certain task by applying a procedure.
6. **Problem solving:**
   - identification of subgoals, use of methods to satisfy subgoals.
7. **Learning Strategies:**
   - learning how to learn, very difficult to teach !

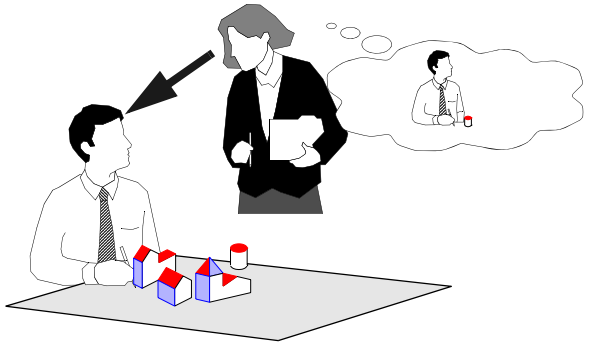
3.2.Levels of Learning

(Bloom's taxonomy of the cognitive dimension)

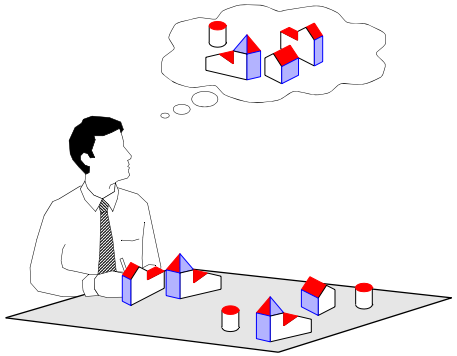
1. **Knowledge: recall data or information**
   - ... describe, identify, recall, arrange, define, duplicate, label, list, ...
2. **Comprehension: be able to translate into own words.**
   - ... give example, classify, describe, discuss, explain, express, identify, ...
3. **Application: Use a concept in a new situation**
   - ... apply, change, construct, compute, choose, demonstrate, write, ...
4. **Analysis: split concepts into parts and understands the structure**
   - ... analyze, break down, relate, appraise, calculate, categorize, compare, ...
5. **Synthesis: Produce something from different elements**
   - ... summarize, arrange, combine, categorize, assemble, collect, compose, ...
6. **Evaluation: Make judgements, justify a solution, etc.**
   - ... appraise, interpret, argue, assess, attach, compare, defend, predict, ...

3.3.Leaning theories

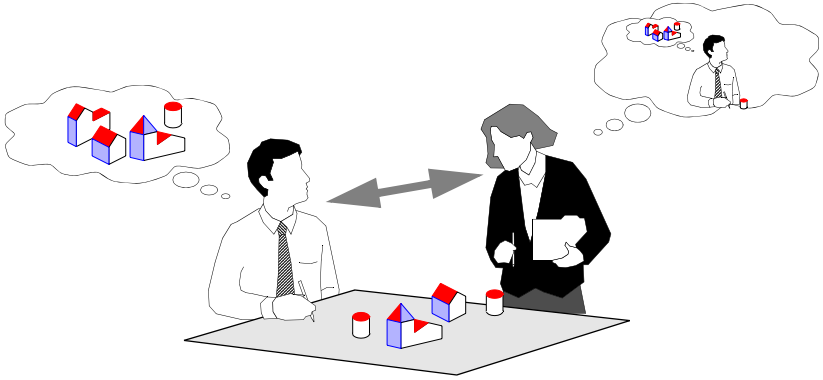
.....look at learning in different ways ... lead to different designs



behaviorism
(change behavior)

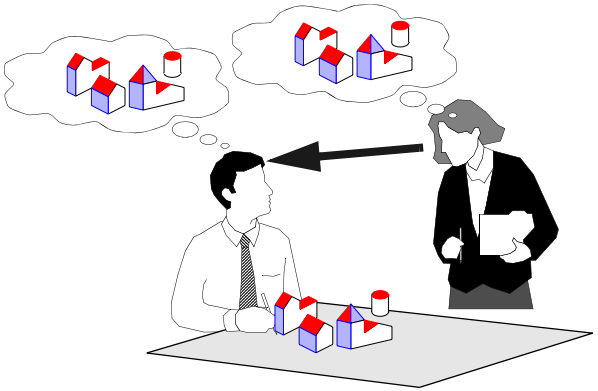


constructivism
(construct knowledge)

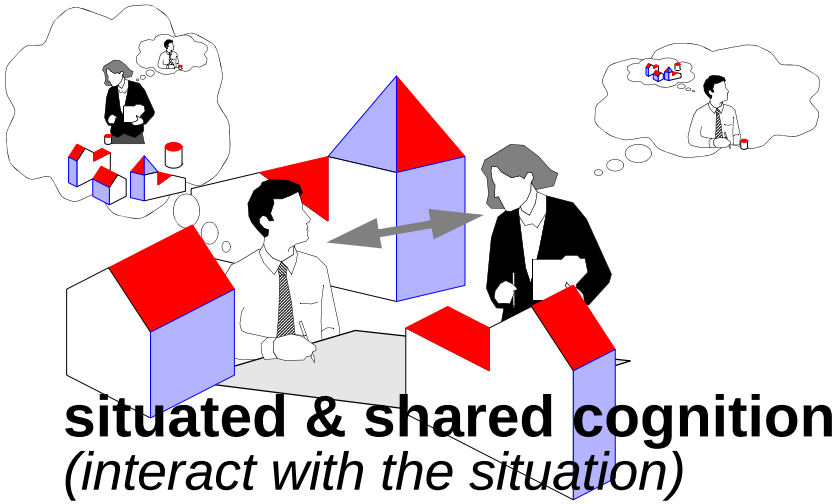


social cognition
(interact with others)

higher learning levels

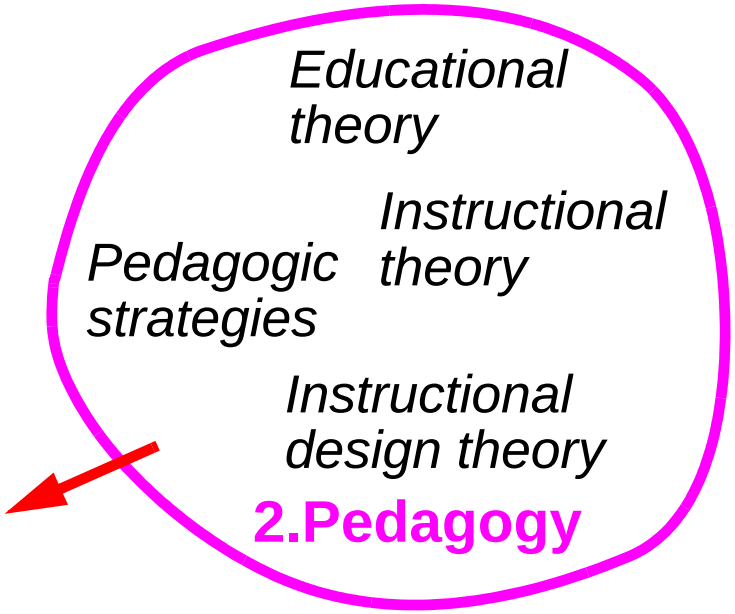


cognitivism
(reach knowledge objectives)



situated & shared cognition
(interact with the situation)

4. Education, pedagogy, instruction



What does pedagogy tell us ?

- There are very few generally applicable principles !
- Different pedagogies relate to different learning goals & types & levels

  ... Therefore try to have a "vocabulary" of strategies and tactics

4.1.Major pedagogical approaches (strategies)

(**Baumgartner & Kalz**), there are many other typologies ...

Learning I ***Transfer***	***Learning II*** ***Tutor***	***Learning III*** ***Coach***
Factual knowledge, "know-that"	**Procedural knowledge, "know-how"**	**Social practise, "knowing in action"**
Transfer of propositional knowledge	**Presentation of predetermined problems**	**Action in (complex and social) situations**
to know, to remember	**to do, to practise**	**to cope, to master**
Production of correct answers	**Selection of correct methods and its use**	**Realization of adequate action strategies**
Verbal knowledge, Memorization	**Skill, Ability**	**Social Responsibility**
to teach, to explain	**to observe, to help, to demonstrate**	**to cooperate, to support**
Teaching I	***Teaching II***	***Teaching III***

(see "Overview of pedagogic strategy models" [p. 18])

4.2.Pedagogical strategies and methods?

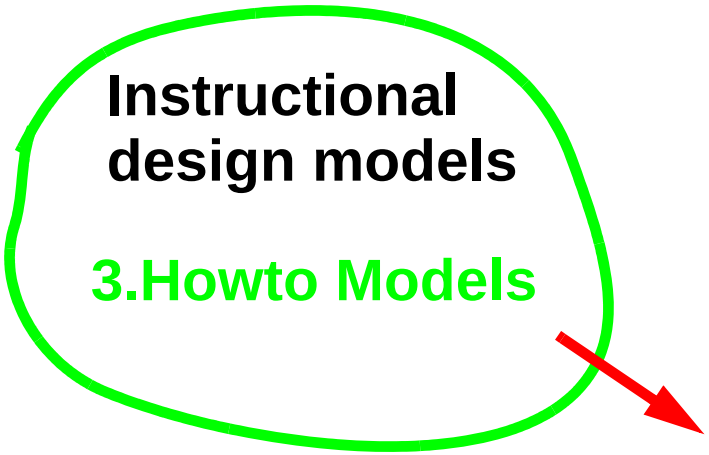
There are many !

E.g. Khan's (2000) list of Methods and Strategies

Presentation	Exhibits
Demonstration	Drill and Practice
Tutorials	Games
Story Telling	Simulations
Role-playing	Discussion
Interaction	Modeling
Facilitation	Collaboration
Debate	Field Trips
Apprenticeship	Case Studies
Generative Development	Motivation

... some (combinations of) strategies work better for some learning goals

5. Instructional design models

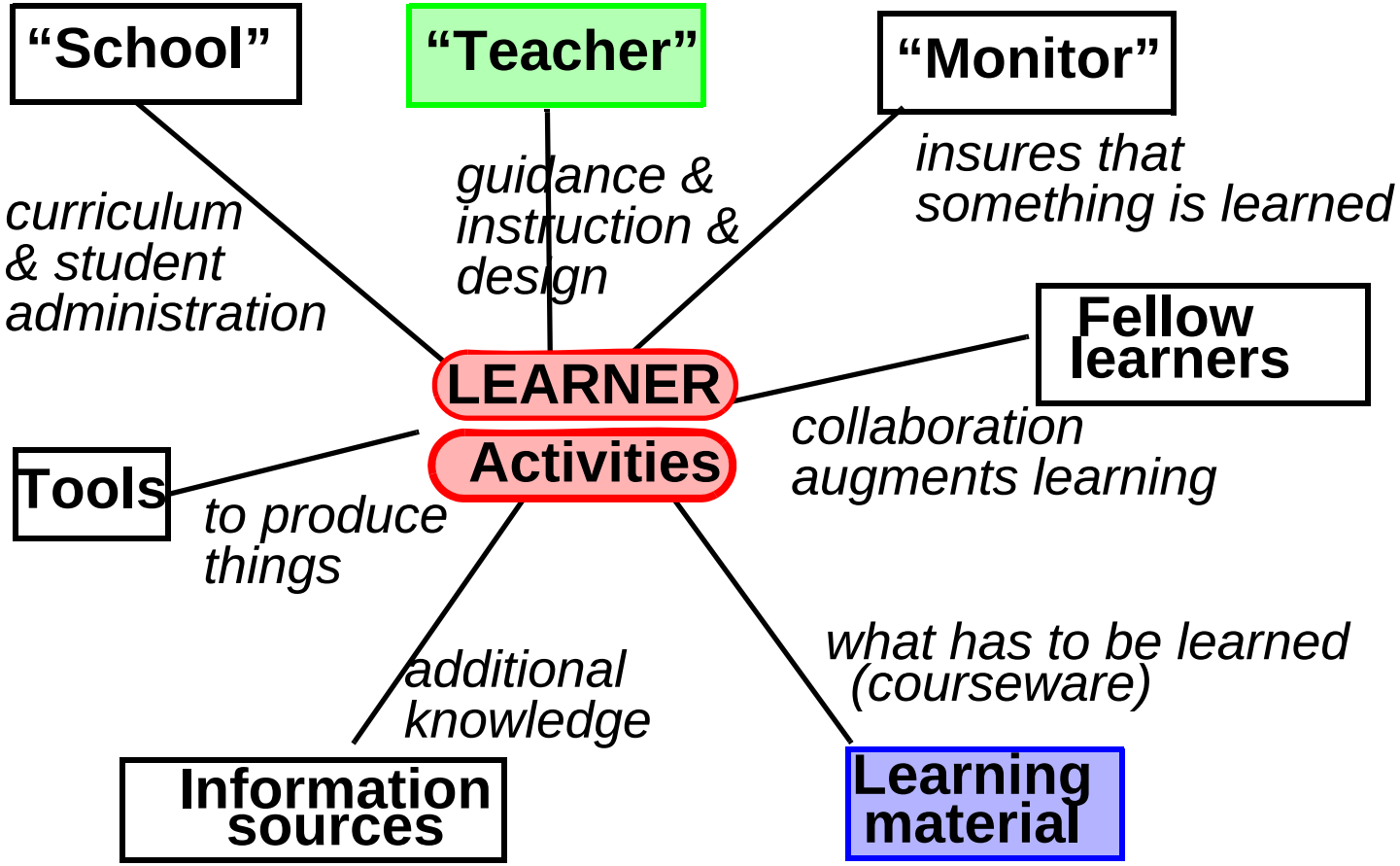


Types of instructional design models:

1. **Functions of a learning environment**
2. **Quality of a design *(not in this talk)***
3. **Complementary models that will enhance a design *(not in this talk)***
4. **Change management models *(not in this talk)***
5. **Pedagogic strategy models**
6. **Instructional design methods: how to implement a design (later)**

EMLs may model 3 and 5 and relate to 1 and 6

5.1. Functions of a learning environment



modified from Sandberg

- Focus can change
  - E.g. **teacher role** is central in activity-based designs
  - E.g. **Learning material** is important for mass-education

5.2.Overview of pedagogic strategy models

- how to organize appropriate pedagogical scenarios to achieve instructional goals.
- a kind of abstract design rule for a given instructional design approach or a given pedagogic strategy.

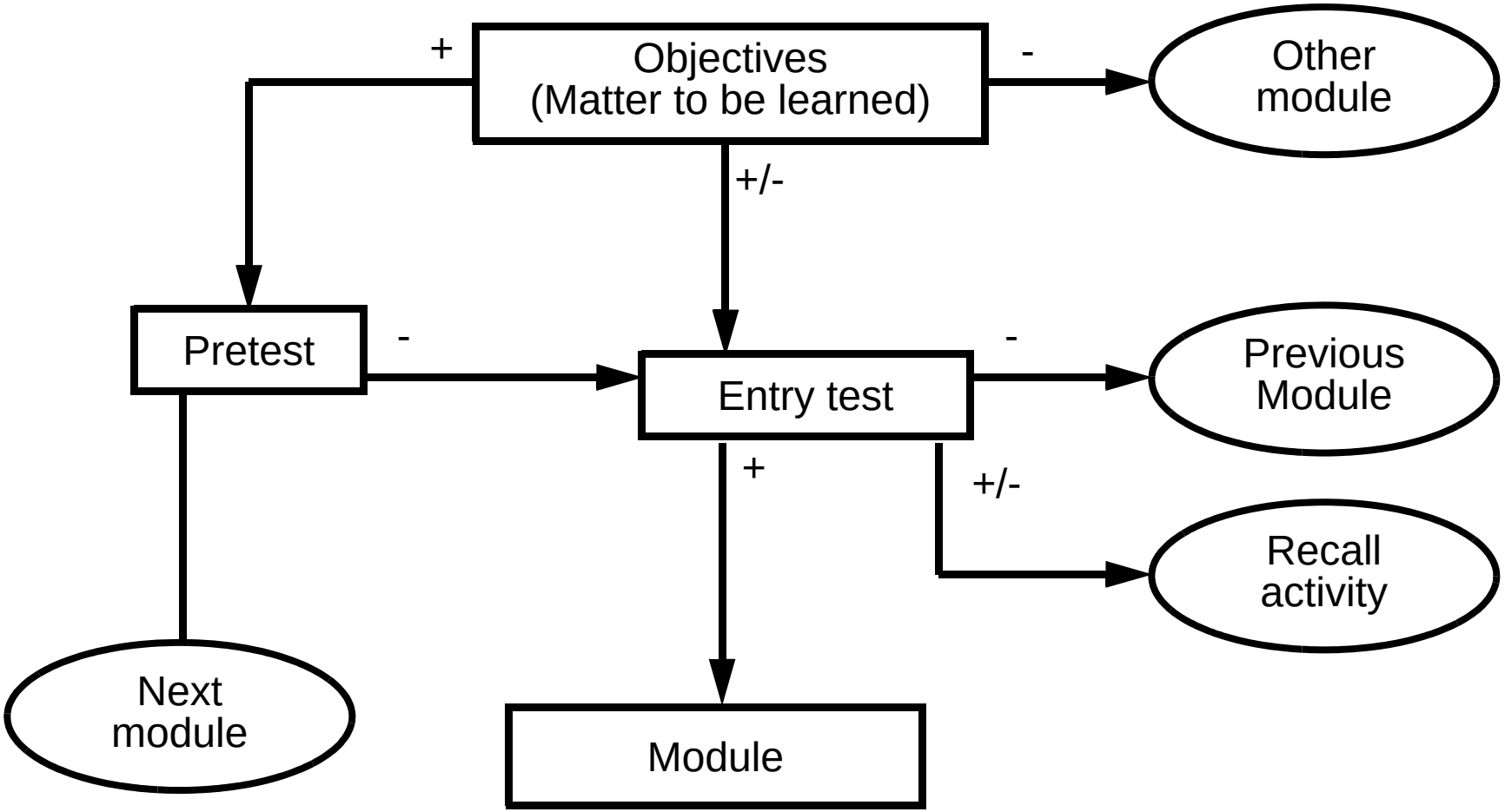
There are dozens: a *few* models sorted by 2 dimensions:

	formal	*open / informal*
Learning I (information)	**Lecturing, "page turners", drill & practise, ...**	**on-demand tutorials, handbooks, ...**
small scale Learning II (know how)	**exercising, e-instruction, simulations, ...**	**on-demand e-instruction, self-learning with textbooks, ...**
large scale Learning II (know how)	**Problem-based learning, Inquiry-based learning, simulation & gaming,...**	**help desk model, on-demand tutoring, knowledge management,...**
Learning III (knowing in action)	**Project-based learning formal e-portfolios ....**	**Communities of practice Mentoring, ...**

5.3.E-instruction model

Part 1: A modular course architecture

- collections of reusable learning objects, i.e. focus on content aggregation



Part 2: Module design

- Should follow (some) sound behaviorist/cognitivist design principles, e.g. Gagne's 9 steps of instruction

a. Gain attention e.g. present a good problem or a new situation.

b. Describe the goal: e.g. state what students will be able to accomplish.

c. Stimulate recall of prior knowledge (facts, rules, procedures or skills) and show how knowledge is connected.

d. Present the material to be learned e.g. text, graphics, simulations, figures, pictures, etc.

e. Provide guidance for learning (instructions on how to learn on a different channel)

f. Elicit performance "practice", let the learner do something with the newly acquired behavior, practice skills or apply knowledge. At least use MCQ's.

g. Provide informative feedback, show correctness of the trainee's response, analyze learner's behavior, etc.

h. Assess performance test and also more general progress information

i. Enhance retention and transfer: inform the learner about similar problem situations, provide additional practice. Put the learner in a transfer situation.

5.4.Inquiry-based learning model

Purpose: concept learning and investigation methodology

Typical cycle of student activities (done several times)

Ask
students ask questions

Investigate
collect information
observe
experiment

Create
synthesize
make links
write report

Discuss
share ideas
confront ideas

Reflect
examine results
and look at questions again

5.5.Project-based learning model

- Main purpose: "deep learning" (applicable knowledge)

Outline of the model defined by "Moursund":

- A typical project has 4 phases:

(1) Getting started

a.Define the topic of overall course

b.Define timelines, milestones and assessment methods

c.Identify resources

d.Identify prerequisites

e.Advance organization (project-methodology, skills that will have to be acquired etc.)

f. form teams

(2) Initial Team Activity - Project Planning

a.Knowledge pooling by team members

b.Initial project specification, e.g. formulate objectives and questions. At university level, this should lead to a research design.

c.Planning, e.g. definition of work packages, milestones and timelines

d.Formal teacher feedback

e.Revision of the project specification and plan (if need return to steps 2 and 3)

(3) Project Implementation

a. **Have students complete one task and milestone at a time. Make sure that students engage in regular meetings**

b. **Refining of project definition**

c. **Sharing between team members (make sure that there is collaboration and cooperation, you decide)**

d. **Provide feedback (this includes peer-to-peer tutoring, global feedback to the class for all projects, etc.)**

e. **Move toward completion.**

f. **Repeat all steps until all milestones have been met**

(4) Completion

a. **Students have to polish the final product and prepare associated presentations.**

b. **Assessment: The whole class should assist at the presentation of the results. Students may have the occasion to integrate a last feedback.**

c. **Closing session with the whole class discussing the experience**

Notes:

- There are many variants of this kind of model
- ICT-enhanced models put more emphasis on intermediary products and exchange activities (see next)

5.6.Structured project-based learning model (variant)

Level 1: Teacher roles and the overall design

- Strong story-boarding, projects are done individually or in small groups
- Teacher role is crucial & complex

Teacher as orchestrator

designs the environment
designs the global project
designs flexible tasks
designs exchange activities

Teacher as monitor

makes audits
reads blogs
controls project plans
evaluates

Teacher as facilitator

gives feedback
answers questions
writes tutorials
provides examples
provides links

Projects

Scenario 1

Goals and questions
stage 1
stage 2
stage 3 ...
output: objectives+
research questions

discussion at class level

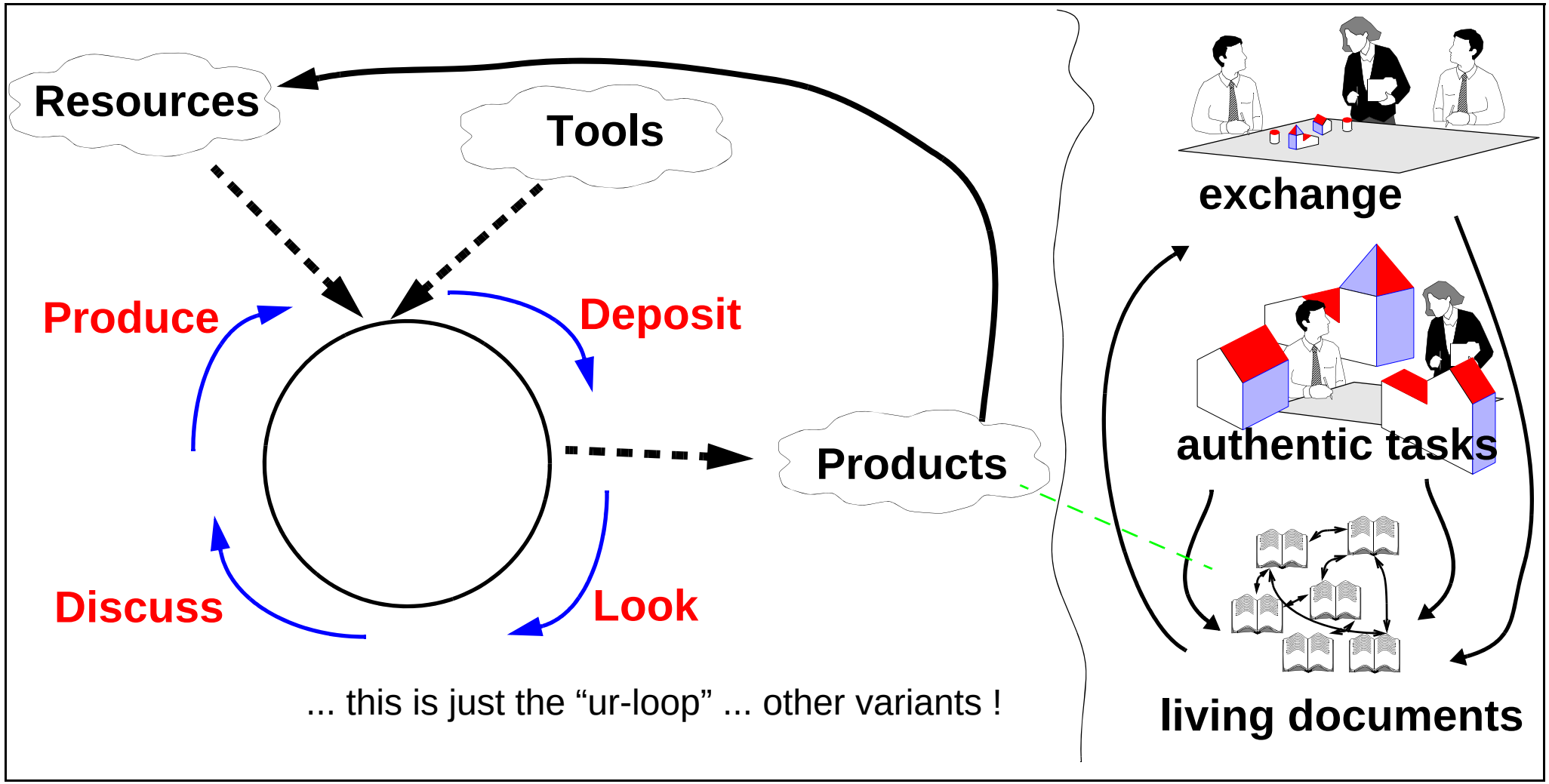
Scenario 2

R & D plans
stage 1
stage 2
stage 3
output: plan v.1

.....

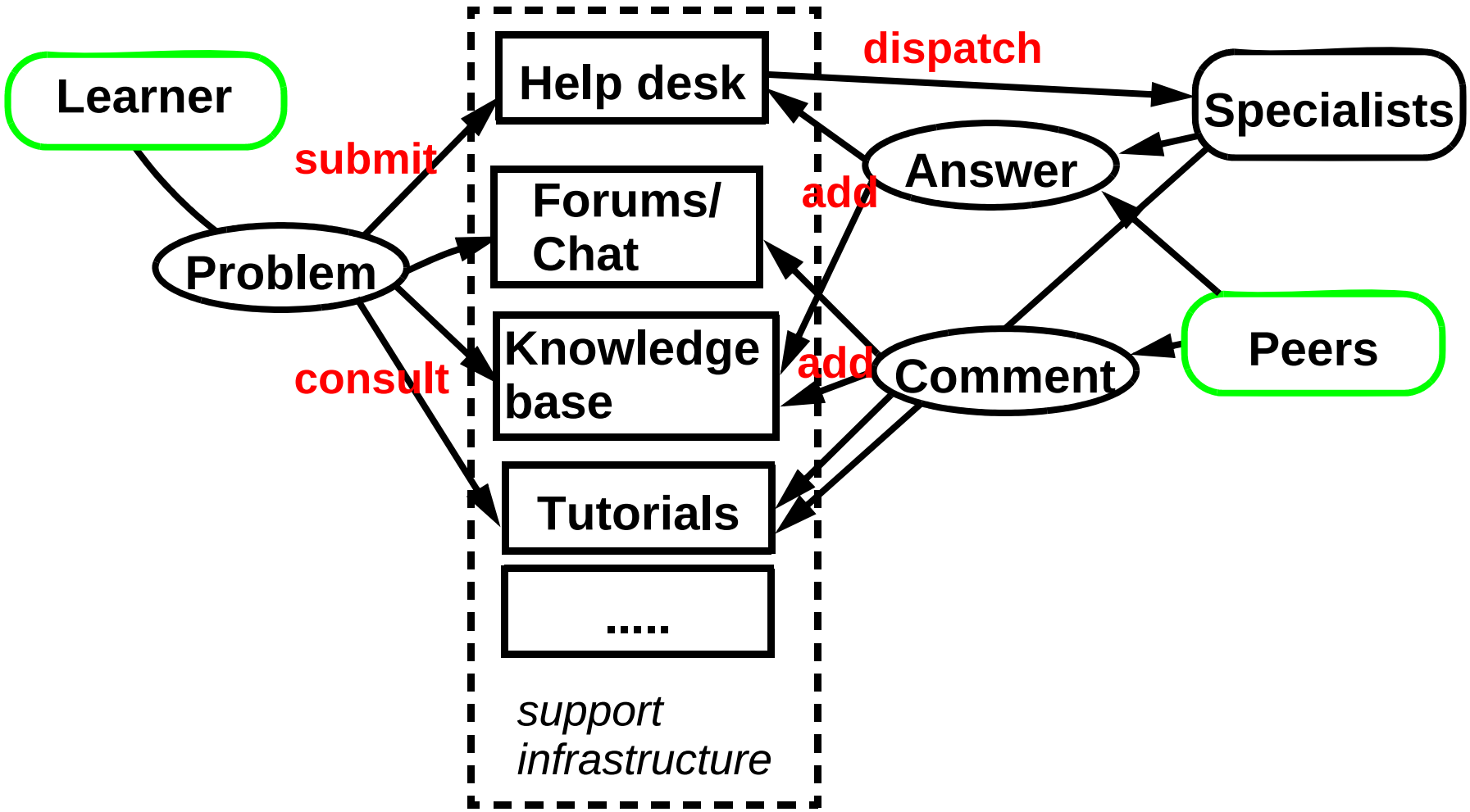
Level 2: Scenario orchestrations with workflow loops and ICT:

- Scenarios are **sequences** of **activity phases (stages)** within which group members **do tasks** and **play specific roles**
- Each activity leads to a product that can be discussed and reused



5.7.A "help desk/knowledge management model" for life-long learning

- Purpose: support informal workplace learning
- Needs involvement of several organizational units



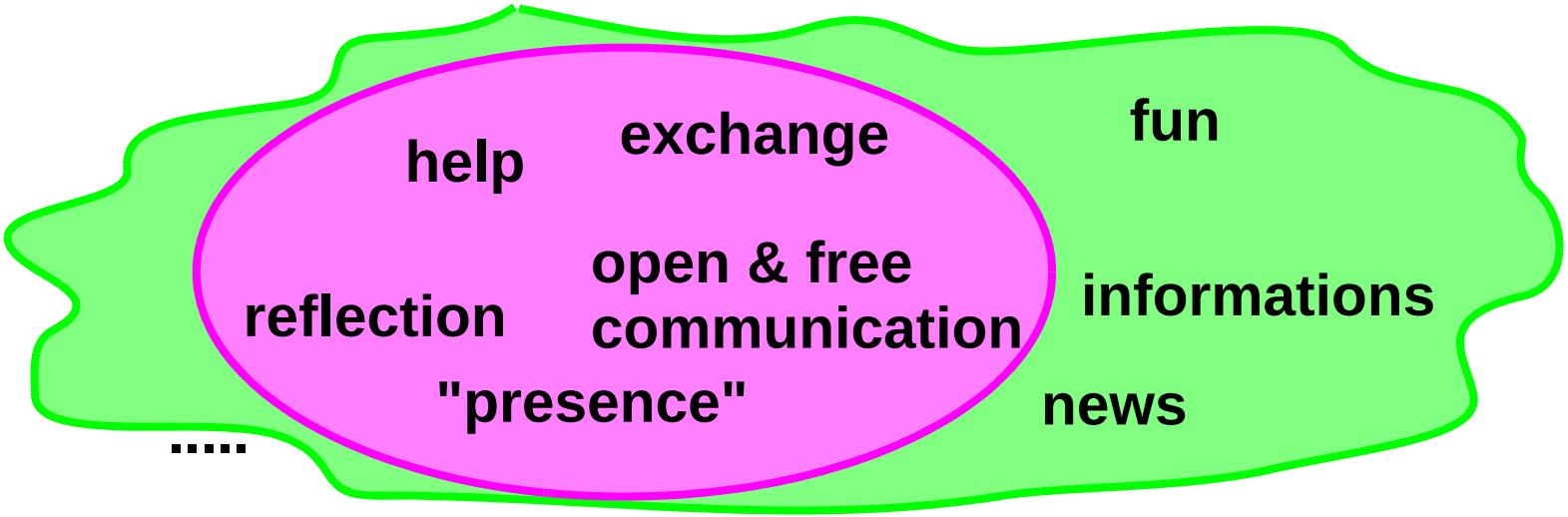
5.8.Community of practise model

Purpose: Engage actors in communities that learn

- often used in professional development, e.g. teacher training
  - members of a community tend to make better progress
  - knowledge through **enculturation** (collective memory)
  - good communities are **knowledge management** aware

Use of portal software

- A place to find informations, news etc.,
- A place to exchange, to reflect, to be, ...



... difficult to set up (people don't collaborate naturally)

5.9.Mentoring

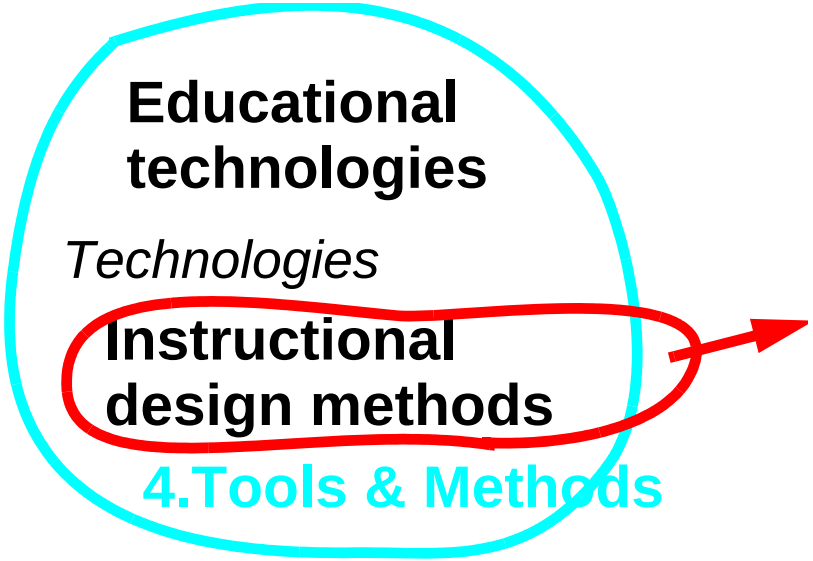
Purpose: Formal or informal "elder" to "younger" training

1. **Conditions: Mentoring works when:**
   - individuals are committed to it,
   - there is a goal (see the mentoring contract),
   - there is a supportive environment.
2. **Stages to set up a mentoring program**
   - Identify development needs of protégé(s)
   - Identify and recruit mentor(s).
   - Prepare/train mentor.
   - Mentor and protégé negotiate a mentoring agreement (see below).
   - Carrying out (implementation can include meetings with a facilitator).
   - Evaluation.
3. **Contract:**
   - Mentors and proteges should agree on a formal contract.
   - Contents: development objectives, roles and expectations, duration, ground rules, other comments.
   - Both have to sign it (and believe in it).

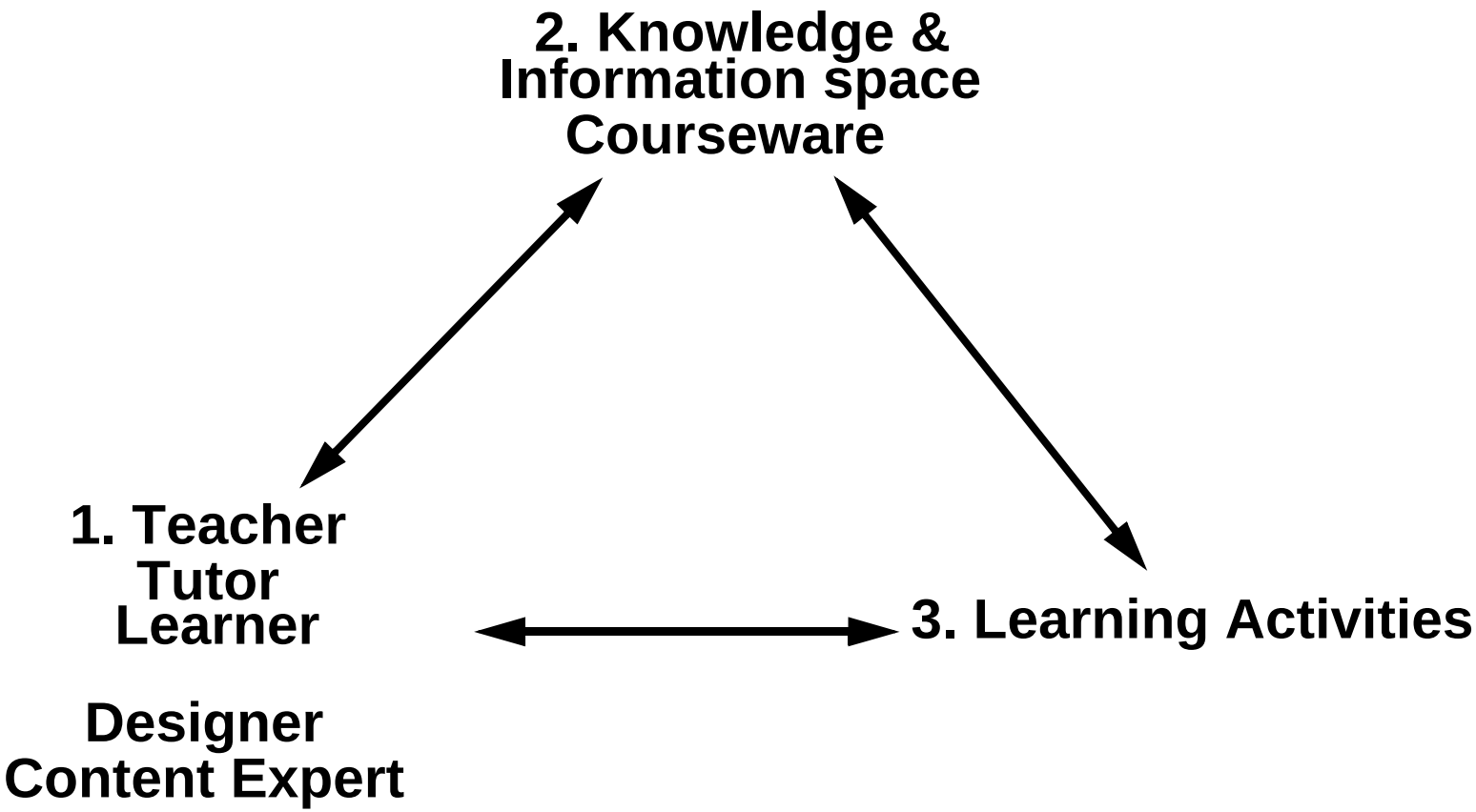
6. Tools & Methods

Methodologies for design and implementation

- There are ***many*** instructional design methods (also called "instructional design models")
- Norms and standards
- Modeling languages (some based on norms and standards)
- Technologies for ***design***, ***authoring***, delivery, and repository



6.1.Plan at least for 3 areas

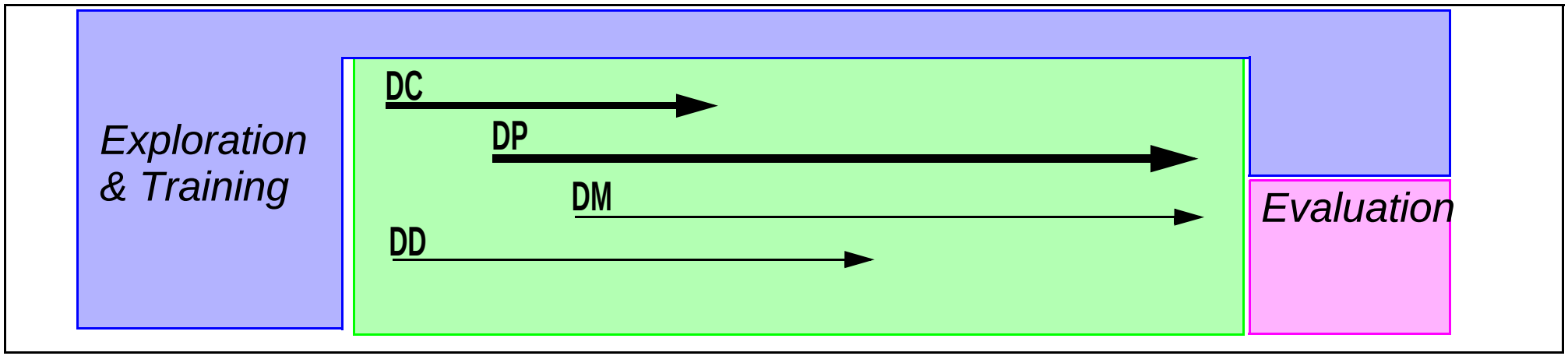


1. **Roles, structures and relations between actors:**
   - who does what and where (before, during and after)
2. **Contents and knowledge**
   - what to learn, materials, knowledge creation)
3. **Learning activities**

6.2.Instructional Design Method (1): heavy duty industrial design ?

MISA/MOT/ADISA: Course designer works on "4 models"

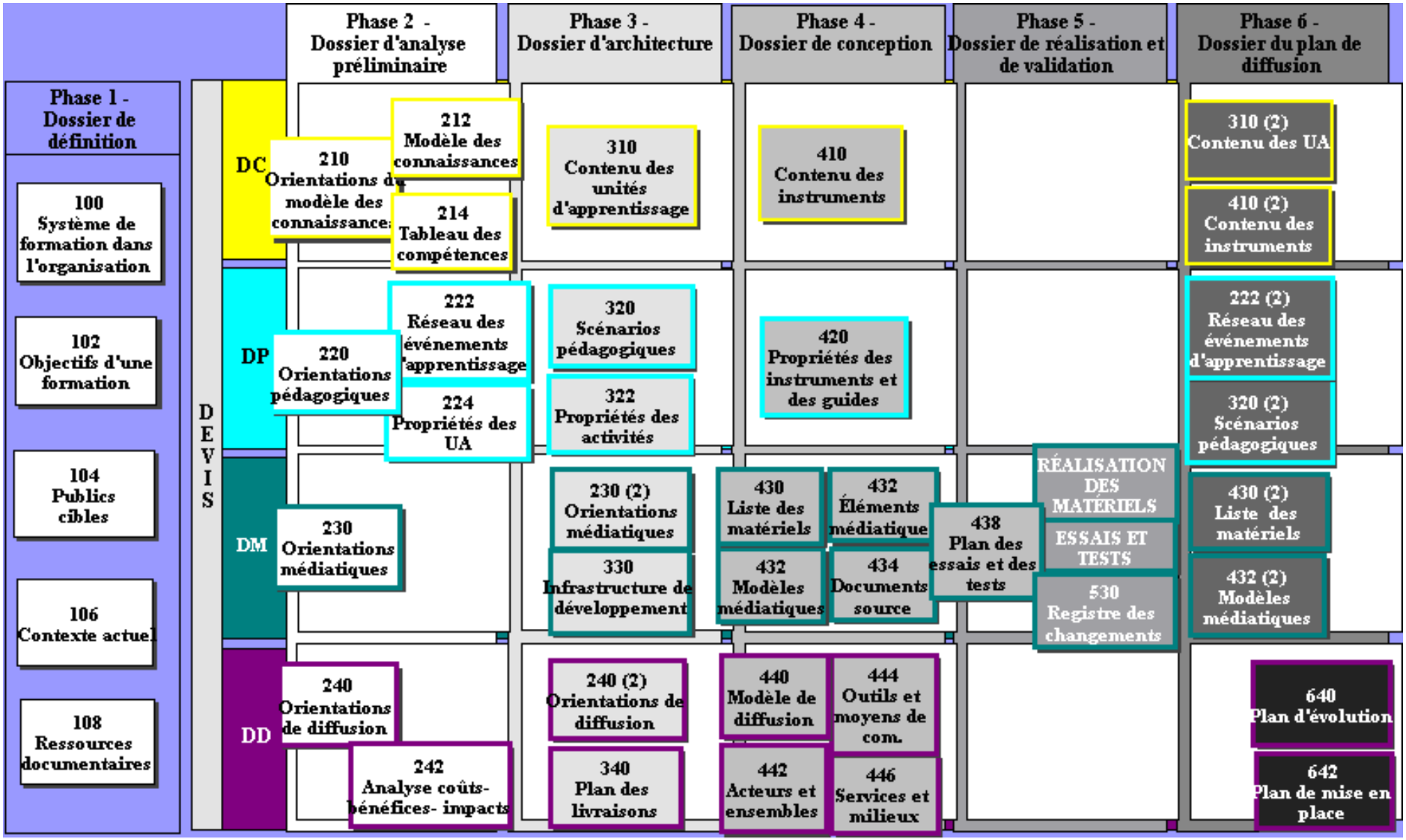
1. **Knowledge and Skill Representation**
   DC: Design of Content (know-that and know-how)
2. **Application of Teaching Methods and Approaches**
   DP: Design of Pedagogical specifications
3. **Specification of Learning Materials**
   DM: Design of Materials
4. **Delivery Planning**
   DD: Design of Delivery



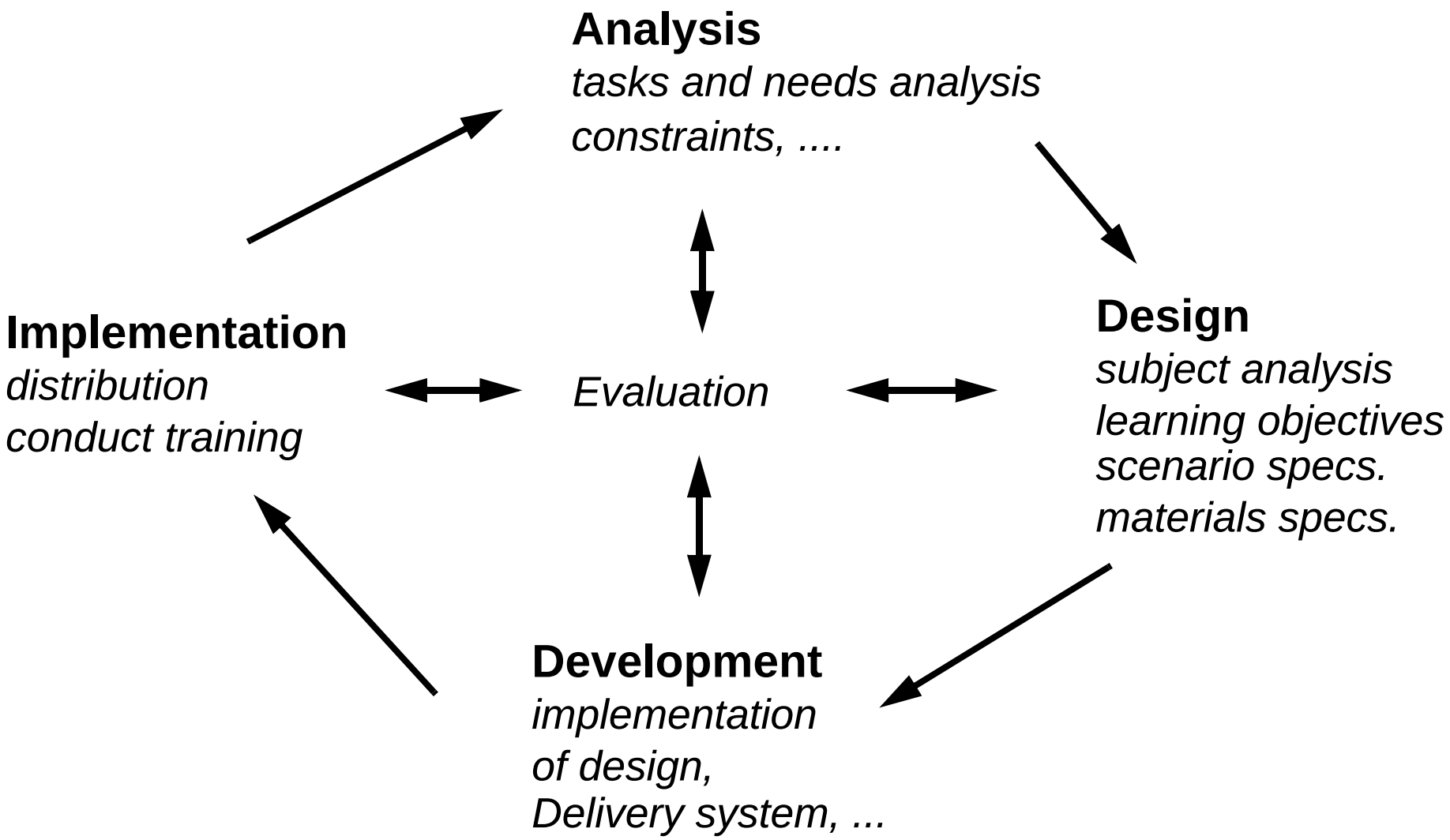
- Worth the effort for large scale projects and if you have manpower

url: **http://www.cogigraph.com**

The complexity of the MISA model ....



6.3. Instructional Design Method (2): Instructional systems design



E.g. ADDIE-like model

6.4.Instructional Design Method (3): Rapid participatory prototyping

Principles:

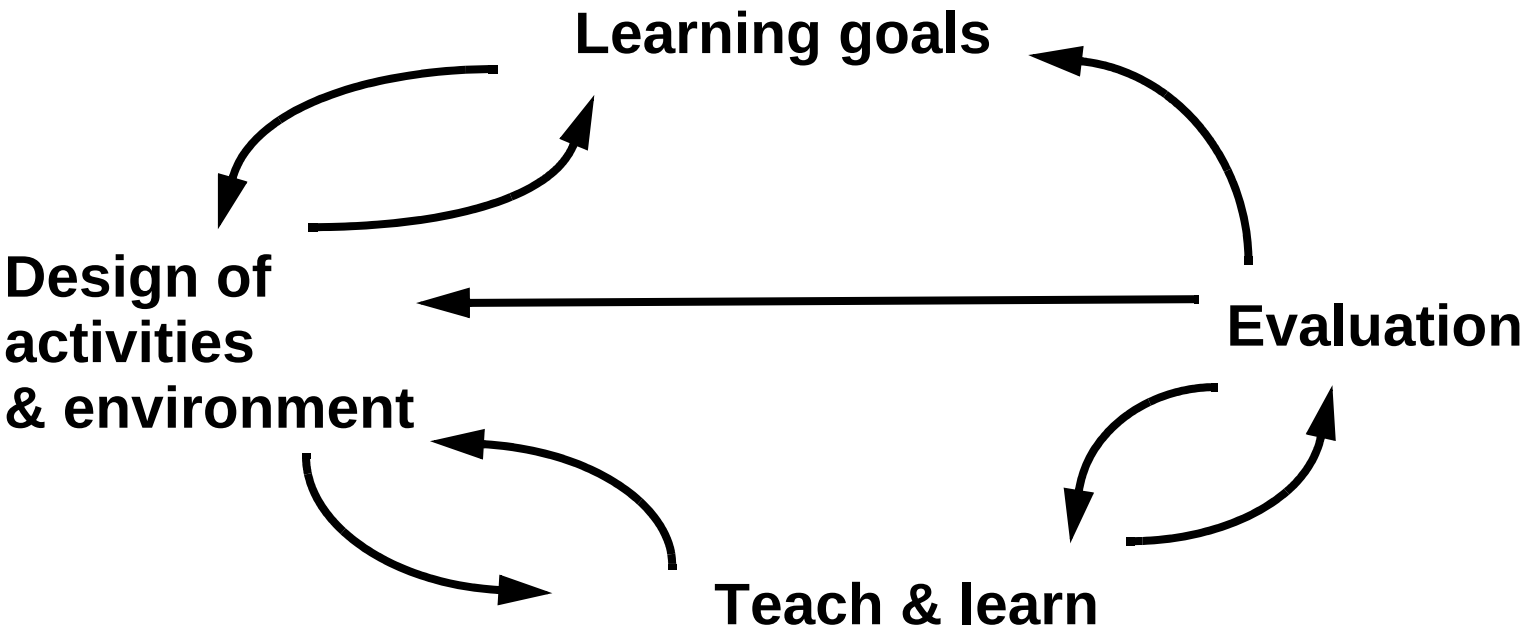
a. Cooperating: all stake-holders (including learners) participate

b. Experimenting: ideas are or may be embedded within current conditions, e.g. they may emerge during teaching

c. Contextualizing: Setting and situations are particular, e.g. each class may evolve differently

d. Iterating: The not-yet-known, draws from the point of view of use

The implementation cycles:



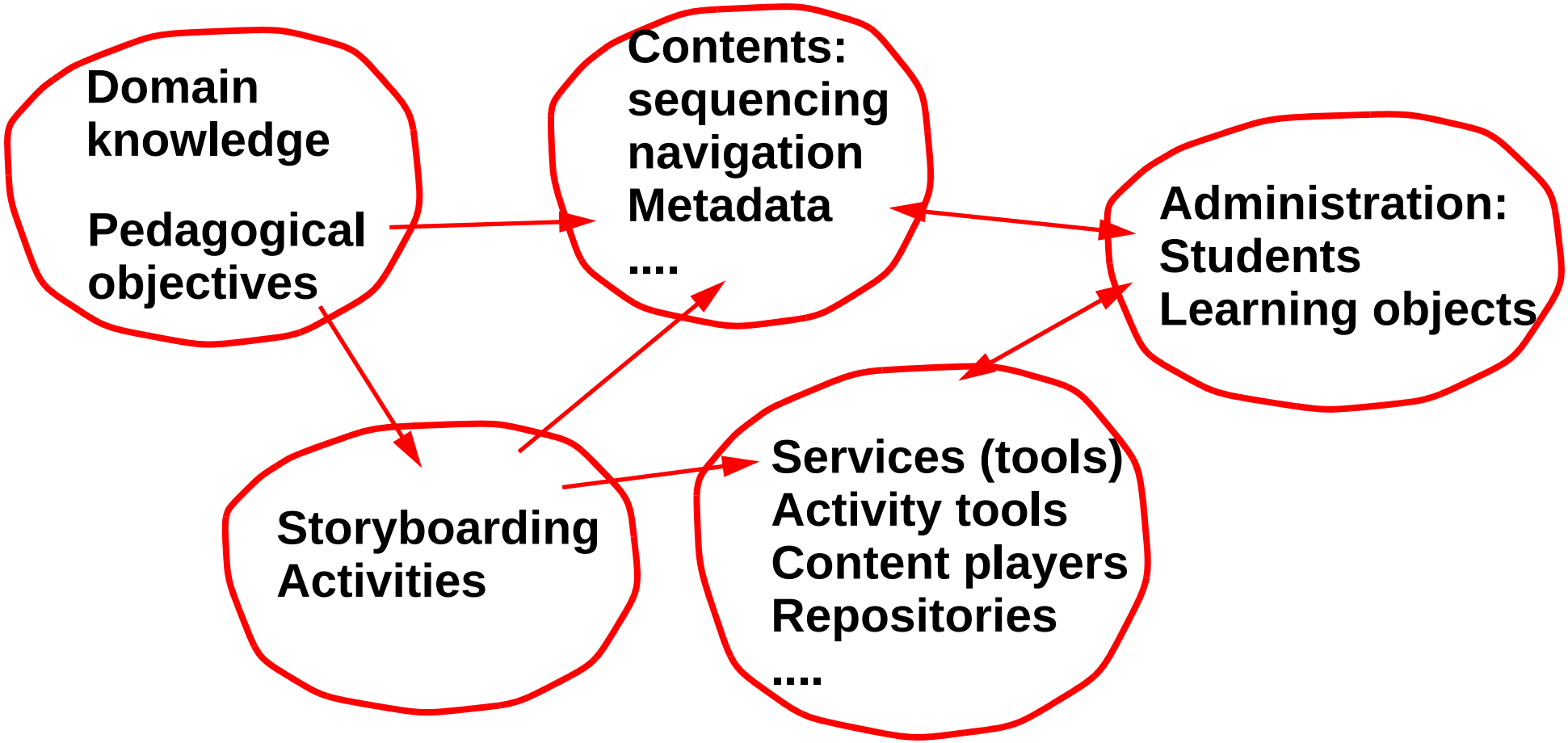
6.5.Technology (not part of this talk)

- **Technology is not innocent !**
- Do not trust vendors who make universal claims ....
- Use different design languages for different problems (see later)

	Teaching I know-that	Teaching II know-how	Teaching III knowing-in-action
E-learning Systems	***	*	
Hypertext, Wikis, CMS (exploring, reading)	***	*	
Groupware (help desk, discussion)	*	***	**
Microworlds (exercising, simulating)		***	*
Hypertext, Wikis, CMS (producing, collaborating)	**	*	***
C3MS (community portals) and web 2.0 services	*	*	***
Comp. Supp. Coll. Learning	*	***	*

7. Standards and EML Overview

7.1. The global picture



- Any of these "elements" can be modeled
- Relations between these "elements" can be modeled

7.2.Are there instructional norms and standards ?

- Pedagogical standards ?
- Instructional design methods like MISA or ADDIE implicitly define what a good pedagogical design might be...
- Curricular standards, e.g.
  - American Association for the Advancement of Science's (AAAS, 1993) Benchmarks for Science Literacy
  - National Research Council's (NRC, 1996) National Science Education Standards,
- Data Standards (e.g. IMS / SCORM introduced later), for example:
  - Learning objects: IMS Content Packaging
  - MetaData: IEEE Learning Object Metadata Standard (LOM)
  - Modeling of sequencing and activities: IMS Learning Design and IMS Simple Sequencing
  - IMS Question and Test Interoperability (QTI)

Educational modeling languages:

- Make pedagogical designs explicit
- Can model learning sequences, learning activities, domain knowledge, etc.
- Do not explicitly impose pedagogical designs, but do so implicitly by supporting given kinds of scenarios, learning materials, etc.

7.3.Major standards players

Organization	***Standard***
IEEE (Learning Technology Standards Committee)	LOM Metadata standard for the reusability and interchangeability of LO's
ADL ***Advanced Distributed Learning (US Gov)***	Define various SCORM profiles In the past: proper standards, today mostly adoption of IMS and IEEE norms. Norms to implement systems.
ARIADNE ***Alliance for remote Instructional Authoring and Distribution Network for Europe (EU)***	Metadata Standard for Interoperability (outdated, integrated and replaced by LOM)
DCMI ***Dublin Core Metadata Initiative (international organization)***	DC Metadata specifications and content structure modeling Light-weight alternative to LOM, popular for document repositories
AICC ***Aviation Industry CBT Committee (International organization)***	Content structure modeling (part of older SCORM profiles)
IMS ***IMS Global Learning Consortium, Inc.***	Metadata specifications Simple Sequencing Information and Behavior Model Content Packaging Specification Learning Design ... more

7.4.What can be modelled in education ?

Three main areas:

1. **Domain knowledge + knowledge and skills to be learnt**
2. **Component aggregation / Sequencing of learning materials (SCORM)**
3. **Learning activities (Learning Design)**

Additional areas:

- Architecture of computer systems
- Repository information (metadata)
- Learner information
- .....

To model, we are looking for

- ***design languages*** ...that can precisely describe a design
- machine readable languages to represent or even run a model
- a way to distribute modeled "stuff" (see: learning objects, units of learning)
- "players"/systems to execute some "stuff"

7.5.Kinds of design languages

- A design language is some kind of formalism to describe a class of artifacts, e.g. a computer software, pedagogical scenarios, knowledge.
- A design language is not necessarily formalized or standardized....

Criteria to look at various design languages

1. **Complexity**
2. **Precision**
3. **Formality & standardization**
4. **Personal vs. shared**
5. **Implicit vs. explicit**
6. **standardized vs. non standardized**
7. **computability (can it be executed ?)**

Source: Gibbons et Brewer

c.f **http://edutechwiki.unige.ch/en/Design_language**

7.6.Kinds of pedagogical design languages

Standardized executable languages

- Explicit standards, including formal languages
- Often: authoring tool, distribution tool and players
  - Most popular: IMS / SCORM
- Some lack good tools support (E.g. IMS Learning Design)
- Execution mechanism is not necessarily standardized
  - Means that you can’t use them yet (SCORM CAM is an exception)

Other executable languages

- 2 Versions: formalized vs. not formalized
- Often they are supported by an authoring tool and a player
- May offer export functionality, e.g. to IMS formats

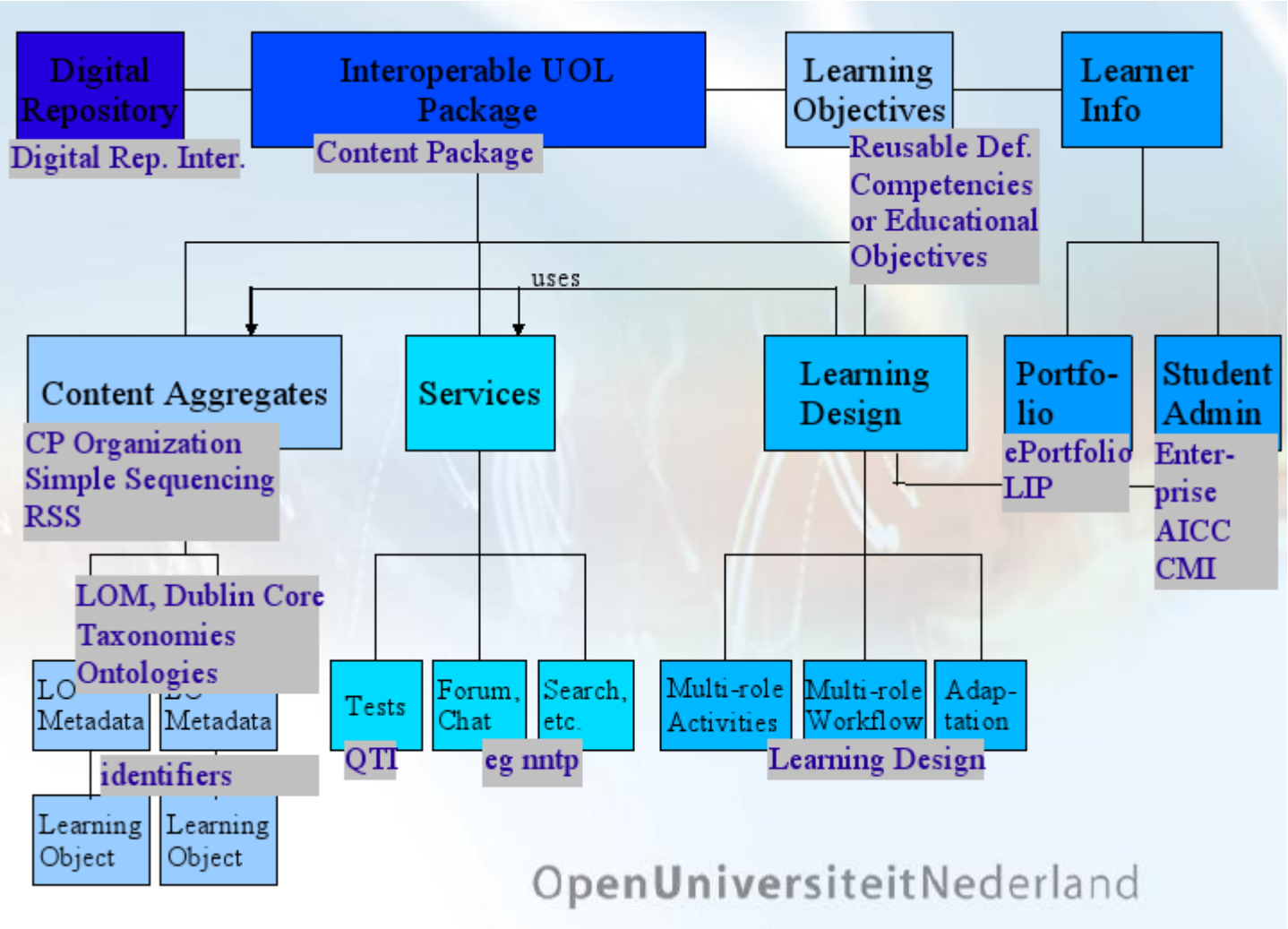
Informal design languages

- Scenario building toolkits
- Many lesson planners ...

Research: .... lots !

7.7.Mainstream standards I

(According to Rob Koper, main inventor of IMS Learning Design)



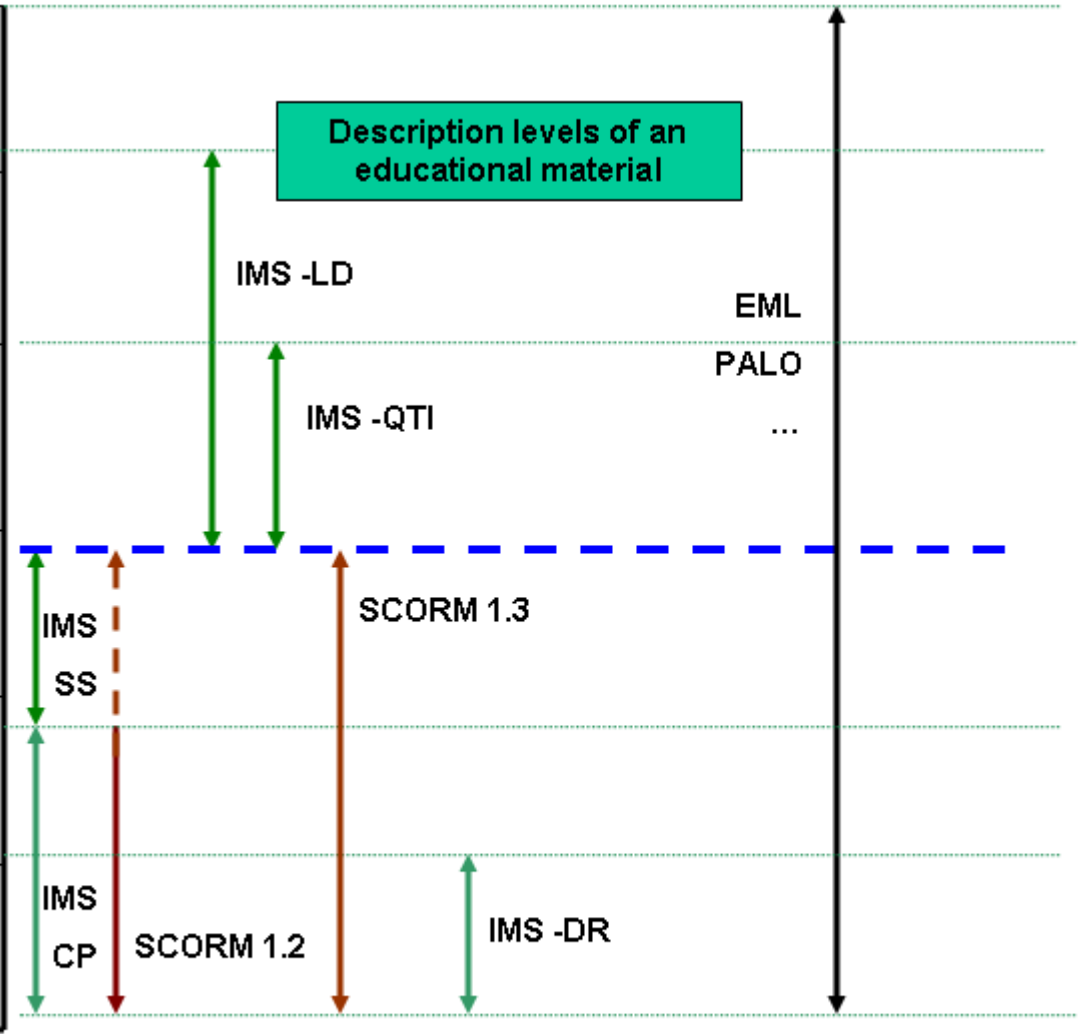
UOL = Unit of learning

Grey boxes represent standards or equivalent

... focus on "IMS Galaxy"

7.8. According to Miguel R. Atacho (creator of PALO)

Management	Interoperability parameters with LMS
Pedagogical/Instructional	Pedagogical Information
Activity/Task	Educative processes y activities. Collaborative tasks and activities
Sequencing	Sequencing, prerrequisites, deadlines, dependencies
Structure	Navigational model
Content	Small LO's, assets and formatted content



Miguel R. Artacho – UNED – miguel@lsi.uned.es

... focus on "IMS Galaxy"

8. Modeling (1) - general purpose design languages

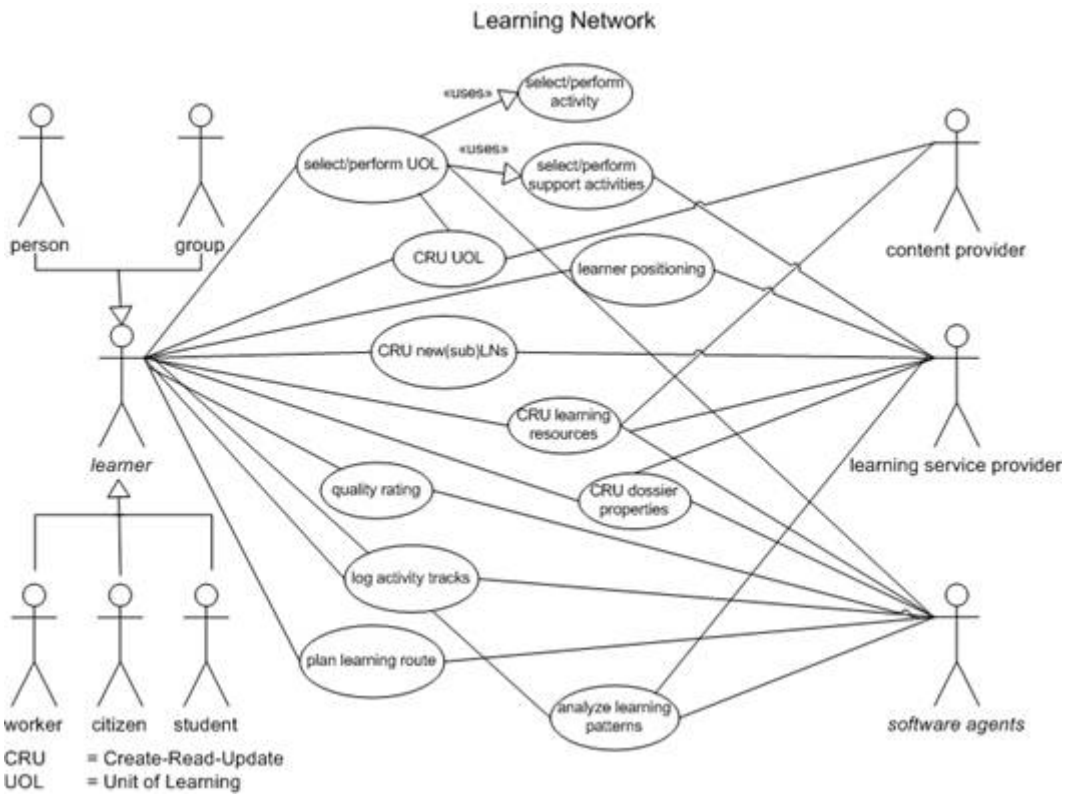
... any appropriate design language can be used to model pedagogical "things"

8.1. UML

- The Unified Modeling Language (UML) is a object modeling and specification language mostly used in software engineering.
- UML includes a standardized graphical notation for several diagram types:
  - ***Use case*** models and scenario's capture the user requirements and functionality of the system.
  - ***Class and object diagrams*** specify the structure of a system.
  - ***Activity diagrams*** to specify workflows.
  - ***State diagrams*** describe the dynamic behavior of an object in a system.
  - ***Interaction diagrams*** (sequence and collaboration diagrams) model how groups of objects collaborate in some behavior.
  - ***Physical diagrams*** (deployment and component diagrams) to model the implementation structure of a system.
- Some educational modeling languages are first of all described as UML diagram, e.g. the semantic information model of IMS Learning Design has been expressed in UML.
  - ... this is why I first present UML ....

8.2.UML Use Case

- Describe behaviors of actors
- Example: Rob Koper's Learning Network diagram



- The figure specifies several use cases, i.e. the activities performed by the actors, represented with oval boxes.

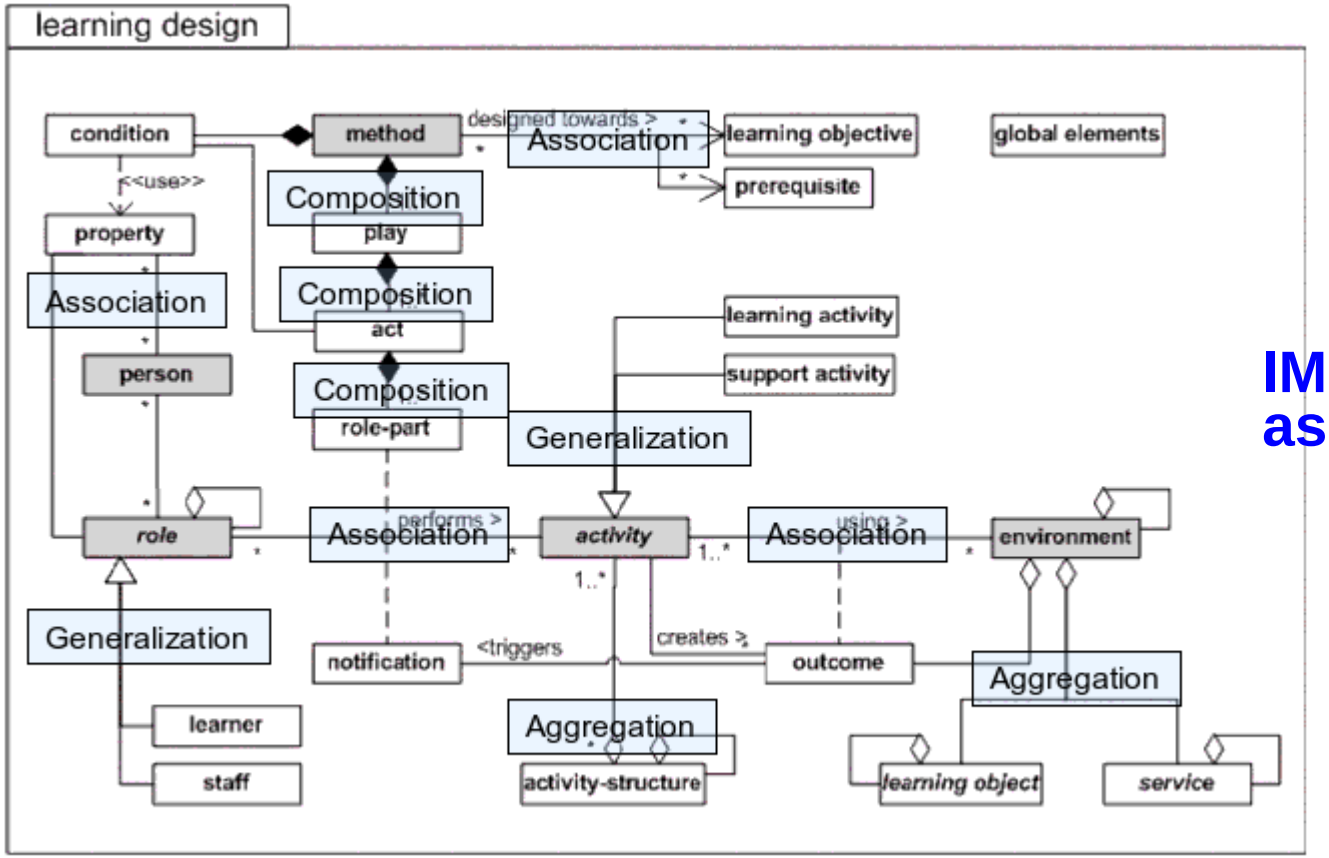
8.3.UML class diagram

url: **http://edutechwiki.unige.ch/en/UML_class_diagram**

- Classes (objects)
  - Represented by: a rectangle with one, two or three "fields": classname, properties and operations
- Association
  - A relationship between instances of the two classes
  - Represented by: a solid line, or if directional with an arrow directed from the source class to the target class.
- Aggregation
  - A part-of relationship
  - Represented by: a solid line, with an empty lozenge at aggregation end and arrow at member end
  - Example: a learning object is part of an environment
- Generalization
  - is-a relationship
  - Represented by: A solid line with a triangular arrow from specialized class to class
  - Example: Learner is a role
- Composition
  - Like aggregation but you can add more constraints.

- An instance of a class can be potentially a component of several classes, but can only be owned by one.
- Represented by: a solid line, with an filled lozenge at aggregation end and arrow at member end

- Multiplicity
  - of an association: x or x...y (x = 0,1,n*) and y = 0,1,n,*)



IMS Learning Design as UML class diagram

- Blue boxes (roughly) give an idea of the relationships involved.

9. Modeling (2) - Learning outcomes

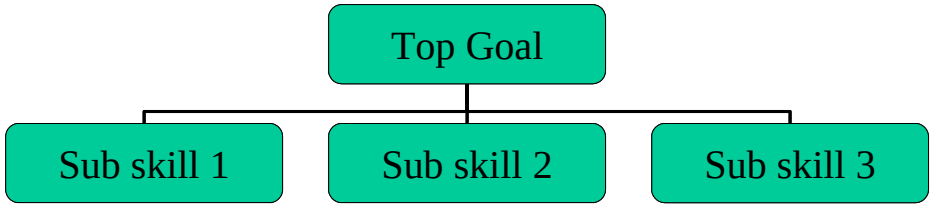
1. Identify learning outcomes (SWBAT = Student Will Be Able To ...)

- Depends on the kind of skills to be learnt

Learning type	*SWBAT ....*
Problem-solving	Generate ...
Rule-using	Demonstrate ....
Concepts	Classify, identify ....
Verbal information	State, list, recite, summarize ....
Affective	Choose ....
Psychomotor	Execute ...

2. Model domain knowledge

3. Identify skills a learner should master

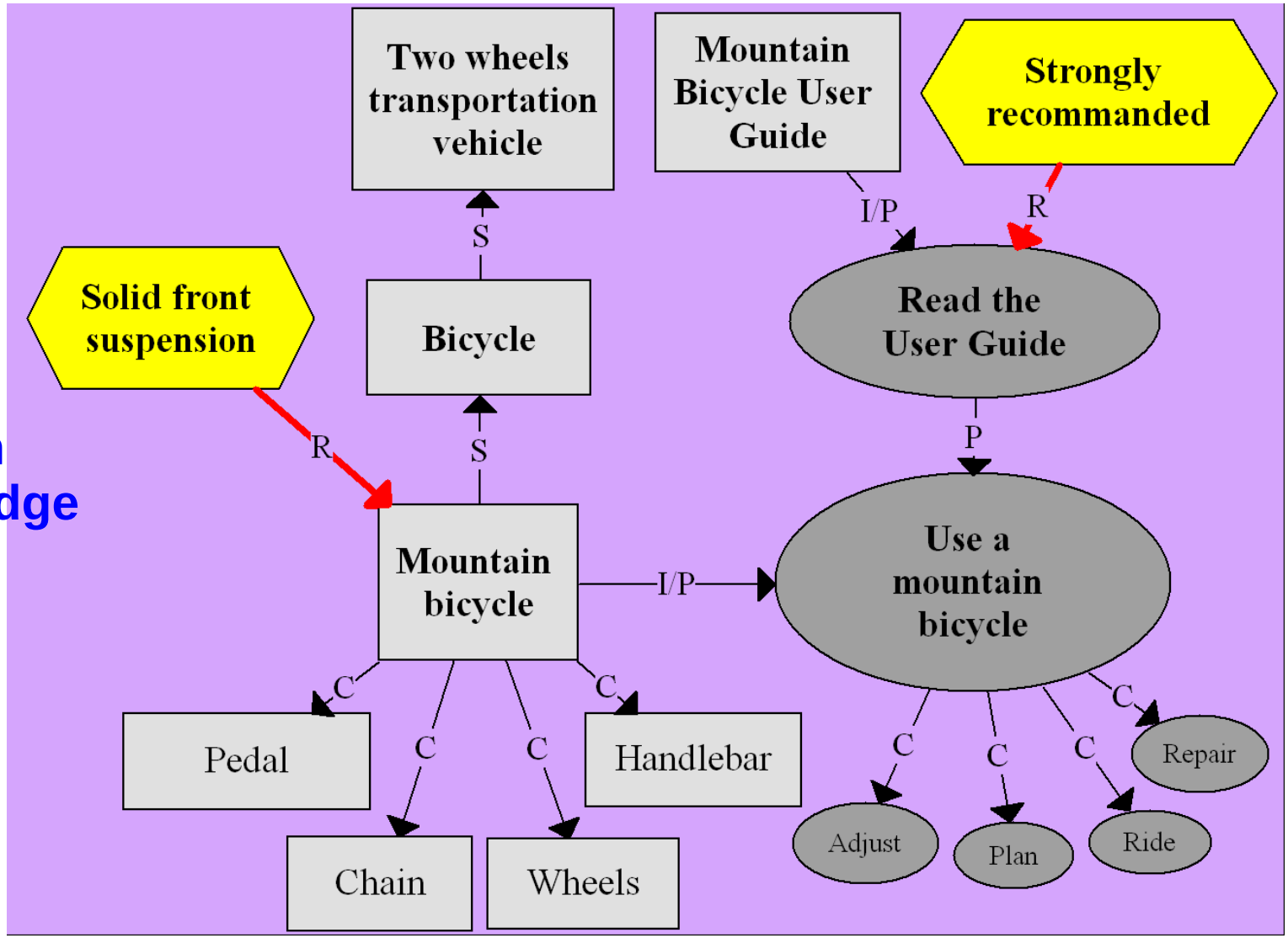


Subordinate skills analysis according to Rieber

Ontology modeling languages can help, e.g. OWL the W3C language

Bicycle knowledge modeled with MOT+ Ontology editor

Domain knowledge



Skills

10.Modeling (3) - Metadata information

- Today's most popular standard is LOM
- LOM is a complex definition
  - In real world, teachers usually refrain from entering MetaData
  - Document repositories often really on simpler Dublin Core
  - Web 2.0 relies on folksonomies (tagging)

10.1.The LOM standard

- LOM = Learning Object Metadata Standard
- LOM was proposed by Ariadne, IMS and then formalized by IEEE

LOM includes 9 sections:

- the General category, the Lifecycle category, the Metametadata category, the Technical category, the Educational category, the Rights category, the Relation category, the Annotation category, and the Classification category

10.2.The LOM Educational category section

Element	*Description*
interactivity Type (IEEE 1484.12.1-2002)	• **active: Active learning (e.g., learning by doing) is supported by content that directly induces productive action by the learner.** • **expositive: Expositive learning (e.g., passive learning) occurs when the learner's job mainly consists of absorbing the content exposed to them.** • **mixed: A blend of active and expositive interactivity types.**
learning Resource Type (IEEE best practice)	**exercise, simulation, questionnaire, diagram, figure, graph, index, slide, table, narrative text, exam, experiment, problem statement, self assessment, lecture**
interactivity Level (IEEE 1484.12.1-2002 but meaningful only in community practice)	**very low, low, medium, high, very high**
semantic Density (IEEE 1484.12.1-2002 but meaningful only in community practice)	**very low, low, medium, high, very high**

Element	Description
intended End User Role (IEEE 1484.12.1-2002)	• **teacher** • **author** • **learner** • **manager**
context (IEEE 1484.12.1-2002)	• **school** • **higher education** • **training** • **other**
typical Age Range	**xx**
difficulty (IEEE 1484.12.1-2002, meaningful only in a context of a community)	• **very easy** • **easy** • **medium** • **difficult** • **very difficult**
typical Learning Time	**open text element**
description	**open text element**
language	**standardized def.**

11.Modeling (4) - Learning Objects, Units and Content Packs

A provisional definition of Learning object and unit of learning

- **Learning objects** are small (relative to the size of an entire course) instructional components that can be reused a number of times in different learning contexts.
- **Units of learning** are course components (or entire courses) that include pedagogy (contents, sequenced contents and/or activities). Both isolated units of learning and entire courses can be distributed as content packs.
- Technically speaking, "learning objects + content packaging" mostly refers to reusable e-learning contents as defined by IMS and SCORM

Most important issues:

- Return on investment (is it worthwhile ?)
- Size (how big is a learning object ?)
- Standards (is IMS the only one ?)

11.1.Motivation: Why learning objects ?

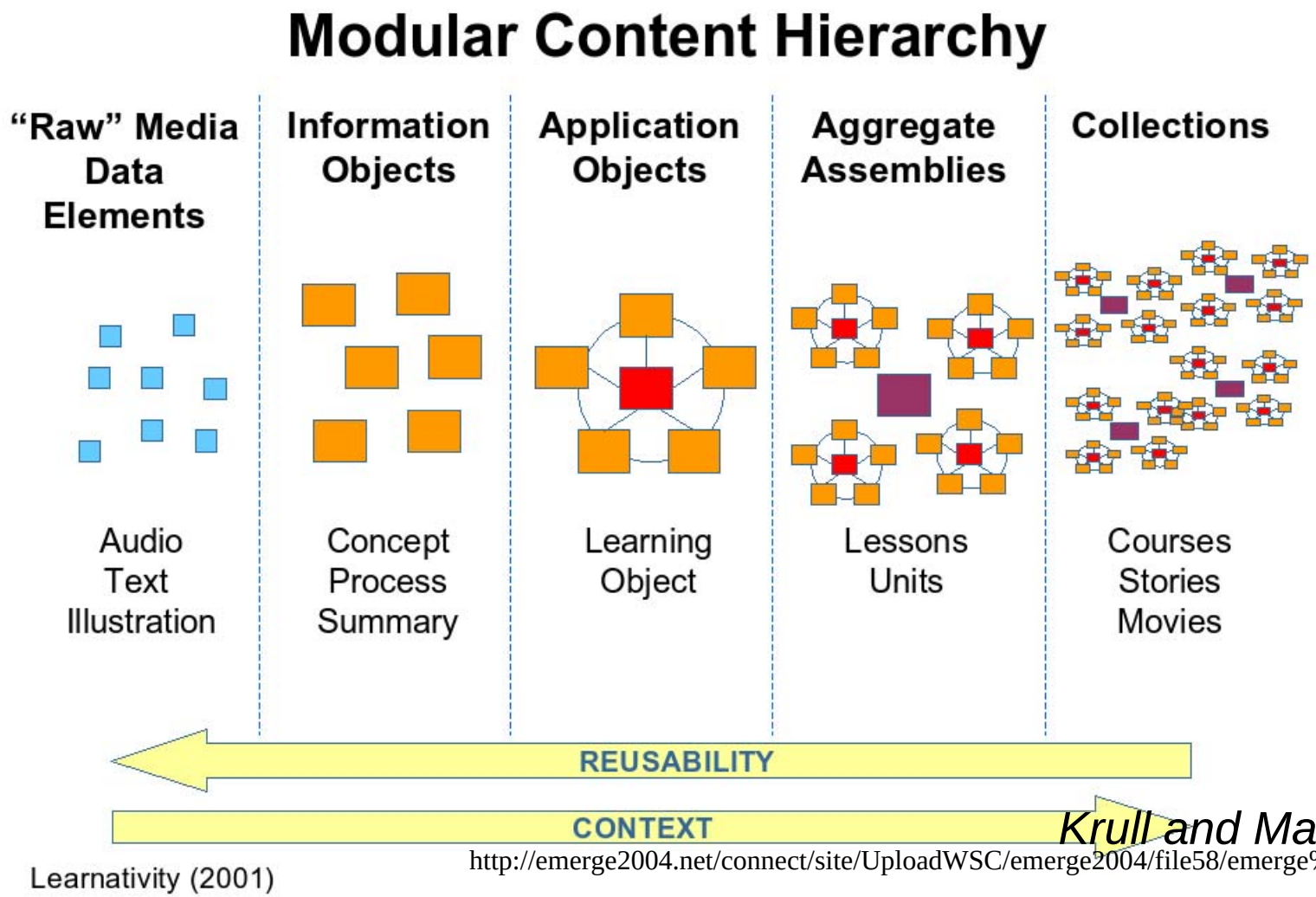
SCORM 2004 3rd Edition Overview (p 1-6)

- **Accessibility**: The ability to locate and access instructional components from one remote location and deliver them to many other locations.
- **Adaptability**: The ability to tailor instruction to individual and organizational needs.
- **Affordability**: The ability to increase efficiency and productivity by reducing the time and costs involved in delivering instruction.
- **Durability**: The ability to withstand technology evolution and changes without costly redesign, reconfiguration or recoding.
- **Interoperability**: The ability to take instructional components developed in one location with one set of tools or platform and use them in another location with a different set of tools or platform.
- **Reusability**: The flexibility to incorporate instructional components in multiple applications and contexts.

11.2.Size of learning object

Two sorts of opinions: (1) Can be ***of any*** size - (2) something ***in between***

A content-oriented view:



Krull and Mallinson PPT slides
http://emerge2004.net/connect/site/UploadWSC/emerge2004/file58/emerge%20slides%20v8%20(final).ppt

11.3.The IMS Content Packaging standard

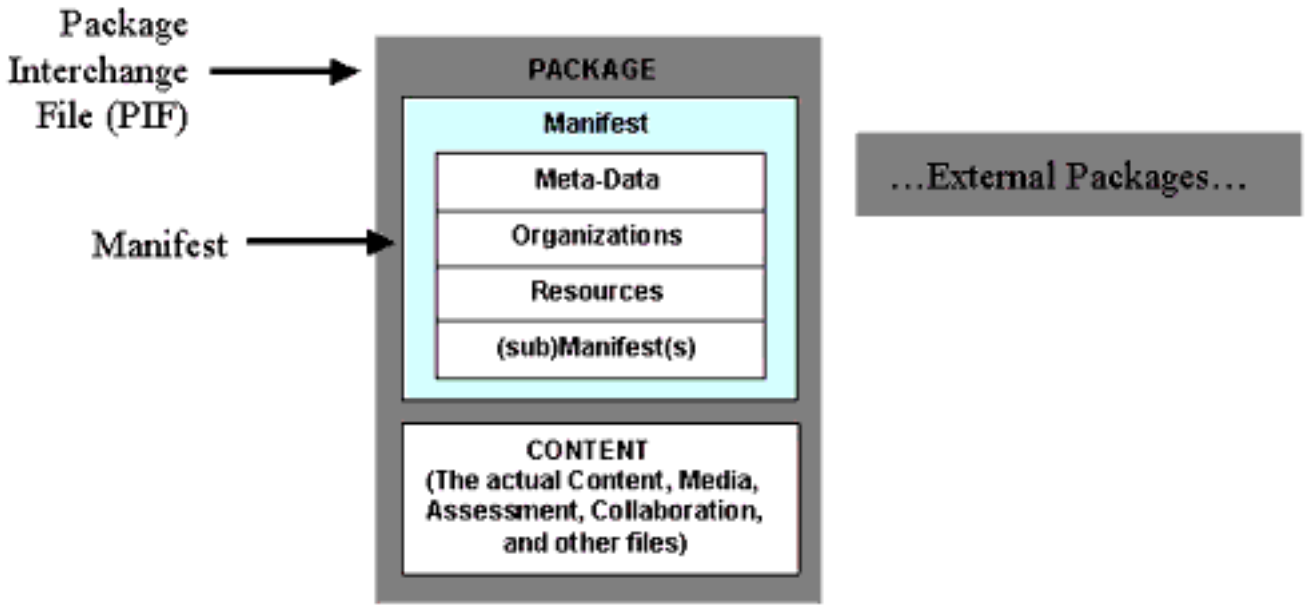
- IMS Content packs are a kind of bigger learning object defined at learning unit or course level

Objectives:

- allow **transfer** of e-learning contents from one system to an other (inter-operability)
- allow to **import/export** contents into/from a learning management system. Therefore support authoring with a stand-alone tool.
- IMS Content Packaging is an evolving and **extensible** standard.
  - Therefore, make sure that you know what version you plan to build / import / export and more importantly what kinds of extensions various profiles adopt !!
- IMS Content packaging is a very popular e-learning standard
  - IMS Content Packaging 1.1.2 is adopted by SCORM 1.2
  - IMS Content Packaging 1.1.4 by SCORM 2004
  - IMS Content Packaging 1.1.4 (?) by IMS Learning Design

11.4.Architecture overview

- IMS Content packaging allows to deliver different sorts of e-learning contents, but all relay on the same PIF architecture
- A Package Interchange File (PIF) is a zip file.



Details

1. File resources that include

- All your **assets** (contents) needed, e.g. HTML pages, pictures, Flash Animations, PPT (whatever, the idea is that you take formats that your client can handle). You may use sub-directories

2. Schema files

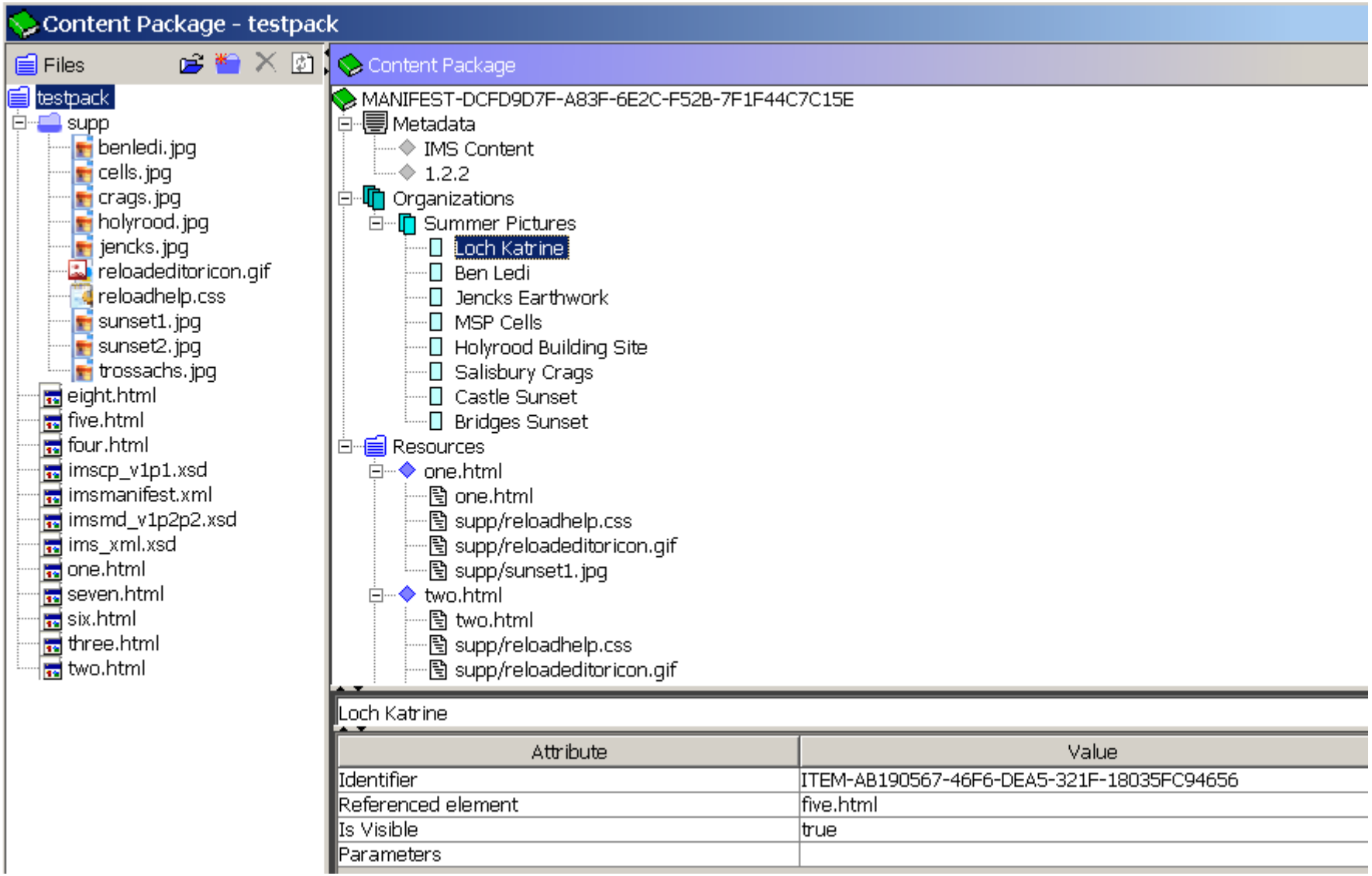
- that formally define all the vocabularies used in the manifest file (see next item).

3. A special file in the zip is imsmanifest.xml containing:

- a **meta-data section** <metadata />: describes the package as a whole.
- a **resource list** <resources />: containing references to all of the actual resources and media elements needed for a manifest (files), including meta-data describing these resources, and references to any external files (URLs).
- an **organization section**: describing zero, one, or multiple learning sequences, i.e. information that is used to "play" the contents.
  The default tree-based default organization refers to simple resources (assets). However, this section can also contain IMS Simple Sequencing or IMS Learning Design organization or any other future standard.
- (sub)manifests, that can do the same for subpackages.

Exemple 11-1: TestPack (made with RELOAD:

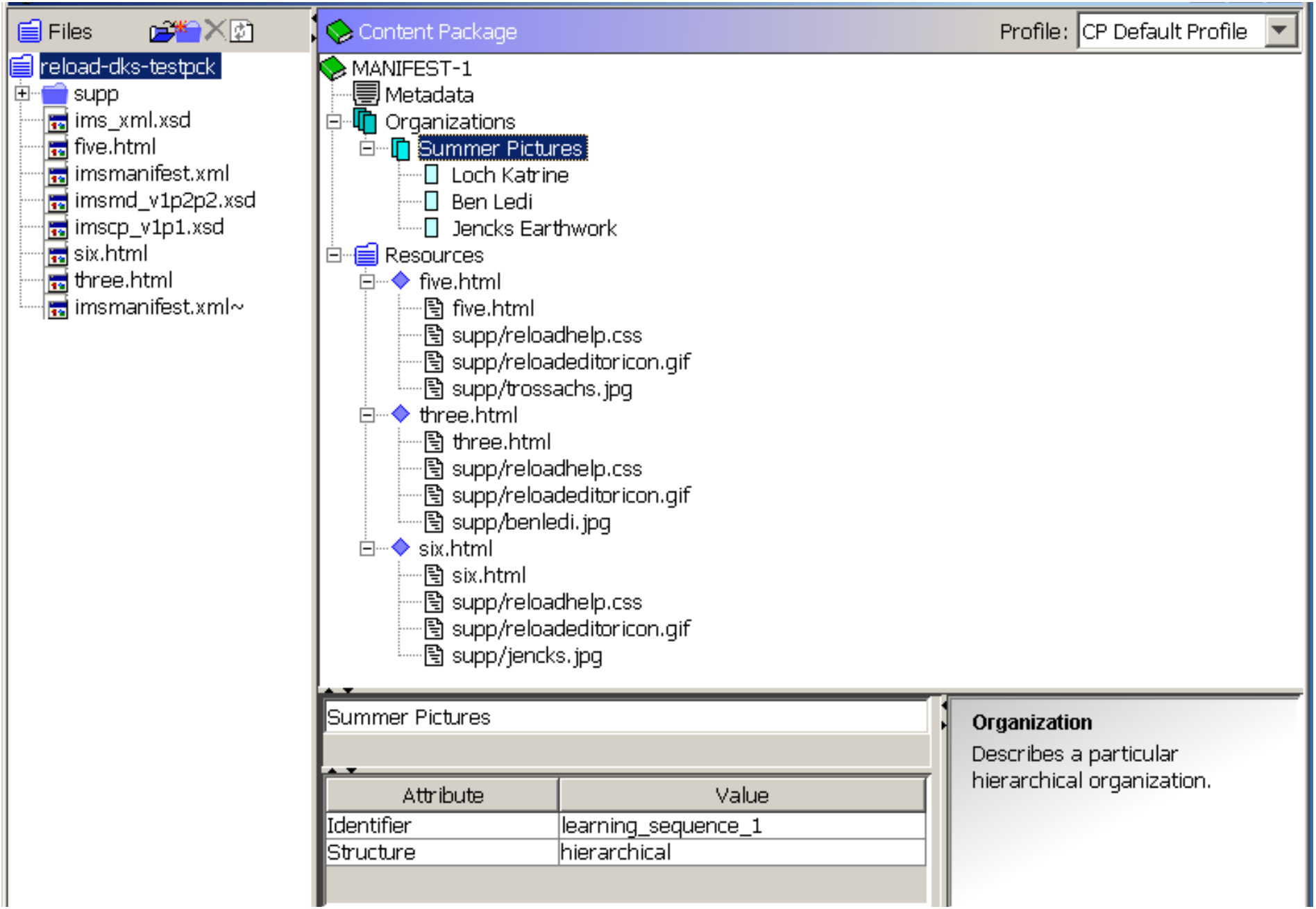
url: **http://www.reload.ac.uk/ex/testpkg.zip - (http://www.reload.ac.uk/ex/)**



Files in a typical zip

supp	--	folder	extra contents (pictures)
imscp_v1p1.xsd	16.3 KB	eXtensible Markup Language document	IMS CP Schema definition
imsmanifest.xml	8.5 KB	eXtensible Markup Language document	The manifest file
imsmd_v1p2p2.xsd	23.7 KB	eXtensible Markup Language document	IMS MD Schema definition
ims_xml.xsd	1.1 KB	eXtensible Markup Language document	
eight.html	745 bytes	HTML document	contents (assets) that will be used by the pedagogical organization
five.html	751 bytes	HTML document	
four.html	756 bytes	HTML document	
one.html	738 bytes	HTML document	
seven.html	762 bytes	HTML document	
six.html	759 bytes	HTML document	
three.html	755 bytes	HTML document	
two.html	739 bytes	HTML document	

Exemple 11-2: A simplified example with code



Header of the manifest

```
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
          xmlns:imsmd="http://www.imsglobal.org/xsd/imsmd_v1p2"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          identifier="MANIFEST-1"
          xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1 imscp_v1p1.xsd
                              http://www.imsglobal.org/xsd/imsmd_v1p2 imsmd_v1p2p2.xsd">
```

Metadata section

```
<metadata>
  <imsmd:lom>
    <imsmd:general>
      <imsmd:identifier>metadata-id</imsmd:identifier>
      <imsmd:title>
        <imsmd:langstring xml:lang="en">Simple IMS 1.1.4 Content Package</imsmd:langstring>
      </imsmd:title>
    </imsmd:general>
    <imsmd:metametadata> [....] </imsmd:metametadata>
    <imsmd:technical> <imsmd:format>text/html</imsmd:format> </imsmd:technical>
    <imsmd:educational>
      <imsmd:interactivitytype>
        <imsmd:source>
          <imsmd:langstring xml:lang="en">LOMv1.0</imsmd:langstring>
        </imsmd:source>
        <imsmd:value> <imsmd:langstring xml:lang="x-none">Expositive</imsmd:langstring> </imsmd:value>
      </imsmd:interactivitytype>
    </imsmd:educational>
    <imsmd:rights>
      <imsmd:description>
        <imsmd:langstring xml:lang="en">This is total freeware.</imsmd:langstring>
      </imsmd:description>
    </imsmd:rights>
  </imsmd:lom>
</metadata>
```

Organization section

```
<organizations default="learning_sequence_1">
  <organization identifier="learning_sequence_1" structure="hierarchical">
    <title>Summer Pictures</title>
    <item identifier="ITEM-1" isvisible="true" identifierref="RES-1">
      <title>Loch Katrine</title>
    </item>
    <item identifier="ITEM-2" [ ....] <title>Ben Ledi</title>
    </item>
    <item identifier="ITEM-3" isvisible="true" identifierref="RES-3">
      <title>Jencks Earthwork</title>
    </item>
  </organization>
</organizations>
```

Resources section

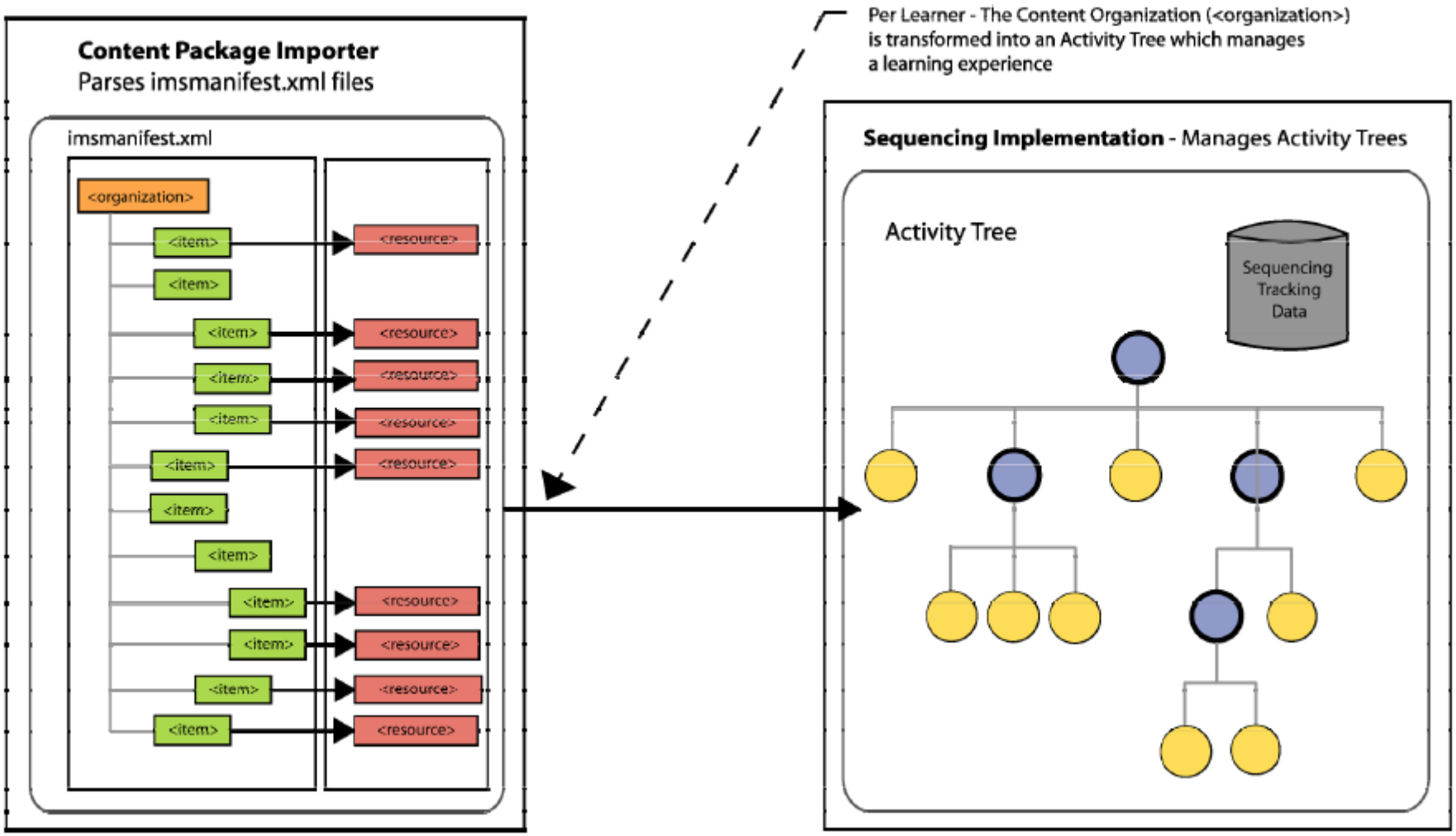
```
<resources>
  <resource identifier="RES-1" type="webcontent" href="five.html">
    <file href="five.html" />
    <file href="supp/reloadhelp.css" />
    <file href="supp/reloadeditoricon.gif" />
    <file href="supp/trossachs.jpg" />
  </resource>
 [.....]
  <resource identifier="RES-3" type="webcontent" href="six.html">
    <file href="six.html" />
    <file href="supp/reloadhelp.css" />
    [....]   </resource>
</resources>
```

11.5.What is SCORM ?

- SCORM = Sharable Courseware Object Reference Model (2000)
- SCORM = Sharable Content Object Reference Model (since 2001)
- Since 2001, SCORM produces few specific standards but adopts "**profiles**" composed from other standards (IMS and IEEE) and norms for the systems architecture that run e-learning packages
- SCORM is the major player respected by industry

Version	*Date*	*Elements*
Version 1.0	**Jan. 2000**	**1. Course Structure Format (CSF) 2. Run-Time Environment 3. Metadata (Course Metadata, Content Metadata, Raw Media Metadata)**
Version 1.1	**Jan. 2001**	**Improved 1.0, clarification of "Run Time" specs**
Version 1.2x (most popular!)	**Oct. 2001**	**Adoption of IMS Content Packaging 1.1.2 (instead of CFS), IMS Learning Resource Metadata 1.2.1 (LOM) instead of MetaData. "Run Time" specifications: CAM/SCO/etc.**
Version 2004 (1.3x)	**Jan. 2004**	**Improvement of 1.2 Inclusion of the "simple sequencing" modeling language**

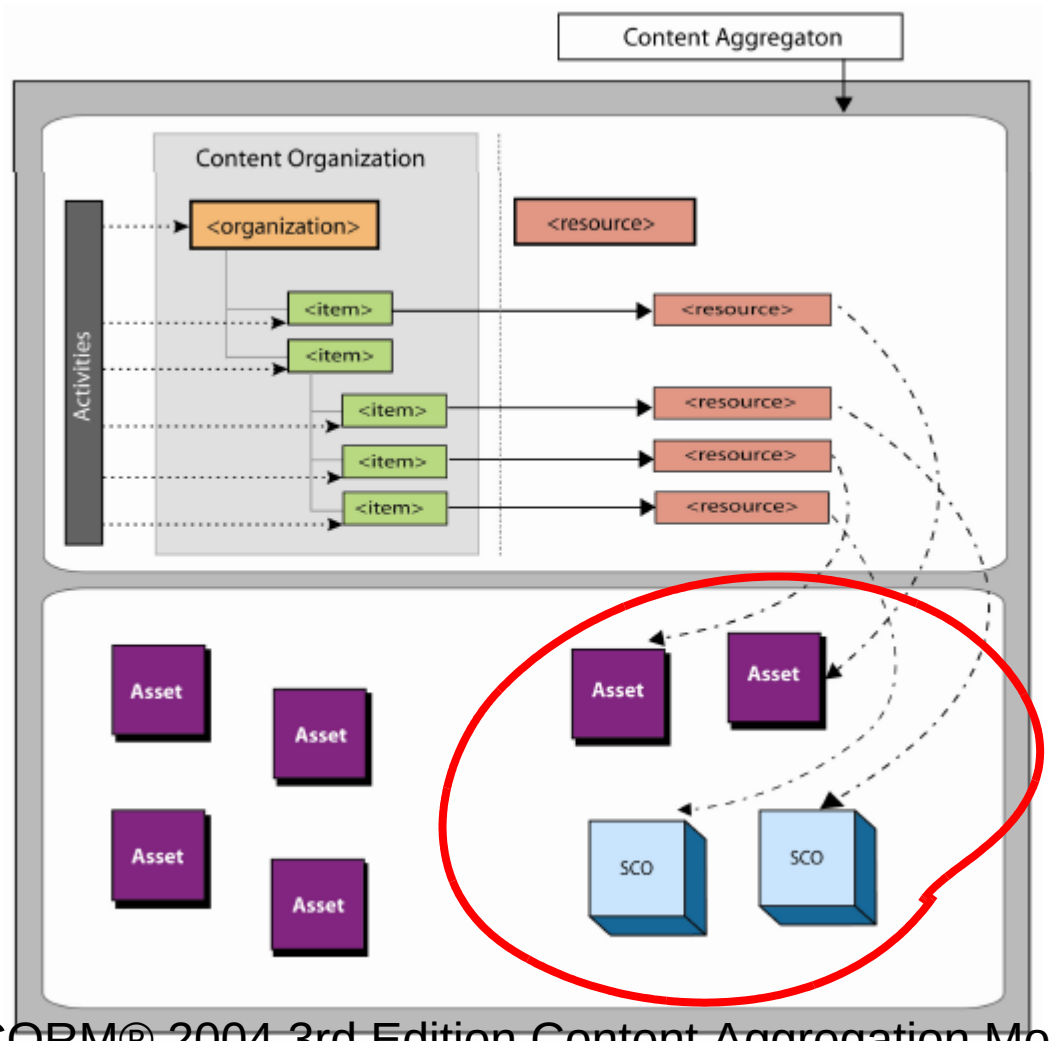
11.6.IMS CP default organization vs. SCORM 1.2 CAM



- IMS CP describes the structure of data (how to organize resources in sequences)
- SCORM Content Aggregation Model (CAM) standardizes how the activity tree will be constructed in a system (RTE) et how student data are to be collected.

SCORM adds so-called SCOs to the Content Pack:

- A SCO is a special kind of resource that can "talk" with an LMS
- Therefore we distinguish: simple "assets" (e.g. an HTML file, an animation) or an Shareable Content Object (SCO).



Scorm adds SCOs

SCORM® 2004 3rd Edition Content Aggregation Model (CAM) Version 1.0

11.7.Summary IMS CP and SCORM

- IMS CP allows to assemble e-learning "data" for distribution
  - MetaData
  - Resources
  - Organizations (i.e. pedagogical sequencing of various kinds)
- SCORM, focuses on execution
  - defines how a resource and an LMS could interact (adds extensions to IMS CP !)
  - defines what execution means in terms of an student-activity tree
- Implementations of IMS CP + SCORM 1.2 players.
  - Most LMS provide support for simple IMS CP.
  - Lots, e.g. MOODLE support both IMS CP and SCORM 1.2
  - Few provide support for SCORM 2004 (including Simple Sequencing)
  - IMS Learning Design is badly supported

Playtime:

- Edit contents with eXe, export as IMS CP, dezip and then look at it
- Import this package into the Reload editor
- Get the IMS CP Test package with Reload editor and modify it

url: **http://edutechwiki.unige.ch/en/Reload_Editor**

url: **http://edutechwiki.unige.ch/en/Reload_Editor_Tutorial**

url: **http://edutechwiki.unige.ch/en/eXe**

12.Modeling (5) - Simple sequencing and QTI

- IMS Simple Sequencing is a norm to define complex learning sequences
- IMS Sequencing is also part of the ADL/Scorm 2004 profile
- IMS QTI Question and Test Interoperability (QTI) defines a sophisticated quizzing language

Simple Sequencing:

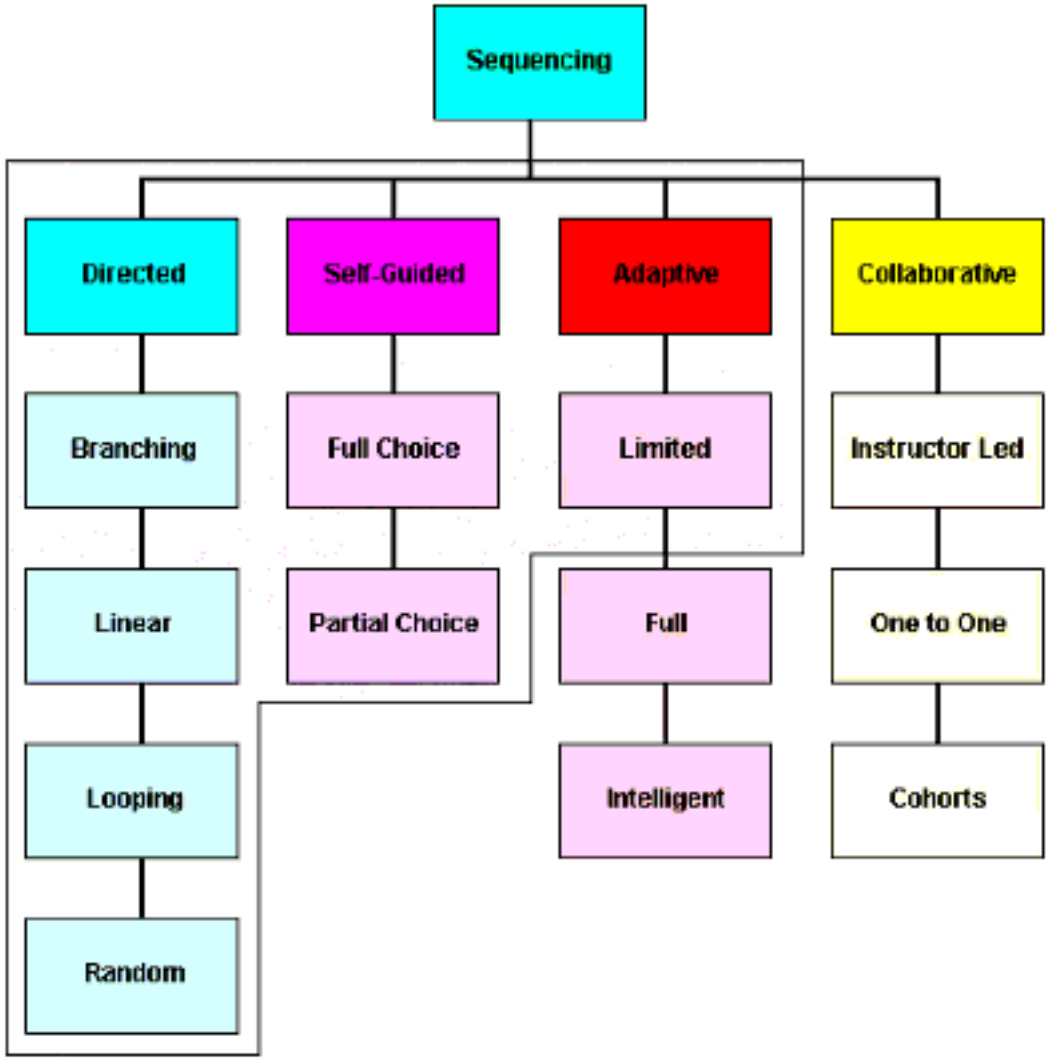
- describes navigation paths for a collection of learning activities
- controls navigation (taking into account performance etc.)

Computational architecture of "simple" sequencing:

1. **Definition information**
   - Definition of an organization in the imsmanifest.xml file
2. **Tracking information**
   - Collected informations from a learner using components
   - allows tracing and conditional navigation
3. **Activity state information**
   - A "snapshot" of the activity tree
   - Allows a learner to "pick up" again an activity.

12.1.Support for educational designs

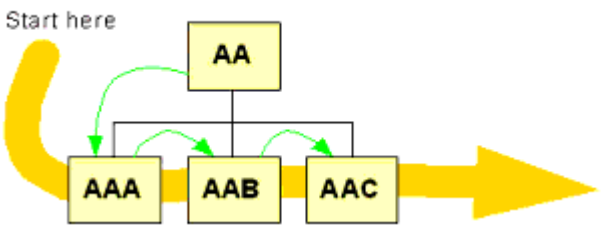
- Simple sequencing is best for mastery learning



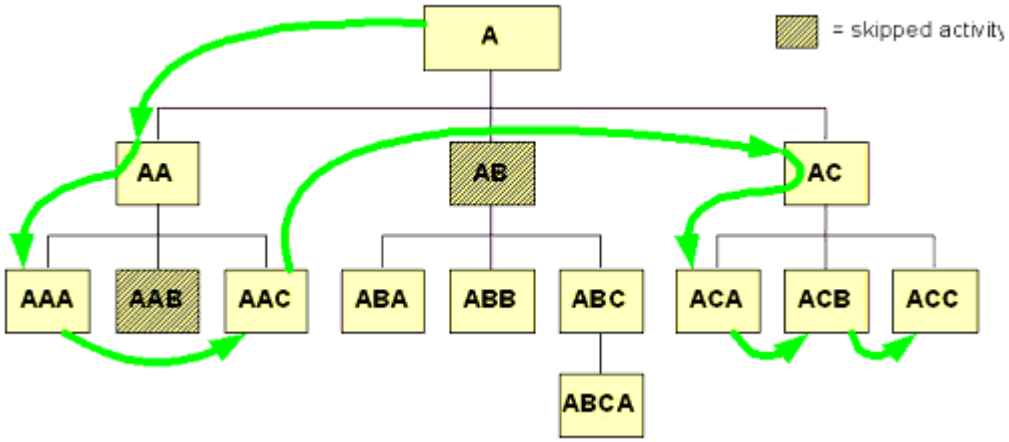
12.2.The sequence in IMS SS

- IMS simple sequencing isn't simple !!

Simple linear activity:



Conditional activities:

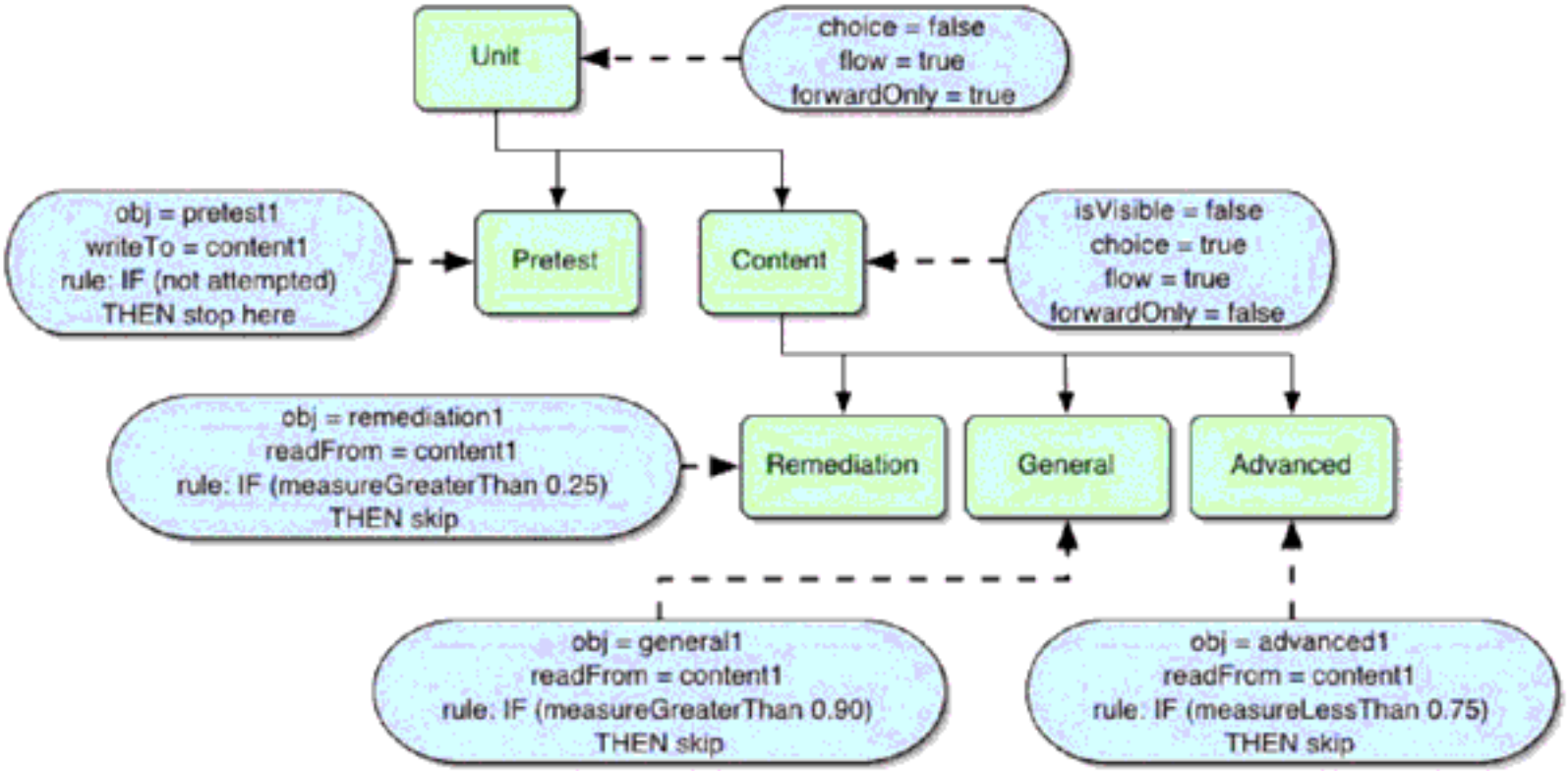


(IMS Simple Sequencing Best Practice and Implementation Guide)

12.3.Example of a sequence and its XML code

(Picture from IMS Simple Sequencing Best Practice and Implementation Guide)

Basic sequencing behavior:



Some imsmanifest.xml code:

```
<?xml version = "1.0" encoding = "UTF-8"?>
<manifest xmlns = "http://www.imsglobal.org/xsd/imscp_v1p1"
      xmlns:imsss = "http://www.imsglobal.org/xsd/imsss"
      xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation = "http://www.imsglobal.org/xsd/imsss Schemas/imsss_v1p0.xsd
         http://www.imsglobal.org/xsd/imscp_v1p1 Schemas/imscp_v1p1p3.xsd"
      identifier = "BBT_TLCExamplePostTest">
<!-- This manifest contains a solution to the following ID Problem:
 * Deliver content, then post-test
 * If student doesn't pass the post-test, deliver remediation, then repeat the content and post-test
-->
      <organizations>
         <organization identifier = "Organization">
            <item identifier = "Unit">
               <item identifier = "Content"/>
               <item identifier = "Posttest">
                  <imsss:sequencing>
                     <imsss:objectives>
                        <imsss:primaryObjective objectiveID = "posttest1" satisfiedByMeasure = "true">
                              <imsss:mapInfo targetObjectiveID = "content1" readSatisfiedStatus = "true"/>
                           </imsss:primaryObjective>
                     </imsss:objectives>
                  </imsss:sequencing>
               </item>
               <item identifier = "Remediation">
                  <imsss:sequencing>
                     <imsss:sequencingRules>
                        <imsss:preConditionRule>
                           <imsss:ruleConditions>
                              <imsss:ruleCondition condition = "satisfied"/>
                              </imsss:ruleConditions>
                              <imsss:ruleAction action = "skip"/>
                           </imsss:preConditionRule>
                     </imsss:sequencingRules>
```

```
                    <imsss:objectives>
                       <imsss:primaryObjective objectiveID = "remediation1">
                          <imsss:mapInfo targetObjectiveID = "content1" readSatisfiedStatus = "true"/>
                       </imsss:primaryObjective>
                    </imsss:objectives>
                 </imsss:sequencing>
              </item>
              <imsss:sequencing>
                 <imsss:controlMode choice = "false" flow = "true" forwardOnly = "true"/>
                 <imsss:sequencingRules>
                    <imsss:postConditionRule>
                       <imsss:ruleConditions>
                          <imsss:ruleCondition condition = "satisfied" operator = "not"/>
                       </imsss:ruleConditions>
                       <imsss:ruleAction action = "retry"/>
                    </imsss:postConditionRule>
                 </imsss:sequencingRules>
                 <imsss:objectives>
                    <imsss:primaryObjective objectiveID = "unit1">
                       <imsss:mapInfo targetObjectiveID="content1" readSatisfiedStatus= "true"/>
                    </imsss:primaryObjective>
                 </imsss:objectives>
              </imsss:sequencing>
           </item>
        </organization>
     </organizations>
     <resources/>
</manifest>
```

12.4.IMS Question and Test Interoperability (QTI)

Objectives of this specification

- models questions and answers (assessment items)
- models data generated by learners (assessment data, user answers)

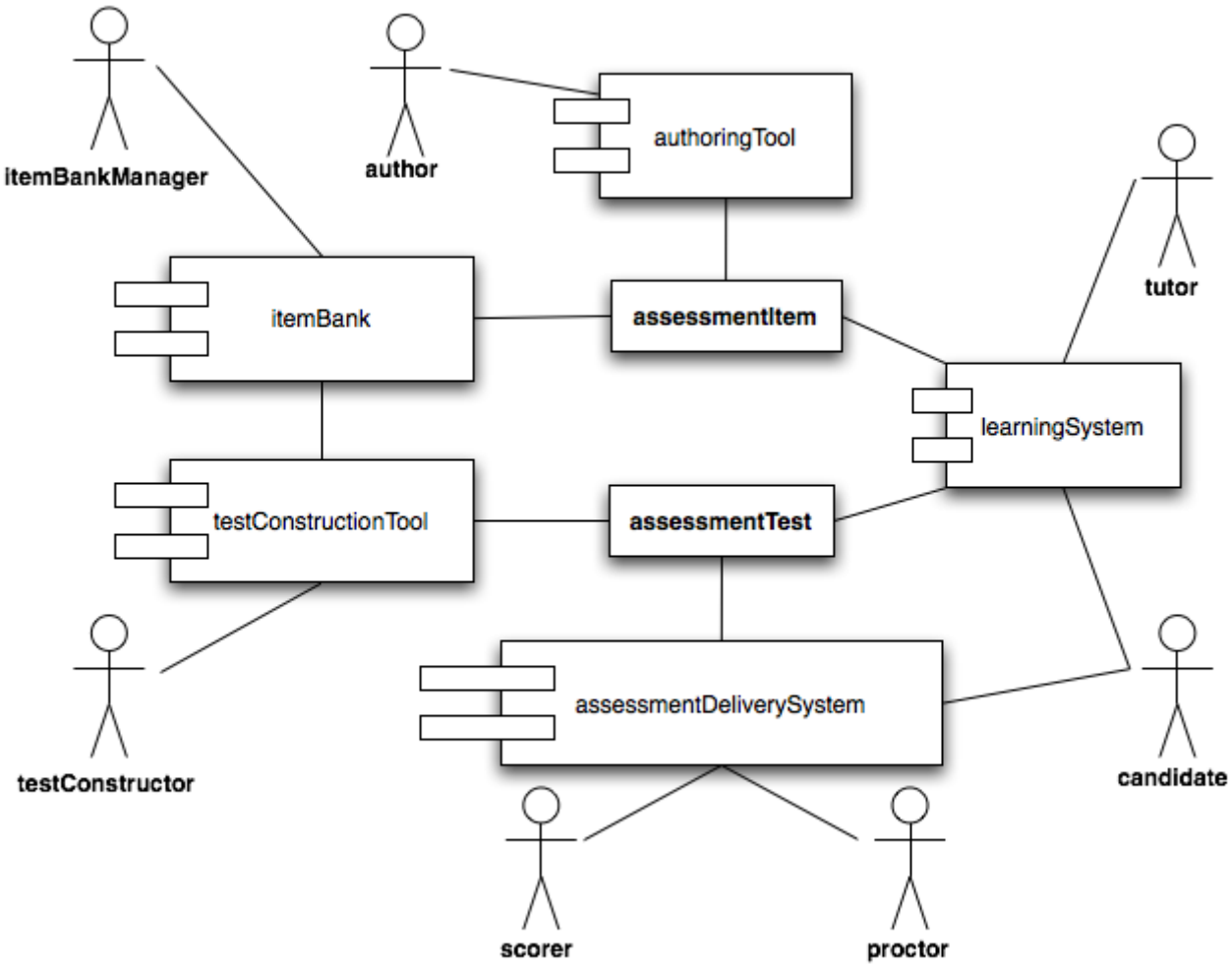
Integration with other norms

- QTI can be combined with IMS Simple Sequencing and IMS Learning Design (under the condition that you use an appropriate player)

Tools

- QTI players can be found in full SCORM 2004 compliant LMSs (some 1.2)
- APIS
  - **http://sourceforge.net/projects/apis/**
- QPlayer
  - **http://www.e-teach.ch/q-player**
- TOIA
  - **http://www.toia.ac.uk/**

12.5.QTI use case analysis



- authoringTool: allows to create an item (questions / answers)
- itemBank: Item repository
- testConstructionTool: assembly tool (should be part of the authoring tool)
- assessmentDeliverySystem: QTI player
- learningSystem: LMS containing the player and that directs the learner to a test

12.6.Example code

```
<assessmentItem xmlns="http://www.imsglobal.org/xsd/imsqti_v2p0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.imsglobal.org/xsd/imsqti_v2p0 imsqti_v2p0.xsd"
    identifier="choice" title="Unattended Luggage" adaptive="false" timeDependent="false">
     <responseDeclaration identifier="RESPONSE" cardinality="single" baseType="identifier">
         <correctResponse>
             <value>A</value>
         </correctResponse>
     </responseDeclaration>
     <outcomeDeclaration identifier="SCORE" cardinality="single" baseType="integer">
         <defaultValue>
             <value>0</value>
         </defaultValue>
     </outcomeDeclaration>
    <itemBody>
       <p>Look at the text in the picture.</p>
       <p><img src="sign.png" alt="NEVER LEAVE LUGGAGE UNATTENDED"/></p>
       <choiceInteraction responseIdentifier="RESPONSE" shuffle="false" maxChoices="1">
           <prompt>What does it say?</prompt>

             <simpleChoice identifier="A">You must stay with your luggage at all times.</simpleChoice>
             <simpleChoice identifier="B">Do not let someone else look after your luggage.</simpleChoice>
             <simpleChoice identifier="C">Remember your luggage when you leave.</simpleChoice>
         </choiceInteraction>
     </itemBody>
    <responseProcessing
       template="http://www.imsglobal.org/question/qti_v2p0/rptemplates/match_correct"/>
</assessmentItem>
```

13.Modeling (6) - Learning Design

13.1.Introduction

- SCORM is about content aggregation and mostly "Learning I" pedagogy
- LD is more concerned by "Learning II" and some "Learning III"

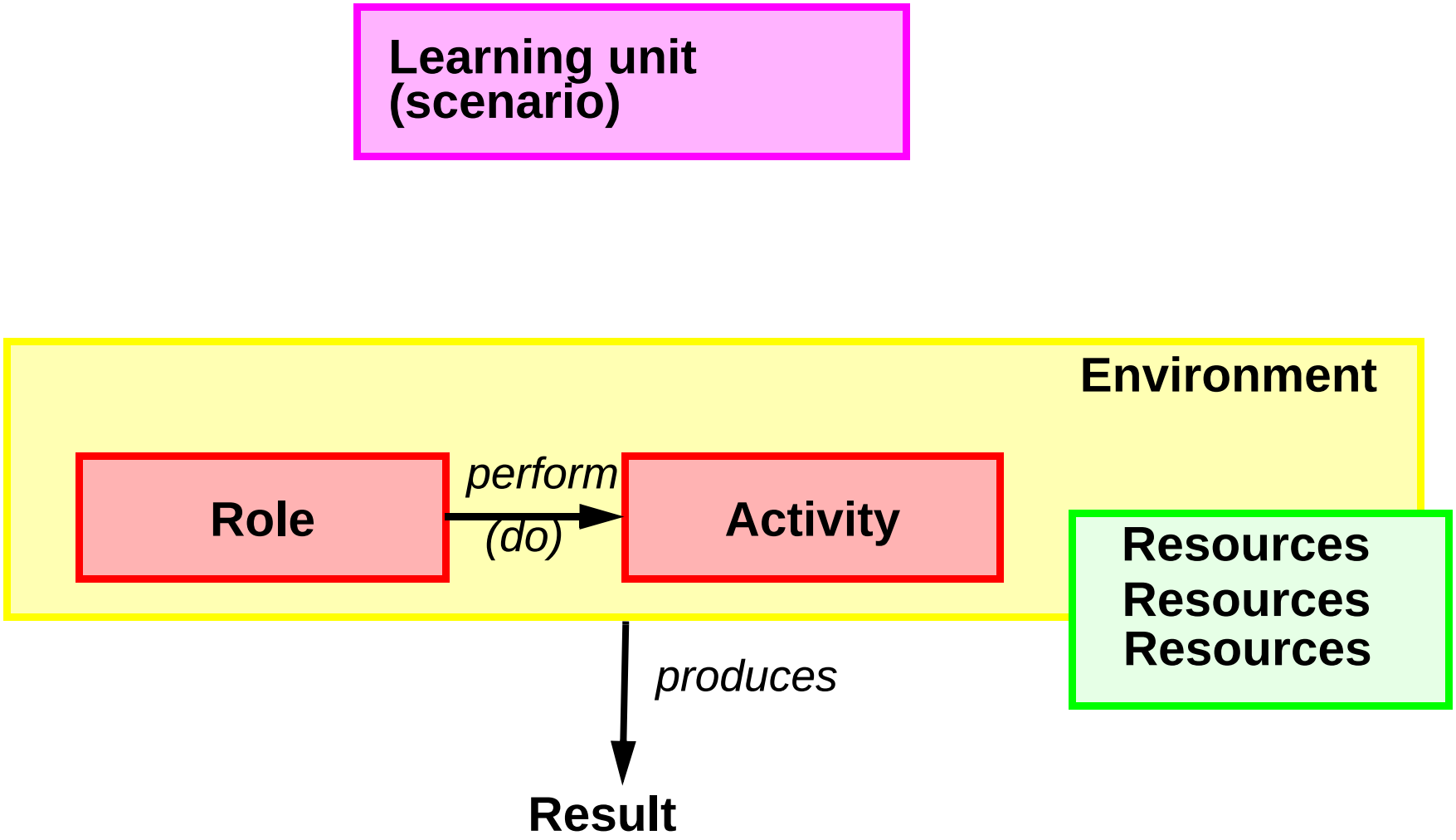
Objectives of Educational Modeling Language (Koper et al, 2001):

- focus on the organization of learning activities
- wants to formalize the "source code" of pedagogy ...

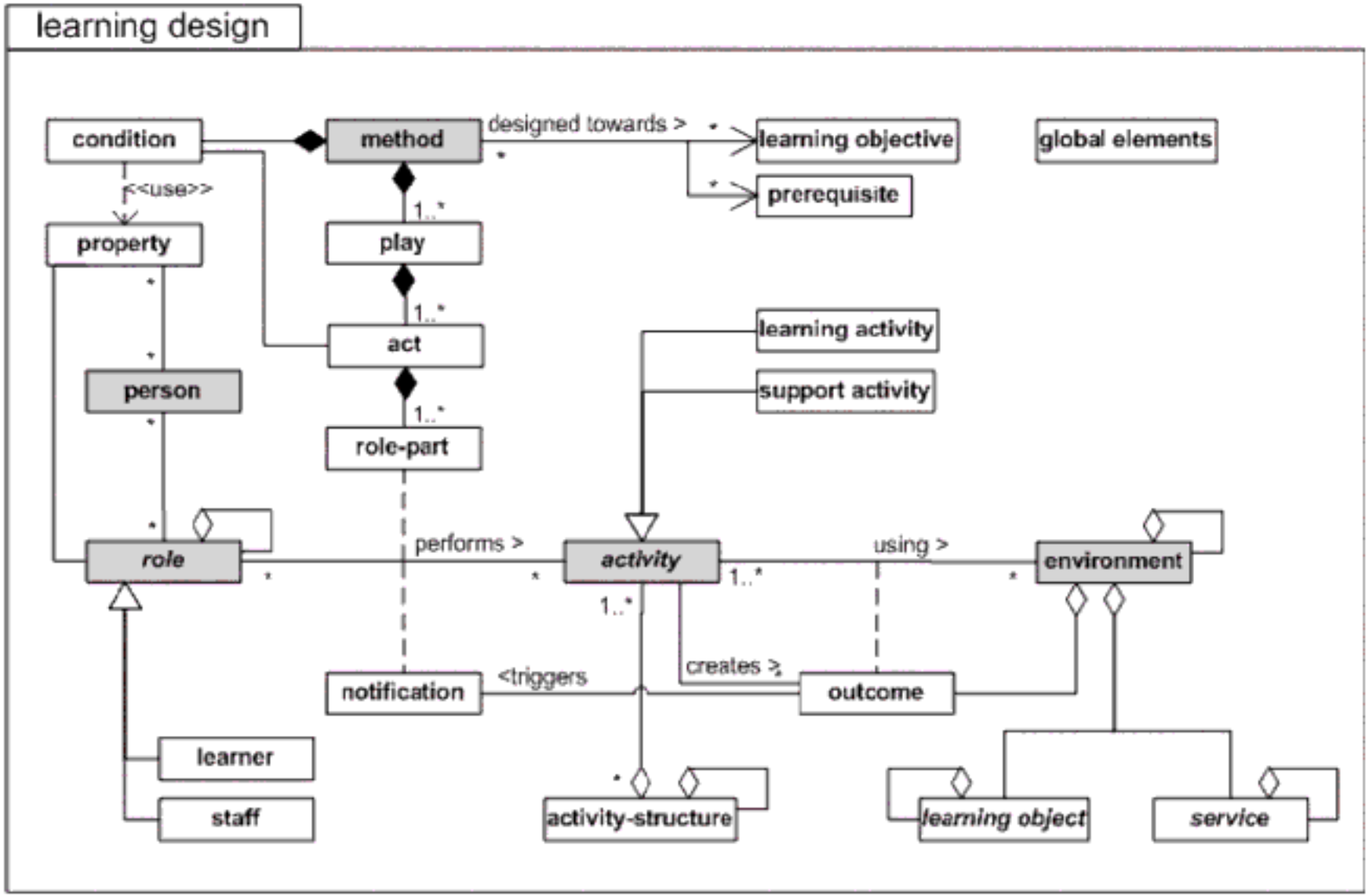
Central elements:

- **Roles** that are performed by learners, teachers, tutors etc.
- **Activities** instantiated by performers.
- **Environments** including services (e.g. a forum) and learning resources
- The **scenario** is called method and contains play, act and role-parts.
- People playing the roles undertake a series of activities within an act.
- An act is completed after all the activities of a specified role, or roles, are finished.

13.2.The LD core activity



The UML class diagram:



source: *http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html#1495548*

13.3.Structure of a LD "unit of learning" (UOL)

In LD, a UOL is a complete learning scenario, organized according to pedagogical principles

Technically: An IMS Content Package, i.e. a *.zip file with the following components:

1. **resources (including materials, tools and links)**
2. **A XML « manifest » :**

describing the « method »:

- play (scenario),
- its acts,
- role-parts of acts

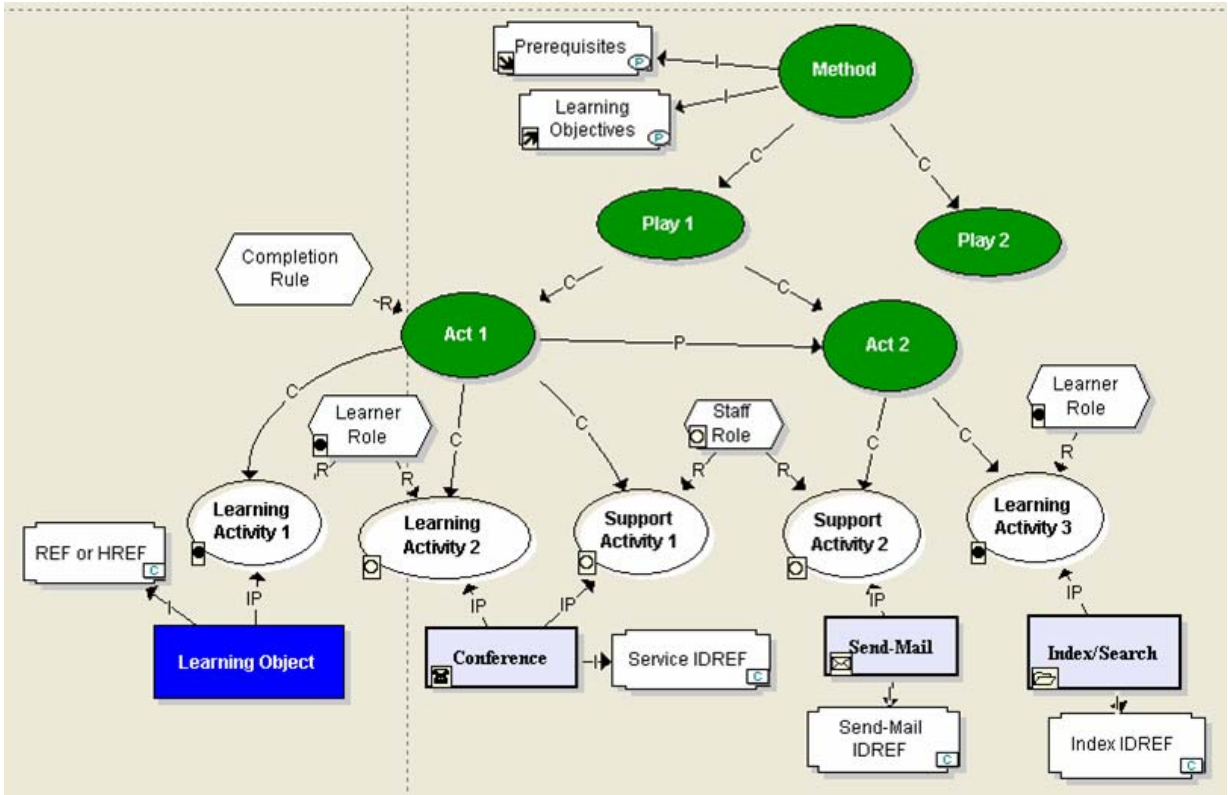
and its components

- roles,
- activities,
- environment,
- properties,
- conditions and notifications (more sophisticated designs allow "jumping")

Difference with simple IMS Content Packaging (CP) and IMS Simple Sequencing (SS):

- LD defines a method (an organization) with acts
- IMS CP (default organization) defines menu-based access to resources
- IMS SS defines conditional learning sequences
- SCORM 1.2 extensions (SCO) are somewhat in between simple CP and SS

A simple LD (formalized in the MOT editor):



Act is central

Engaging in an act means doing something:

reading, conferencing, sending mail, search ....

13.4.Major Elements of Learning Design

In IMS CP terminology a LD is an organization. It has the following sections:

```
learning-design
  components
    roles
      learner
      staff
    activities
      learning-activity (*)
      support-activity (*)
      activity-structure (*)
    environments
      learning-object (*)
      service (*)
  method
    play
      act (*)
        role-part (*)
(*) means multiple instances are possible
```

- LD makes a distinction between **components** of a play and its organization, i.e. the **method** (the play).
- Role-parts ***refer*** to components with links

  Looks a bit like a cooking recipe that lists ingredients first, then directions how to cook ...

Overview Table of LD elements (there are some extra elements, e.g. title)

learning-design	The base level container
title	A title for the learning design
learning-objectives	What this unit of learning achieves
prerequisites	Whether there are dependencies?
components	The reusable elements of the learning design - this is the key level of granularity
roles	The Role List
learner*	Learner-role
staff*	Tutor-role
activities	The Activity Container: Activities (can) have objectives, prerequisites and metadata. They have an activity description (typically a web page containing instructions for how to perform the activity). If the activity is offline, then no further content is needed. if online, there would also normally be reference to an environment.
learning-activity*	e.g. view this learning object
environment-ref*	A reference to the environment for this activity
activity-description	A narrative description of the activity
support-activity*	e.g. pose question to class
environment-ref*	A reference to the environment for this activity
activity-description	A narrative description of the activity - usually a web page, This is kept separate from the resources in the environment, and so the runtime system can treat it differently - perhaps keeping it always available as a tab.

activity-structures*	A grouping of activities (with attributes to determine whether individual activities are presented as selection or in sequence). At this point there is no facility for coordination of different users doing different things - that has to be done one level up.
environment-ref*	A reference to the environment for this activity-structure
environments	The Environment Container: which contains learning objects and/or services to be used in that activity
environment*	Container for an individual environment (an environment is the collection of resources, services etc necessary for an activity)
title	A short-name for the environment
learning objects*	Learning content utilised within this environment
services*	A service needed for this environment to be utilised
environment-ref*	ref to another environment in the package
metadata	metadata about the environment
method	The key container - cf simple sequencing.
play*	Usually only one, but more than one would run in parallel.
act*	Acts run in sequence, with start triggered by the end of the preceding act. Transitions between acts form synchronisation points for roles. any coordination of events has to be done at this level - it can't be done at the activity level.
role-parts*	Run in parallel - so different roles do different things at the same time. Usually used for learners and teachers, but can be sophisticated - e.g. to support group-setting and role-play
role-ref	ref. to a specific role for this role-part.
activity-ref	ref to activity(-structure) for this role-part.
metadata	Descriptive Metadata for the LD

13.5.LD implementation levels

- There are 3 levels: A, B, C (makes gradual tools development easier)

Level A: Contains the core idea of learning design:

- Ingredients:
  - a series of activities (assessment, discussion, simulation),
  - performed by one or more players (learners, teachers etc.) - roles,
  - in an environment consisting of learning objects or services.
- Coordination of ingredients with method, play, act
- This provides a model for a series of time ordered learning activities to be performed by learners and teachers (one act after each other)

Level B: adds conditions and properties

- conditions (like in SS) allow to program evolution of a scenario in terms of evaluation functions.
- properties are informations about a group of learners (external or generated by activities)

Level C: adds notifications

- Allows to start another activity (jump out/goto)

13.6.Tools and implementations overview

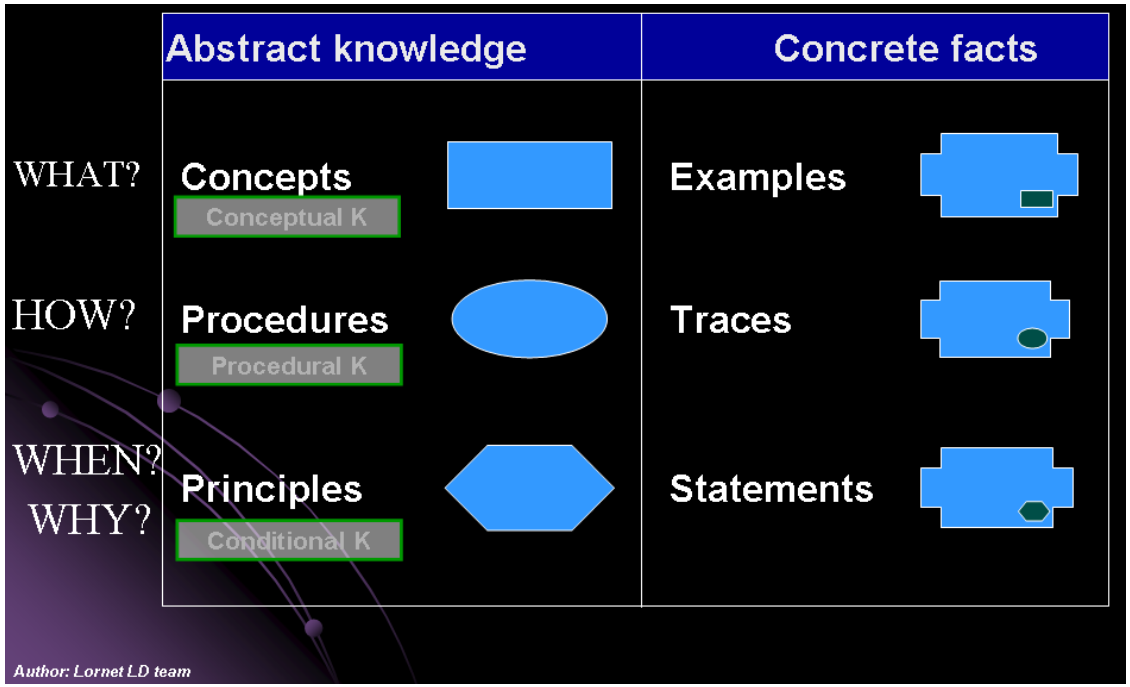
- Most authoring tools and players are not end-user ready so far (5/2007)

url: **http://www.unfold-project.net/general_resources_folder/tools/currenttools** **(list)**

13.7.LD with MotPlus

url: **http://www.cogigraph.com/**

- MotPlus is a hybrid between a specialized Mind Map and a "Case Tool".
- Besides LD, this tool also can model knowledge ontologies and other things.
- MOT can export to LD level A. The manifest can then be imported into Reload in order to add level B,C instructions.
- The MOT LD design language has 6 knowledge types and 7 link types:



Six knowledge element types

(Michel Léonard, Walkenburg workshop, 2005)

Elements of the LD editor component:

RESSOURCES (Concepts)

- Environment
- Learning object or outcome
- Conference
- Send-Mail
- Search by index

ACTIONS (Procedures)

- Method
- Play
- Act
- Activity Structure
- Learning Activity
- Support Activity
- External unit of learning

ACTORS (Principles)

- Learner role #1
- Staff role #1

RULES (Principles)

- Number to select #1
- Time limit #1
- On completion #1

BASIC SPECIFIC OBJECTS (Facts)

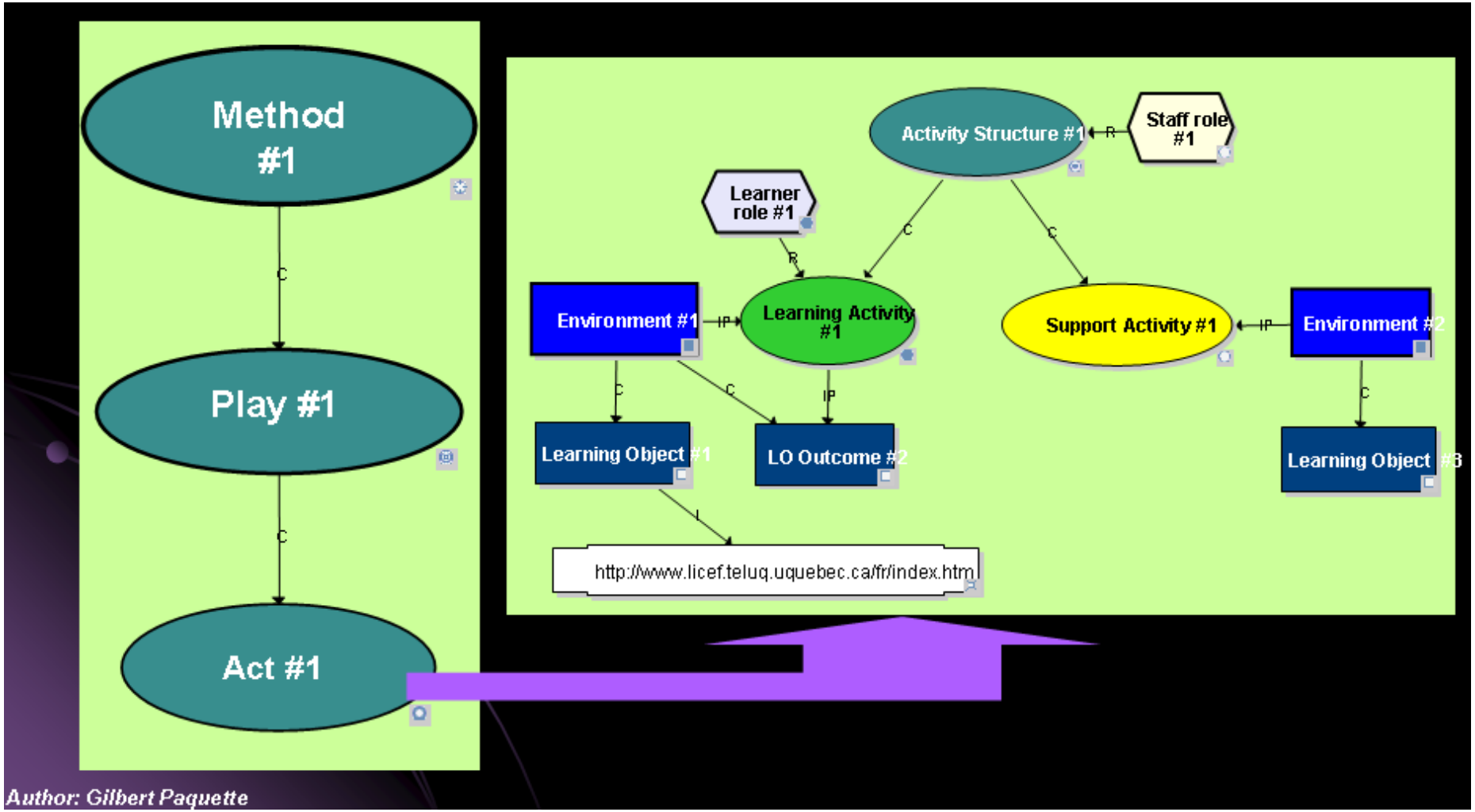
- Learning objectives
- Prerequisites

EXTENDED SPECIFIC OBJECTS (Facts)

- Metadata
- Item
- Class
- Index by Class
- Index by Element
- Index by Type of Element

Example of an activity structure

- MOT activity structure = LD Act
- Each learning activity uses environments that can contain contents or services



MOT links

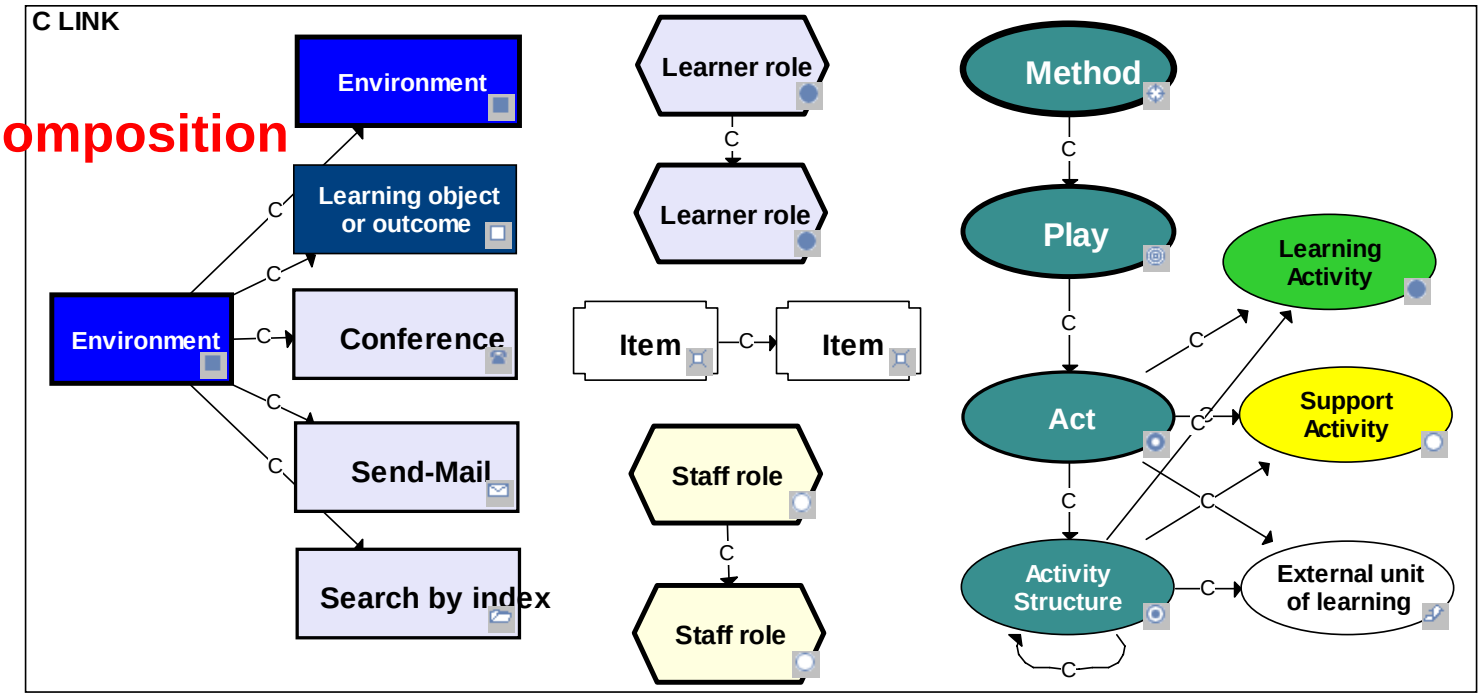
The seven link types:

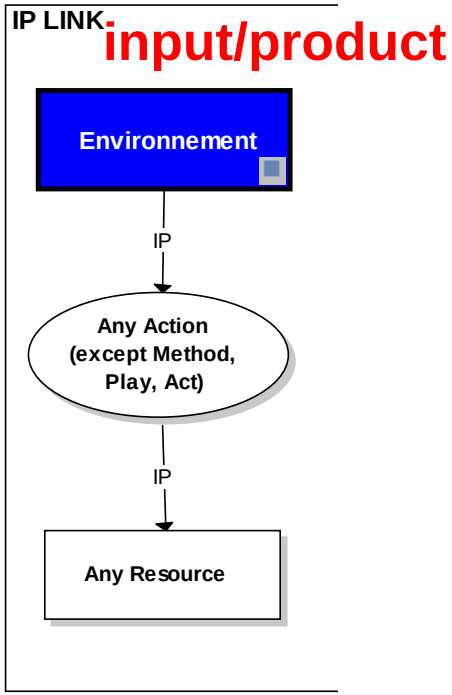
- ***composition*** (C): "x is decomposed in", "y is part of x"
- ***specialization*** (S): "is a sort of"
- ***regulation*** (R): x regulates y (Concept, procedure, etc.)
- ***precedence*** (P), x before y
- ***input /product*** (IP): links a concept and a procedure
- ***application*** (A): links a fact to knowledge
- ***instantiation*** (I) fact is an instance of, object X is a y"

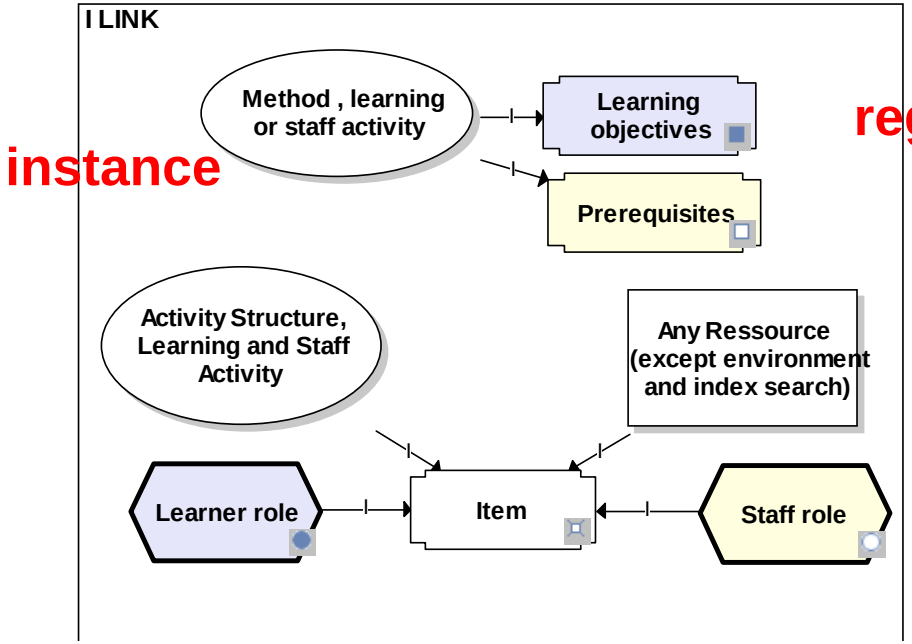
Links grammar

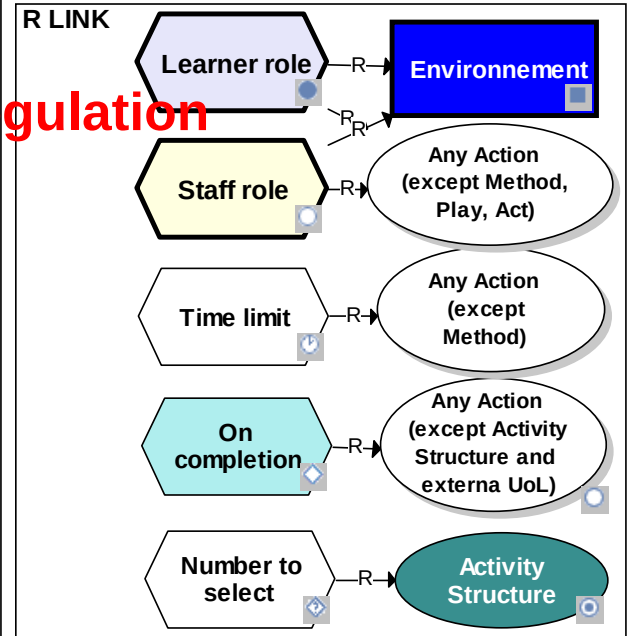
- Direction of arrows: "A -->c-->D" means "D is a component of A"
- There are restrictions on how to use links
  (see next slide)

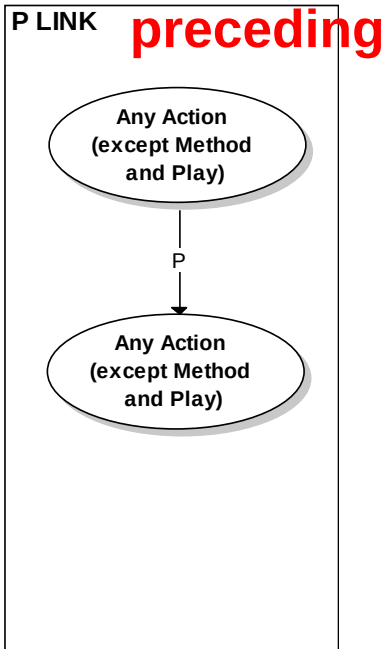
Links summary





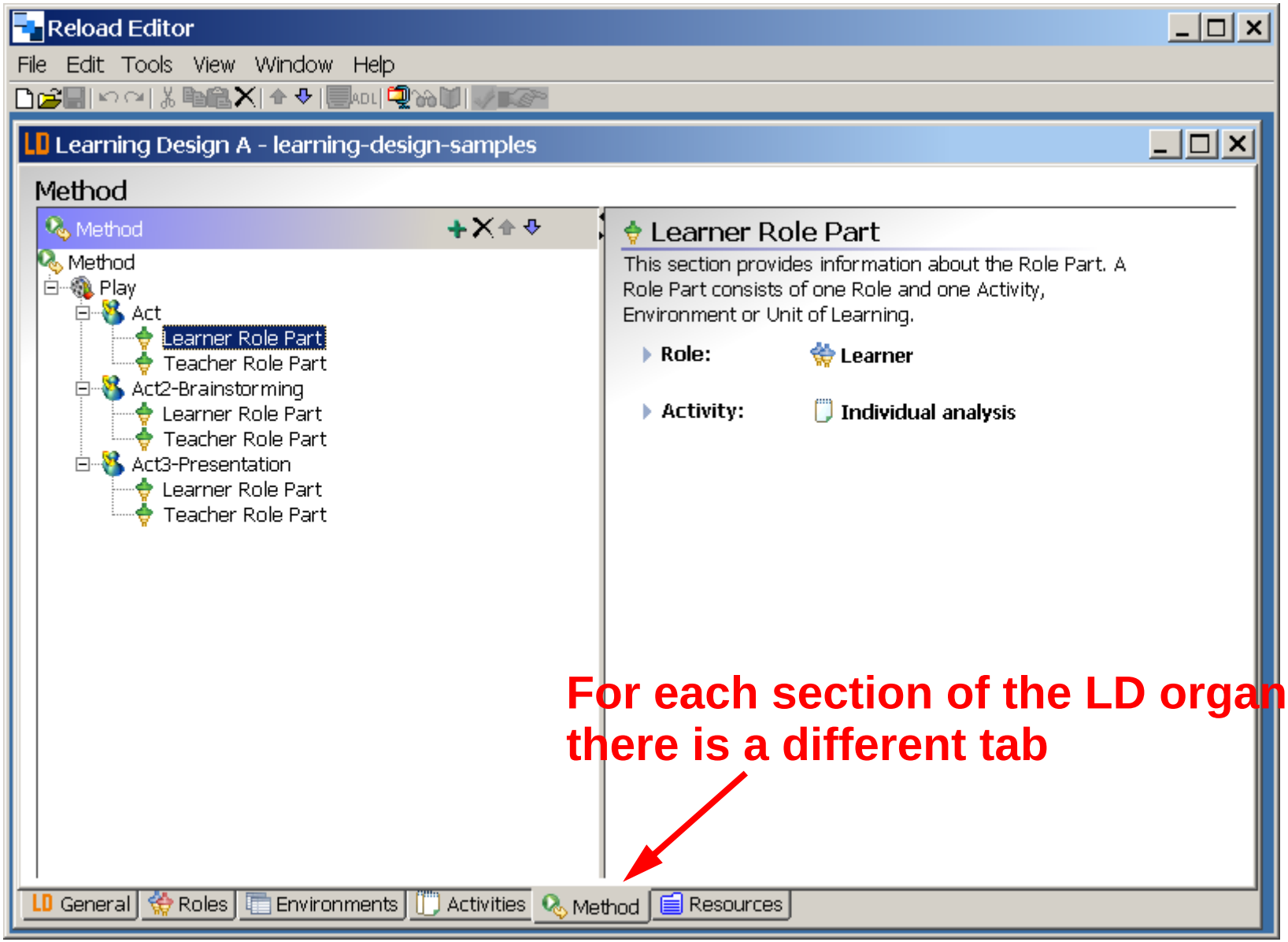






13.8.LD with the Reload Editor

- The last version of the Reload Content Packaging Editor can edit LD level A



For each section of the LD organization there is a different tab

13.9.Reload Learning Design Editor

url: **http://www.reload.ac.uk/tools.html**

Can edit levels A,B,C. It has several tools:

- Overview: define titles, objectives and prerequisites
- Roles: to define roles
- Properties
- Activities: single activities and activity structures
- Environments: Tools and services
- Methods: The play, i.e. the learning design
- Files: Resource management à la IMS CP
- Export: Creation of IMS CP for distribution

Discussion

- Difficult to use because
  - on has to start defining components
  - do a lot of links
  - then define the method (and again do many links)

Screendump

Reload Learning Design Editor

File Edit Window Help

R... Pr...

Learning Designs
- Brainstorming
- test

Brainstorming | test | *Brain Storm Test

Method
- Play
  - Act
    - Learner Role Part
    - Teacher Role Part
  - Act2-Brainstorming
    - Learner Role Part
    - Teacher Role Part
  - Act3-Presentation
    - Learner Role Part
    - Teacher Role Part

Learner Role Part

Role Part

Metadata

Role

- Learners
  - Learner
- Staff
  - teacher

Activity/Environment

- Brainstorming
- Review and clarification
- Presentation of the ideas
- Support analysis and comparing
- Supervision of the Brainstorming
- Guide the review
- Supervision of the presentation
- Giving feedback

Unit of Learning HREF. If this is set, do not select an Activity or Environment.

HREF:

Overview | Roles | Properties | Activities | Environments | Method | Files | Export

14.Modeling (7) - Other learning design environments

- These tools are not IMS LD, although you can store data in LD format (but e.g. for LAMS there is no way to play LAMS contents elsewhere ...).

14.1.LAMS

LAMS is a on-line authoring and activity management system

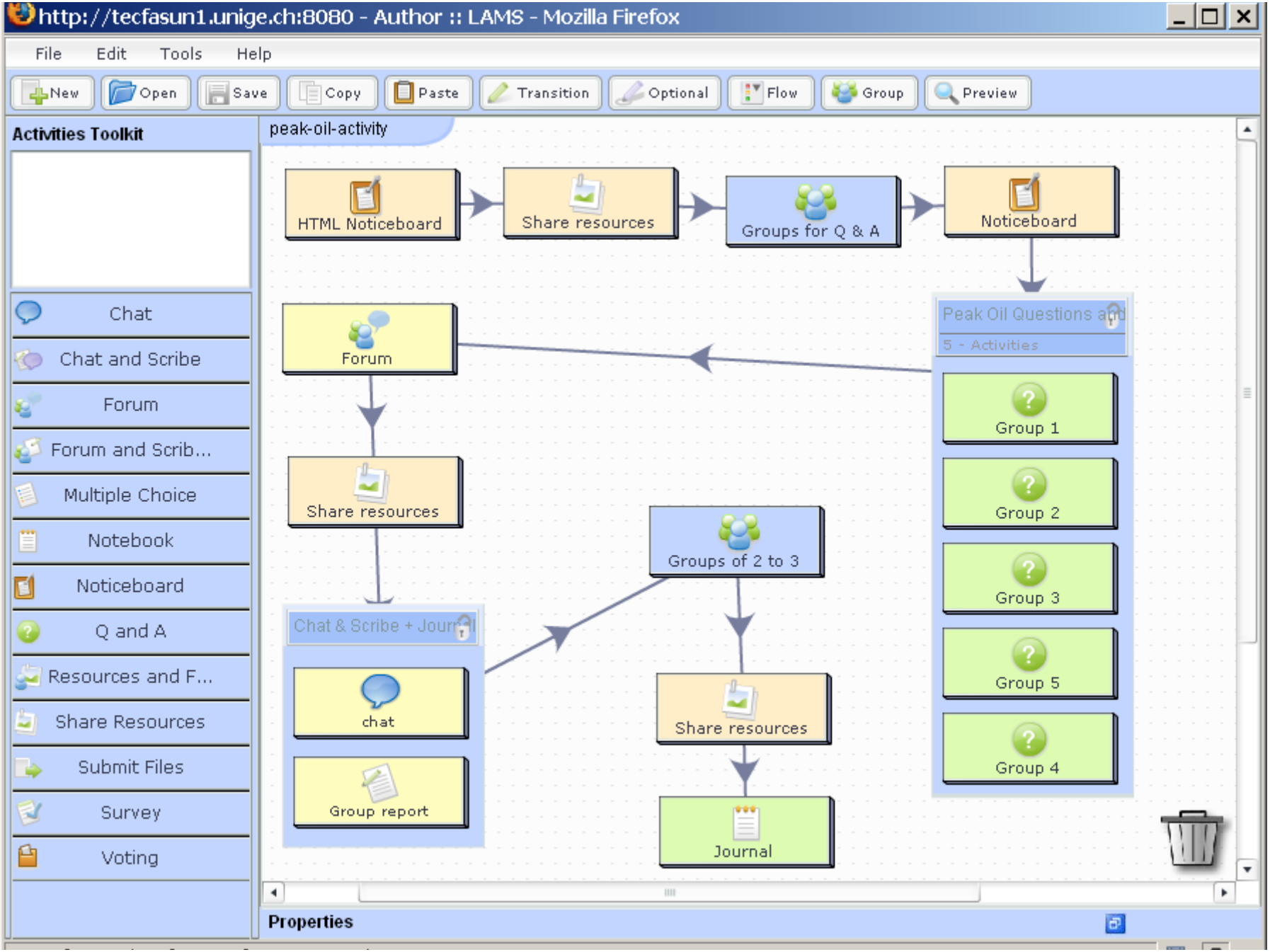
- Authoring / scenarization with a graphical editor
- User and role management
- Scenario execution with the same on-line tool (now that’s nice !)

Inspired by IMS LD

Authoring

- The scenario is defined with activities
- Each activity can be parametrized
  - ex. add contents to a reading activity
  - ex. decide which group has to discuss what
  - ex. people can vote on question
- Activities are sequenced

Screendump of LAMS



15.Modeling (8) - Scenario planners

15.1.DialogPlus

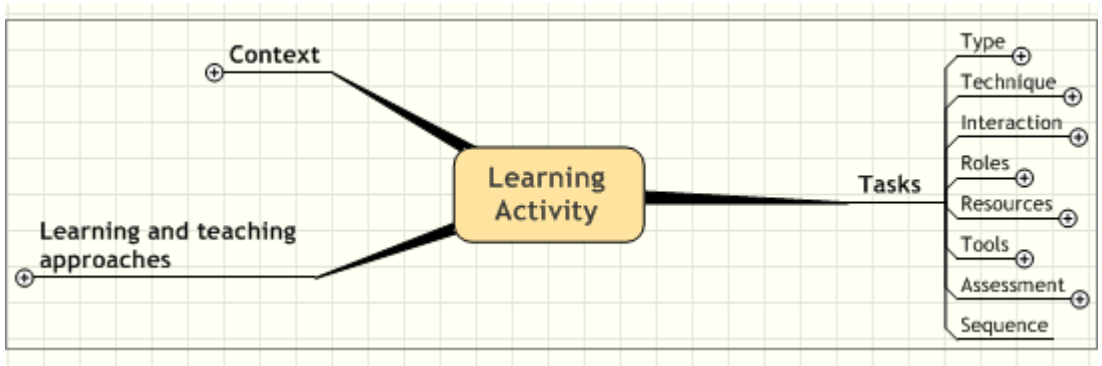
On-line scenario building toolkit (including sharable

url: **http://www.nettle.soton.ac.uk/toolkit/**

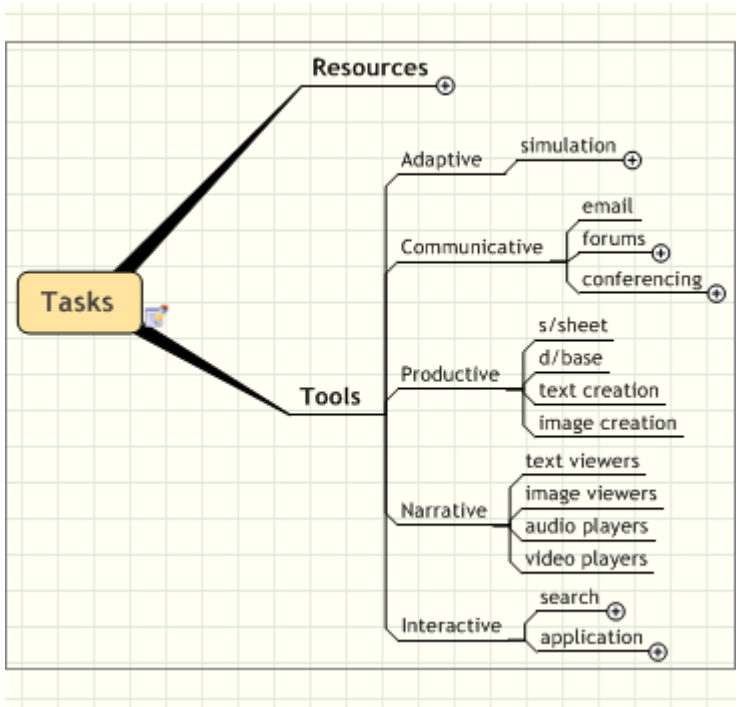
Key concept: the nugget = learning activity

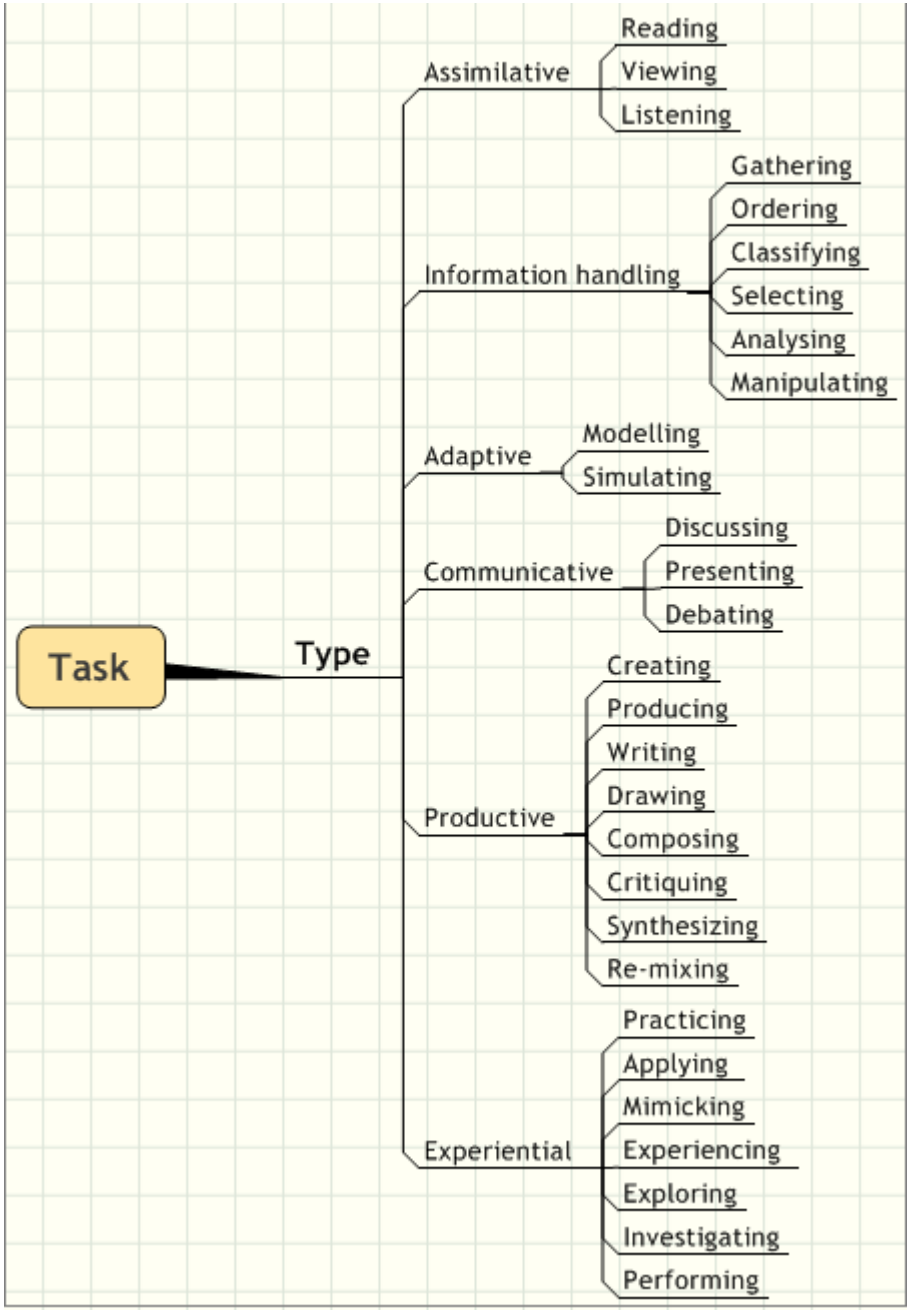
3 elements:

- The ***context*** of the activity: e.g. subject, level of difficulty, intended learning outcomes and the environment within which the activity takes place.
- The ***learning and teaching approaches***: including theories and models.
- The learning ***tasks***: This includes type of task, techniques used, associated tools and resources, interaction and roles of those involved and learner assessment.

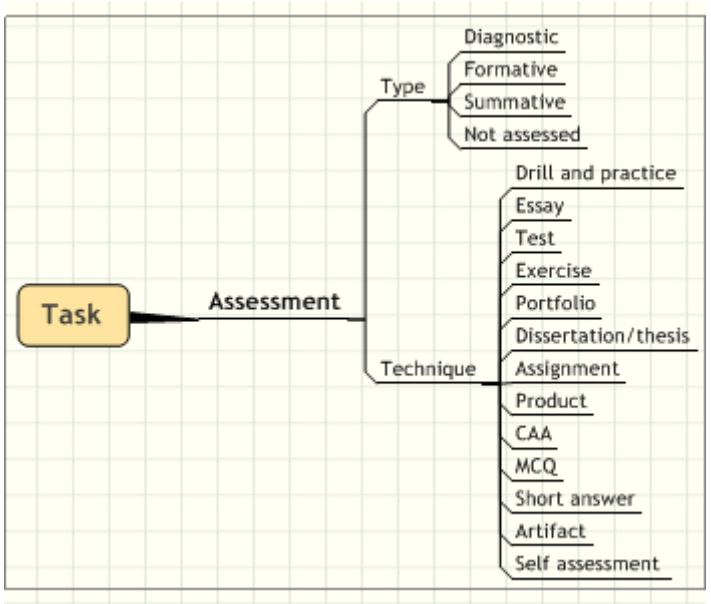


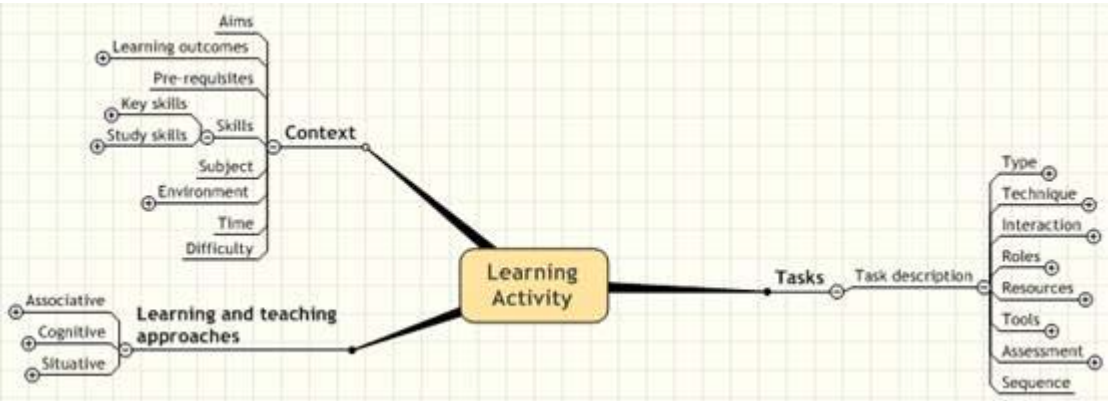
15.2.Tasks in DialogPlus



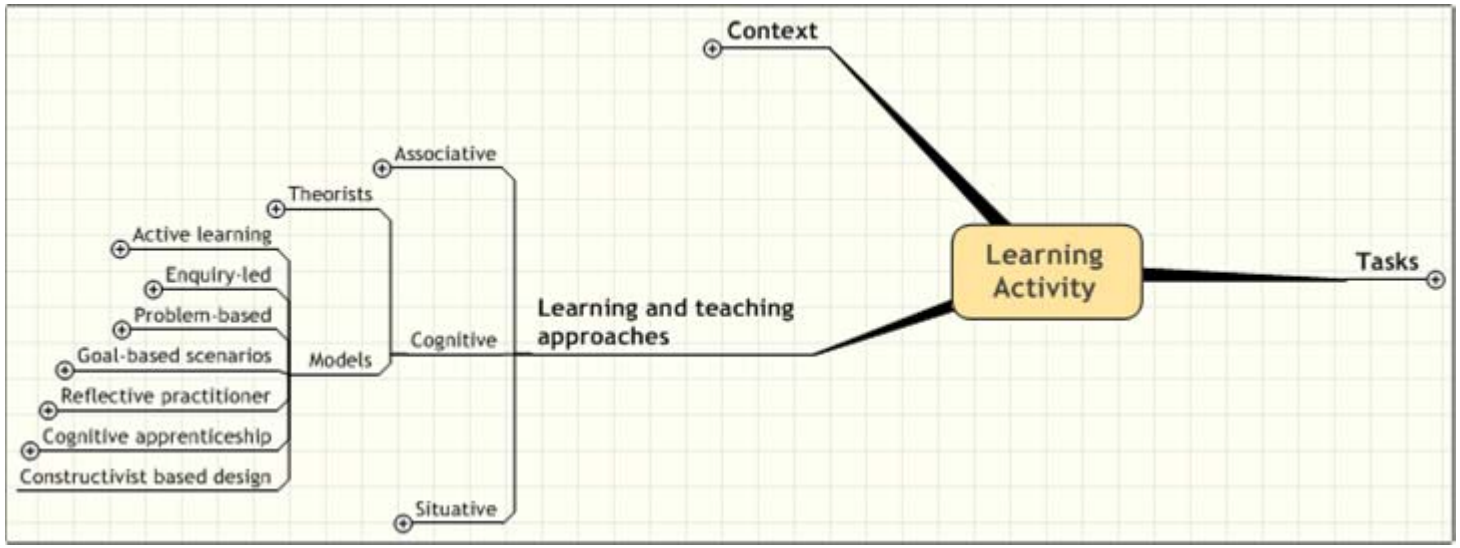


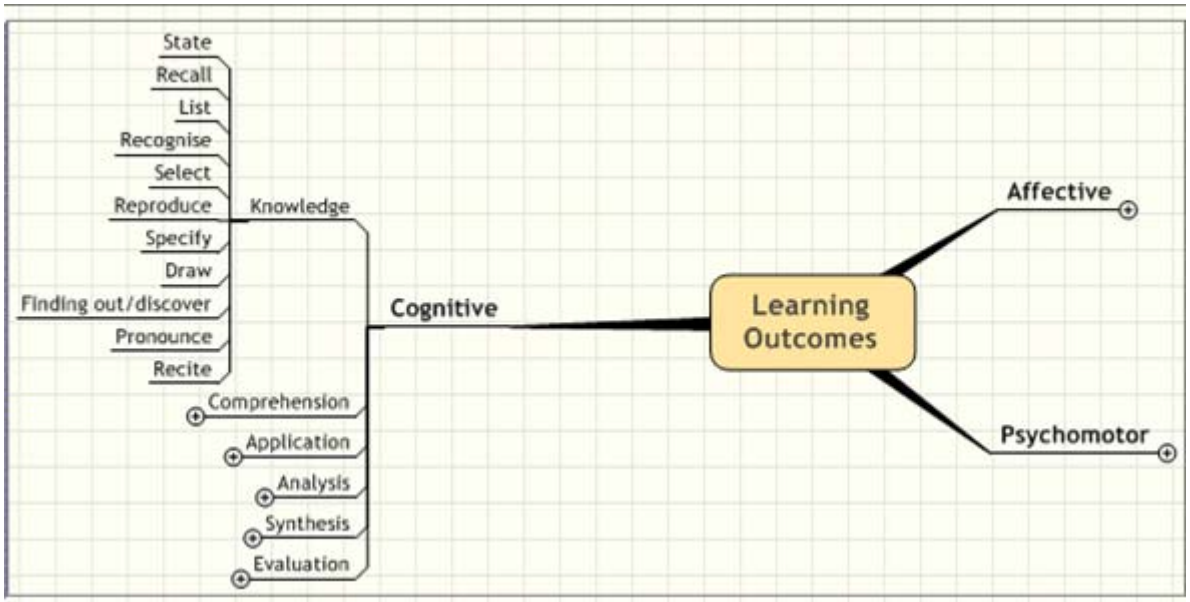
Tasks (2): Evaluation and assessment





Learning and teaching approaches / learning outcomes

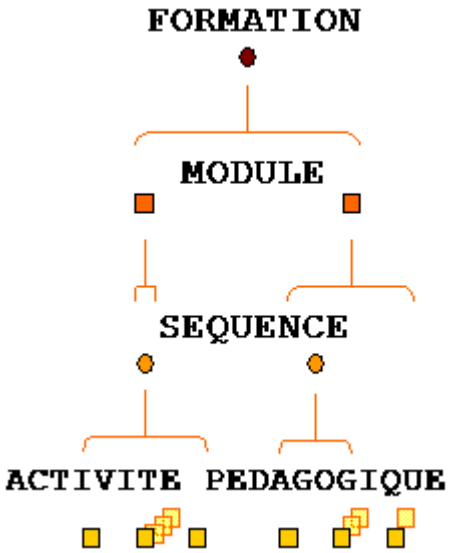




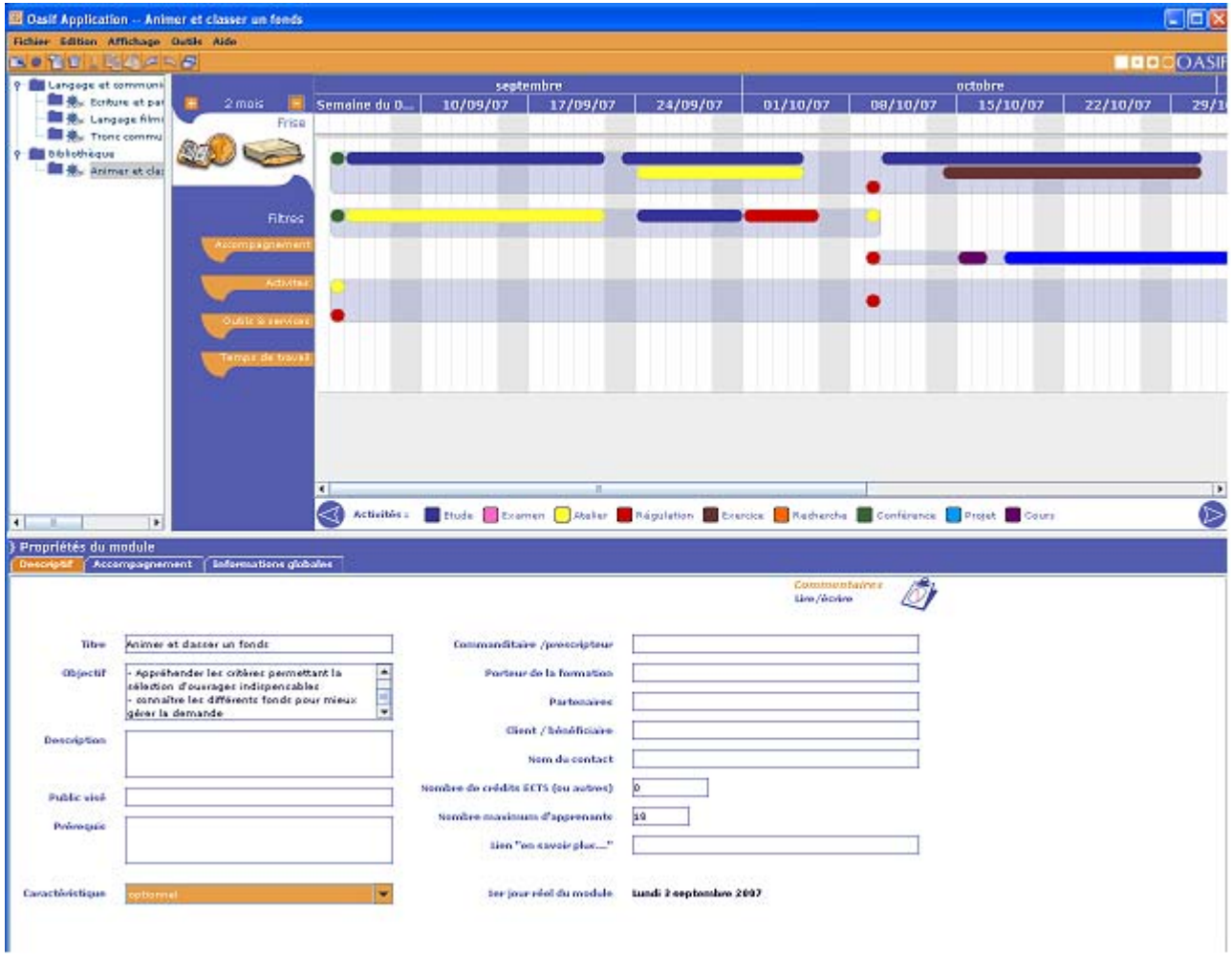
15.3.OASIF

- French scenario planner made for ODL.
- Activity oriented

  url: **http://oasif.educagri.fr/**
- Organization of a teaching unit in with four levels:



Organization and tool looks a bit like a project planner



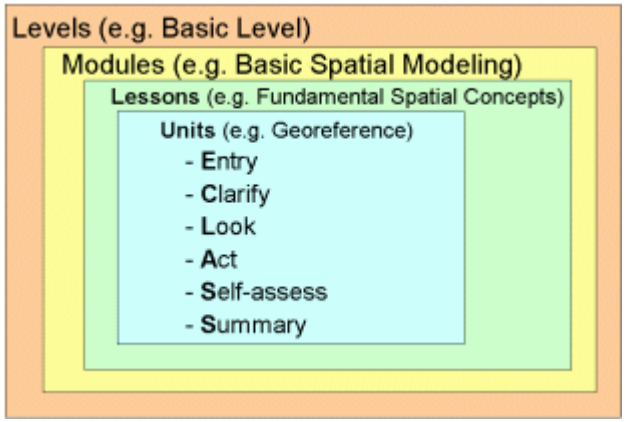
16.Modeling (9) - Back to Contents

- IMS CP/Scorm are not content models, they are aggregators of smaller assets.
- If you plan to produce on-line textbooks, maybe IMS is not a good idea ...
- See also: DocBook, Latex, Semantic XHTML, DITA (e.g. document formats)

16.1.ELML

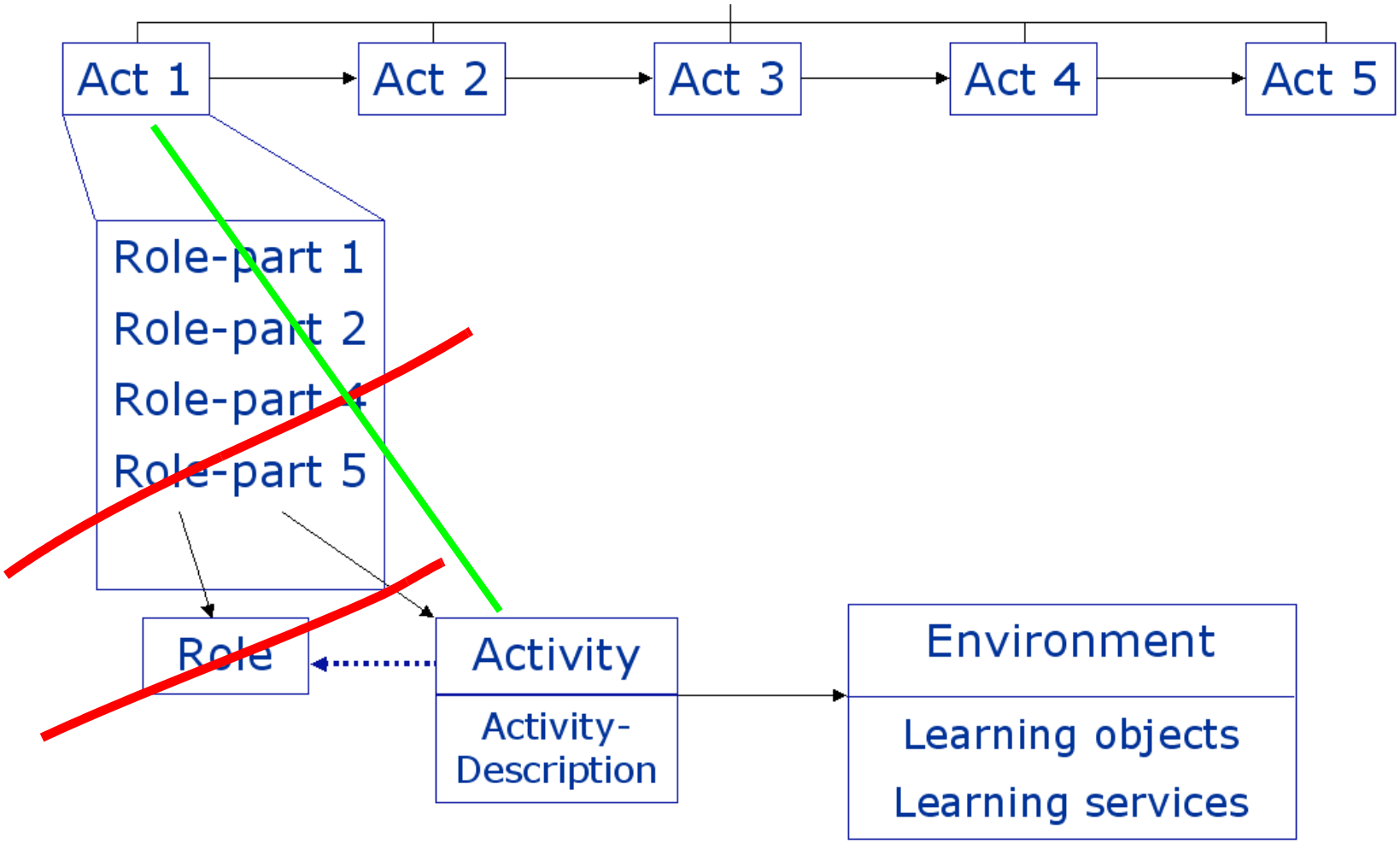
- The eLesson Markup Language (eLML) is an open source XML framework for creating structured eLessons using XML.
- elML is an authoring toolkit that doesn't need a specific environment to run the contents

url: **http://edutechwiki.unige.ch/en/ELML**



17.Conclusion

17.1.You don't need to be formal and you can simplify



17.2.There are many tools to build an activity-based environment

- Just do it ... but think in terms of storyboarding