

# Instruktionsdesign und Kursgenerierung/ Instructional design and tutorial planning

Special Course “Educational Technologies”

Carsten Ullrich  
cullrich@dfki.de



**2.3.7 Definition** Let  $a \in \mathbb{R}$  and  $\varepsilon > 0$ . Then the  $\varepsilon$ -neighborhood of  $a$  is the set  $V_\varepsilon(a) = \{x \in \mathbb{R} : |x - a| < \varepsilon\}$ .

For  $a \in \mathbb{R}$ , the statement that  $x$  belongs to  $V_\varepsilon(a)$  is equivalent to either of the statements

$$-\varepsilon < x - a < \varepsilon \iff a - \varepsilon < x < a + \varepsilon.$$

(See Figure 2.3.2.)



FIGURE 2.3.2 An  $\varepsilon$ -neighborhood of  $a$ .

**2.3.8 Theorem** Let  $a \in \mathbb{R}$ . If  $x$  belongs to the neighborhood  $V_\varepsilon(a)$  for every  $\varepsilon > 0$ , then  $x = a$ .

**Proof.** If a particular  $x$  satisfies  $|x - a| < \varepsilon$  for every  $\varepsilon > 0$ , then it follows from 2.2.9 that  $|x - a| = 0$ , and hence  $x = a$ . Q.E.D.

**2.3.9 Examples** (a) Let  $U := \{x : 0 < x < 1\}$ . If  $a \in U$ , then let  $\varepsilon$  be the smaller of the two numbers  $a$  and  $1 - a$ . Then  $V_\varepsilon(a)$  is contained in  $U$ . Thus each element of  $U$  has some  $\varepsilon$ -neighborhood of it contained in  $U$ .

(b) If  $I := \{x : 0 \leq x \leq 1\}$ , then for any  $\varepsilon > 0$ , the  $\varepsilon$ -neighborhood  $V_\varepsilon(0)$  of 0 contains points not in  $I$ , and so  $V_\varepsilon(0)$  is not contained in  $I$ . For example, the number  $x_\varepsilon := -\varepsilon/2$  is in  $V_\varepsilon(0)$  but not in  $I$ .

(c) If  $|x - a| < \varepsilon$  and  $|y - b| < \varepsilon$ , then the Triangle Inequality implies that

$$\begin{aligned} |(x + y) - (a + b)| &= |(x - a) + (y - b)| \\ &\leq |x - a| + |y - b| < 2\varepsilon. \end{aligned}$$

Thus if  $x, y$  belong to the  $\varepsilon$ -neighborhoods of  $a, b$ , respectively, then  $x + y$  belongs to the  $2\varepsilon$ -neighborhood of  $a + b$  (but not necessarily to the  $\varepsilon$ -neighborhood of  $a + b$ ).

### Exercises for Section 2.3

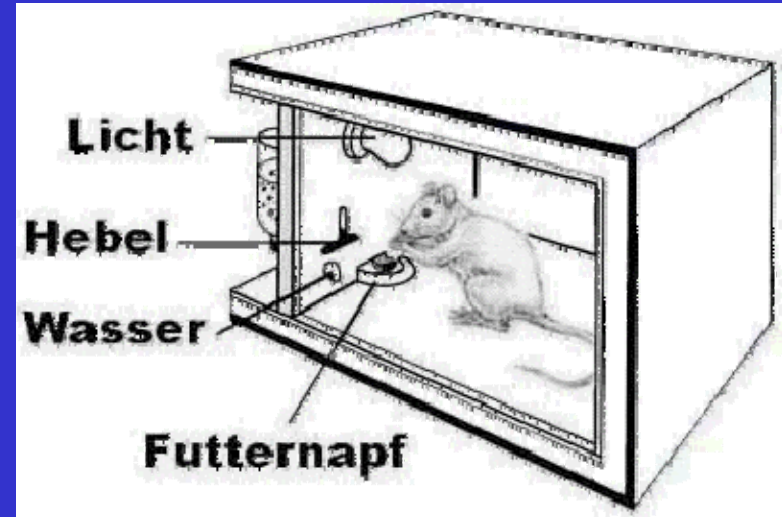
1. Let  $a \in \mathbb{R}$ . Show that we have:

- After this lecture you should know
  - what is **tutorial knowledge**
  - how to **represent it**
  - how to **apply it**



# Theory: Learning Theories

- Behaviorism (Skinner)
  - Stimulus–Response coupling
- Programmed Instruction
  - Intermittent reinforcement:
    - Reward correct answers
    - Punish incorrect answers
    - Break learning material down into smallest units possible



# Programmed Instruction: Example

- The important parts of a flashlight are the battery and the bulb. When we "turn on" a flashlight, we close a switch which connects the battery with the \_\_\_\_\_ .
  - bulb
- When we turn on a flashlight, an electric current flows through the fine wire in the \_\_\_\_\_ and causes it to grow hot.
  - bulb
- When the hot wire glows brightly, we say that it gives off or sends out heat and \_\_\_\_\_ .
  - light
- The fine wire in the bulb is called filament. The bulb "lights up" when the filament is heated by the passage of a(n) \_\_\_\_\_ current
  - electric
- When a weak battery produces little current, the fine wire, or \_\_\_\_\_ , does not get very hot.
  - filament



# Test

- The scientist \_\_\_\_\_ is one of the fathers of Behaviorism

Would you like to learn this way?

- For what subject matters is this appropriate?

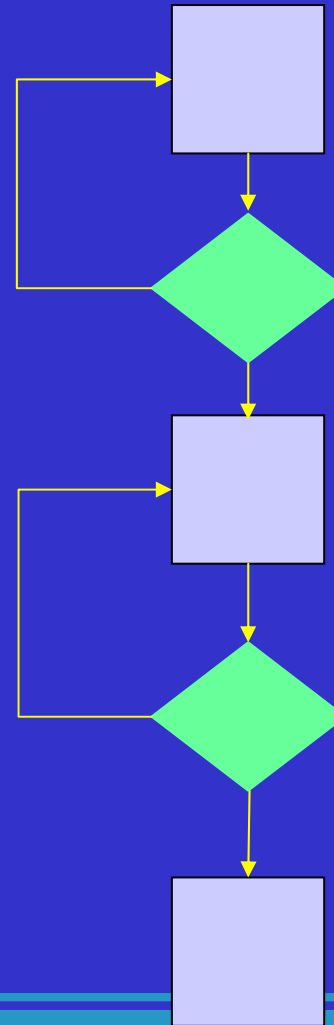
- What is the difference between these questions and the Programmed Instruction questions?

– Rewarding, punishing



# Practical Issues

- Implementation:
  - Fixed sequence of steps (flow chart)
- Attractive because seems to be an easy way of creating learning material...
- ... but is expensive if you really follow the rules:
  - decomposing the learning material in the smallest units possible
- Evaluation with mixed results



# Theories of instruction (or instructional design)

- Gagné: “Conditions of Learning and Events of Instruction”
  - based on Bloom’s taxonomy of learning objectives
  - assigns specific instructional events to the objectives
- ID is prescriptive: Suggesting optimal methods for the achievement of certain learning objectives
- Merrill: „Component Display Theory“
  - operationalisation of Gagné
  - ID Expert: Authoring Tool



# Example: TICCIT (Merrill, 1980)

Here is the general rule  
for grammar-referent agreement.

A pronoun agrees in number  
with its REFERENT. Singular  
referents take singular  
pronouns. Plural referents  
take plural pronouns.  
Singular referents which  
have no sex indicated take  
the generic pronouns him/he/his.

RULE

page 1/1

# Example: TICCIT (Merrill, 1980)

Edit any pronoun in the passage below that doesn't agree in number with its referent. If all pronouns are correct, press ENTER.

Several of the mechanics brought his tools.

PRACT

3 easy page 1/1



# Instructional Design: Assumptions

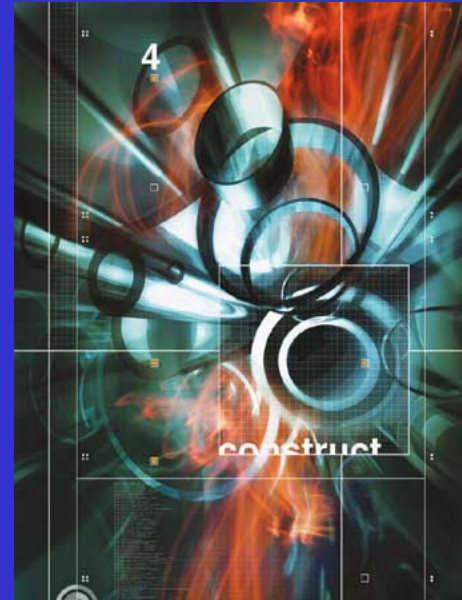
- Instruction, in large measure, **communicates accepted meaning**. The developer of instruction explicitly desires **that the learner adopt the meaning intended by the developer**, and **not reach a separate and personal interpretation** of that meaning. ... most instruction... concerns **transferring**, as effectively and efficiently as possible, **determined interpretations**

Jones, Li and Merrill (1990)



# Learning Theories: Constructivism

- Knowledge is a **result of cognitive processes**
- Constructivism emphasizes **active interpretation by the perceiving subject**, the process of the **actual construction of meaning**
- No transfer of knowledge from one head to another



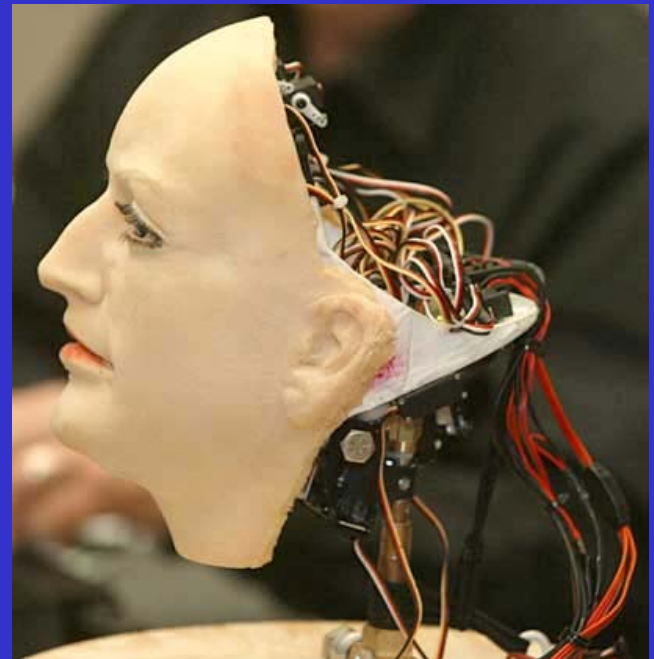
# Tutorial Knowledge for Course Generation

- Adapt learning materials (Course) with respect to
  - the individual (learning goals, knowledge, emotions, ...)
  - the context (location, ...)



# Pedagogical Knowledge

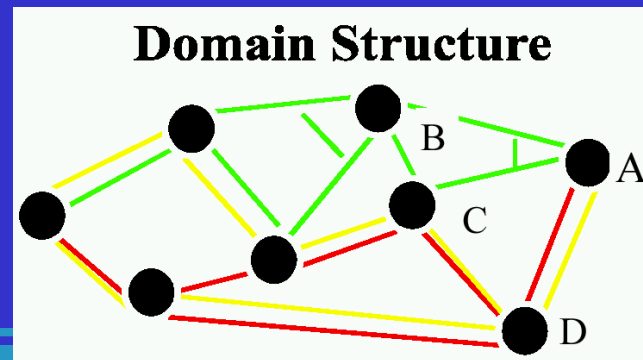
- Represent pedagogical knowledge at abstract level, independent of
  - learned domain
  - concrete learning environment
  - domain knowledge representation



# Dynamic Course Generator



- Julita Vassileva 1992
- Dynamically plans **content of course**
- Separation between
  - Content (**Learning Objects**)
  - **Domain Concept Structure: AND/OR-Graph**
    - Nodes: Concepts
    - Edges: Relationships (e.g., generalization ...)

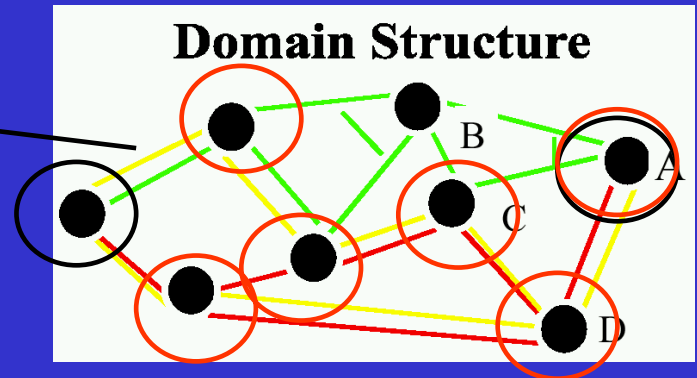


# Content Planning in DCG

- Generates a sub-graph of domain structure

- Input:

- relationship
- student's knowledge
- teaching goal
- heuristic function



- Generated content plan is passed to Executor
  - selects teaching materials
  - updated student model
  - re-invokes planner in case of failure

# Delivery Planning: GTE

- Generic Tutoring Environment  
1989–1995 Kris van Marcke
- Assumption:
  - **Generic Instructional Knowledge exists and can be modeled**
- Instructional Task
- Instructional Methods
- Instructional Objects

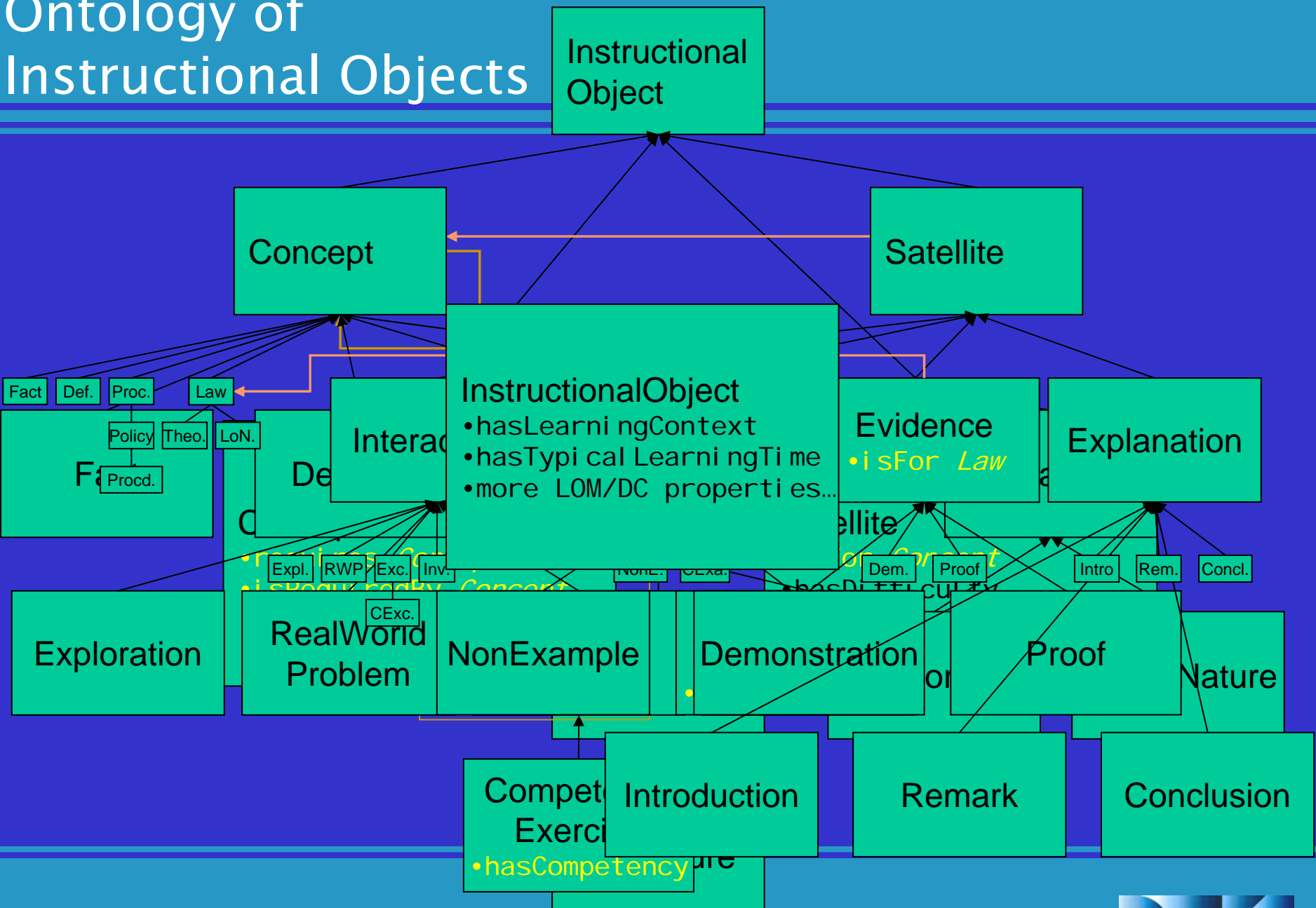


# Ontology of Instructional Objects

- Captures relevant
  - objects
  - relationshipsfor describing content from pedagogical perspective.
- **Design goals:** Domain independence, pedagogical flexibility, completeness, compatibility



# Ontology of Instructional Objects



# Ontology of Instructional

**2.3.7 Definition** Let  $a \in \mathbb{R}$  and  $\varepsilon > 0$ . Then the  $\varepsilon$ -neighborhood of  $a$  is the set  $V_\varepsilon(a) = \{x \in \mathbb{R} : |x - a| < \varepsilon\}$ .

For  $a \in \mathbb{R}$ , the statement that  $x$  belongs to the  $\varepsilon$ -neighborhood of  $a$  is equivalent to the statement

$$-\varepsilon < x - a < \varepsilon \iff a - \varepsilon < x < a + \varepsilon.$$

(See Figure 2.3.2.)

**FIGURE 2.3.2** An  $\varepsilon$ -neighborhood of  $a$ .

Definition  
"epsilon-neighborhood"

Conc

ellite

**2.3.8 Theorem** Let  $a \in \mathbb{R}$ . If  $x$  belongs to the  $\varepsilon$ -neighborhood of  $a$  and  $\varepsilon > 0$ , then  $x = a$ .

**Proof.** If a particular  $x$  satisfies  $|x - a| < \varepsilon$ , then from 2.2.9 that  $|x - a| = 0$ , and hence  $x = a$ .

Theorem  
"neighborhood-equality"

Proof  
for "neighborhood-equality"

Explanation

**2.3.9 Examples** (a) Let  $U := \{x \in \mathbb{R} : x < 1\}$  and  $V := \{x \in \mathbb{R} : x < 1/2\}$ . Then  $V$  is smaller of the two numbers  $a$  and  $1 - a$ . Then  $V_\varepsilon(a)$  is contained in  $U$ . Thus each element of  $U$  has some  $\varepsilon$ -neighborhood of it contained in  $U$ .

(b) If  $I := (x : 0 \leq x \leq 1)$ , then for any  $\varepsilon > 0$ ,  $I$  contains points not in  $I$ , and so  $V_\varepsilon(0)$  is not contained in  $I$ . For example, the number  $x_\varepsilon := -\varepsilon/2$  is in  $V_\varepsilon(0)$  but not in  $I$ .

(c) If  $|x - a| < \varepsilon$  and  $|y - b| < \varepsilon$ , then

$$\begin{aligned} |(x + y) - (a + b)| &= |(x - a) + (y - b)| \\ &\leq |x - a| + |y - b| < 2\varepsilon. \end{aligned}$$

Thus if  $x, y$  belong to the  $\varepsilon$ -neighborhoods of  $a, b$ , respectively, then  $x + y$  belongs to the  $2\varepsilon$ -neighborhood of  $a + b$  (but not necessarily to the  $\varepsilon$ -neighborhood of  $a + b$ ).

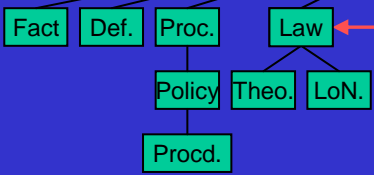
Example  
for "epsilon-neighborhood"

Intro Rem. Concl.

**Exercises for Section 2.3**

- Let  $a \in \mathbb{R}$ . Show that we have
  - $|a| = \sqrt{a^2}$ ,
  - If  $a, b \in \mathbb{R}$  and  $b \neq 0$ , show that  $|a/b| = |a|/|b|$ .
  - If  $a, b \in \mathbb{R}$ , show that  $|a + b| \leq |a| + |b|$ .

Exercise  
for "epsilon-neighborhood"

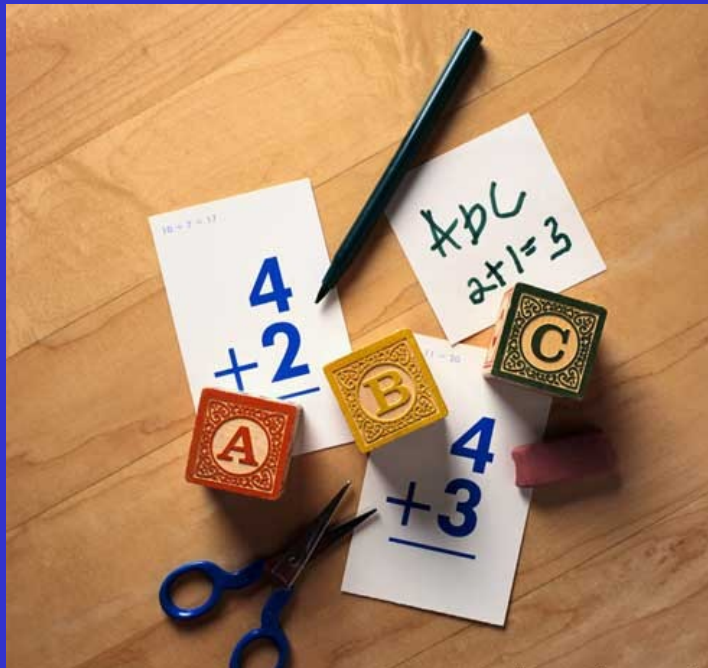


Expl.



# Representing Pedagogical Knowledge

- Pedagogical Task
- Pedagogical Methods
- Pedagogical Strategy: collection of tasks and methods



# Pedagogical Tasks

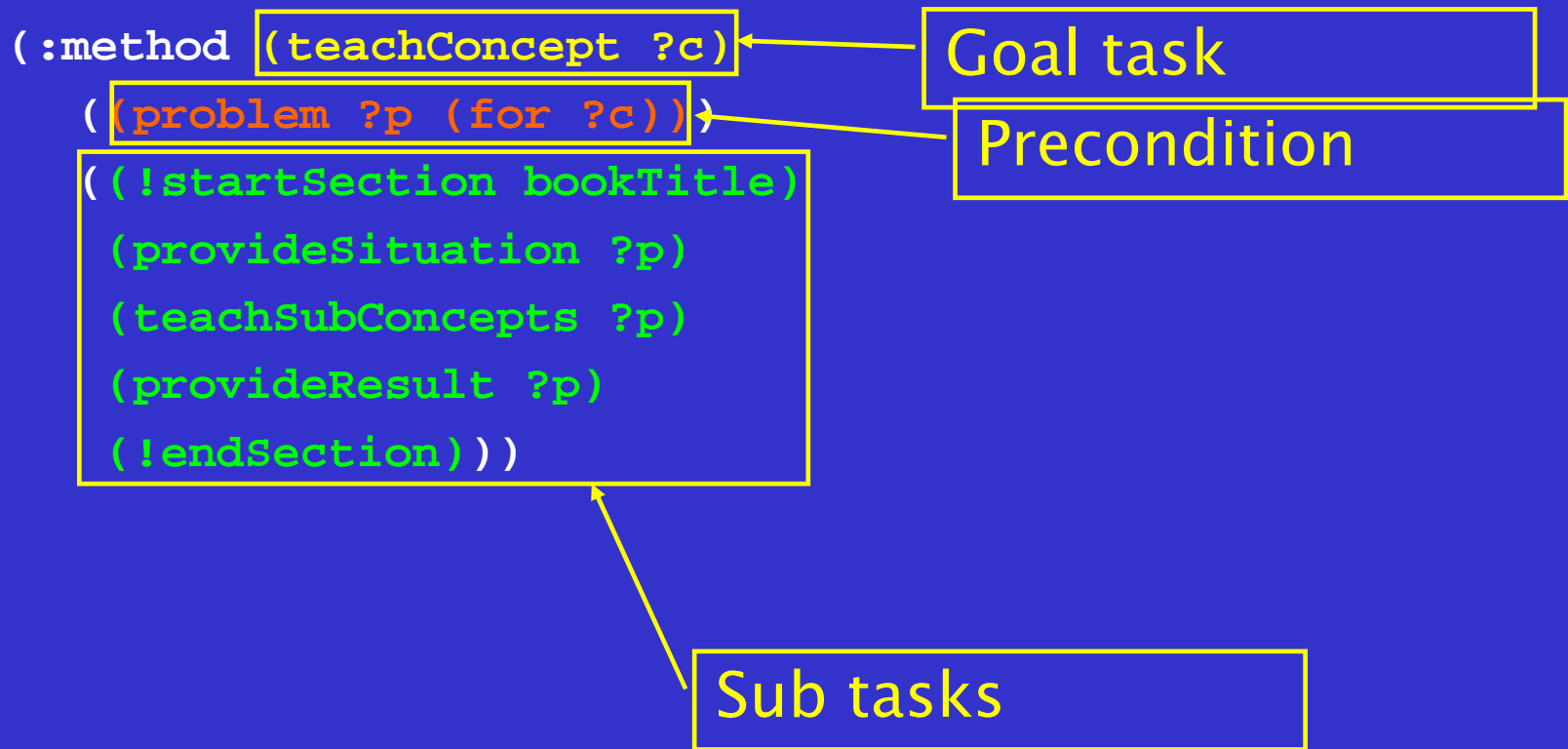
Pedagogical activity accomplished by instructor during pedagogical process

- `teachConcept`
- `selectAdequateExercise,`  
`selectDifficultExercise,`  
`selectEasyExercise`
- `motivate`
- `helpAboutConcept`
- `teachPrerequisites`
- `challenge`
- `illustrate`



# Pedagogical Method

- Describes how to achieve a task



# Adapting to the Learner's State

- Mastery:
  - Train insufficiently mastered competencies
  - Selecting an exercise from the adequate competency level

- Motivation

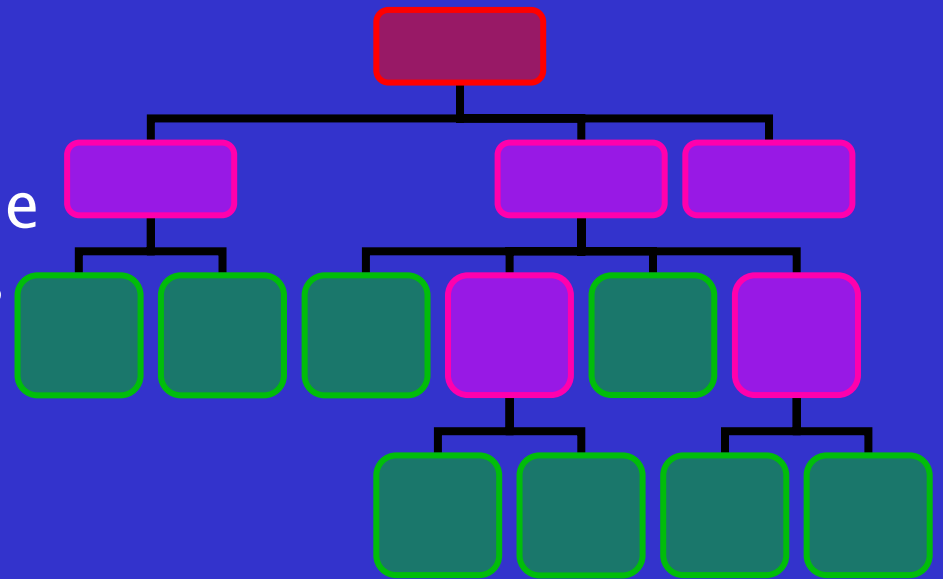
```
(:method (motivate ?c)  
  ((problem_solving_comp high))  
  ((presentEasyExercise ?c)))
```

```
(:method (motivate ?c)  
  ((problem_solving_comp low))  
  ((presentEasyExample ?c)))
```



# Implementation

- Hierarchical Task Network planning (HTN)
  - very efficient
  - straight-forward way for representing human expert knowledge.
- Planning problem:
  - task networks
  - methods decompose non-primitive tasks into sub-tasks



# Pedagogical Scenario: Example

```
(:method (teachSingleConcept ?c)
```

```
  ()
```

```
  (!startSection ?c)
```

```
  (tryMotivate ?c)
```

```
  (tryIntroduce ?c)
```

```
  (insertElement ?c)
```

```
  (tryExplain ?c)
```

```
  (tryExemplify ?c)
```

```
  (tryExercise ?c)
```

```
  (tryConclude ?c)
```

```
  (!endSection)
```

```
)
```

```
)
```



# Pedagogical Scenario: Example

```
(:method (exemplify ?c)
  ((learnerProperty mastery ?c low))
  ((insertAllExamples ?c very_easy)
   (tryInsertExample ?c easy)
   (tryInsertExample ?c easy)
   (tryInsertExample ?c easy)
   (tryInsertExample ?c medium)
   (tryInsertExample ?c medium)
  )
  ((learnerProperty mastery ?c medium))
  (
    (tryInsertExample ?c very_easy)
    (tryInsertExample ?c very_easy)
    (tryInsertExample ?c easy)
    (tryInsertExample ?c easy)
    (insertAllExamples ?c medium)
    (tryInsertExample ?c difficult)
    (tryInsertExample ?c difficult)
  )
  ...
)
```



# Pedagogical Scenario: Example

```
(:method (tryInsertExample ?c ?difficulty)
  ()
  ((insertExample ?c ?difficulty))
)
```

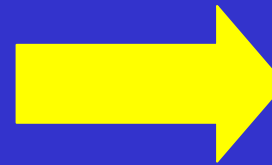
```
(:method (tryInsertExample ?c ?difficulty)
  ()
  ()
)
```



# Pedagogical Scenario: Example

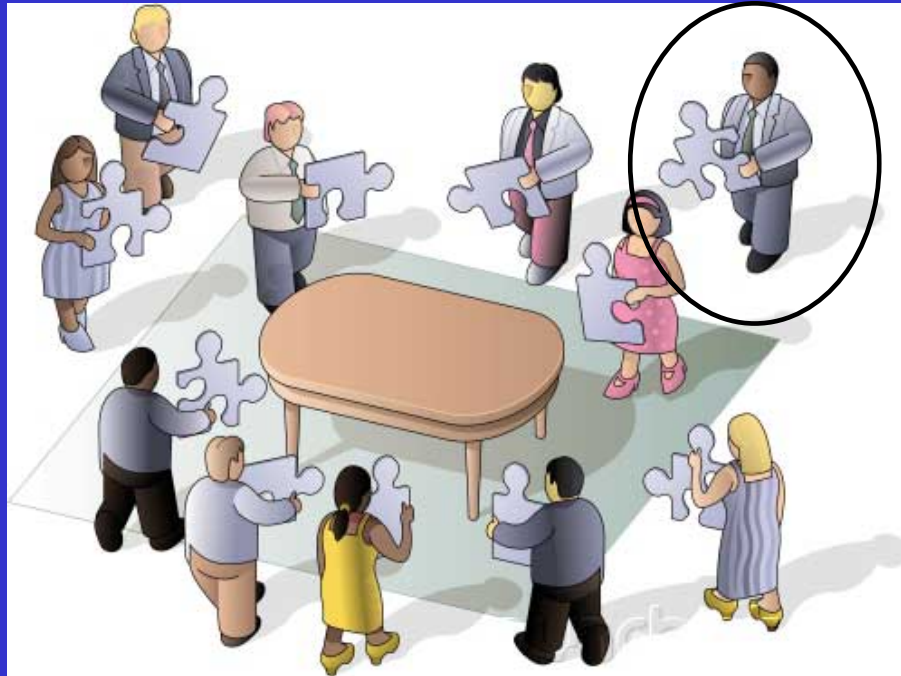
- Result of planning process is a **tree of nested sections** with the leaves being pointers to LOs

```
<section>
  <title>Grundbegriffe...</title>
  <section>
    <title>Grundbegriffe...</title>
    <ref xref="c1">
    <ref xref="e1">
    <ref xref="e2">
  </section>
  <section>
    <title>Übungsaufgaben</title>
    <ref xref="c2">
    <ref xref="e3">
    <ref xref="e4">
  </section>
  ...
</section>
```



▼ Grundbegriffe der Mengenlehre und der Logik
■ Grundbegriffe der Mengenlehre und der Logik
■ Übungsaufgaben
■ Schwerpunkte
▼ Reelle Zahlen
■ Einführung
■ Eigenschaften der reellen Zahlen - Axiome
▼ Rechnen mit reellen Zahlen
■ Beweis Existenz m-te Wurzel
■ Absolutbetrag von reellen Zahlen
■ Mengen von reellen Zahlen
■ Übungsaufgaben
■ Schwerpunkte
▼ Folgen von reellen Zahlen
■ Einführung

# Integration of Distributed Content



## Knowledge Sources

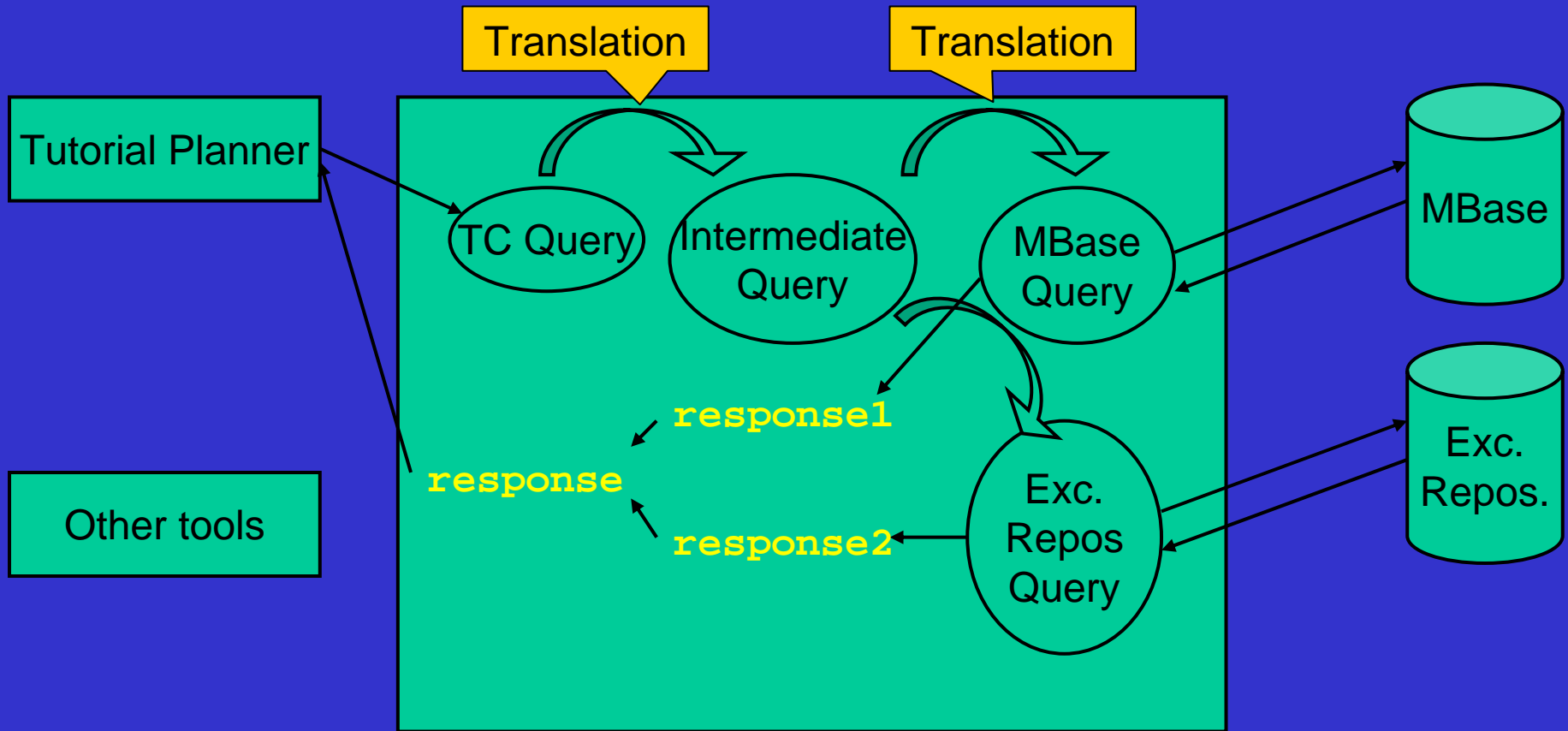
Planning?  
Knowledge  
representation?  
On-demand  
generation?

# Information Integration using a Mediator

- Information Integration using a **Mediator**
  - Reformulate,
  - Query,
  - Collect,
  - Send back
- Methods query the mediator
- A **single interface** to several repositories



# Sketch of the Mediator




# Integration of Distributed Content: Example

```
(:method (insertExample ?c ?difficulty)
  ((assign ?element
    (call
      GetElements
      ((class example)
        (relation for ?c)
        (property difficulty ?difficulty)
      ))
    ))
  ((insertElement ?element))
)
```

Problem: XML RPC expensive! From 0.4 sec. to 2.5 sec.!



# Dynamic Adaptivity

- Early course generation supports
  - orientation
  - self-organization
- Problem: **changing capabilities** of the learner 
- ⇒ Delayed Task Execution:
  - planning stops at the level of specially marked tasks
  - instantiation at first visit
- New possibilities for authors!

## ▼ Grundbegriffe der Mengenlehre und der Logik

■ Grundbegriffe der Mengenlehre und der Logik

■ Übungsaufgaben

■ Schwerpunkte

## ▼ Reelle Zahlen

■ Einführung

■ Eigenschaften der reellen Zahlen - Axiome

## ▼ Rechnen mit reellen Zahlen

■ Beweis Existenz  $m$ -te Wurzel

■ Absolutbetrag von reellen Zahlen

■ Mengen von reellen Zahlen

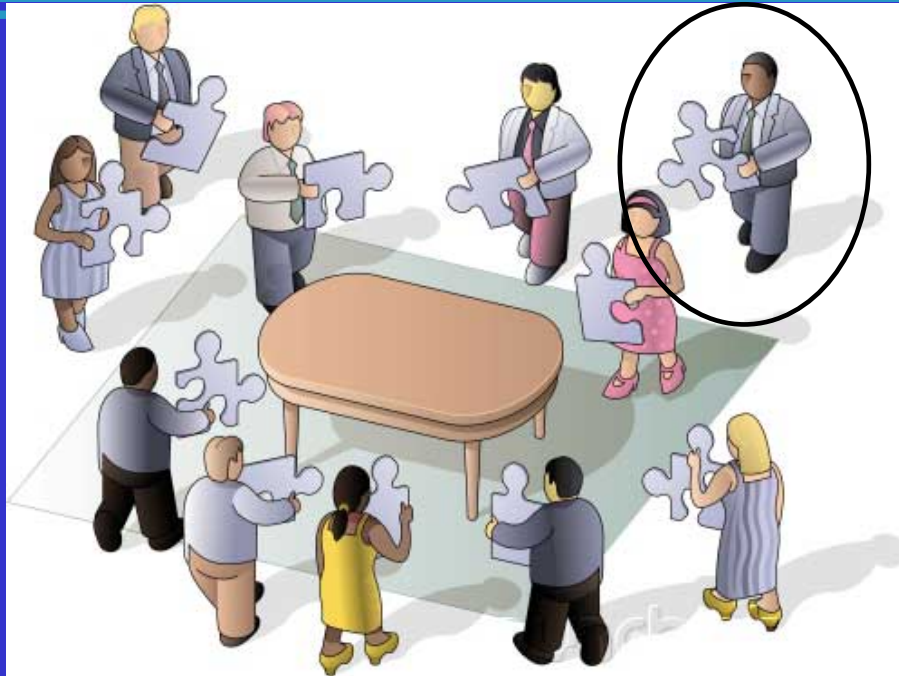
■ Übungsaufgaben

■ Schwerpunkte

## ▼ Folgen von reellen Zahlen

■ Einführung

# Integrating Intelligent Applications



Intelligent  
Applications

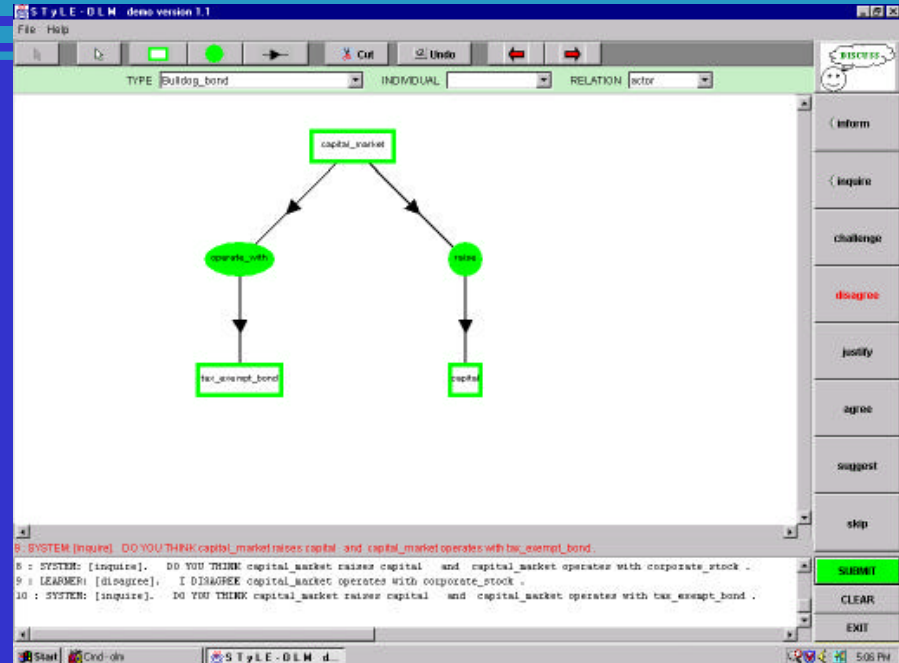
Service  
availability?

Learning  
process?

- Treat applications similar to Learning Objects:  
Methods encode
  - service query
  - pedagogical knowledge when to use tool
- Delayed Tasks!

# Integrating Intelligent Applications: Example

- Metacognitive Skills:
  - Reflection on learned knowledge
- Open Learner Modeling (V. Dimitrova, STyLE-OLM)
- But: Not available in every environment



```
(:method (doReflect ?c)
  ((toolAvailable? OpenLearnerModel))
  ((insertInvitation OpenLearnerModel))

  )
((insertText Reflection ?c))
)
```



# Novel Forms of Interaction

- Learner accesses pedagogical expertise:
  - triggers processing of a tutorial task,
  - results integrated on-demand
- No knowledge metadata required
- Active learner: **dynamic and extensible workbook**



Complete Content of LeAM\_calculus

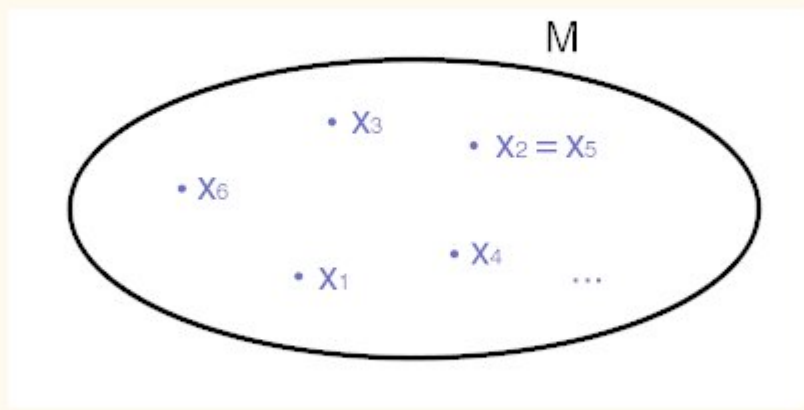
### Definition einer Folge

Sei  $M$  eine Menge. Eine **Folge in  $M$**  ist eine Abbildung  $\mathbf{N} \rightarrow M$  (oder  $\mathbf{N} \rightarrow M$ ). Man schreibt dafür meist:  $x_n[n]$  und meint damit die Folge  $x_1, x_2, x_3, \dots$

Show me an example

## You asked for: Example

### Ein Beispiel für eine Folge in einer Menge



Add to Course



Complete Content of  
LeAM\_calculus

Grundlagen

- Grundlegendes zu mathematischen Beweisen
- Grundlagen zu Geraden und ihrer Steigung
- Die binomischen Formeln
- Trigonometrische Formeln
- Beschränkte Mengen

Folgen, Reihen und Grenzwerte

- Grundlagen von Folgen
- Grenzwerte von Folgen
- Beschränkte Folgen
- Rechnungen mit Grenzwerten
- Teilfolgen und Umordnungen
- Cauchy-Folgen
- Reihen
- Absolut konvergente Reihen
- Konvergenzkriterien für Reihen
- Potenzreihen

### Definition einer Folge

Sei  $M$  eine Menge. Eine **Folge in  $M$**  ist eine Abbildung  $\mathbf{N} \rightarrow M$  (oder  $\mathbf{N} \rightarrow M$ ). Man schreibt dafür meist:  $x_n[n]$  und meint damit die Folge  $x_1, x_2, x_3, \dots$

Für die Analysis spielen vor allem Folgen reeller oder komplexer Zahlen eine Rolle, also Folgen in  $\mathbf{R}$  oder  $\mathbf{C}$ .



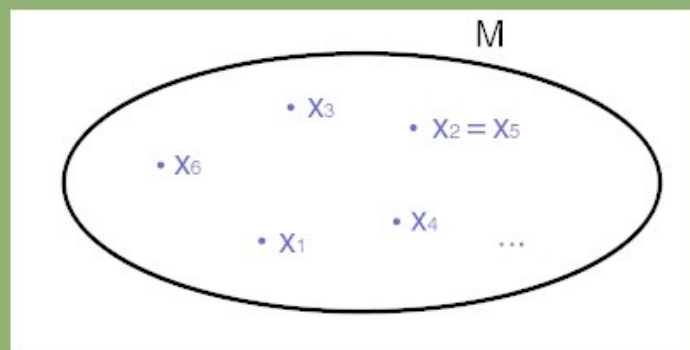
### Wann sind zwei Folgen gleich?

Zwei Abbildungen sind gleich, wenn sie in all ihren Bildern übereinstimmen. Folglich sind auch zwei Folgen  $x_n[n]$  und  $y_n[n]$  gleich, wenn  $x_n = y_n$  für alle  $n$  gilt. Das ist nicht immer sofort zu sehen. Beispielsweise sind folgende Folgen gleich:

$$x_n = \sin((2 \cdot n + 1) \cdot 90^\circ) \text{ und } y_n = (-1)^n.$$



### Ein Beispiel für eine Folge in einer Menge



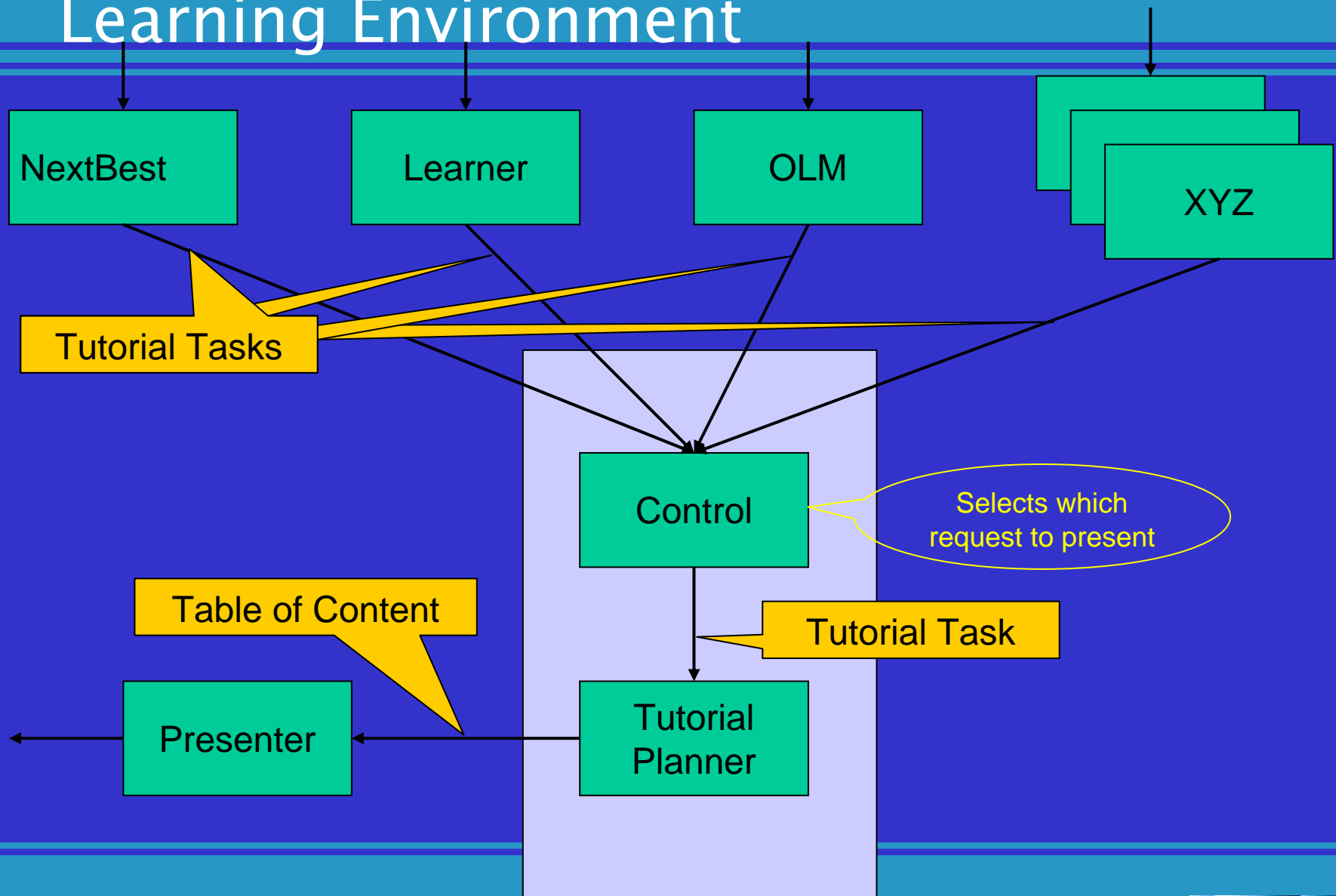
# Tutorial Planning as a Web Service

- Goal: Service Provision & Composition

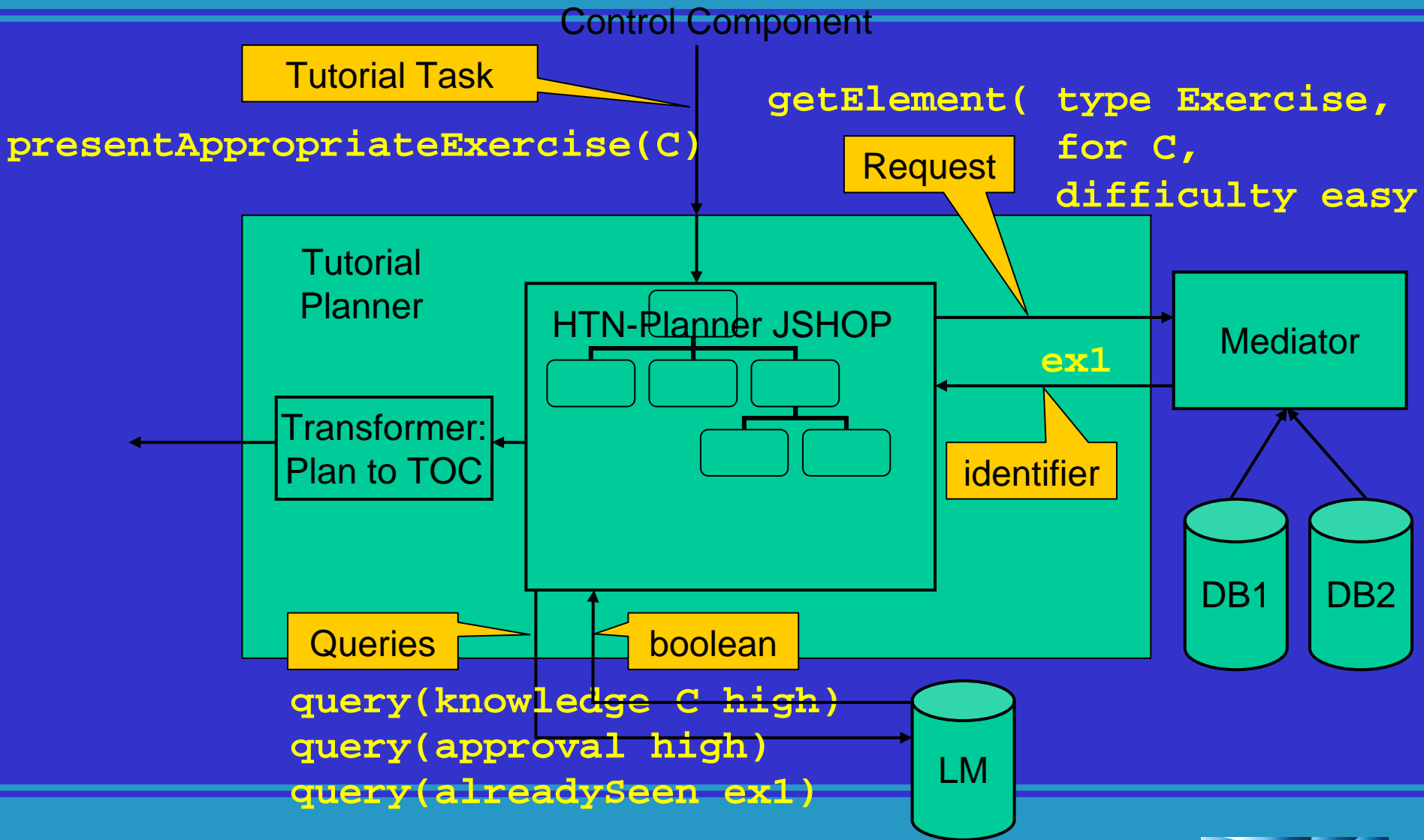


⇒ **Public tasks** describe functionalities of tutorial planner

# Tutorial Planning in a Learning Environment



# Sketch of the Architecture



# Summary

- Different theoretical approaches to learning
- Different ways of implementing course generation
- An example of a task-based approach that provides a framework that solves several of today's challenges



# Early Work

- W. R. Murray. Control for intelligent tutoring systems: A blackboard-based dynamic instructional planner. Proc. 4th International Conference of AI and Education, 1989
- B. J. Wasson. Determining the Focus of Instruction: Content planning for intelligent tutoring systems. PhD thesis, University of Saskatchewan, 1990.
- K. van Marcke. Instructional expertise. Proc. 2cd International Conference on Intelligent Tutoring Systems, 1992.
- J. Vassileva. Dynamic courseware generation: at the cross point of CAL, ITS and authoring. Proc. of ICCE, 1995.



# Related Work

- M. Specht and R. Oppermann. ACE – adaptive courseware environment. *The New Review of Hypermedia and Multimedia*, 1998.
- N. Henze and W. Nejdl. Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education*, 2001.
- P. Karampiperis and D. Sampson. Adaptive instructional planning using ontologies. *Proc. of the 4th IEEE International Conference on Advanced Learning Technologies*, 2004.
- K. Keenoy et. al. Personalisation services for self e-learning networks. *Proc. of 4th International Conference on Web Engineering*, 2004.



# Additional References

- V. Dimitrova. STyLE-OLM: Interactive open learner modelling. *International Journal of Artificial Intelligence in Education*, 13:35–78, 2002.
- R. Schulmeister. *Grundlagen hypermedialer Lernsysteme. Theorie – Didaktik – Design*. R. Oldenbourg Verlag: München (available in English language online: [http://www.izhd.uni-hamburg.de/paginae/Book/Frames/Start\\_FRAME.html](http://www.izhd.uni-hamburg.de/paginae/Book/Frames/Start_FRAME.html)).



# Wizard of Oz

- Goal: Design pedagogical task and methods that generate courses providing the learner the necessary knowledge to answer the questions
- First design the task and methods, write them down, then apply them in the WOZ experiment

