Introduction

- LIA is an Intelligent Tutoring Engine
- LIA integrates Learner Modelling and Learning Experience Individualisation in “traditional” e-learning systems
- LIA is based on:
  - a set of Models able to represent the main entities involved in the process of teaching/learning
  - a set of Algorithms, leveraging on such models, for the generation of individualised learning experiences
- LIA is currently included in a complete solution for e-learning named Intelligent Web Teacher (IWT)
LIA Models

- Formally represent main actors and objects involved in the learning process

Domain Model
- Describes the knowledge that is object of teaching in terms of concepts and relations between concepts

Learner Model
- Describes learning preferences and acquired knowledge of each learner

Learning Activity Model
- Describes an activity that must be performed by a learner to acquire one or more domain concepts

Unit of Learning Model
- Describes a sequence of learning activities needed to understand a set of concepts with respect to a set of given constraints
The Domain Model

- Describes the knowledge that is object of teaching in terms of concepts and relations between concepts
- It is concepts graph $G(C, R_1, \ldots, R_n)$ where:
  - $C$ is the set of nodes representing domain concepts
  - each $R_i$ is a set of arcs corresponding to the $i$-th kind of relation.
- Two categories of relations are supported:
  - hierarchical relations are used to factorise concepts
  - ordering relations are used to impose partial orderings

Example of Domain Model

- $G(C, BT, IRB, SO)$ where:
  - $BT(a, b)$ i.e. $a$ belongs to $b$
    - $b$ is understood iif every $a$ so that $a$ belongs to $b$ is understood
    - hierarchical relation
  - $IRB(a, b)$ i.e. $a$ is required by $b$
    - a necessary condition to study $b$ is to have understood $a$
    - ordering relation
  - $SO(a, b)$ i.e. the suggested order between $a$ and $b$ is that $a$ precedes $b$
    - to favour learning, it is desirable to study $a$ before $b$
    - ordering relation
Example of Domain Model

Teaching Preferences

- May be added to define feasible teaching strategies for concepts
- It is an application $TP (C \times Props \times PropVals) \rightarrow [0, 10]$ where:
  - $C$ is the set of nodes representing domain concepts
  - $Props$ is the set of didactical properties
  - $PropVals$ is the set of feasible values for such properties

<table>
<thead>
<tr>
<th>Properties</th>
<th>Feasible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>didactic method</td>
<td>deductive, inductive, etc.</td>
</tr>
<tr>
<td>activity type</td>
<td>text reading, video clip, simulation, discussion with a peer, discussion with the teacher, etc.</td>
</tr>
<tr>
<td>interactivity level</td>
<td>high, medium, low</td>
</tr>
</tbody>
</table>
Teaching Preferences

- Example:
  - TP ("formal systems", "didactic method", "deductive") = 10
  - TP ("formal systems", "didactic method", "inductive") = 4
  - TP ("outline of set theory", "didactic method", "deductive") = 8
  - TP ("outline of set theory", "didactic method", "inductive") = 8

<table>
<thead>
<tr>
<th>Properties</th>
<th>Feasible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>didactic method</td>
<td>deductive, inductive, etc.</td>
</tr>
<tr>
<td>activity type</td>
<td>text reading, video clip, simulation, discussion</td>
</tr>
<tr>
<td></td>
<td>with a peer, discussion with the teacher, etc.</td>
</tr>
<tr>
<td>interactivity level</td>
<td>high, medium, low</td>
</tr>
</tbody>
</table>

The Learner Model

- Describes learning preferences and acquired knowledge of each learner
- The Cognitive State represents the degree of knowledge reached by a learner at a given time
  - It is an application $CS(C) \rightarrow [0, 10]$ where:
    - $C$ is the set of concepts of a given domain model
  - If $CS(c) > 0$ then $c$ is considered as known
- Examples:
  - $CS("outline of set theory") = 8$
  - $CS("formal systems") = 4$
The Learner Model

- The Learning Preferences provide an evaluation of learning strategies that may be adopted for a given learner.

- It is an application
  \[ \text{LP} (\text{Props} \times \text{PropVals}) \rightarrow [0, 10] \]
  where:
  - Props is the set of didactical properties
  - PropVals is the set of feasible values for such properties

- Example:
  - \( \text{LP} ("\text{didactic method}", "\text{deductive}") = 5 \)
  - \( \text{LP} ("\text{didactic method}", "\text{inductive}") = 8 \)

The Learning Activity Model

- Describes an activity that must be performed by a learner to acquire one or more domain concepts.

- It is composed by:
  - a Set of Concepts \( C_A \) that are covered by the learning activity
  - a set of Didactical Properties \( \text{DP}_A \) \( (\text{property}) = \text{value} \) representing learning strategies applied by the learning activity
    - \( \text{DP}_A \) domain is the set of didactical properties Props
    - \( \text{DP}_A \) range is the set of properties feasible values PropVals
  - a set of Cost Properties \( \text{CP}_A \) \( (\text{property}) = \text{value} \) that must be taken into account in the optimisation process
    - \( \text{CP}_A \) domain is the closed vocabulary \{price, duration\}
    - \( \text{CP}_A \) range is \( \mathbb{R}^+ \)
The Learning Activity Model

- Example:
  - $C_A = \{ \text{“formal systems”} \}$
  - $DP_A \ (“\text{didactic method}”) = \text{“inductive”}$
  - $DP_A \ (“\text{activity type}”) = \text{“simulation”}$
  - $DP_A \ (“\text{interactivity level}”) = \text{“high”}$
  - $CP_A \ (“\text{price}”) = 1.5$
  - $CP_A \ (“\text{duration}”) = 5$

Testing Activity Model

- It represents a Testing Activity i.e. a Learning Activity used to verify the knowledge acquired by a learner
- It is composed by:
  - a Set of Concepts $CT_A$ that are verified by in the testing activity
- Once executed a testing activity returns an Evaluation $ET \in [0, 10]$ indicating the degree of fulfilment of the test by the learner
The Unit of Learning

- Describes a sequence of learning activities needed to understand a set of target concepts with respect to a set of given constraints

- It is composed by:
  - a set of Target Concepts TC that have to be mastered in order to successfully accomplish the unit of learning
  - a set of Cost Constraints CC (property) = value that must be taken into account in the generation of the unit of learning
  - a Learning Path LPath (c₁, ..., cₙ) i.e. an ordered sequence of concepts that must be taught to a learner in order to let him master target concepts
  - a Learning Presentation LPres (a₁, ..., aₘ) i.e. a sequence of learning activities that a learner must perform in order to learn the target concepts

Example:
- TC = \{“logics”\}
- CC (“price”) = 100
- CC (“duration”) = 360

Generates an Unit of Learning that:
- explains the concept of “logics”
- has a maximum total price of 100 €
- has a maximum duration of 6 hours (= 360 minutes)
The Learning Life-Cycle

Learning Path Generation

- This algorithm generates a Learning Path starting from:
  - a Domain Model $G (C, BT, IRB, SO)$
  - a set of Target Concepts $TC \subseteq C$

- Example
  - $TC = \{ \text{“first order logic”} \}$
First Step

- Builds the graph $G'(C, BT, IRB', SO')$ by propagating ordering relations downward the hierarchical relation

Second Step

- Builds the graph $G''(C', R)$ where $C'$ is the subset of $C$ including all concept that must be taught according to $TC$
Third Step

- Finds a linear ordering of nodes of $G$
- The obtained list $L$ is a first approximation of the learning path

A Further Example

- $TC = \{\text{"first order logic"}\}$
Milestone Setting

- Milestones are placeholders for testing activities:
  - they can be placed directly by teachers
  - they can be placed basing on a list of percentages given by the teacher
  - they can be placed dynamically after each sub-list of concepts belonging to the same higher-level

- Each milestone covers all preceding concepts in the learning path apart concepts already known

- A milestone at 0% of the learning path indicates a Pre-Test:
  - MR indicates a pre-test on requirements
  - MC indicates a pre-test on content
  - MI indicates an integrated pre-test

Presentation Generation

- This algorithm generates a fragment of Learning Presentation, part of an Unit of Learning, suitable for a specific learner basing on:
  - a Learning Path LPath that have to be covered
  - a set of Teaching Preferences TP belonging to a domain model
  - a Cognitive State CS belonging to the learner
  - a set of Learning Preferences LP belonging to the learner
  - a set of optional Cost Constraints CC
  - a set of available Learning Activities (including tests)
Presentation Generation

- **Step 1** - select the sub-list $L$ of $L_{Path'}$ that have to be converted in a presentation
  - $L$ includes all unknown concepts of $L_{Path'}$ from the beginning to the first milestone preceded by at least one unknown concept
  - If $L$ is empty then stop

- **Step 2** - define the best sequence of learning activities $P$ (not including tests) covering $L$ on the basis of $T_P$, $L_P$ and $C_C$

- **Step 3** - add testing activities at the end of $P$ so obtaining the final learning presentation $Pres$
  - Testing activities are selected in order to cover all concepts of $L$

Details on Step 2

- How to find the optimal set of learning activities $P$ covering $L$ on the basis of $T_P$, $L_P$ and $C_C$.

- **Definition of distance measures:**

  \[
  d_{T_P}(A, c) = \sum_{\text{property}} |(10 - \text{level})| D_P(\text{property}, \text{value}) \cap T_P(c, \text{property}, \text{value}, \text{level}) \text{ if } c \in C_A, \infty \text{ otherwise};
  \]

  \[
  d_{L_P}(A) = \sum_{\text{property}} |(10 - \text{level})| D_P(\text{property}, \text{value}) \cap L_P(\text{property}, \text{value}, \text{level});
  \]

  \[
  d(A, c) = \beta_{T_P} d_{T_P}(A, c) + \beta_{L_P} d_{L_P}(A).
  \]

- Then the problem becomes a **Facility Location Problem**
Details on Step 2

- **How to solve the Facility Location Problem**

\[
\min \sum_{d \in D} \sum_{c \in C} d(x, c) \cdot x_{dc} \quad \text{so that:}
\]
\[
\sum_{c \in C} x_{dc} = 1 \quad \forall c;
\]
\[
x_{dc} \leq \sum_{c \in C} x_{dc} \quad \forall A, c;
\]
\[
x_{dc} \in \{0, 1\} \quad \forall A, c \text{ and } y_{dc} = \{0, 1\} \quad \forall A.
\]
\[
\sum_{d \in D} \sum_{c \in C} \text{value}_{d,c} \cdot y_{dc} \leq \text{value} \quad \forall C_{d} \text{ (property, value)}
\]

- **The solution can be found with a Greedy Algorithm**

Learner Model Updating

- Each performed testing activity returns an evaluation \( ET \in [0, 10] \)
- For each covered concepts \( c \) the **Cognitive State** of the learner is modified in this way:
  - if \( \text{CS (c)} \) is not defined then \( \text{CS (c)} = ET \)
  - else \( \text{CS (c)} = (\text{CS (c)} + ET) / 2 \)
  - the evaluation may be propagated on required concepts in the concepts graph
- **Learning Preferences** are modified in the following way:
  - each time a learning activity \( A \) is successful (unsuccessful) for a learner
  - the system increases (decreases) learner preferences \( \text{LP (property, value)} \) of a given constant \( \delta \)
  - for each didactical property \( \text{DPA (property)} = \text{value} \) belonging to \( A \)
Experimentation

- An experimentation of LIA was made with 28 volunteers:
  - the 1st Group used a Learning System without LIA
  - the 2nd Group used a Learning System with LIA

Integration in IWT

- IWT (Intelligent Web Teacher) is a complete e-Learning System providing a comprehensive set of traditional and intelligent features
IWT Snapshots

A UNIT OF LEARNING

www.diima.unisa.it - ncapuano@unisa.it
Conclusions

- IWT is currently used by about 30 Italian Organisations (more than 40,000 Users) including:
  - big companies like Atos Origin Italy, Italdia, and Metoda
  - departments and faculties from the Universities of Roma “La Sapienza”, Milano “Bicocca”, Salerno, Bari, Calabria and Napoli “Parthenope”
- IWT was selected by the Italian Ministry of Education for the project DigiScuola purposed to experimentally introduce e-learning in 550 Italian Schools involving 3000 Teachers and 33000 Learners
- IWT won the prize Best Practices for Innovation from the General Confederation for Italian Industry
- The research on LIA is active and further improvements are under study