Representing CSCL macro-scripts using IMS LD lessons learned

Representing Computer-Supported Collaborative Learning macro-scripts using IMS Learning Design

Davinia Hernández-Leo1, **Daniel Burgos**2, **Colin Tattersall**2, **Rob Koper**2

1 School of Telecommunications Engineering, **University of Valladolid**, Spain
dherleo@ulises.tel.uva.es

2 Educational Technology Expertise Centre, **Open University of the Netherlands,** The Netherlands
{daniel.burgos, rob.koper}@ou.nl

Universidad de Valladolid

OpenUniversiteitNederland

Motivation

Technology-Enhanced Learning:
Computer-Supported Collaborative Learning

Free collaboration does not necessarily produce fruitful interactions

CSCL Scripting
Micro-scripts vs. macro-scripts

(Structure collaborative learning scenarios by defining the composition of groups, the distribution of roles and resources, and the coordination of activities that make up the learning process)

"Scripts" are "hardwired" in specific environments

Formalizing the scripts so that they are computer-interpretable by learning environments

- Educational Modelling Languages (EML)
  - Specifying teaching-learning processes that stress the importance of the activities (Rawlings et al., 2002)
- ✓ IMS Learning Design (LD) (IMS, 2003)
  - Open specification agreed upon by the IMS consortium: interoperability prospects
  - Lack of systematic analysis of its support to express CL situations (demanding requirements: e.g. group hierarchies)

Objective

Analyze the suitability of IMS LD for computationally representing CSCL macro-scripts

Methodology

- Identification of significant requirements of CSCL macro-scripts (NISE, 1997; Johnson et al., 1999; CoSSICLE, 2005; Dillenbourg et al., 2007)
  - Group composition: hierarchy of groups, group size, amount of groups, group formation policies, dynamic group formation
  - Role and resource distribution
  - Coordination: flow of (collaborative) learning activities, floor control, flow of artefacts
  - Flexibility: flexible group composition
- "Coding" cases that significantly feature the requirements
  - Well-known scripts: **Universanté, ArgueGraph** (Berger et al., 2001; Jermann et al., 2002; Dillenbourg, 2002)
  - Collaborative Learning Flow Patterns (**CLFPs**): **JIGSAW, PYRAMID, SIMULATION, BRAINSTORMING, TPS, TAPPS** (Aronson et al, 1992; Gibbs, 1995; Paulsen, 1995; NISE, 1997; Davis, 2002)

Requirement	IMS LD notation	Related specifications and tooling
Group composition	LD role component and related elements and attributes, plus the joint use of properties and conditions Good support to: **hierarchy of groups, group size** and dynamic group formation	Enhance the realization of the requirements (including **amount of groups** and **group formation policies)** with administrative tools (grouping services, player utilities) in combination with group composition specifications
Role/ resource distribution	**Role distribution**: LD role component and role-parts	Supporting tools ("group-services") may define specific roles with different privileges
	Rotation of roles: rotating activities	or using mechanisms provided by players
	Resource distribution: role-parts, environments, activity descriptions and properties	Supporting tools can be used for resource distribution
Coordination	**Flow of activities**: LD method and conditions	Data flow between tools (new spec, workflow languages?)
	Flow of artefacts between activities: properties, global-elements and monitoring services **Shared artefacts**: properties	Sophisticated floor control mechanisms achieved by means of supporting tools (new spec for group-services?)
Flexibility	**Flexible group composition**: LD role attributes min-persons, max-persons, create new Distinction between abstract descriptions (UoLs) and instantiation	Handling specifications of script constraints that are allowed to be modified (by teachers and learners) Considering specific interface features related to particular scripts

The flow of activities is expressed in the *method*. A method contains one or more *plays*, which are modelled according to a theatrical play with *acts* and *role-parts*. The plays run in parallel. Acts together with *conditions* (and also *notifications*) determine whether, when, and for what *roles activities* and *resources* need to be available. Different types of conditional expressions can be used to orchestrate the flow.

Illustrative excerpts (or explanations), supposing that there will be 2 countries and 4 participants per country, i.e. 2 thematic groups comprising 2 case groups

This script requires a method with five acts. Each act contains a role-part per role of the "type" of role that corresponds to each phase. In the cases that the activities are performed by persons belonging at the same time to two groups (e.g. "within each thematic group, the members of each country group create a fact sheet"), it is necessary to add conditions with two expressions of type *is-member-of-role*.

```
<if>
  <and>
   <is-member-of-role ref="R-thematic-group-cancer" />
   <is-member-of-role ref="R-country-group-switzerland" /> </and> </if>
<then>
   <show> <class  class="C-fact-sheet-cancer-switzerland" /> </show>
   <hide>  <class  class="C-fact-sheet-cancer-cameroon" />
          <class  class="C-fact-sheet-aids-switzerland" />
          <class  class="C-fact-sheet-aids-cameroon" /> </hide> </then>
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
<div class="C-fact-sheet-cancer-switzerland">
   <p>Please create fact sheet concerning the "Cancer"
   status in your country, Switzerland: </p>
   <ld:set-property ref="L-fact-sheet-cancer-switzerland"/>
</div>
```

Figure 1. Screenshot of a run corresponding to the UoL that represents the **Universanté Script.**
In this activity the members of the same thematic group (cancer) and the same country group (Switzerland) create a shared fact sheet concerning the thematic status in their country

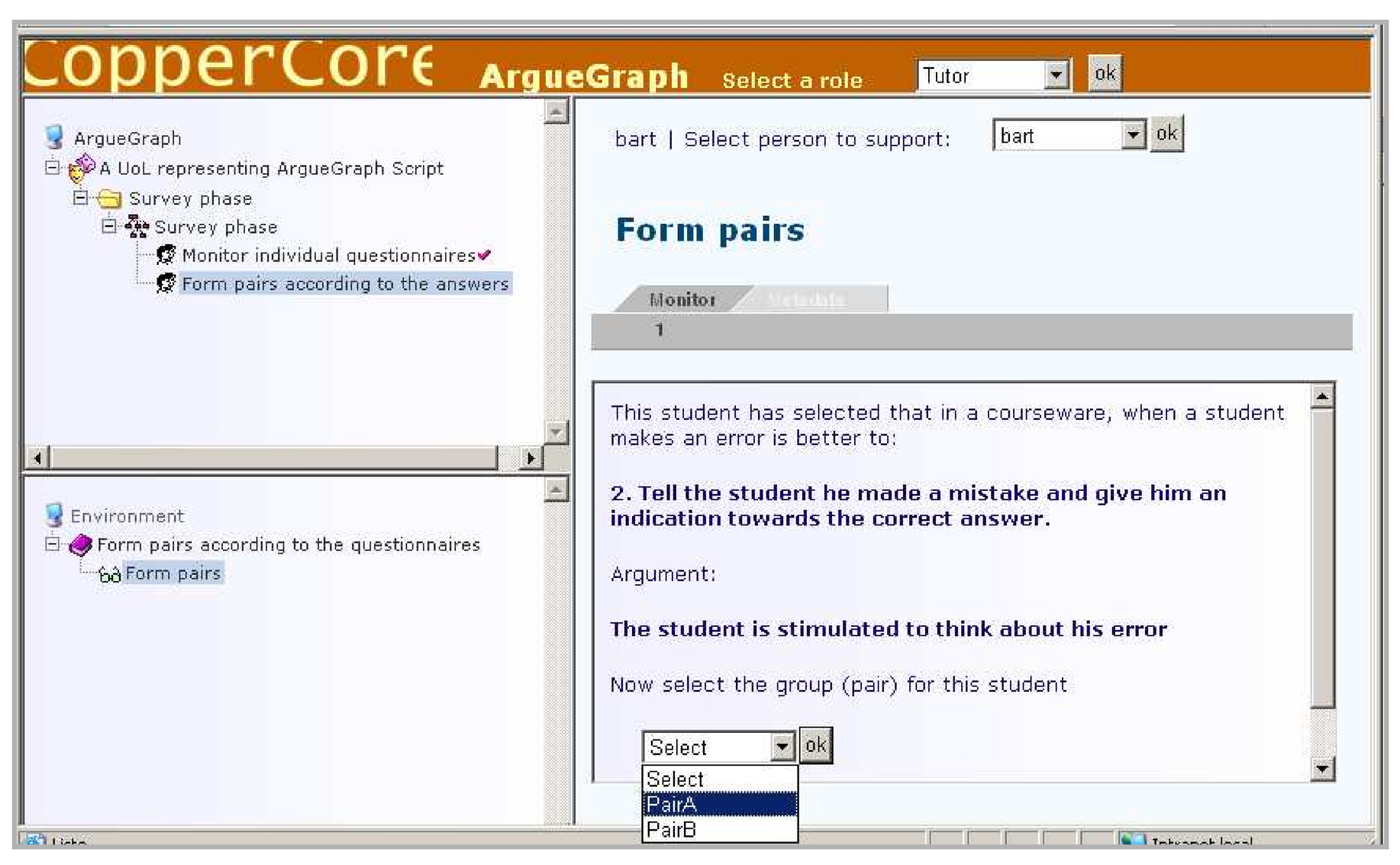
Local personal properties can be used to model groups so that their value can be determined at runtime using *global-elements*. The persons with the same value of the property are in the same group. *Conditions* are in charge of coordinating the activities and artefacts according to this value of each participant's property (i.e. according to their group).

Illustrative excerpt, supposing that in ArgueGraph there will be only 4 participants, i.e. 2 pairs

The tutor dynamically determines at runtime to which group each student is bound. Depending on the group, access to an activity containing a different instance of the shared questionnaire is provided.

```
<locpers-property identifier="LP-pair"> [...]
  <datatype datatype="string"/>
  <restriction restriction-type="enumeration">PairA</restriction>
  <restriction restriction-type="enumeration">PairB</restriction>
</locpers-property>
[...]
<if>  <!-- If the student is in Pair A -->
  <is> <property-ref ref="LP-pair"/>
       <property-value>PairA</property-value>
  </is>
</if>
<then>
  <show>
  <learning-activity-ref ref="LA-fill-in-pairs-questionnaire-PairA"/>
</show>
  <hide>
  <learning-activity-ref ref="LA-fill-in-pairs-questionnaire-PairB"/>
</hide>
</then>
```

Figure 2: Screenshot of a run corresponding to the UoL that represents the **ArgueGraph Script**.
In this activity the tutor forms pairs of students according to their answers so that each student will argue with somebody having opposite opinions



Main conclusions

- Analysis considering the capacity of the LD notation but also the support of related specifications and tooling
  - The manifestation of most requirements in the cases can be expressed with LD
    - Involved LD elements and attributes and illustrative XML excerpts
  - Discussion of how the limitations could be supported with eventually developed specifications and tooling
    - Group-services, grouping characteristics (e.g. group formation policies) for administration tools that instantiate the scripts

European Conference on Technology Enhanced Learning
Crete, Greece

Sept 18th, 2007