

# COML (Classroom Orchestration Modelling Language) and Scenarios Designer: Toolsets to Facilitate Collaborative Learning in a One-to-One Technology Classroom

Jitti Niramitranon, Mike Sharples, Chris Greenhalgh

► **To cite this version:**

Jitti Niramitranon, Mike Sharples, Chris Greenhalgh. COML (Classroom Orchestration Modelling Language) and Scenarios Designer: Toolsets to Facilitate Collaborative Learning in a One-to-One Technology Classroom. Ulrich Hoppe, Judith Schoonenboom, Mike Sharples, Barbara Wasson. Kaleidoscope Convergence Workshop, 2006, Amsterdam, Netherlands. Les cahiers du laboratoire Leibniz, pp.17-24, 2006. <hal-00190507>

**HAL Id: hal-00190507**

**<https://telearn.archives-ouvertes.fr/hal-00190507>**

Submitted on 23 Nov 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# COML (Classroom Orchestration Modelling Language) and Scenarios Designer: Toolsets to Facilitate Collaborative Learning in a One-to-One Technology Classroom

Jitti Niramitranon, Mike Sharples, Chris Greenhalgh  
Learning Science Research Institute (LSRI)  
University of Nottingham  
{ttxn3, Mike.Sharples}@nottingham.ac.uk, c.greenhalgh@cs.nott.ac.uk

## ABSTRACT

In a one-to-one collaborative learning classroom supported by ubiquitous computing, teachers require tools that allow them to design of learning scenarios, and to manage and monitor the activities happening in the classroom. Our project proposes an architecture for a classroom management system and a scenarios designer tool, both based on a Classroom Orchestration Modelling Language (COML), to support these requirements. We are developing and testing this with the GroupScribbles software using a set of learning scenarios for classrooms equipped with pen tablet computers.

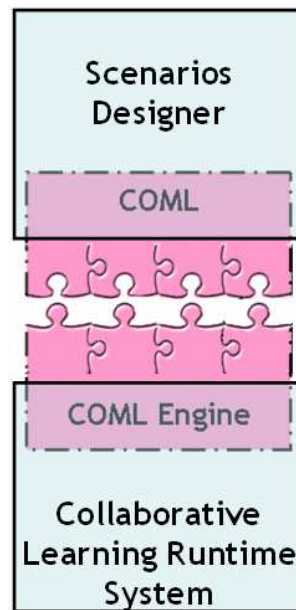
## Keywords

Collaborative and group learning; Learning systems platforms and architectures; learning design, wireless and mobile technologies

## 1. INTRODUCTION

The emergence of one-to-one technology (one computer device or more per student) has the potential to enhance learning in the classroom [1]. A literature survey reveals two main areas of interest. The first is based on a pedagogical approach, with a focus on the study of possible patterns in face-to-face classroom activity. The aim is to capture and describe common learning situations, particularly for collaborative learning [2] [3] [4]. The other area is focused on the technology and is concerned with creating innovative software and hardware tools that will be able to enhance specific learning scenarios [5] [6] [7]. A key problem with this research is how to balance the pedagogical and computational approaches, so that technologies for classroom interaction can support effective collaboration in real classrooms [8]. On one hand, teachers should be easily able to design and manage collaborative learning supported by ubiquitous computing (such as a combination of tablet PCs, desktop PCs and electronic whiteboards). On the other hand, the computational system, including hardware and software technologies should be able to support a range of effective learning scenarios. This paper proposes to design a Classroom Orchestration Modelling Language (COML) and a scenario designer tool, with the aim of providing a single platform that brings together the pedagogical and technological aims we have highlighted.

## 2. PROPOSED DESIGN AND ARCHITECTURE



**Figure 1: Design Architecture with 3 additional proposed components, Scenarios Designer, COML and COML engine**

As seen in **Figure 1**, the architecture of the system has been divided into two major parts, the Scenarios Designer and collaborative runtime system. These are integrated by the use of a common COML definition of the learning scenario, as supported by the COML constructor on the scenarios designer side and a COML execution engine on the runtime system side.

The Scenarios Designer has been developed through the study of pre-existing scenarios for effective interactive learning in the classroom. These scenarios were analysed to capture common components such as actors, learning materials, technical and non-technical related activities, which were then defined as elements of the formal notation. A sequence of prototyping has produced a software system that supports notations for interactive learning design, including actors and learning artefacts. These components can then be combined into activity diagrams that cover the original scenarios (with iterative refinement as required). The real value of these learning diagrams is that they can be exported as a COML document to be executed by the computer supported collaborative learning *run-time* system. COML is based on generic XML description with similar aims to CML [9] (although CML is currently only a conceptual design rather than an actual technology). The COML engine manages the interaction of a teacher and learners with ubiquitous computing in a classroom guided by the COML scenario definition.

### **3. LEARNING SCENARIO AND SCENARIOS DESIGNER TOOL**

The screenshot shows the Scenarios Designer interface. At the top, there is a menu bar with 'New', 'Open', and 'Save' buttons, and a 'Designer Tools Bar'. Below this is a 'Scenarios Designer' panel with a 'Components Properties' box on the left. The main workspace is a grid with five columns: 'Deliverer', 'What to do?', 'Receiver', 'Electronic Resources', and 'Presentation Spaces'. The grid contains seven rows of components, numbered 1 to 7. A red arrow points from the 'Deliverer' column to the 'Components Properties' box, and another red arrow points from the 'Electronic Resources' column to the 'COML Document' window. The 'COML Document' window shows XML code for a scenario named 'Chemical Formula Game'.

Deliverer	What to do?	Receiver	Electronic Resources	Presentation Spaces
Teacher	Q Give the abbreviation for each element	Class	Chemical_Element.JPG	Public 1
Class	A Answer Step 1	Teacher	T Text [In Class]	Public 1
Teacher	Q Draw H2O	Group 1	T H2O	Group 1
Teacher	Q Draw CO2	Group 2	T CO2	Group 2
Teacher	Q Draw N2O	Group 3	T N2O	Group 3
Group 1, 2, 3	A Answer Step 3, 4, 5	Teacher	Drawing [In Class]	Public (Group 1, 2, 3)
Teacher	D Conclusion	Class	Any [In Class]	Public (Conclusion)

```

<coml ver="1.0">
  <scenario name="Chemical Formula Game">
    <description>
      1. Students answer the abbreviation of elements
      2. Students draw molecular
    </description>
    <step No="1">
      <Name>question</Name>
      <description>This step is to ask for a whole class</description>
      <deliver Type="teacher"/>
      <activity Type="question">
        <Topic>Give the abbreviation for each element</Topic>
        <Memo>Start with the first three row on the table</Memo>
      </activity>
    </step>
  </scenario>
</coml>

```

Figure 2 : Screen shot of the Scenarios Designer and COML document

The current version of the Scenarios Designer has four main parts: a designer toolbar, a components properties box, a working space and a COML document window which is resizable. On the designer toolbar, common components captured from the scenario analysis process are listed, which have been divided into three sets: actors, activities and learning artefacts. Actors have four different roles: teacher, students, groups and presenters (the computational actors that present the learning artefacts on spaces such as a display screen). The activities have four components: question, answer, discussion, other (user defined). The learning artefacts comprise four types of electronic resource: text, file, drawing and any, where all of these are data to be presented in particular presentation spaces such as a computer screen.

To create a learning scenarios, a teacher drags components from the designer tool bar and drops them onto the appropriate blocks in five 'swim lanes': deliverer, what to do?, receiver, electronic resources, and presentation space. Once any component is placed in the working area, the COML interpreter engine, which is running in the background of the Scenarios Designer, will convert the diagram into the COML document corresponding to the designed diagram.

As an example, we show three main learning scenarios running in order: brainstorming, followed by group work, followed by conclusion. The brainstorming scenario begins with a teacher showing the picture of an abbreviated name of some chemical elements to all students and then asking students as a whole class for the full name of each element. In the next scenario, the students are formed into three groups and the teacher gives each group a chemical formula. The teacher then asks them to draw the correct molecular relation corresponding to the given question. The teacher concludes by discussing the results of the group activity with the whole class.

Therefore, the meaning of the scenario design pictured in Erreur ! Source du renvoi introuvable. from the first to the seventh row can be interpreted as the following

- 1<sup>st</sup> row: a teacher shows a “prepared file (Chemical\_Element.JPG)” which is a chemical element table to “all students in the class” on the public presentation space (such as an electronic whiteboard) named “Public 1” and asks students verbally to “please answer with the correct abbreviation for each element”;
- 2<sup>nd</sup> row: all students answer the teacher by typing “text” on their personal computing device and the all answers are then shown on the presentation space “Public 1”;
- 3<sup>rd</sup> row: the “teacher” splits the students into three groups and verbally asks the “group 1” to draw a molecular form of H<sub>2</sub>O. The prepared question “H<sub>2</sub>O” which was created by using “text tool” is also displayed on the presentation space of “Group 1” (an environment for “group 1” to work collaboratively);
- 4<sup>th</sup> the “teacher” verbally asks the “group 2” to draw a molecular form of CO<sub>2</sub>. The prepared question “CO<sub>2</sub>” which was created by using “text tool” is also displayed on the presentation space of “Group 2” (an environment for “group 2” to work collaboratively);
- 5<sup>th</sup> the “teacher” verbally asks the “group 3” to draw a molecular form of N<sub>2</sub>O. The prepared question “N<sub>2</sub>O” which was created by using “text tool” is also displayed on the presentation space of “Group 3” (an environment for “group 3” to work collaboratively);
- 6<sup>th</sup> group 1, 2 and 3 reply to the teacher with “drawing tool in the class” and all answers are displayed on the public space named “Group 1, 2, 3”;
- 7<sup>th</sup> the teacher has a conclusion session to “all students” and also allows students use “any tools (text, file or drawing)” on the public space named “Conclusion”.

The Scenarios Designer is able to identify the actor(s) who will be in charge of creating or completing each certain task and to specify the flow direction of the tasks from one to another actor (e.g. from a teacher to all students or from group A to groups B). These design techniques have made the Scenario Designer different from LAMS [10] whose design is based on the concept of the tasks stepping forward through available tools provided by LAMS itself, for example, from chatroom to vote and then from vote to question(s) and answer(s).

#### **4. COML, COML ENGINE AND RUNTIME SYSTEM**

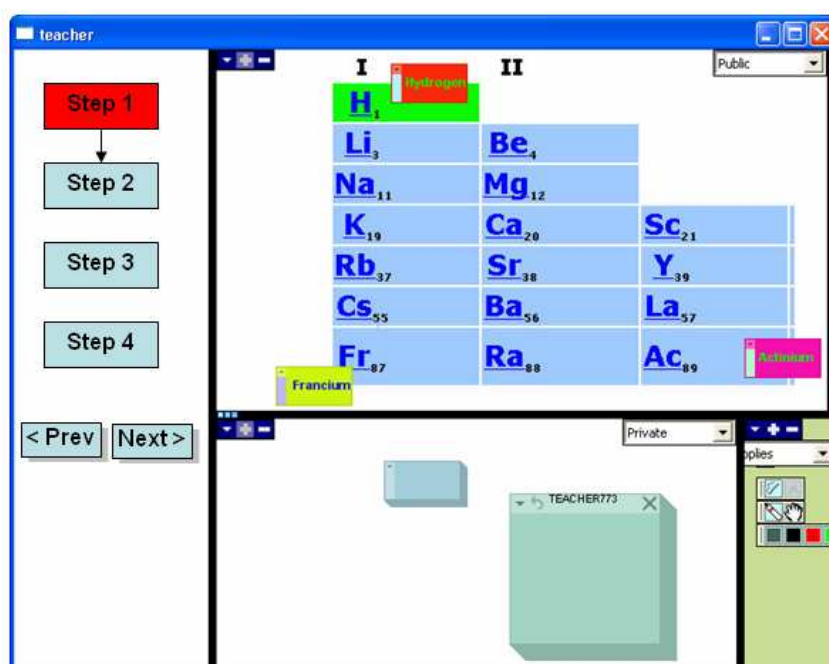
Although COML has a similar basic learning design model to EML [2], that is “*Role performs Activities within an Environment*”, the main idea underlying COML is derived from trends in the field of collaborative learning supported by one-to-one technology and distributed system runtime environment. COML therefore has begun with its own language structure and specification.

As our test bed for this research we are using an existing distributed system for collaborative learning called GroupScribbles (GS) [11] as the runtime system. In order to make GS understand and work in coordination with COML, a COML engine is being developed which

works as an interpreter and interface between COML and GS. The COML engine is able to call all GS functions to prepare the designed lesson ready for the class.

```
<coml ver="1.0">
  <scenario name="Chemical Formula Game">
    <Description>
      1. Students answer the abbreviation of elements
      2. Students draw molecular
    </Description>
    <Step No="1">
      <Name>Question</Name>
      <Description>This step is to ask for a whole class</Description>
      <Deliver Type="teacher"/>
      <Activity Type="question">
        <Topic>Give the abbreviation for each element</Topic>
        <Memo>Start with the first three row on the table</Memo>
      </Activity>
      <Receiver Type="student"/>
      <Resource Type="file" Name="chemical_element.JPG"/>
      <Presentation Type="Public" name="Public 1"/>
    </Step>
  </scenario>
</coml>
```

**Figure 3: Sample of COML document for the first step of Brainstorming scenario**



**Figure 4: A design of GS when it works together with COML engine**

In the example scenario, teachers display chemical elements and verbally ask students “What are the full names of these chemical elements?” This step uses a COML document which is generated by Scenarios Designer as shown in

**Figure 3.** Once the COML document is parsed and interpreted by the COML engine it loads the prepared picture onto the GS public space as shown in **Figure 4.** We can regard the COML engine as an actor that prepares the technical learning environment, and monitors, controls and gives relevant information to the teacher so that they know when to transition to subsequent stages of the learning scenario.

## 5. CONCLUSIONS

We are currently developing and refining the Scenarios Designer with Macromedia Flash, the COML definition based on XML schema, the COML engine and some additional functions in GS based on Java technology with the aim of balancing pedagogical and computational orientations. Our goals are to:

- allow teachers design or prepare the lessons beforehand
- enable monitoring, tracking of learning activities/sequences and transition changing
- potentially make the designed scenarios exchangeable in a collaborative learning object format, so that teachers can share lesson elements.

The system will be evaluated in the lab and through teacher walk-throughs and interviews to explore whether these tools are supporting the research aims.

## 6. ACKNOWLEDGEMENTS

The authors wish to thank SRI international and GS design team: John Brecht, Jeremy Roschelle, and Patti Schank for the many fruitful discussions and ideas they have shared including the offer of the GroupScribbles software to be used as a runtime system in this research.

## REFERENCES

- [1] Chan, T.W., Roschelle, J., Hsi, S., Kinshuk, Sharples, M., and Brown, T. (2006). One-to-one technology enhanced learning: and opportunity for global research collaboration. *Research and Practice in Technology Enhanced Learning* 1(1): 3-29.
- [2] Koper, R. and Manderveld, J. (2004). Educational modelling language: modelling reusable, interoperable, rich and personalised units of learning. *British Journal of Educational Technology* 35(5): 537-551.
- [3] Chris DiGiano, L.Y., Charlie Patton, Jeremy Roschelle, Deborah Tatar, Matt Manley (2003). Conceptual Tools for Planning for The Wireless Classroom. *Journal of Computer Assisted Learning* 19(3): 284.
- [4] Roschelle, J., Schank, P., Brecht, J., Tatar, D., and Chaudhury, S.R. (2005). From Response Systems to Distributed Systems for Enhanced Collaborative Learning. In *Proceeding of the 13th International Conference on Computers in Education (ICCE 2005)*: 363-370.
- [5] Liu, T.C., Wang, H.Y., Liang, J.K., Chan, T.W., Ko, H.W., and Yang, J.C. (2003). Wireless and mobile technologies to enhance teaching and learning. *Journal of Computer Assisted Learning* 19(3): 371-382.
- [6] Pinkwart, N., Hoppe, U., Milrad, M., and Perez, J. (2003). Architectures and educational scenarios for cooperative use of PDAs in wireless networks. *Journal of Computer Assisted Learning* 19(3): 383-391.
- [7] Liu, T.-C., Liang, J.-K., Wang, H.-Y., and Chan, T.-W. (2003). The Features and Potential of Interactive Response System. In *Proceeding of the International Conference on Computers in Education (ICCE)*: 315-322.
- [8] Sharples, M. (2000). The design of personal mobile technologies for lifelong learning. *Computer & Education* 34(3-4): 177-193.
- [9] Brecht, J., Chung, M., and Pea, R. (2002). CML - The ClassSync Modeling Language, *Presented at the Computer Support for Collaborative Learning (CSCL) conference, Boulder, CO*. Retrieved 28th November, 2005, from [www.stanford.edu/~roypea/RoyPDF%20folder/A108\\_Brecht\\_etal\\_02.pdf](http://www.stanford.edu/~roypea/RoyPDF%20folder/A108_Brecht_etal_02.pdf)
- [10] Dalziel, J. (2003). Implementing Learning Design: The Learning Activity Management System (LAMS). In *Proceeding of the 20th Annual Conference of the*

*Australasian Society for Computers in Learning in Tertiary Education (ASCILITE), Adelaide, 7-10 December 2003. Retrieved 26th January, 2006, from <http://www.ascilite.org.au/conferences/adelaide03/docs/pdf/593.pdf>*

- [11] Patton, C. and Schank, P. (2006). Group Scribbles Innovation Flyer. *To be submitted to Popular Science Best of What's New 2006.*