Demo of Orchestration Graph Engine: Enabling Rich Social Pedagogical Scenarios in MOOCs

Louis Faucon
Ecole Polytechnique Fédérale
de Lausanne
louis.faucon@epfl.ch

Stian Håklev
Ecole Polytechnique Fédérale
de Lausanne
stian.haklev@epfl.ch

Thanasis Hadzilacos
Open University of Cyprus and
Ecole Polytechnique Fédérale
de Lausanne
thanasis.hadzilacos@ouc.ac.cy

Pierre Dillenbourg
Ecole Polytechnique Fédérale
de Lausanne
pierre.dillenbourg@epfl.ch

ABSTRACT

This demo submission is associated with the Work-in-Progress submission "Orchestration Graphs: enabling rich social pedagogical scenarios in MOOCs".

We present our implementation of a web application for designing, running and orchestrating social pedagogical scenarios. The application is based on Orchestration Graphs, an educational modeling language. We plan to demonstrate our technology by automatically simulating user activity, and offering visitors the opportunity to interact with the graph editor.

Author Keywords

MOOCs; Scripting; Orchestration.

INTRODUCTION

The gathering of students with very different backgrounds and origins in MOOCs has the potential to provide a rich setting for collaborative learning and exchange. However, current MOOC platforms provide only rudimentary social features, which offer little space for exploration of rich collaborative scripts. Often, the only social interaction in MOOCs is the discussion forums and peer-review of assignments.

We show our implementation of a web learning platform allowing the creation, editing, and running of complex social scenarios as Orchestration Graphs (described in our Work in Progress paper). We believe this technology offers new opportunities for richer collaboration and orchestration in MOOCs, and can support a broad research agenda for learning at scale.

PLATFORM

Graph Editor

The first interface of our application is the graph editor (shown in Figure 1). It allows teachers or instructional designers to design their own pedagogical scenarios.
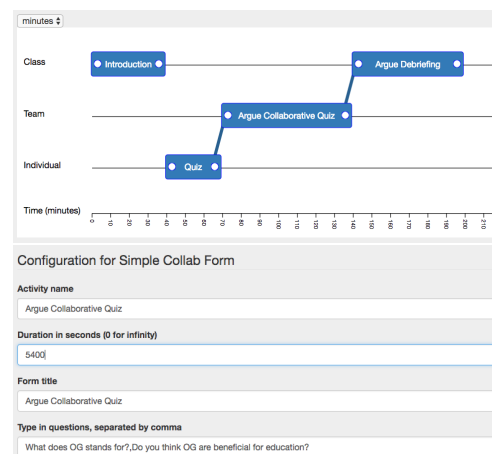

Figure 1. Orchestration Graph Editor (top) with the activity configuration interface (bottom).

The main functionality includes:

- Creating learning activities as nodes, and moving and resizing them to change their starting time and duration
- Creating operators to generate social structures or send the student product from one activity as input for another activity
- Creating edges to depict the pedagogical relationship between activities
- Configuring learning activities and operators

Learning Activities

Once the teacher starts a session based on a specific Orchestration Graph, registered students are able to join the session

and are presented with the current learning activity. We have implemented several simple types of learning activities in the prototype, such as Quiz, Video, Collaborative Quiz with a chat, and Visual Brainstorming. The activity API enables the design of new rich collaborative activities with little code.

For example, Figure 2 shows two students answering a quiz collaboratively. Their answers are synchronised and they can use a chat to reach agreement. The students would have been matched in a group by an operator based on some criteria, which could range from very simple, like generating random pairs, to very complex, like calculating the maximum divergence in opinion (from a previous questionnaire) to form teams.
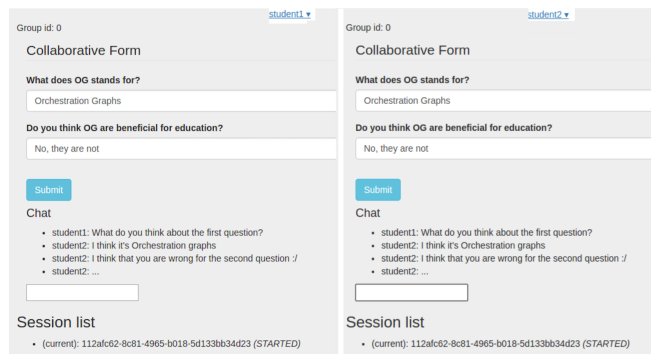

Figure 2. Two students participating in a collaborative learning activity.

Orchestration and analytics dashboard

One of the main strengths of our implementation is the ability for the teacher to orchestrate a learning session. We provide a live-streaming learning analytics dashboard for the teacher to access information needed to make decisions for his course. The teacher can also update the graph to orchestrate the session while it is running, within integrity constraints (as an example of an integrity constraint, a group activity cannot run if the groups have not been previously defined through an operator).

Figure 3 shows a custom teacher dashboard displaying the progress of individual students' video watching. This dashboard is only an example on a small classroom, but we are currently in the process of implementing new visualisations for monitoring larger groups of students.

The aim of this dashboard is to allow the teacher to decide whether he needs to intervene with the script or if it is running satisfactorily. An ongoing line of research focuses on effective representations of student state at scale, allowing teachers to see at a glance how the whole class is performing, identify outliers, and zoom in or sub-select based on various criteria to study student progress in more detail.

As well as utilizing the stream of learning traces to update the live dashboard, details are also captured for future study, using the xAPI standard [1].

EXECUTION OF THE DEMONSTRATION

Our demonstration will aim at showing the main features of our technology. As our application is hosted on a server, we
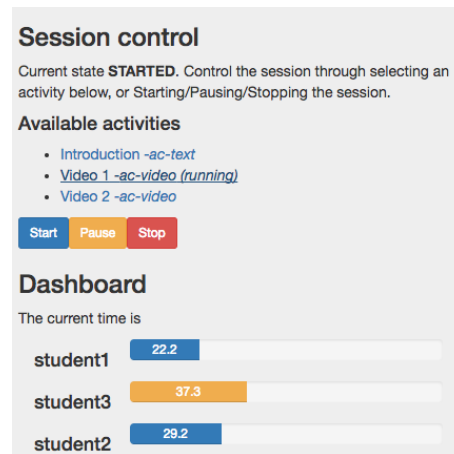

Figure 3. The teacher's analytics dashboard and orchestration controls.

will only require to use the web browser of 3 to 4 computers with Internet connection.

As the design of a non-trivial orchestration graph or the completion of a sequence of learning activities are processes that can last several hours, we plan to automatically simulate the interaction of a teacher and a group of students within our application. The simulation will first show the teacher designing a collaborative scenario using our orchestration graph editor, then the students following the pedagogical script, including individual and collaborative activities and finally the teacher monitoring and orchestrating student learning with the help of the analytics dashboard.

Visitors willing to spend more time with us will be able to explore our application, either by designing their own pedagogical scripts with the graph editor or by testing as a student one of some well-known collaborative learning scenarios such as ArgueGraph and Jigsaw [2].

CONCLUSION

We will present a web learning platform using the concepts of Orchestration Graphs to build and run rich social pedagogical scenarios with the goal of bringing collaborative learning in MOOCs. We expect to collect new ideas about scripts, learning activities and operators.

REFERENCES

1. Aneesha Bakharia, Kirsty Kitto, Abelardo Pardo, Dragan Gašević, and Shane Dawson. 2016. Recipe for success: lessons learnt from using xAPI within the connected learning analytics toolkit. In *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*. ACM, 378–382.

2. Pierre Dillenbourg and Patrick Jermann. 2007. Designing Integrative Scripts. In *Scripting Computer-Supported Collaborative Learning*, Frank Fischer, Ingo Kollar, Heinz Mandl, and Jörg M. Haake (Eds.). Vol. 6. Springer US, Boston, MA, 275–301.