FROG: rapid prototyping of collaborative learning scenarios

Stian Håklev[1], Louis Faucon[1], Thanasis Hadzilacos[2,1], Pierre Dillenbourg[1]

1. École Polytechnique Fédérale de Lausanne, Switzerland
2. Open University of Cyprus, Cyprus

Abstract. We describe FROG, an integrated environment for authoring and running collaborative learning scenarios, called Orchestration Graphs. We describe the pedagogical background and the technical architecture, and present a case study of a teacher using FROG to experiment with a variation of a jigsaw script.

Keywords: scripting, orchestration, group formation, learning analytics, collaboration, CSCL

1 Pedagogical background

This paper presents FROG, an integrated environment with tools implementing Orchestration Graphs (OG) [2]. OG model teaching and learning scenarios as a structured network of learning activities devoted to a set of learning objectives. FROG[1], one of several possible implementations of OG, allows educators to design in advance, teachers to orchestrate in real time, and researchers to study afterwards what students actually do while learning.

Orchestration Graphs depict the structure (what is done when by whom), the pedagogical rationale behind the activity flow, and the work flow created by data transformations and student groupings ("social structure"). An "Orchestration Graph Engine" allows the teacher to run educational scenarios with her students following the activities as specified in the OG – or as modified real-time by her as she monitors their work – to provide input data and obtain output products. Finally, products and traces of the learning process are collected and processed; a stochastic model of the scenario uses these traces to predict (or confirm) learner states [2, p 95-100].

OG constitute a useful modelling language for any form of organized education (e-learning, large auditorium, MOOC, seminar, elementary school classroom) independent of learning theory and pedagogical paradigm. FROG is a tool for the action researcher and the design researcher. By translating educational scenarios into computational structures and providing an environment to test and run them, the modelling formalism (OG) provides a formal model for education. The implementing tool (FROG) takes care of the 'algorithmic' aspects of teaching – a significant part of a much broader whole. They do not constrict towards a specific learning theory and the ensuing pedagogical practice; OG are pedagogically neutral, but certainly more useful when the pedagogy and the social interactions among students are rich.

[1] https://github.com/chili-epfl/FROG

2 A platform for building and running pedagogical scripts

FROG is a web application for building and running collaborative pedagogical scripts. Teachers can design scripts by building an Orchestration Graph through a graph editor, and then run it. An OG is a structured view of a pedagogical script composed of connected Activities and Operators. Activities range from reading a text, watching a video, contributing ideas in a brainstorm or writing a report collaboratively. Activities receive input data and produce learning data. The role of Operators is to carry flows of data from an activity to another, while transforming, aggregating or distributing it. Operators can also create social structures which describe how students should be grouped during group activities.

Our system contains three main components; The first is a graph editor, which allows teachers to create, configure, and connect Activities and Operators. The second is the Engine that interprets graphs, provides the Activities to the students and runs the Operator algorithms. The last component is the library of types of activities and operators for which our system provide an API to be used both within the graph editor and the engine. Each activity and operator implemented in the library can be reused in different scripts. The only constraint is that input and output data of connected elements matches, which is checked by the graph editor.

As one of the main purpose of FROG is to run collaborative pedagogical script, the Activity API provides functionality for synchronizing complex data among sub-groups, collaborative text editing, and other tools required for online collaboration. Learning traces are integrated into the core of FROG, which allows us to implement context-specific meaningful analytics and visualization enabling the teacher to monitor student and group progress in diverse activities. Activities can also implement analytics algorithm for example to measure the quality of collaboration in the learning process or to extract the main topics in student's essays.

3 Scenario

Marie has used variations of the jigsaw-script [1] in her class many times, to let students specialize in different sub-topics, and then come together, combining their insights to solve challenging problems. Typically students get assigned different roles, and then meet in their role-specific expert groups to discuss first, before they regroup into mixed groups, to try to solve a problem together. She has the idea to let students continue to stay in touch with their expert groups, while they are discussing in their mixed groups, but is worried that the cognitive overload will be detrimental.

She decides to test her new design in a class session. Here are her draft notes for her lesson plan: *"5 teams, each representing a company with CEO, production, financial and marketing members – if one is missing, financial and marketing are merged to one role; all students receive one page with the problem; students in each role receive one issue to optimise from their point of view and a few web links to explore – too much for the allocated time, they will have to choose what to read; they read for 5-10' individually; they group in similar roles for 10' and exchange ideas and explore options keeping notes; then they group as companies do decide on strategy..."*

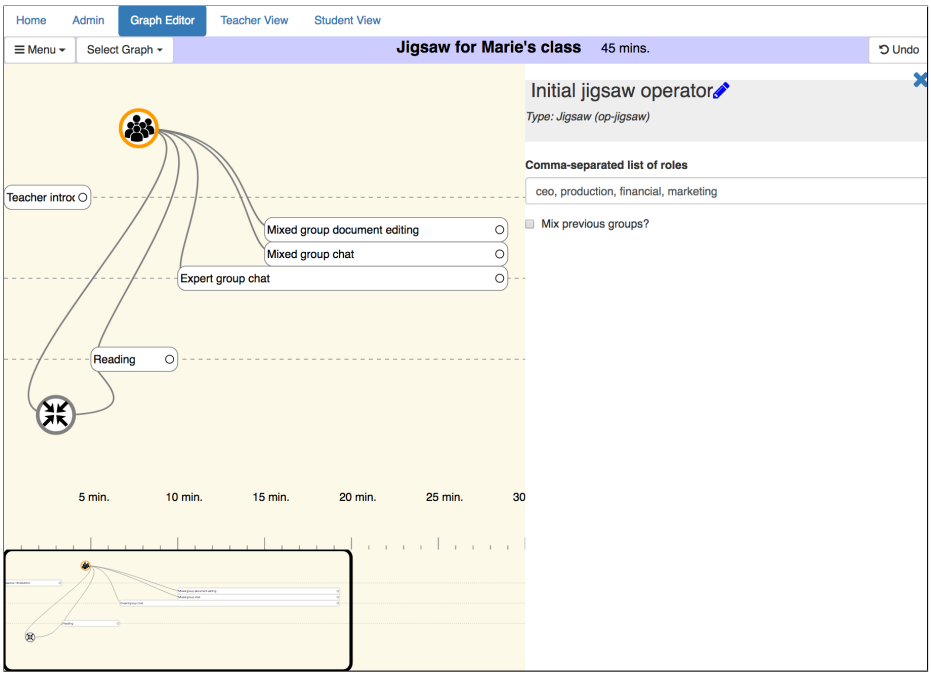

Fig. 1. Pedagogical graph of Marie's script.

She opens the FROG designer, and begins adding activities. Let's start with a whole-class activity in which the teacher for 5' explains what will happen. Then, there will be a reading activity, where students see different material depending on their expert role. Then, students will meet in expert groups, and she chooses four chat activities with discussion prompts. For the next activity, the students will meet in their mixed groups. She adds a collaborative editing document for each group to author the company strategy for the problem, and a chat for the mixed group to collaborate. To try out her new idea, she selects the chat activity from stage two, and expands it to cover stage three as well, that way students keep access to their fellow experts, even after the third activity in their mixed group has begun.

Then it is time to wire up the dataflow and social flow. First, she has to generate the groups. She chooses the jigsaw operator, and specifies the roles (in this case, the challenge is related to urban development, and she wants a CEO, production, financial and marketing roles). This operator will generate a list of students, with two attributes, group: {1, 2, 3} and role: {ceo, production, financial, marketing}. She links this social structure to all the different activities.

In order to distribute the different readings based on role, she needs a product operator, which takes a groupingKey (in this case, 'role'), and a list of attribute values and contents. Since she is using an iFrame activity to display the readings, this means a list of pairs (role, URL). This operator connects to the initial reading activity. Now,

all she has to do is to specify the groupingKey for each activity (role or group), and the graph is ready to run. See Figure 1.

The students arrive in the classroom, and log in to FROG. Mary starts the graph engine. While students are working in groups, she can follow their progress on different dashboards such as a set of word clouds highlighting the different directions in which different group discussions go (while hiding the words common across all groups), or collaborative text analytics, showing how well groups are collaborating while editing their final responses. Based on this insight, she might choose to move the script along more quickly, pause all the student screens to make a quick announcement, or even virtually visit a certain team to provide input and guidance. See examples of working screens for four students in Figure 2.

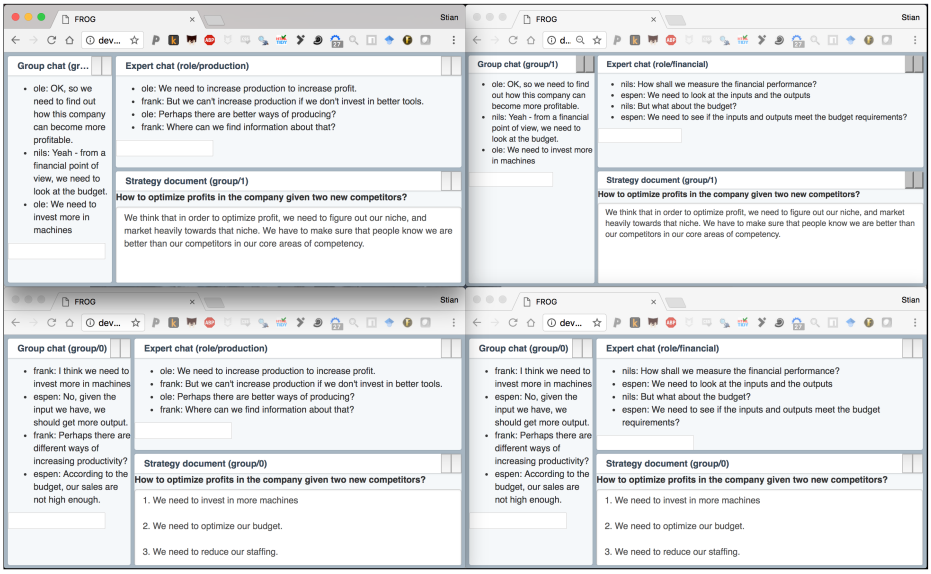

Fig. 2. Four browser windows representing four students logged in, each student is member of group one or two, and simultaneously either in production, or finance.

After the class is over, Mary asks the analytics engine for a table showing for each student how often and how much he contributed; she will use it to encourage low-engagement students. She also asks for the same information in anonymous statistical chart; she will pass it along and discuss it with other teachers.

References

1. Aronson, E., Blaney, N., Stephin, C., Sikes, J., and Snapp, M. *The Jigsaw classroom*. Beverly Hills, CA: Sage Publishing Company, 1978.
2. Dillenbourg, P. *Orchestration Graphs: Modeling scalable education*. EPFL Press, 2015.

Demo description for FROG: rapid prototyping of collaborative learning scenarios

Stian Håklev[1], Louis Faucon[1], Thanasis Hadzilacos[2,1], Pierre Dillenbourg[1]

1. École Polytechnique Fédérale de Lausanne, Switzerland
2. Open University of Cyprus, Cyprus

Abstract. We describe the demo of FROG, an integrated environment for authoring and running collaborative learning scenarios, called Orchestration Graphs.

Keywords: scripting, orchestration, group formation, learning analytics, collaboration, CSCL

1 Organization of the demo

We will set up two work-stations where people interested can experiment with the graph editor and modify existing graphs, like the advanced jigsaw mentioned in our paper. We can quickly run the graphs with several browser windows to simulate multiple students.

We will also make the tool available online during the conference, to let anyone with their own computer access our graph editor, and author and run graphs.