FROG, A Tool to Author and Run Orchestration Graphs: Affordances and Tensions

Stian Håklev, Louis Faucon, Jennifer Olsen, Pierre Dillenbourg
stian.haklev@epfl.ch, louis.faucon@epfl.ch, jennifer.olsen@epfl.ch, pierre.dillenbourg@epfl.ch
École Polytechnique Fédérale de Lausanne

Abstract: Orchestration Graphs (OG) consist of sequences of activities placed in a two-dimensional coordinate system, with the x-axis denoting time (absolute or relative), and the y-axis the social planes. This contribution to the Special Interactive Session will discuss FROG, a tool based on the OG notation, and the affordances and tensions between the theoretical framework and the theory instantiated as a tool.

Orchestration graphs

Pierre Dillenbourg introduced Orchestration Graphs (OGs) in 2015 as a new notation for modeling the design and orchestration of sequences of learning activities. OGs consist of sequences of activities placed in a two-dimensional coordinate system, with the x-axis denoting time (absolute or relative), and the y-axis the social planes, usually individual, team and whole class, but expandable to also include periphery, community and world.

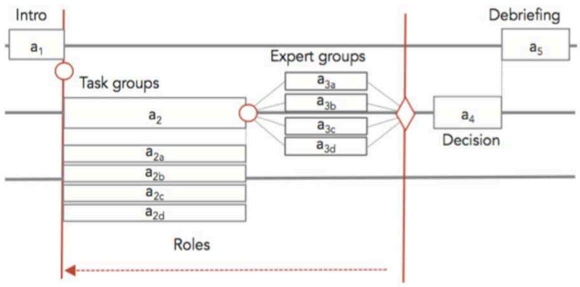

Figure 1. Example of an Orchestration Graph.

Graph activities (nodes) are connected by edges, which represent both pedagogical and data dependencies. The pedagogical relationship between two activities might be described as motivation, advanced organized, induction, etc. However, in order to enable rich collaborative sequences, edges can also express data flow through the use of operators, which transform the data structures produced during a learning activity into the data structures needed to run the next activity. For example, student opinions about an ethical dilemma could be aggregated and converted to a visual representation that the teacher can show the class, but can also be used to automatically group students with different opinions. Edges can also contain control structures, such as conditional operators or loops.

Several researchers have used OGs to analyze and communicate their experimental designs, such as Acosta (2018), who added a learning analytics layer, White (2018) used OGs as an analytical tool to analyze shifts between social planes in a math classroom, and Prieto, et al. (2018) attempted to automatically reconstruct an orchestration graph from multi-modal teacher data. In Singapore, some teacher education courses use OGs to train teachers in learning design (Samuel Tan, personal communication, December 5, 2018). In this paper, we will introduce a platform called FROG, which lets users design and run learning designs based on OGs.

Learning design notations

There have been a series of proposals for educational modeling languages (Botturi, 2007). While a data format for encoding instructional designs, such as IMS-LD, would aim to capture all information relevant to the design, graphic notations must choose judiciously which aspects of the design to make explicit. Many proposals visualize sequences of activities, and some introduce swim lanes (such as CADMOS, Katsamani, Retalis & Boloudakis, 2012) to distinguish between different entities, such as students, teachers, and resources. Others, like the Learning Designer (Laurillard, 2013) emphasize the kind of learning process an activity is designed to foster through colors. The elements that are visually emphasized in OGs, on the other hand, are the activity

 © ISLS

sequence relative to time, the social planes, and the data flow between activities. Pedagogical information is focused on the relationship between activities (edges) rather than the purpose of individual activities, and resources or tools are not explicitly modeled.

FROG

FROG is our group's attempt at building a platform that lets users author executable Orchestration Graphs. It is an open source (https://github.com/chili-epfl/FROG) browser-based application where teachers use the graph editor to author graphs, and the orchestration view to run and monitor sessions. It is currently focusing on synchronous sessions, both in one-to-one classrooms, and in fully online settings, and has so far been tested in a Swiss middle school, and several university lectures in Switzerland, the US, Norway and Greece, ranging from 150-300 student undergraduate lectures, 30 student graduate seminars to small online courses.

Open-ended learning design versus authoring a concrete FROG lesson

One interesting tension is between on one hand enabling teachers to rapidly and correctly input all the information necessary to let an "orchestration engine" actually instantiate a learning design, and on the other hand to support an open-ended and creative design process for a lesson that might even happen elsewhere, or fully offline. FROG has so far focused on the former, which means that you can only add activity types that already exist in the system, and there is not much space to capture unstructured ideas, or pedagogical information that is not strictly necessary for the orchestration engine to run (see the graph editor in Figure 2). Although we often engage in learning design work around FROG, whether when working with teachers to co-design collaborative learning lessons, or when students in a graduate course on Digital Education use FROG to design lessons that reflect the learning theories that they have been examining, we tend to do the actual design process on paper or in a Google Doc, while referencing the OG notation ("this activity is on plane 3, and we will use this operator to connect with this other activity").

In order to capture enough specific information for the information flow within a graph to be instantiated, we have also had to deviate somewhat from the original design. For example, we represent operators not as parts of edges, but as entities that can have multiple inputs and outputs, to make the information flow much more explicit. We also added an additional plane for teacher tasks, to explicitly model things that teachers might need to do as part of the graph workflow (grading, classifying examples, etc.). According to OG theory, these would be "interactive operators", but they importantly have a time dimension, which justifies modeling them as activities.

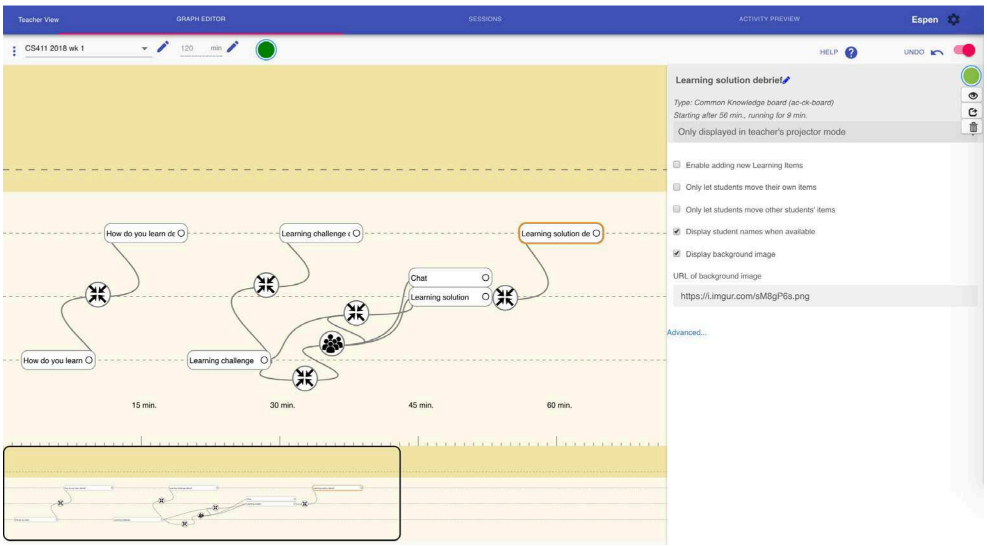

Figure 2. The FROG Graph Editor.

How the theoretical framework impacts technology design

Although there are other orchestration tools that enable some flow of information between different social planes and different activity types, the OG framework guided us to focus strongly on designing the activity type APIs and operators, as well as the underlying data structures, in a way to maximize flexibility and connectivity. We believe that this principled approach to technology design leads to a framework that is not only very flexible, but also learnable through some key abstractions. However, there are some tensions inherent in this approach. One is that we necessarily had to make some adaptations to the OG notation when implementing it in an editor meant to output runnable graphs. As we proceed to popularize FROG, we ask ourselves to what extent we should emphasize the OG theory when onboarding teachers, and how to manage the divergence between the theoretical representations and the actual interface (it is of course possible that in the future, we could revise the theoretical representations based on our experiences with implementing FROG).

Because of our desire to maximize flexibility and configurability of scenarios, we have also ended up with a somewhat complex interface, as seen in Figure 2. In the future, we hope to simplify this interface, by implementing better defaults and hiding some of the wiring for very common cases (like having a single group structure persistent throughout a session), while still enabling and exposing the possibilities of doing very complex designs.

Theory-free use of FROG technology

FROG was from the beginning conceptualized as a pluggable ecosystem, with the goal of having multiple groups contributing activity types (threaded chats, collaborative concept mapping or physic simulations) and operators (group formation, semantic analysis and inferring student progress). Given our focus on synchronous collaborative learning, the FROG engine provides an activity-type API which makes it very easy to build innovative activity types that support synchronous collaboration, and these activity types themselves could be interesting to other learning projects that might not share our theoretical stance (see Figure 3 for an example of an innovative synchronous collaborative activity types in FROG).

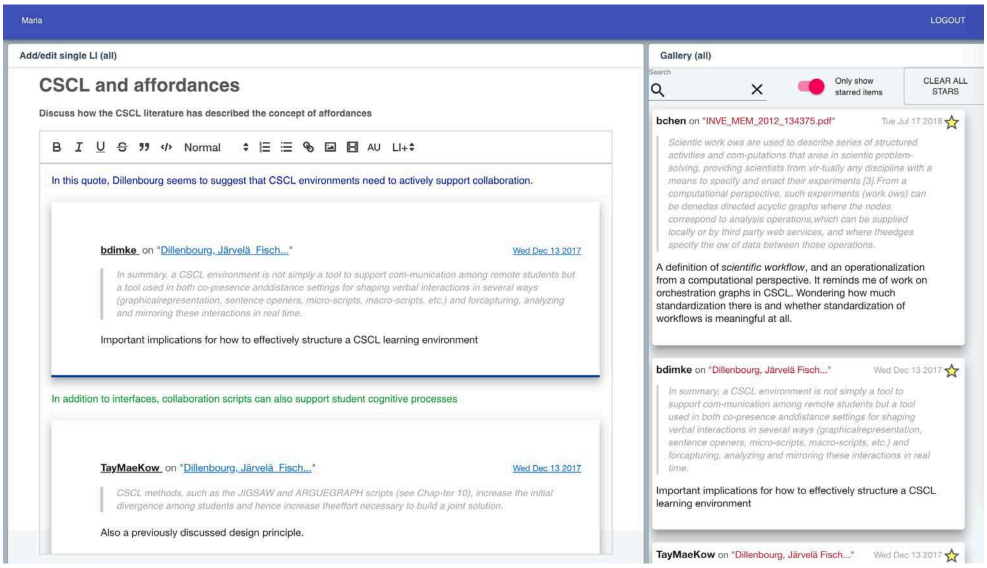

Figure 3. A searchable gallery with Hypothes.is annotations, and a collaborative rich text editor, which supports embedding rich content (like the annotations).

We are currently prototyping APIs to expose individual FROG activity types to other systems, enabling remote configuration, student grouping, embedding of dashboards, and live-streaming of learning analytics. The Graasp platform is focused around science inquiry spaces (Bogdanov, *et al.* 2012) is currently integrating FROG activity types and making them available to teachers in the EU GoLabz project.

Conclusion

We have described Orchestration Graphs as an approach to learning design notation, and how it has inspired the FROG platform. We will demonstrate this in practice, letting people experiment with the graph editor, and participate in a graph as learners.

References

Laurillard, D. (2013). *Teaching as a design science: Building pedagogical patterns for learning and technology*. Routledge.

 © ISLS

Acosta, A. (2018). *Scripting and Orchestration of a Knowledge Community and Inquiry Curriculum for Secondary Biology.* (Unpublished doctoral dissertation). University of Toronto, Canada.

Prieto, L. P., Sharma, K., Kidzinski, Ł., Rodríguez-Triana, M. J., & Dillenbourg, P. (2018). Multimodal teaching analytics: Automated extraction of orchestration graphs from wearable sensor data. *Journal of computer assisted learning*, *34*(2), 193-203.

White, T. (2018). Connecting levels of activity with classroom network technology. *International Journal of Computer-Supported Collaborative Learning*, *13*(1), 93-122.

Katsamani, M., Retalis, S., & Boloudakis, M. (2012). Designing a Moodle course with the CADMOS learning design tool. *Educational Media International*, *49*(4), 317-331.

Bogdanov, E., Limpens, F., Li, N., El Helou, S., Salzmann, C., & Gillet, D. (2012). A social media platform in higher education. In *Proceedings of the Global Engineering Education Conference (EDUCON)* (No. EPFL-CONF-178720, pp. 1-8).

Botturi, L. (Ed.). (2007). *Handbook of Visual Languages for Instructional Design: Theories and Practices: Theories and Practices*. IGI Global.