An Engine producing Orchestration Graphs

Samuel Bélisle

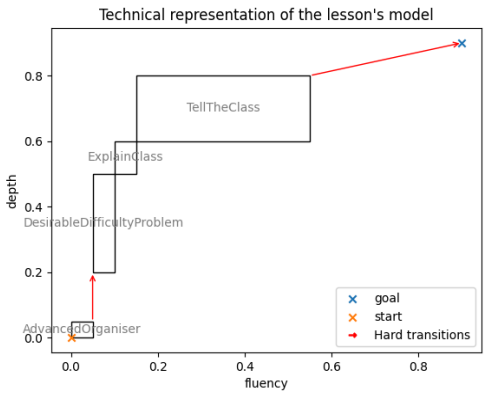
Supervisor: Pierre Dillenbourg

MOTIVATION

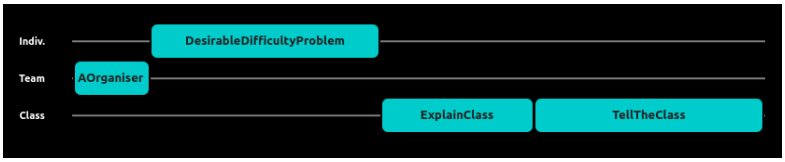
In this semester project, we designed and implemented parts of an application that could help teachers in their lesson planning, in particular the creation of an orchestration graph. The scope of the project was to develop a prototype integrating some parts of the required elements.

METHODS

The project has two parts. The first is a Python engine that manipulates activities within a lesson and produces recommendations of which activity to use and when. This engine reasons with an extremely simplified model we developed which represents the state of the class. This state encodes their understanding of the topic and their fluency doing the associated tasks.



The second part is an interactive application built with PyQt and QML, using the engine. This application makes it intuitive to interact with the engine. It represents the lesson using an orchestration graph:



RESULTS

The most interesting research question was about the recommendation of the activities by the engine. We explored a greedy algorithm making choices according to a local heuristic. Different heuristics were explored, and one of them left satisfactory results. However the main finding is that the locality of the heuristics makes them lack a key property. We theorised another approach but had not enough time to implement it.

On a technical and personal side, the project involved the creation of a graphical interactive application. Creating this application was a new process for me. I discovered a new language (QML) and some very useful python libraries (pickle, pipreqs, pyinstaller). Linking all of this was very instructive.

Link to the code repo: github.com/Katokoda/OG_QML/. See README.md and the Reports/ subfolder.