COMS 6901 Midterm Report

Spring 2018

Chao-Yang Lo

1. Abstract

This report discusses the current progress of Gameful Intelligent Tutoring system. It first focuses on the progress of Scratch Analyzer to Assessment Server Connectivity [292], including the architecture and implementations. In the latter part of this report, it explores the next possible proposals and implementation of Programming Behavior Persistence [293].

2. Architecture

Intelligent Tutoring system is a proposed system designed to help students to perform better and learn more efficiently by providing appropriate hints. To do so, we need to classify students by their behavior. While there have been some works done to analyze and classify student with different learning types, the whole solution is executed locally and student behavior data are fetched and stored in local file system. User story [292] proposed the idea to migrate the analyzer on the server.

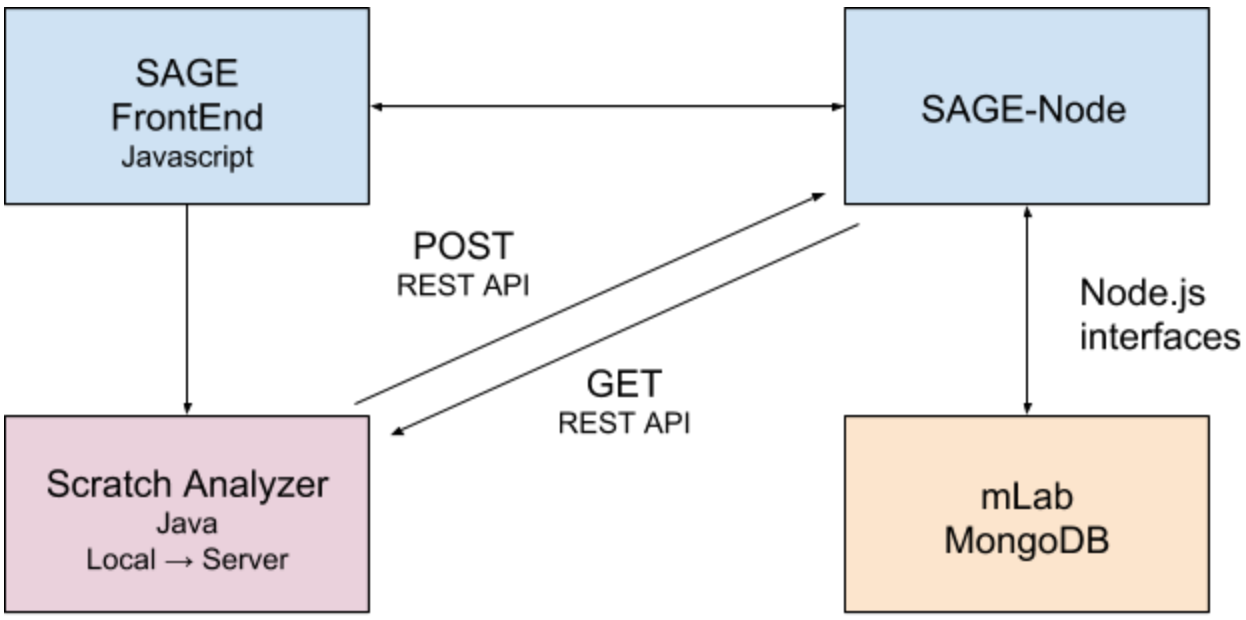

Fig. 1

Fig. 1 show the proposed architecture to integrate scratch analyzer system. RESTful APIs will be used to fetch data and upload processed data back to the server. Once the communication is modified to be fully dependent on APIs instead of the file system, the whole working process can be deployed and executed on the server.

3. Implementation

Instead of storing the game JSON files in the local file system, I first created a Java program with Spring framework, which provides a series of REST consuming APIs. Also, I investigate the data structures stored in mLab database and was able to fetch game JSON files using game IDs as shown in Fig.2. The JSON files can then be fed to Scratch Analyzer and student behavior sequence files can be generated.

Fig. 2

4. Assumptions and Limits

The main assumption made at this stage is that the system only emulates the possible solutions but it still runs the program locally. Whether it is feasible server-side solution still needs testing after deploying the system on a real server.

5. Future works

4.1 Completing Assessment Server Connectivity

Currently, the development is still not complete. There are two important components to be finished. The first is to refactor the existing scratch analyzer so that the analyzer can be directly applied to JSON data structure. The other is that the POST method needs to be implemented to store the processed data in SAGE-Node.

4.2 Programming Behavior Persistence

This feature requires the data regarding student classification to be stored also in SAGE. As there are already a collection of Student data in mLab and it is the format of JSON, it is possible to add another attribute indicating the classification of a certain student. Also, the RESTful consuming application in the previous feature can be reused to manipulate database via SAGE-Node.