Measuring the Centralized Mindset in Scratch

Dimitris Nikolos, dnikolos@upatras.gr
Dept of Educational Sciences and Pre-school Education, University of Patras, Greece

Vassilis Komis, komis@upatras.gr
Dept of Educational Sciences and Pre-school Education, University of Patras, Greece

Abstract
This paper describes an effort to measure the centralized control of Scratch projects using social networks analysis software. We developed software to transform the structure of Scratch projects into networks. The theory of centrality and centralization of social networks can then be applied to the structure of a Scratch project providing measurements for the centralized mindset of the creator.

Keywords
Scratch, Centralized mindset, Social Network Analysis

Introduction
The centralized mindset is a theory developed by Resnick (Resnick, 1992). His work on StarLogo showed that students tend to apply centralized control when they model distributed phenomena. Using StarLogo a student can create thousands of turtles with programmed behavior. Even though each turtle acts autonomously patterns appear in the behavior of the whole system. Students incline towards explaining these patterns assuming leadership or some kind of inherent seed (Resnick, 1996).

Centralized mindset is also present in Object-Oriented Programming where students develop centralized solutions to problems at the cost of reusability. In some cases centralized models are suitable for problems but a decentralized approach would result in managing complexity more efficiently (Guzdial, 1995).

A visualization of a centralized model resembles a star network where a central object is connected with all the other objects. A visualization of a decentralized approach resembles a circle where each object communicates only with its neighbors. The star and the circle network are networks of interest when centrality (for vertices) and centralization (for the whole network) are computed in social network analysis (Scott, 2000).

This paper presents an effort to quantify the centralized control of an arbitrary Scratch project. In order to measure the centralized control of a Scratch project the structure of the project was transformed into a network using an automated tool that we developed. Sprites, variables, messages, lists and the scene are the vertices of the network. A sprite can be connected to the variables, messages, lists or to the scene if it affects them. A path between two sprites (A,B) means that a process on sprite A can trigger an action on sprite B. If there is no path between two sprites (A,B) their behavior is independent of each other. The software that is used to generate the networks is developed in Python and it is distributed via Github (http://github.com/dimnikolos/Ghoul/) under the MIT License.
Figure 1. An example of network generation

For example consider a remix of a project that was used in the Creative Computing Online Workshop (Brennan, 2014) the scripts of the project and the corresponding network are shown in figure 1. The Scratch Cat sends a single message to Giga, Nano, Pico and Tera. The message is central to the communication of this project.

The centralized control can be measured using methods that apply to Social Network Analysis. Two are the main measures of network centralization: closeness centralization and betweenness centralization, they are computed as values from 0 to 1 (Freeman, 1979). For the network of figure 1, closeness centralization is 1 (star) and betweenness centralization is 0.2.

The methodology is presented in the next section then we discuss our observations and finally conclusions are drawn.

Methodology

In order to determine how informative closeness and betweenness centrality are in measuring the centralization of a Scratch project we used the projects that thirty-six third year students of a Pre Service Early Childhood Education Department created. They enrolled in an elective laboratory course about Scratch programming. At the end of the 20 hours laboratory course they were asked to submit a final project that could be used by their pre-school students. All the students can be considered novice programmers and they had no previous experience in any kind of programming. Two projects featured minimal interactions between sprites and were omitted from the study. The nineteen projects that remained comprise our data.

These projects are not answers to small programming exercises, the students had to design the projects themselves. The laboratory lessons emphasized on the communication between sprites but no particular emphasis was placed on centralized and decentralized approaches to programming. If they wanted they could design a completely centralized project and implement it.

We computed the two main centralization measures, closeness centralization and betweenness centralization for the projects using the Pajek software (Batagelj & Mrvar, 1998). We used the normalized measures to account for the differences in the network sizes. The results were interpreted in terms of project structure.
Discussion

The centralization measurements for the projects are shown in figure 2. Two projects were omitted from this study, project 11 and project 21.

![Centralization measures for the projects](image1)

*Figure 2. Centralization measures for the projects*

The second and the twelfth projects feature the highest centralization measures. Screenshots and corresponding networks are shown in figure 3. The projects feature centralized control where a single variable stores the current state of the program and everything is synchronized with it. The two projects are very different in size. However they both have a central variable that controls the whole project. In that sense, closeness centralization is very informative for the centralized control of the projects. On the other hand, betweenness centralization does not seem to be very informative in that sense, since betweenness centralization for project 2 is around 0.9 and for project 12 is 0.5.

![Maximum closeness centralization](image2)

*Figure 3. Maximum closeness centralization*
In figure 4 other networks with high closeness centralization are shown (projects 1, 7). They all form the star structure and they all have centralized control with the use of either variables (project 1) or the scene (project 7). In project 7 we notice that the betweeness centralization measure is very low, in contrast with the centralized nature of the project.

The networks of the projects that feature high centralization measures form the star pattern; however the projects that feature low centralization can have various forms.

The fourth project is characterized by very small closeness centralization. The network of this project is shown in figure 5. We can see that the whole project is not centralized but the variables (blue vertices) seem to be focal points for the behavior of the sprites. A closer look on the project confirms this observation since the game had three levels and each level was controlled by a different variable.
The fifth project according to centralization measures does not seem to feature centralized control. Both in the network and in the project itself (figure 6) the variable “number” seems to play an important role but there are other means of communication equally important.

Another project with very low centralization measures is the tenth project (figure 7) where the student named a variable “control” but this variable is not in the center of things, it actually controls only two sprites of the project. The project is decentralized but it is not structured in any way.

Closeness centralization measure was much more informative than betweenness centralization in our context. Projects with closeness centralization of 1 were highly structured with only one medium controlling the whole project. The same was not true for betweenness centralization.
Conclusions

We observed that projects that featured high closeness centralization have a centralized nature. The same was not true for betweenness centralization.

Students submitted projects with both centralized and decentralized control. All centralized projects were highly structured and a single object controlled the whole project. The majority of the decentralized projects did not feature a concrete structure where each sprite is communicating with its neighbors; they seemed to form unorganized patterns of communication.

The transformation of the structure of a Scratch project to a network provides new possibilities in teaching and learning Scratch. Future work may include the use of this transformation tool for debugging or visualizing some aspects of Scratch projects that remain hidden.

References


