Andreas Landgrebe, Victor Zheng, Lucas Hawk

Professor Wenskovitch

April 29, 2015

Final Project

Computer Science 250 Write Up

Class scheduling have always been a subject for debate. Some people may enjoy the way that the scheduling of Allegheny College works and some may not like the way that the class scheduling is working throughout Allegheny College. We believe that class scheduling can be improved. In order to improve such a system, we have looked at what students would be looking at when scheduling for classes.

One of the things that students would like to look into is the idea of getting results of locally optimal class schedule that the students has proposed. In order for students to get results for a optimal class schedule, we have looked into the inputs that must be used. The inputs we have decided to implement is professors, students, courses, classrooms and time slots. After these variables have been inputted, our system will allow us to output a optimal class schedule. In order to do show this the most effective way, we have decided to use a genetic algorithm to be able to give students a optimal class schedule the most effective way.

In this system, we have 7 java source code algorithms. Class.java has different objects to be used to organize this system the most effective way. In this file, we also get and setters and return these objects to get information about the professor, number of students already enrolled, the information about a course, and classroom that these classes will be held at.

The next file, Classroom.java layouts another part of the class scheduling system. This java file has getter and setter method to be able to return the ID of a student, get the name of a student, and also return the number of seats that a class has to offer.

Course.java is yet another file to be used for the design of this project. This will return the ID and the name of the specific course offered.

Professor.java will include yet more objects to be used. This java file will return the information about objects for the ID of the professor, the classes that the professor is teaching, the maximum number of classes that the professor could possibly teach, and the current amount of classes that a professor is teaching.

The next java file that is in the designed system is Schedule.java. This java file is meant to have students be able to organize their schedule by knowing the specific times that courses are being offered. This will have students be able to add and remove classes from their schedule given the time that it is being offered. We have also added the fitness to determine how well this schedule will work with a student. We still have to discuss how we will write this algorithm to determine how well a proposed schedule will work with a student.

Student.java will return a list of courses that a student is taking.

The last java file in the current proposed project is TimeTabler.java. This file is our main file to include a number of ArrayList to present all of the information from the objects that have been declared in the past files.

We have organized a system to use genetic algorithms to determine a class schedule between a professor and a student. There was a great amount of programming to be done in order to complete this proposed project.

There was also a great amount of data collections to be gathered as well. In order to run this program the most effective way, it is important to be able to gather as much data to be able to display the implemented system most effectively. The variants that were collected to implement this system were information about the professor, classroom, classes, and students.

In relation to the professor, there were a number of variables that were collected for the professor. Some of these include hard restrictions. In the implementation of our system, it is important to be able to make sure that professors do not teach two classes at the same time. We have also collected all of the names of the professors here at Allegheny College. All of these names will be collected and gathered into a XML file.

In relation to the classroom, there is a hard constraint that has been implemented. This hard constraints is making sure that two classes cannot be in the same classroom. We have also collected all of the possible classrooms at Allegheny College. We were able to find this information by going to the allegheny.edu website and finding all of the information about the possible classrooms that are at Allegheny College. All of the information about classrooms have been collected and a sample of these will be put into an XML file to be parsed.

In relation to classes, this is another area where there are a substantial amount of data collection. This was due to the fact that students will be looking at these classes to be able to find a schedule that is best suited for them. We have decided to go through the courses catalogue of Allegheny College and took a sample of these classes being offered. We have decided to only take a sample of classes being offered due to the fact that not every class will be offered in a given semester. There are not enough resources in relation to professors or classrooms at Allegheny College for every class to be offered at Allegheny College. Due to this, we have decided to only given a sample of classes that will be offered in a given semester.

In relation to students, we have also decided to give a sample of information about classes that students would like to take to Allegheny College. When looking at students, the main idea of our project is to have a tool to output a locally desired schedule for students. In order to evaluate that, it is important to create a sample of desired schedules that students would like to take in a given semester.

We have collected the names of the professors and possible classrooms that are at Allegheny College. Collecting this data is important to be able to give an ID to the professor to later be stored in an XML file. In order to read the information in the Xml file, we have decided to parse the XML file using a SAX approach. We have e died to use a SAX approach to parse the XML file compared to parsing the Xml file with a DOM approach for the number of reasons. one of these reasons include not having to worry about memory usages. Using a DOM parsing approach will load the whole XML file into memory. Due to having so much data collection in our system, it would have taken a longer amount of time to parse the XML file. The SAX parser is designed for large XML files to be parsed. The only downside to using a SAX approach compared a DOM approach in parsing XML file is the idea of having flexibility of changing structures and contents. Using a DOM parsing will allow use to do so but the system that has implemented has been designed to have fixed structures and contents.

When all of the implementation that we have described, we are also able to create a solution to be able to assess these tool. In order to do so, we have to be able to generate 100 random schedules to be evaluates. After these random schedules have been created, it is important to make sure that the hard constraints are held. By this, it means that these schedules needs to made so that professors cannot be teaching two classes at the same time or having two classes in the same classroom at the same time. After it is made sure that hard constraints of our tool are held, it is time to evaluate the soft constraints. By this, we are going to calculate the fitness of the desired courses for the students. In order to evaluate, we have decided to implement an algorithm to be able to keep the best desired schedules and remove the least desired schedules. To do this, we have decided to implemented what has been described as such. After the random schedules have been generated, we will sort these schedules and evaluate the fitness of each schedule. To evaluate this, we will kill the bottom 20 schedules and replace these schedules with children of two random top-ten solutions. After this, we will then randomly crossover each remaining solution with a random top-ten solution. After this, we have a implement a mutate overlap to find the best schedule for a given student.

From this all system that has been implemented, there has been some challenges to be able to implement such a system. The main idea of this project is to implement that makes it easier for students to find a desired schedule. Since that was the main idea of the system, we did not think about the possible time complexity of the project until we ran it. After looking through the source code, we saw that we had several nested for loops to go through and due to this, the time complexity of this project was poor. One area of research to look into is to implement a binary search so that the time complexity of this system is not so bad. This way, we would be able to use logarithms to be able to organize this system to be a little quicker compared to what it is now. Throughout the implementation of this project, we have learned the idea of group work and being able to communicate to group members is imperative in order to make the desired system at the correct time. We have also learned the idea of patience. Each group member learns to different paces, so it is important to be able to patient with group members that may take a longer amount of time to understand an idea of this system.

One of the biggest challenge that we had faced as a group is communication and time management. At times, it was difficult to communicate with group members via text messages or e-mail so being able to check e-mails and stay up to date with the progress of the project is important to order to finish the system at the correct time.

Another big challenge that we had faced as a team is compiler errors. At the point, this designed system has roughly about 100 errors that needed to be fixed. Looking through this source code to find these errors were tedious. However, looking through the source code for these errors was also rewarding. One of the biggest rewards was compiling this system and seeing no errors show up in the terminal windows on our desktops was the most rewarding idea we were able to look at.

For the future, it is important to think about working in a group for a project. It is important to follow a similar idea to the software engineering process where each group member has a specific role to finish the project at the correct time. In the software engineering process, it is important to have a leader in the group. Having this leader in this group will ensure that this system will be finished at the correct time. Another idea of the software engineering process is to group members to have specific tasks. One group members may have the responsible to have data collection to be gathered and another group member may have the responsibility of collecting all of these data and have this data be generated in an XML file. For the future, it is important to thoroughly discuss this system and assign tasks to each group member to make sure that everything is organized and completed at the correct time.