Report: Exploring Cyber-Physical Crowd Behaviour in an Agent-based System Using Data from Location-Based Social Networks First Stage

By Paulo Balbino Campos

st20048261

BSc Computing

Cardiff School of Management

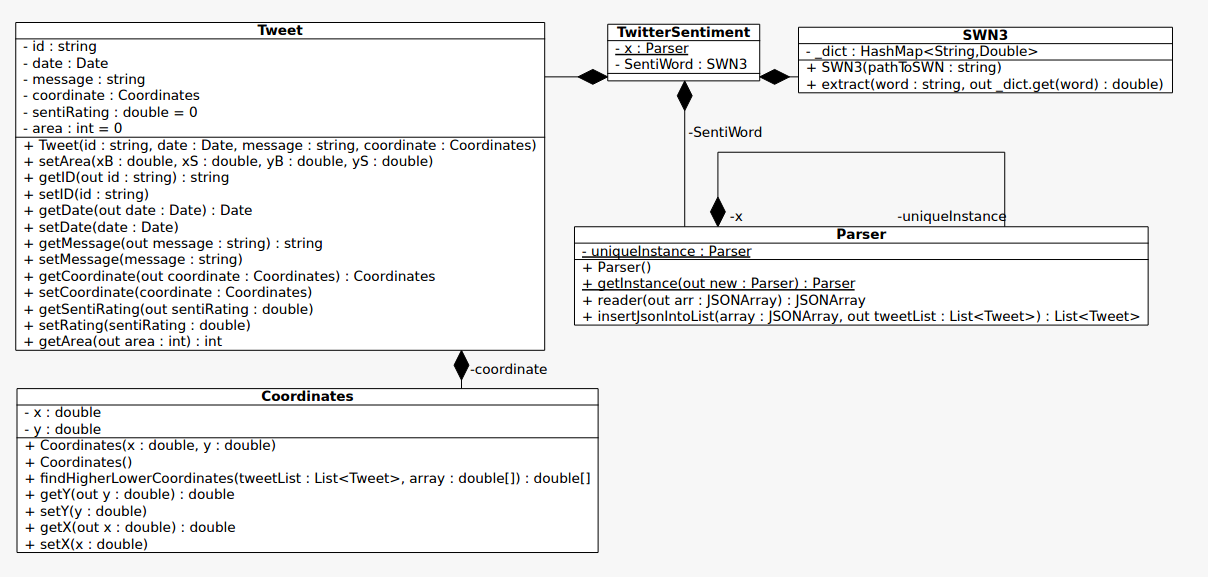
Cardiff Metropolitan University

1st of December of 2014

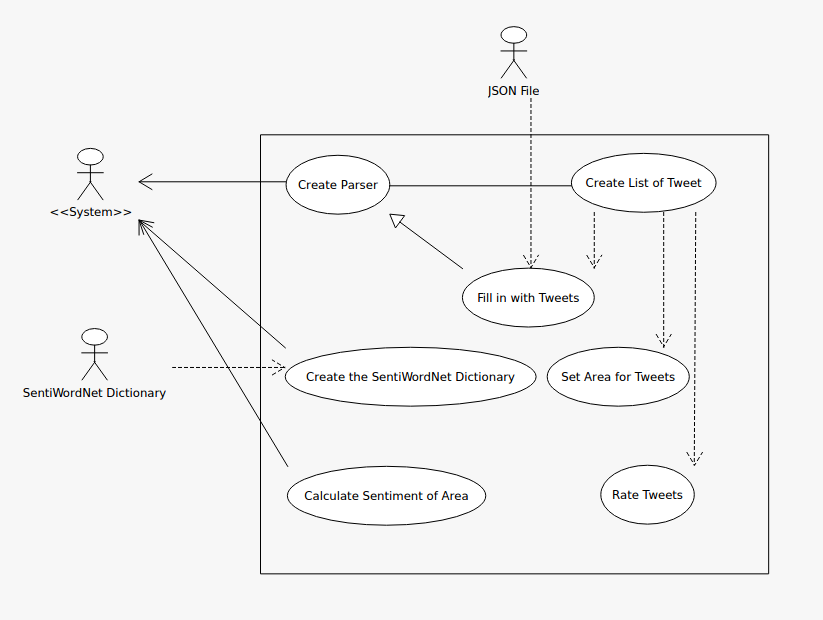
Use Case / Class Diagram / Sequence Diagram

* Class Diagram

The class diagram's purpose is to model the statical view of the application, it describes the attributes and operations of a class and also the constraints imposed on the system. It is used for visualizing, describing and documenting different aspects of a system.

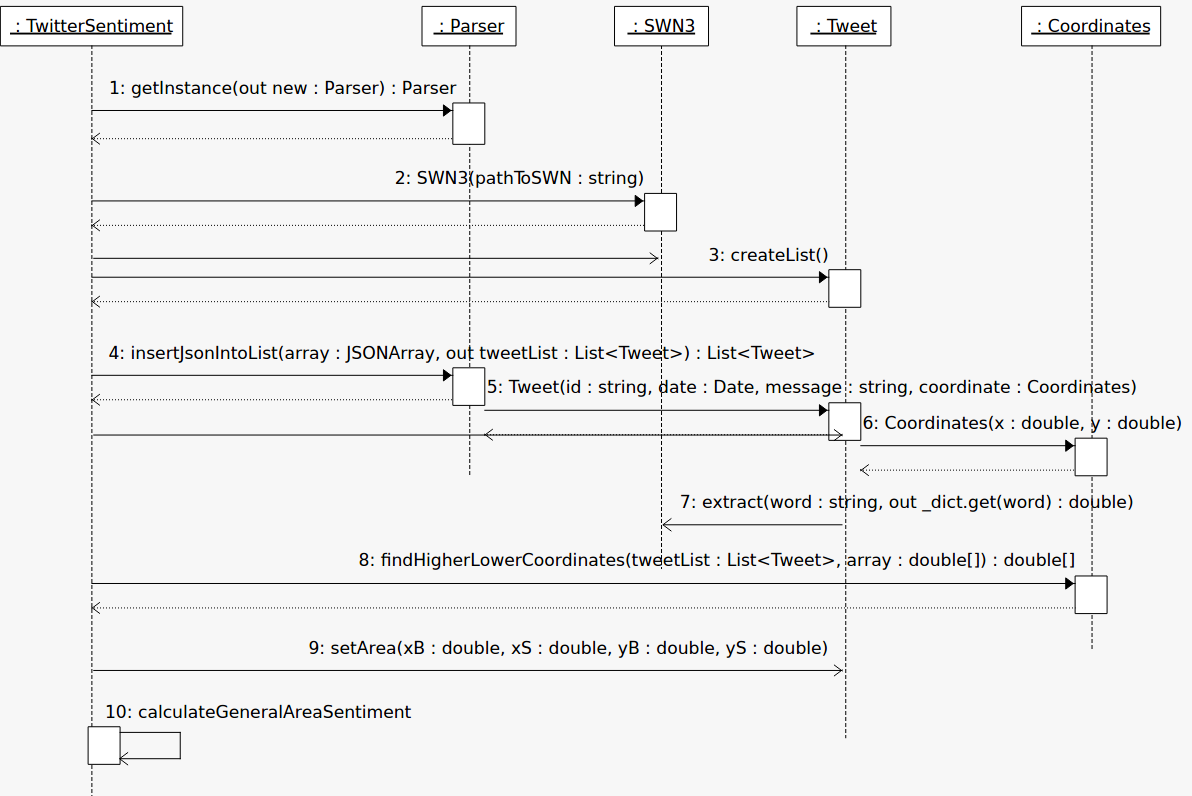
Illustration 1: Class Diagram

* Use Case Diagram

Illustration 2: Use Case Diagram

The purpose of a use case diagram is to gather the requirements of a system, including external influences. It get an outside view of a system, identify external factors that influence the system and show the interaction between elements.

* Sequence Diagram

Illustration 3: Sequence Diagram

The Sequence Diagram is used to describe some type of interactions between elements in the model. It captures dynamic behaviour of the system, describe the message flow in the system and structural organization of the objects.

Re-factoring

* Handling the Coordinates

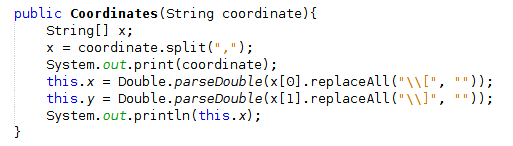
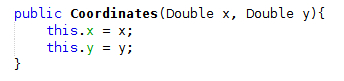
Illustration 4: Previous Code

Illustration 5: Current Code

At first, instead of getting a JSONArray from the JSONObject tweet, I was getting the JSONArray of coordinates as a String, and handling it in the Coordinates constructor, using regular expressions to remove the brackets and splitting the String by the comma. Now the new code, gets the JSONArray and store it in a Array of Doubles, which then call the constructor getting the first coordinate and second.

* Renaming

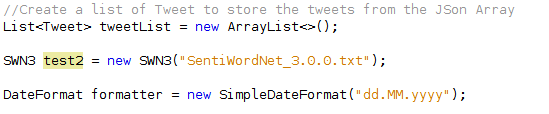
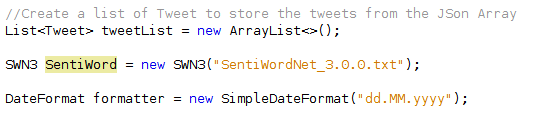
Illustration 6: Previous Code

Illustration 7: Current Code

On the previous code, I was trying different methods and giving them different names to test which one of them would be best, test2 was the chosen method, and then it was renamed to a more appropriate and obvious name.

* Moving operations and methods to functions

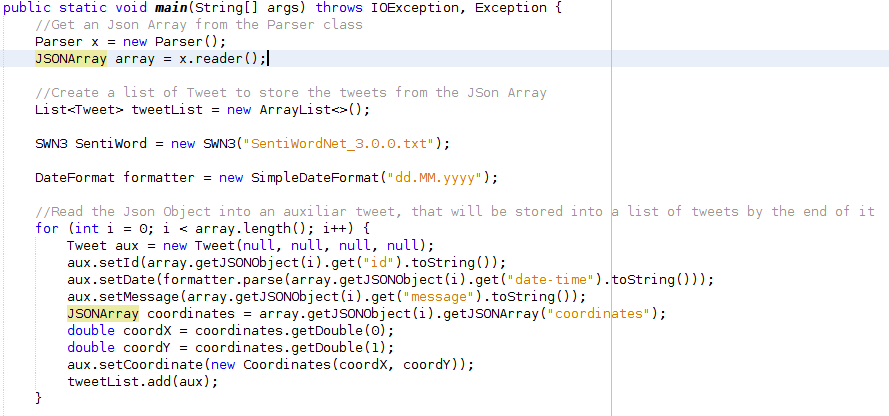
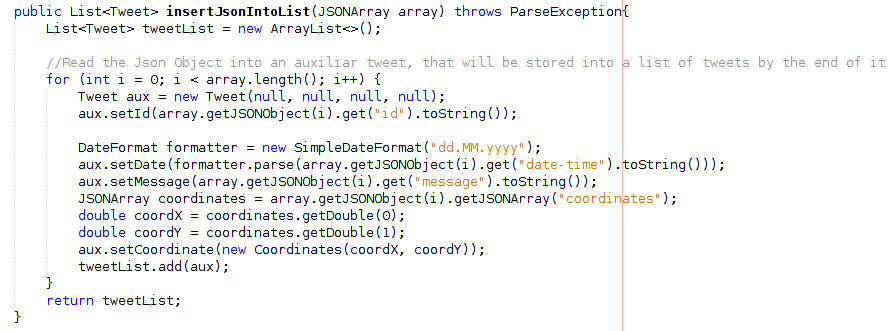
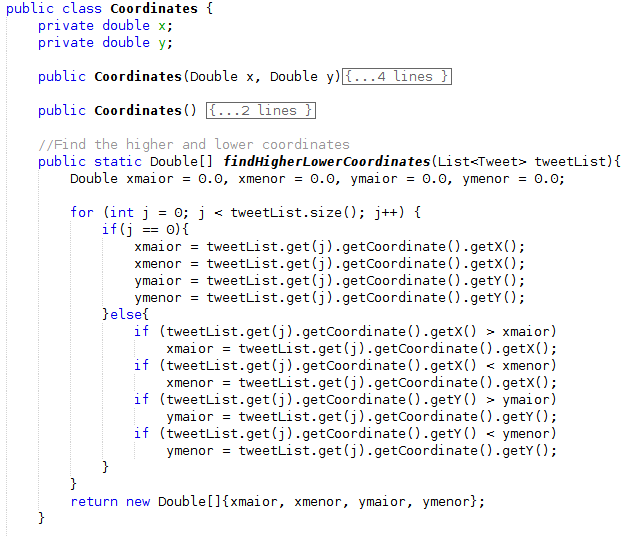
Illustration 8: Previous Code

Illustration 9: Current Code

The Previous Code in Illustration 5 was in the main class, and to clean out the main class and make it cleaner and more organized, so I decided to move the piece of code that insert the JSONTweets in a list to the Parser Class, the method returns a List of Tweets, this method is called in the Main Class.

Illustration 10: Previous Code

Illustration 11: Current Code

The situation for Illustration 5 and 6, works on the same way for Illustration 7 and 8. To clean the main class of extra information and just keep what is needed, I moved the function that finds the Lower and Higher Coordinates to the Coordinates class, the method now returns an array of double with the coordinates, the bigger and smaller x and y.

Collections

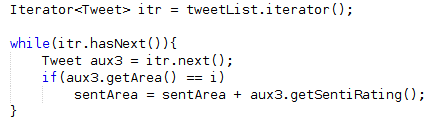
* List

Illustration 12: List

In the software a List was chosen to store the tweets since it's convenience to handle the data when adding and getting it. It can be added duplicated data and the elements could be accessed using a zero-base index, which is convenient when testing the software.

* Iterators

Illustration 13: Iterator

Illustration 14: Iterator

Iterator have been used in two different ways, explicitly (Illustration 11) and a implicitly (Illustration 10). On Illustration 10, the TweetList1 receives the next object of the tweetList each time through the loop. On Illustration 11, a Iterator itr is created and then used to handle the loop.

* Array

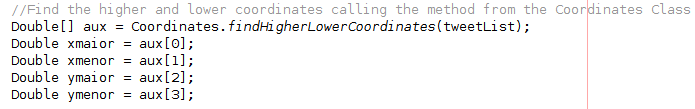
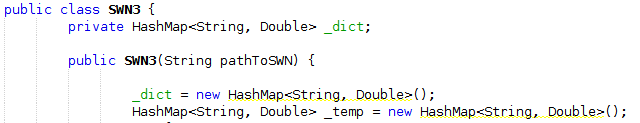
Illustration 15: Array

Illustration 16: Array

The array was chosen because it easy to handle when storing the same type of data, on both cases the Double type. On Illustration 12, the Array aux receives the value returned by the function findHigherLowerCoordinates of the class Coordinates, which has the return statement shown on Illustration 13.

* Hash Map

Illustration 17: HashMap

A HashMap is used in the SWN3 class, this class was not created by me, but taken from a website on the internet who provided the code. But the use of HashMap for the situation can be explained by it's use, which stores a big amount of data and HashMap is one of the best Collections which for basic operations such as get(), remain constant for larger sets of data. Using the word of the Dictionary as a key of the HashMap is really useful and smart to find the sentiment of each value using only the get() method.

Libraries

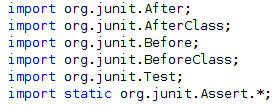
* JSON Library

Illustration 18: JSON Library

JSON or JavaScript Object Notation is a lightweight text-based open standard designed for human-readable data interchange. Conventions used by JSON are known to programmers which include C, C++, Java, Python, Perl, etc. (Tutorialspoint.com, 2014)

JSON is primarily used to transfer data between web and servers, is easy to read, lightweight text based format and language independent.

* JUnit

Illustration 19: JUnit Library

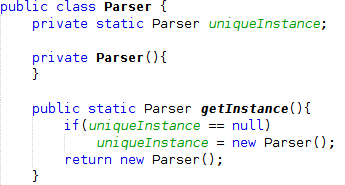
JUnit is a unit testing framework that aims to increase the programmer productivity ans stability of program code since the programmer can code and test alongside without having to finish the whole program so then he can test it, reducing time on debugging.

Design Patterns

* Iterators

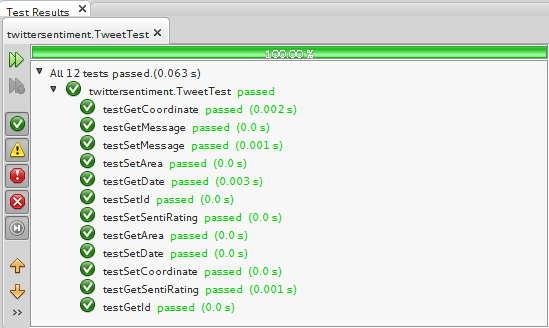
It can be seen on Illustration 13 and 14.

* Singleton

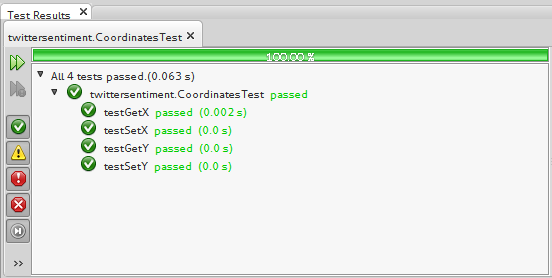
Illustration 20: Design Pattern: Singleton

Unit Tests

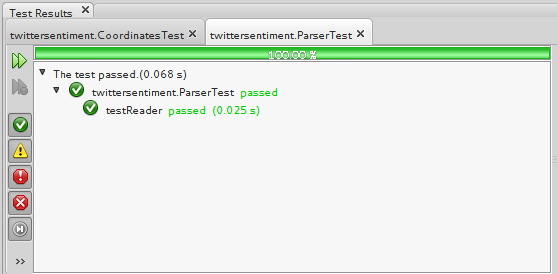
* Tweet Class

Illustration 21: JUnit Tweet Test

* Coordinates Class

Illustration 22: JUnit Coordinates Test

* Parser Class

Illustration 23: JUnit Parser Test

Reflection

Working on this assignment has been a challenge and instructive. The JSON library has proved to be useful, and I have learnt how to use it properly, understanding how it works and the relevance of it when it comes to transferring data via web. Re-factoring was quite difficult and tricky, realizing where it could have been applied was a challenge, but I could apply the knowledge acquired in class on this assignment. Collections has proved to be useful and practical, making the process of coding easier since the most important and basic methods are ready to use, it may take awhile to understand how it works and how to manipulate it, but after I got used to it, it was easy and direct.

In the program two external libraries were used: the JSON and the JUnit library. The JSON library was required to manipulate the JSON file, objects and arrays, it was downloaded from the website provided in Blackboard and worked properly if using an older version, the new version of it appeared to have problems while working older Java Versions, like the 7th version. The JUnit library was required to execute test while programming, using the JUnit provided with NetBeans proved to be handy, easy and clear to understand.

In conclusion, the assignment has proved to be instructive and a great opportunity to apply the knowledge acquired in lectures in a practical way and developing it even more.

Bibliography

Tutorialspoint.com, (2014). *JSON Overview*. [online] Available at: http://www.tutorialspoint.com/json/json\_overview.htm [Accessed 5 Dec. 2014].

Tutorialspoint.com, (2014). *UML Tutorial*. [online] Available at: http://www.tutorialspoint.com/uml/index.htm [Accessed 5 Dec. 2014].