

Architecture Design Genome Explorer

Programming Life: Context Project T12806

19-6-2015

Group 2

Supervisors: Dr. A. Bacchelli & Dr. T. Abeel

Gerben Oolbekkink (gjwoolbekkink, 4223896)



Jasper Boot (jboot, 1516272)



Jasper Nieuwdorp (jnieuwdorp, 4215796)



Jim Hommes (jimhommes, 4306090)



René Vennik (rvennik, 4102959)



Table of contents

1. Introduction	p. 2
2. Design goals	p. 2
3. Software architecture views	p. 2
4. Hardware/software mapping	p. 2
5. System design	p. 3
5.1 Geex Models.....	p. 3
5.2 Geex Core.....	p. 3
5.3 Geex Services.....	p. 3
5.4 Geex Workspace.....	p. 3
5.5 Geex Effects.....	p. 4
5.6 Geex Logger.....	p. 4
6. Glossary.....	p. 4

1. Introduction

This document should provide guidance alongside our project build in the context project programming life. This document will be updated throughout the project and will explain all the systems and their subsystems to give an overview of the software engineering aspects. For a quick view of the project also visit <http://geex.hup.blue/>, <http://sonar.hup.blue/> and <https://github.com/Vennik/contextproject/wiki>.

2. Design goals

- **Availability**
Always a working product at master. The client should be able to download a build at any time from the release page so the client can test it and possible changes can be made quick.
- **Interactivity**
We want an efficient to use, quickly responding program to visualise multiple genome strands and see their underlying relations and differences (mutations) in an interactive way.
- **Scalability**
We would like to be able compare around 500 strands of DNA on a single workstation. But 6000 strains should also be usefull in the future; this is a matter of scaling the hardware with the data.
- **Modifiability**
The program should be easily modifiable since extra filter could be added after we are done with it.
- **Reusability**
The components of the program should be made with reusability in mind. Abeel indicated that he wants to use the best parts of each program, so it should not be hard to reuse parts of the project.

3. Software architecture views

The system preloads data stored in textual files. Text files for the edges and nodes of the DNA strands and Newick tree format for the phylogenetic tree. These files are stored locally.

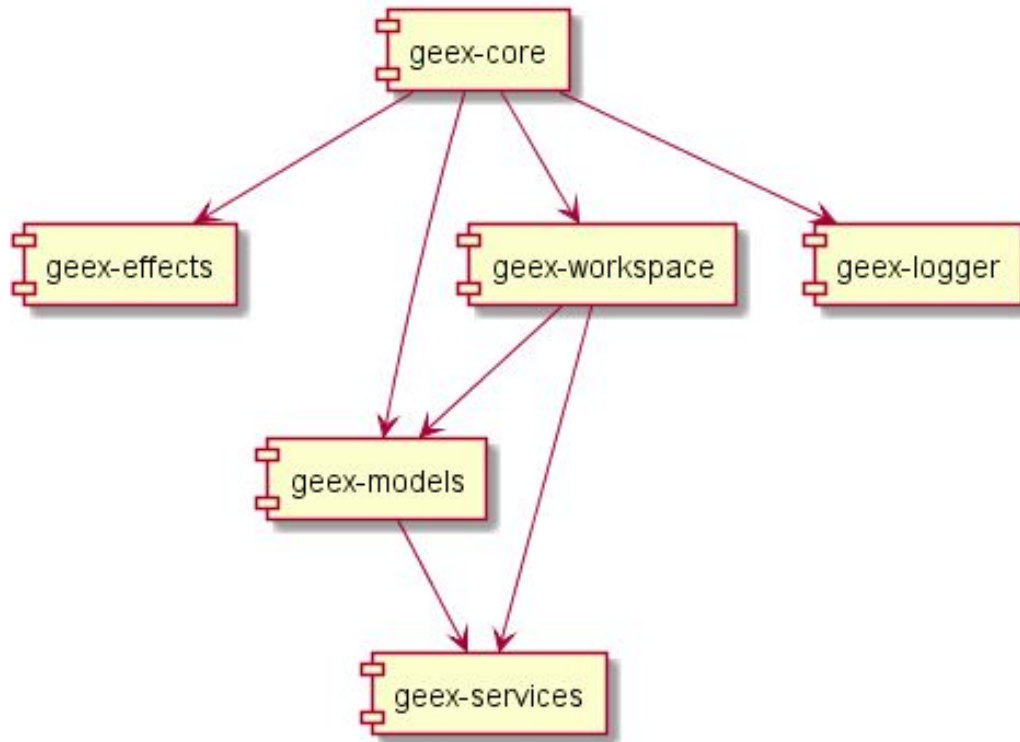
There will be a controller to select different options and a viewer that will display the graphs. There is only one interface to this program.

4. Hardware/software mapping

The software should run on one single workstation, of which the specifications are unknown at this moment. We suspect an Unix based operating system with at least 2GB of RAM and a current processor. We will use Java so it can run on any operating system. A single researcher

will use a single workstation so data persistency should not be an issue, since files are loaded only once and not edited.

5. System design



There are currently six modules in the geex project. Dependencies between modules must be unidirectional, preventing cycles between modules.

5.1 Geex Models

Geex models is the module containing all models and parsers. It contains models for the Newick, graph, annotation data and resistance data. Each type of model has its own parser.

5.2 Geex Core

Geex core is the main entry point of the application. It is responsible for showing and building the views and routing the views. This module also contains the App class that runs the entire application.

5.3 Geex Services

Geex services contains helper classes for loading data asynchronously. Any class implementing `Loadable` can be loaded in a separate thread by `LoadService`.

5.4 Geex Workspace

Geex workspace is the module which is responsible for managing the current workspace. It also includes a small database class for saving previous workspaces. The workspace provides `ReadOnlyObjectProperty` objects for each type of model it provides. This way any controller can bind to the change of this property and will be notified when the workspace is loaded.

5.5 Geex Effects

Geex effects is a helper module which is currently only used for creating a zoom effect on nodes in the graph.

5.6 Geex Logger

Geex logger is a small module which is responsible for logging data. It includes a `Log` class which is able to log messages to several different Loggers, there are currently two loggers. The first logger is the `StdOutLogger`, which is part of Geex Logger, it writes data to `System.out` when the software is run with the `-Ddebug=true` jvm option. The second logger is `MessageController`, which displays info and warning messages on the bottom of the screen.

6. Glossary

- Genome: the haploid set of chromosomes in an organism
- Strands: a reading of the genome expressed as a list of A, G, C and T's.
- Mutation: an alteration of the DNA, this can be deletion, insertion, translocation & inversion
- Newick tree format: a file format to store trees in a textual representation.
- Phylogenetic tree: diagram showing the inferred evolutionary relationships among species.