

Architecture Design

Programming Life
Team Dynamites & Butterflies

Table of Contents

Introduction	2
Design Goals	2
Usability.....	2
Maintainability.....	2
Scalability.....	2
Software Architecture Views	3
Subsystem Decomposition.....	3
Hardware/Software Mapping.....	3
Persistent Data Management.....	3
Concurrency.....	3
Glossary	3

Introduction

This document describes the architectural design of the multi-genome interactive visualizer. The product was created for the Programming Life context project and serves as an extension of current genomen viewing tools.

Design goals

In the design goals section we explain our design goals. These are goals that stay in the back of our head during development of the entire product. These design goals are there to ensure good coding and good teamwork.

Usability

The goal of our product is to be useful. For the customers we want to make sure the application feels intuitive and modern. Our goal to ensure that happens is by the constant feedback loop we receive by the use of the SCRUM principle. With constant usability feedback and the must-have of a working application at the end of each sprint we can ensure this happens.

Maintainability

Our goal is to ensure that the product is maintainable. Due to the open-source nature of our project we encourage others to aid in updating features. To make sure they can quickly aid us and don't waste too much time on understanding our code we want to make sure it is maintainable. A lot of comments, clear variable names are all ways of making sure this happen. We try to ensure this by reviewing each other's code after a feature is made.

Scalability

Genomes come in all shapes and sizes. The initial runs and products will be built for genomes with 'only' 1000 base pairs. The idea is that at the end of the product cycle our application will successfully parse and load genomes up to 1 million nodes. With that in the back of our mind we want to ensure our application scales well.

Software architecture views

Subsystem decomposition

Hardware/software mapping (mapping of sub-systems to processes and computers, communication between computers)

Persistent data management (file/ database, database design)

Concurrency (processes, shared resources, communication between processes, deadlocks prevention)

Glossary