



UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 301 Main Project

Architecture Design

ThinkTech

Group Members:

Lelethu Zazaza 13028023

Git repository link:

<https://github.com/COS301-ThinkTech>

Version 0.1

May 29, 2015

Contents

1 Architecture design

1.1 Overview

The Model-View-Controller allows for dividing the application in three components, making the problem domain independent from the user interface. A controller has the ability to send commands to the model to update the model's state (e.g., adding a component to the flowchart). It can also send commands to the associated view to change the view's presentation of the model (e.g. maximizing the flowchart window) A model stores data that is retrieved by the controller and displayed in the view. A view requests information from the model that it uses to generate an output representation to the user.

1.2 Architectural tactics addressing quality requirements

This section discusses the architectural tactics which are used to concretely address the quality requirements for the Flowchart and Simulation Tool application.

1.2.1 Contracts based development

Contracts allow the application to adhere to the following quality requirements: Testability and maintainability. The contracts will be enforced for services with pre and post-conditions which are assessed in unit tests, and data structure constraints enforced through data structure validation.

1.2.2 Dependency Injection

Dependency injection allows for the application to adhere to the following quality requirements: Flexibility, deployability, testability.

1.2.3 Minimize technology suite

In order to improve maintainability the software architecture will minimize the number of programming languages used.

1.3 Architectural components

This section discusses the architectural components and technologies used to address the architectural responsibilities and the architectural tactics chosen to address the quality requirements

1.3.1 Java

Java is chosen as a single programming language used for the application in order to implement the tactic of minimizing the technology suite. Using a single programming language reduces complexity and improves maintainability.

1.4 Development architecture

The development architecture is the architecture designed to support qualities in the development process itself. This includes qualities like reliability.

1.4.1 Version control

The developer of the project make use of a git repository. Any new feature or bug should be developed in a new branch which is only merged into the trunk once the feature or passes its unit tests.

1.5 IDE

Each developer has the freedom to choose whichever IDE they prefer.

1.6 Unit testing

Unit testing will be done using JUnit.

1.7 Bug tracking

The projects will use GitHub's issue tracker for bug tracking.