LATEX EDITOR

DESIGN RECOVERY AND QUALITY ASSESSMENT REPORT

VERSION <1.0>

Βασιλειάδης Μιλτιάδης 2944

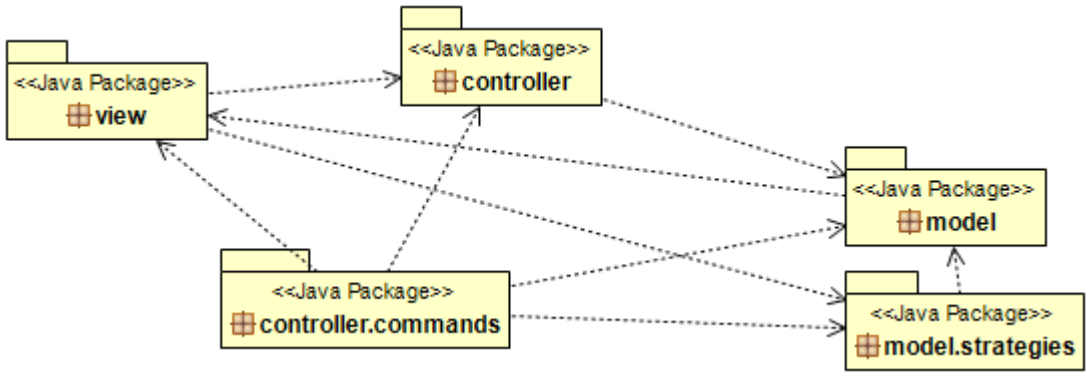
Θωμά Αθανάσιος 2979

INTRODUCTION

The goal of this project is to reengineer a Java application. At a glance, the objective of this project is to develop a simple Latex editor for inexperienced Latex users. Latex is a well known high quality document preparation markup language. It provides a large variety of styles and commands that enable advanced document formatting. Typically, a Latex document is compiled with a tool like MikTex, Lyx, etc. to produce a respective formatted document in pdf, ps, etc. Formatting documents with Latex is like a programming process as it involves the proper usage of Latex commands which are embedded in the document contents. The goal of the Latex editor is to facilitate the usage of Latex commands for the preparation of Latex documents. One of the prominent features that distinguishes the LatexEditor from other similar applications is its multi-strategy version tracking functionalities that enable undo and redo actions.

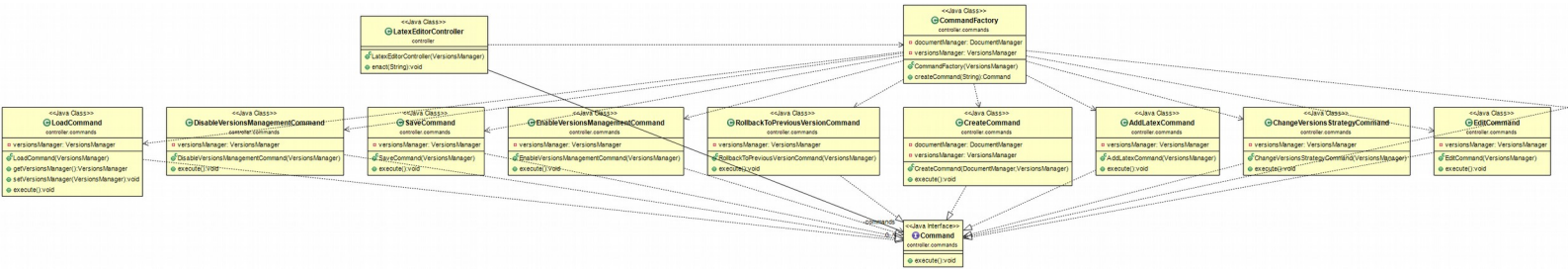
DESIGN RECOVERY

ARCHITECTURE

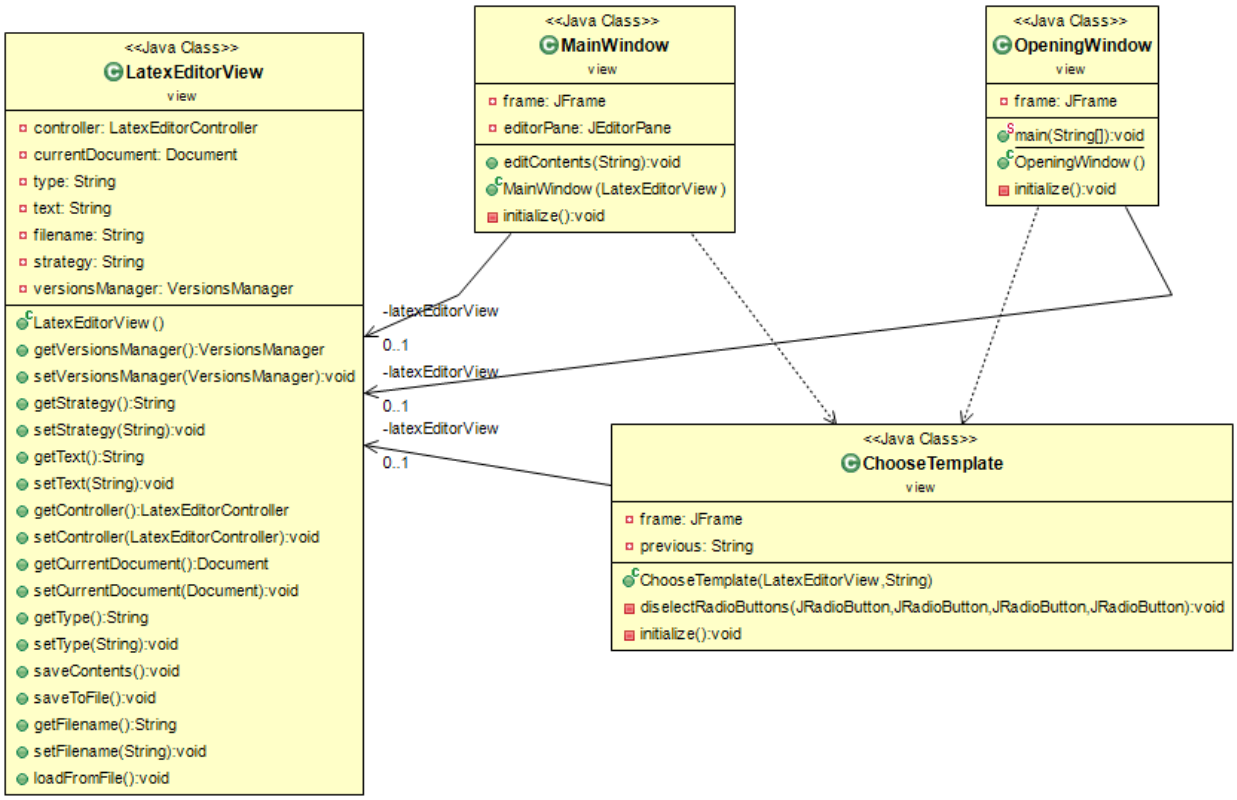


DETAILED DESIGN

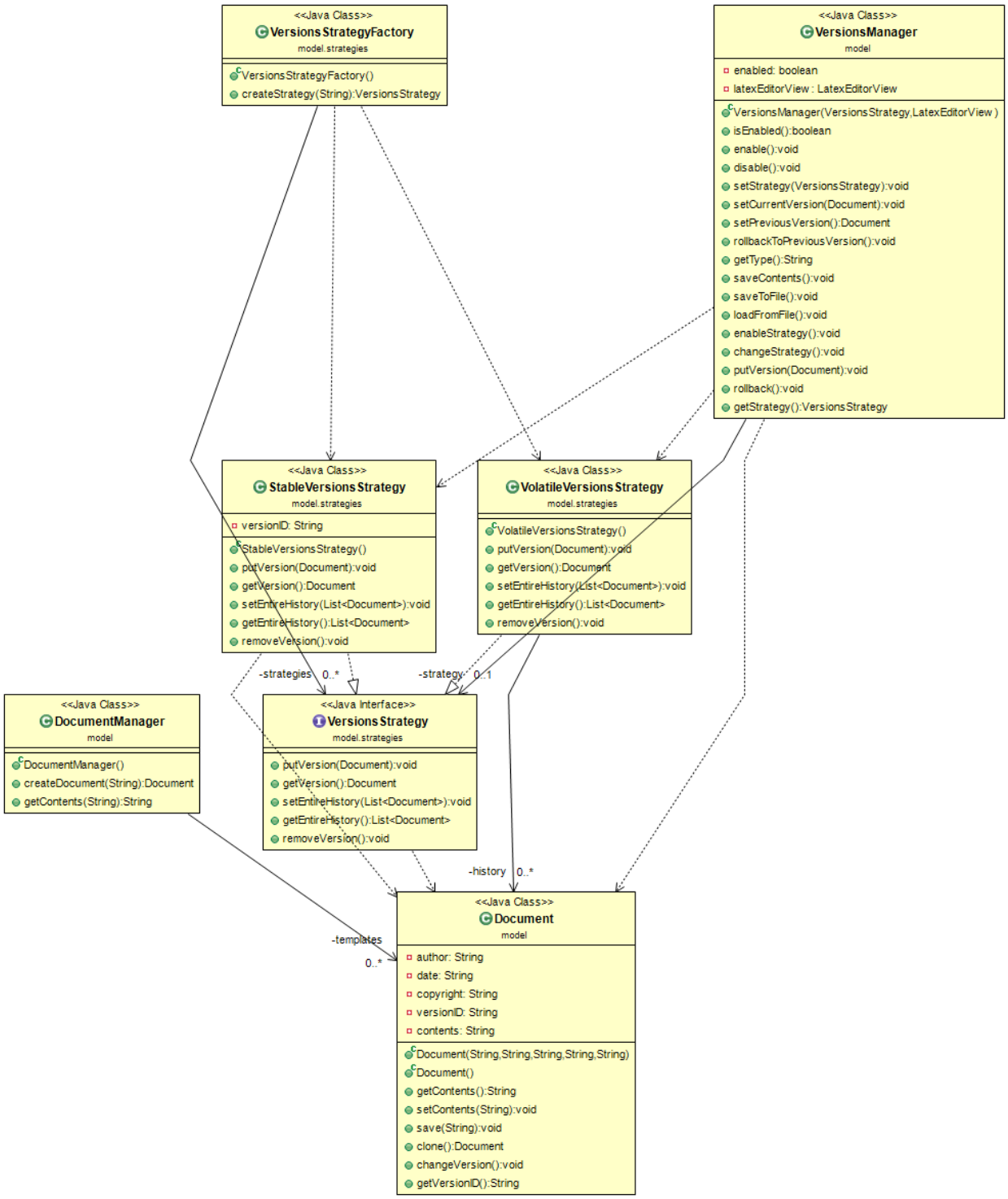
- controller package



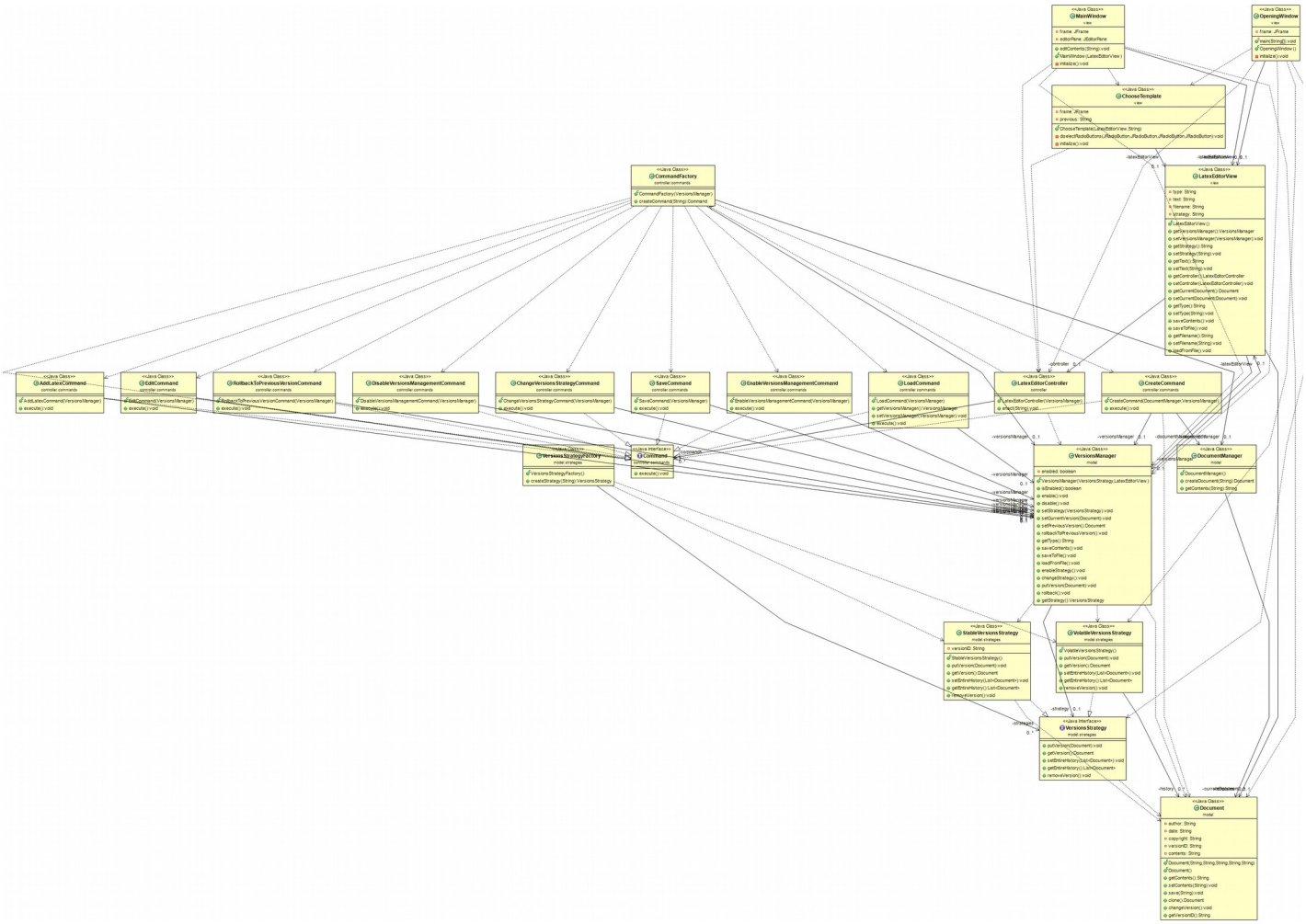
- view package



- model package



- overall graph



IMPLEMENTATION

QUALITY ASSESSMENT

Problematic Methods:

- Document.clone() ~ creates shallow instead of deep copy.
- DocumentManager.constructor() ~ Document constructor is not used correctly, instead the contents of each document template is filled through setter method.
- VersionsManager.rollbackToPreviousVersion() ~ does nothing.
- StableVersionsStrategy.putVersion() ~ calls Document.save() method instead of implementing its own mechanism to write to the disk.
- LoadCommand ~ has setter and getter methods. Dead code.
- ChooseTemplate.initialize() ~ has duplicate code inside.
- LatexEditorView ~ has setter and getter methods. Dead code.

Classes with many responsibilities:

~Model Package

- Document: also has save() method.
- DocumentManager: getContents() method should be in Document class.
- VersionsManager: is a God class. It is not only responsible for the versioning system. This is the class that all the command classes send their job to. It also handles some of the communication with the GUI.

~View Package

- MainWindow: method editContents() should be in AddLatexCommand class.
- LatexEditorView: this is the “controller” class of the GUI package. Responsibilities such as loadFromFile() and saveContents() are shoved in here. Has some duplicate code.

Classes with very few responsibilities:

~Controller Package

- LatexEditorController: this class should be the one communicating with the GUI.

~Controller.Commands Package

- AddLatexCommand: the job of this class is implemented elsewhere. The whole class is a duplicate of EditCommand.
- EditCommand: doesn't do anything, just calls the VersionsManager.
- ChangeVersionsStrategyCommand: doesn't do anything, just passes calls VersionsManager.
- LoadCommand: doesn't do anything, just calls the VersionsManager.
- RollbackToPreviousVersionCommand: doesn't do anything, just calls the VersionsManager.
- SaveCommand: doesn't do anything, just calls the VersionsManager.