

MTAT.03.083 – Systems Modeling

Assignment 3: Code Generation (10 points, due on 22.10.2013 at 10 am)

Purpose

The purpose of this homework is to help you to develop the ability to link UML class models with code, and to produce a simple running application starting from a UML class model.

Task

Design a UML class model of PERT charts, ready for implementation. Use this class model to develop a command-line tool that takes as input a file in “.pert” format (described below) and that produces a Graphviz file (“.dot” format) or an image file (“.png” format), containing a representation of the PERT chart with the critical path highlighted in bold style or using red color.

The input “.pert” file is formatted as a comma-separated-value file (using commas as the delimiter) with each line formatted as follows:

```
task, duration,precedingTask1, ... precedingTaskN
```

where field *task* is an identifier of a task composed of alphanumeric characters only, field *duration* is an integer, and precedingTask1, ... precedingTaskN is a sequence of comma-separated identifiers of tasks that immediately precede taskID in the PERT chart. The PERT charts we deal with in this assignment do not have a notion of “sub-task” (i.e. they are flat).

A correct input PERT chart should satisfy the following criteria:

1. Every node has at least one predecessor or one successor
2. There is a single node without predecessors (i.e. start node)
3. There is a single node without successors (i.e. end node)
4. The graph of tasks is acyclic

The output (PERT chart with critical path) should consist of nodes representing tasks, and edges encoding precedence dependencies. The name of the task and the duration must appear inside each node. A task or edge must be in bold or red if it is in the critical path. A sample output is the given in Figure 1.

Deliverable

The deliverables are a MagicDraw project file and a Java command-line application that we can invoke as follows: “java -jar myApplication.jar NameOfPERTFile.pert”. The application should produce either an error message stating that the input file is incorrect (if this is the case), or a file “NameOfPERTFile.dot” or “NameOfPERTFile.gif” in Graphviz or Gif format.

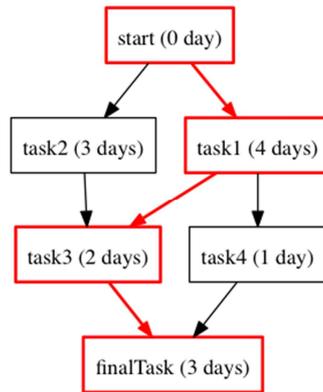


Figure 1. Sample Output

Non-functional requirements

- The UML class model should be designed using MagicDraw;
- The command-line tool should be developed in Java – use MagicDraw code generation and round-trip engineering features as much as possible;
- Right from the start of the assignment, every team must maintain their models and code in a publicly-accessible version control system. Teams should use one of the following services: Assembla, Google code, Bitbucket or GitHub. Every team member must use his/her own username when making commits to the version control system – making commits using someone else’s username is forbidden. Our preference is that you use an SVN repository in Assembla, or Google code as second choice;

Practical Notes:

- The task is to be completed in pairs;
- One member per team should log in and submit the assignment using the link “Submit” on the course webpage. Please specify in the “Comments” field in the submission form the members of the group. The submission should take the form of a “zip” file containing the MagicDraw project file (“.mdzip”), the compiled “.jar” file, and a README.txt file containing the link to the public repository.

Grading criteria

- UML class model (correctness and completeness): 2 points
- Alignment between UML model and code: 2 points
- Use of version control system: 1 point
- Ability of the tool to parse and check the input PERT chart file: 2 points
- Ability of the tool to calculate critical path: 2 points
- Ability of the tool to output the PERT chart and critical path as a DOT or a PNG file: 1 point