

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
ECOLE POLYTECHNIQUE DE LOUVAIN



LINFO2252
SOFTWARE MAINTENANCE AND EVOLUTION

TESTING THE DESIGN OF A
FEATURE-BASED
CONTEXT-ORIENTED SYSTEM WITH
MUTATION TESTING

LAB SESSION 1 (EXTENDED)

Professor: Kim Mens
(kim.mens@uclouvain.be)

Teaching assistants: Julien Liénard and Pierre Martou
({julien.lienard,pierre.martou}@uclouvain.be)

October 11, 2023

1 Approach of the lab sessions

During the previous lab sessions, you designed a feature-based context-oriented systems using feature, mapping and context models. When you designed these systems, you could maybe experience how tedious it could be to verify the correctness of the models as well as verify all possible behaviours of your system.

For example, at the design level, each feature should be activated by at least one valid configuration of the system, each valid configuration of the context model should be mapped to a valid configuration of the feature model, etc.

In this lab session, we will ask you to analyse the modelling of your smart *messaging* system in order to correct and/or improve it. For that, we will provide you with a testing tool that will challenge the design of your models.

2 Testing your design

We divide this lab session in three parts. We first introduce the theory of the testing tool, then how to use it, and finally what we expect you to do with it.

Tool process

The provided test tool uses mutation testing to generate mutants of your context-feature models. A mutation can be defined as a small change in the model characteristics. For instance, we can create a mutant by changing an *OR* relationship into an *AND* relationship or a *MANDATORY* feature into an optional one. Traditionally, mutation testing is used to test the efficiency of a test suite by introducing a small bug in the program and observing whether the suite catches the error or not. In the case of a context-feature model, we will use the mutations as alternative designs that could be more appropriate. With all these mutants, we generate a list of yes/no questions. Depending on the answers, the system can then propose recommendations to improve your model.

For more information on this recommendation tool, please refer to the related master thesis (<https://dial.uclouvain.be/memoire/ucl/object/thesis:40644>).

Using the tool

You can find the executable under the folder *Testing tool* on Moodle. It is made to run on Ubuntu 18.04 or higher, or equivalent. If your group does not have Ubuntu, please either use a computer from the Intel room, or use a virtual machine on your own device.

In order to feed the tool with your models, you will need to write three text files named *features.txt*, *contexts.txt* and *mapping.txt*, containing respectively your feature, context and mapping models.

Your feature (context) model file must contain one line for each relationship of your feature (context) model. Each line contains the parent feature (context) of the relationship to the left, a keyword representing the type of the relationship, and the child features (contexts) to the right (the relationships supported are: *MANDATORY*, *OPTIONAL*, *OR*, *ALTERNATIVE*). The features and keyword are separated by a forward slash: *'/'*. The features are separated by a hyphen: *'-'*. A line where F is the parent feature and F1 and F2 are in an *ALTERNATIVE* relationship looks like: *"F/Alternative/F1-F2"*

The mapping model file has one line by mapping relationship. Each line contains the contexts on the left and the features triggered by these contexts on the right, separated by the keyword *ACTIVATES*. Each context, keyword and feature are separated by a hyphen: *'-'*. If C activates F, then the line is: *"C-ACTIVATES-F"*.

Be consistent with the names of your features/contexts throughout **all the files**. They must be a perfect match for the tool to recognise them. Avoid spaces and special characters in the names of your features/contexts.

You can find an example of these three files on Moodle, next to the file containing the tool. Would you encounter errors that are not documented by the tool, please call us, so we can improve it with new error feedbacks.

Once you downloaded the tool and wrote your three text files, please follow these steps in order to generate a set of scenarios:

1. Place the file *recommendation_system* (found on Moodle) and your files *features.txt*, *contexts.txt* and *mapping.txt* in the same empty folder.
2. Open a Linux command line in the folder.
3. Type *chmod +x recommendation_system* (once is enough).
4. Type *./recommendation_system* to run the tool. The computation time can take some time, depending on the resources allocated to your virtual machine.

To answer to these questions, you will have to imagine features or contexts enabled at the same time or being exclusive. Try imagining an application or a user: is it normal that these contexts can be combined ? Is it normal that these two features are never both available to the user ? Once you answered all the questions asked by the program, you will receive a mutation score rating the correctness of your model and possibly recommendations to improve your models.

When using the recommendation system and correcting your design, please take note of your mutation score, of the changes you make (e.g. if you change a relationship from *or* to *alternative*). Even if you do not follow a recommendation but find an error thanks to the questions of the tool, that could be useful feedback for us. The use of this tool will not be graded in a Mission. However, in one to two weeks, you will be asked to answer a questionnaire on how you used the tool and your thoughts on it. The answers will be entirely anonymous and will be used for research purpose only, in the context of a current master thesis.

Please also note that later in the year, you will have to use another testing tool whose input is the same format (using text files), so do not lose your files until then.