Software Product Line Engineering

Summer 2019, Assignment 2

Bauhaus-Universität Weimar

Prof. Dr.-Ing. Norbert Siegmund
Stefan Mühlbauer, M. Sc.

Assignment issued:	Friday, 26 April, 2019
Submission due:	Thursday, 9 May, 2019, 23:59 CEST
Presentantion:	Friday, 10 May, 2019

Task 1: Domain and Application Engineering (6 marks)

(a) Complete the following overview about "domain and application engineering". (3 marks)

Domain Engineering

Application Engineering

(b) Explain the differences between the classical software life cycle (e.g., waterfall model) to domain and application engineering. (4 marks)

Task 2: Design Patterns (8 points)

In the lecture, different design patterns were presented. For the following four design patterns, briefly describe their basis purpose, and how they could be used in the chat product line from the previous assignment. (2 points each)

(a) Observer Pattern

(b) Template Method Pattern

(c) Strategy Pattern

(d) Decorator Pattern

Task 3: Feature Model Extraction (DE students/bonus; 8 marks)

The application *GNU xz utils*[1] is a compression software commonly used in operating systems of the Unix family. The application is used via the command-line and provides a number of different configuration options (features) that can be configured upon run. The application provides two modes, *compression* and *decompression*.

From the manual page, extract the features that correspond to the compression mode (2 points), and construct a feature model (6 points) with FeatureIDE.

Note: The application contains numerical features, i.e., options that are not binary (selectable), but options that are assigned numerical values. FeatureIDE does not allow modeling numerical features though, so it is sufficient to treat them as mandatory features and report the valid value range.

Hint: The application's feature model has cross-tree constraints and numerical features are involved in it.

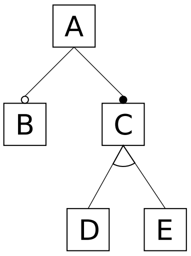
[1] The project site is `https://tukaani.org/xz/`, a manual page can be found at `https://www.freebsd.org/cgi/man.cgi?query=xz`

Task 4: Feature Modeling (16 marks)

Feature models are used to model and represent variability.

a) Why are feature models represented typically as trees and not as list, graphs or logical expressions? (2 marks)

b) Given the product line A with the following feature diagram.



(i) Determine *all* valid variants of product line A. (1 mark)

(ii) Design and describe a simple algorithm which calculates all possible variants of some product line (without source code). (2 marks)

(iii) Provide a logical expression for the variability model of A. (1 mark)

(iv) Which consequences has the implication $E \implies B$ to the logical expression. Develop the resulting feature diagram. Is that easily possible? (2 marks)

c) Create a feature diagram for the chat product line (see assignment 01). Model the features: color, authentication, history and encryption (encryption should provide the possibility to pick neither of them, each one individually or both together) as optional features. (4 marks)

d) Extend this feature model with the input features GUI and CLI, which are mutually exclusive. (2 marks)

e) Create the final feature model with FeatureIDE[2]. (2 marks)

Submit your answers (PDF) and implementation (jar + source code) as an archive with name and matriculation number until 9 May, 23:59 CEST to `stefan.muehlbauer@uni-weimar.de`.

[2] https://featureide.github.io