David Tschirschwitz Digital Enginnering 119383

Task 1: Theory of Feature-Orientation

a)

feature: a functionality that the software wants to provide, that in case of sple is also exchangeable

collaboration: classes that together implement some functionality/feature

mixin: are abstract classes that inherit from their superclass but they can be applied to this superclasses, this is not possible with the normal behaviour. They can be considered as a static counterpart towards the Decorator pattern (Apel et.al.).

jampack: composing classes with superimposition

class: is maybe a part of many collaborations and therefore playing multiple roles and is also a concluded piece of code

b) Jampack: Assignment of generated code to roles after generation no longer available

Mixin: Slower by Indirection with Method Calls and Method Overhead

(lecture slide tu-darmstadt https://www.es.tu-darmstadt.de/fileadmin/lehre/spl/ss17/05b-SPLE-SS-VP.pdf)

c) FOP allows the subclasses to inherit behavior to there superclasses. Features are not scattered around classes they can be combined into moduls.

d) AHEAD solves this issue by the flexible refinement of the basic class, this changes are possible on very fine granularity level, hence this allows us to make changes in the modularity that were not planned previously.

e) Yes especially the feature logging has a bad feature traceability. This makes refactoring or changes difficult, since it is easy to oversee parts of the code.

Task 2: Components and Features

a) The library scaling problem describes the difficulty of scaling reuse libraries in both component size and feature variation. (Biggerstaff T, 1994) The issue is described as a trade of between the amount of features provided by a component and the performance at that component. Both cannot be achieved at the same time.

b) Fine granularity for the libraries and smart tooling to combine libraries, hence a minimum of libraries is linked.