Software Engineering

WS 2022/23, Sheet 01

Prof. Dr. Sven Apel
Yannick Lehmen, Maurice Vincon

Handout: 31.10.2022

Task 1

a) What do you conceive of a configurable software system? Give a short description or definition.

b) Name examples of programs that are not configurable software systems. Why aren't they? Why wouldn't one develop these as such?

c) What other domains (inside and outside of IT context) are suited for configurable system technologies?

d) Check whether or not the following software systems are product lines. Explain your decision!

- BusyBox: `www.busybox.net`
- Candy Crush: `https://www.king.com/de/game/candycrush`
- Android: `www.android.com`

Solution

a) A configurable software system comprises a set of *system variants* that share a *common set of functionalities* tailored for one *specific domain* with the goal of *reusing common software artifacts*.

Some characteristics of a configurable software system are:

- A configurable system can be tailored to a specific use-case.
- Configurable systems can differ in size, configuration, cost, ...
- Developing new variants costs less effort (compared to developing each variant individually) because of reuse.
- A configurable system offers specific functionalities (features).
- ...

b) Some examples of programs one would not develop as a configurable software system are:

- video games
- admin scripts
- single-vehicle software (Ariane)

A program does not benefit from being developed as a configurable software system if:

- it cannot be tailored to specific needs.
- it fulfills only a single use-case.

c) Examples for other domains that are suited for configurable system technologies:

- planes, cars
- PCs, Laptops
- printers
- furniture
- jewelry
- websites
- ...

d) **BusyBox** : Yes, because one can customize the binary at the time of compilation (mainly to save space).

Candy Crush : No, because all functionalities are always present.

Android : Yes, because different manufacturers can tailor andriod to work with their specific hardware.

Task 2

Imagine you work at a company that wants to develop a new chat application. Since this project was given to you, you want to implement it as a configurable software system.

Analyze the domain of such a chat application. Based on your experience, which functionalities could users want? Which of these functionalities may only be desired by some users?

Solution

Must-have features:

- contact list
- history
- group chats
- spell checking
- data transfers
- profile pictures
- themes
- ...

May-have features:

- protocols/plugins
- games
- encryption
- colors/formatting
- audio and video chats
- ...

Task 3

What are some challenges when it comes to designing a configurable software system? Explain them.

Solution

Combinatorial Explosion The more features you have the more system variants are possible. With only 33 optional and independent features it is possible to create one tailor-made variant for each human on earth.

Sensibility and Consistency It is hard to predict whether all combinations of features are sensible. For example, a selecting two different compression algorithms at the same time would not make much sense.

Correctness It is impossible to check the correctness of all possible combinations of features in a large system (because of the combinatorial explosion).

Performance There can be interactions between certain features that influence the performance of the program significantly. However, finding all such combinations is next to impossible (again, because of the combinatorial explosion).

Maintainability and Program Comprehension Some widely used implementation techniques for variability, like preprocessors, can make programs very hard to understand when used carelessly (cf. slides 01/37-38). This can have a negative impact on the maintainability of a configurable software system.