The critiques for a feature-oriented reuse method with domain-specific reference architectures

Name: Yachen Zhao

This article talks about the feature-oriented reuse method can support to design the software architecture which could be reused later by developers. And which process does feature oriented reuse method has. Here are some opinions about this method.

I agree that feature-oriented reuse method can somehow decrease the development time. Because developers can reuse some resources from the existing system to create a new system, probably some features, code, architecture styles. It helps to reduce some works to do. But on the other hand, if they want to reuse something from existing systems, they have to do a lot of research to find the resource that can be used. And also, once they find something to use, they have to do some analysis to figure out if this can be used or which one is better to be used in a system. This would take a lot of time. Also, once they choose something to reuse, for an example, like code. They should understand the code for using it into the new system. They also need to change some parts of the code, probably. This would consume time, too. So Feature-Oriented Reuse Method in some way makes the development both easier and harder.

I think the reuse of features isn’t popular nowadays because the reuse of features is a little harder, because to develop features many people would be involved, such as, users developers and analysts of system. This is an unstable cause for reuse of features. Because everyone has their own opinions about the features of a system. Users would focus more on the function of a feature, like what they can do using this feature. Developers would focus more on the technologies they are using. And system analysts would focus more on the whole system instead of one feature, like the architecture style and stuff. In this case, they would have different thoughts about reuse a feature or not. For an example, for users a feature is reusable. But analysts and developers think they should use another implementation technology or another architecture style for a whole system. Then this feature can not be reused, even it has the requirements for all users.

In domain analysis and feature modeling, the first step is feature identification. In this step, developers would get advices from domain experts. I think the analysis result might not be very precise, because different expert has different opinions. After feature identification, the feature classification would be done. I think some features might not be easily classified. They probably have overlaps. For an example, for a chat application, it should be a friend feature and a chat feature. For the friend feature, users can add or delete friends something like that. In the chat feature, the friend feature would be used too. Because it needs to show a friend list for chatting. The chat feature would be classified, but it include part of the friend feature inside.