

Research Problems for CS612

Alexander Egyed, Draft 2/9/1999

This is primarily for all those who don't have a term project. However, even if you have one, you might still find the one or other item here worth investigating.

Architectural Life-Cycle

Go through a full cycle of architectural synthesis, analysis and view reconciliation:

A. Architectural Synthesis

1. Model the Customer 1st Bank/Telecommunications problem in UML and/or other ADL(s) (not using the C2-UML technique). Use Scenarios, Classes, State Diagrams, etc.
2. Model your cs577 project architecture in C2 and other ADLs (UML to ADL).

B. Architectural Analysis

Identify Mismatches in UML/ADL views of the cs577 project architectures/ Customer 1st Bank project architectures/ Telecommunications architectures. Basically, what I mean with mismatches are cases of incompleteness between views. Also investigate things like:

- Is the C2/ADL view saying the same as the UML view and vice versa
- What can be seen in one view which cannot be seen (or expressed in the other) – What do they have in common? How do they differ? How do they address similar design issues
- Generalize your idea so that it might work for other projects of similar type.

C. Reconcile Architectural Views

Reconcile identified mismatches so that those mismatches are resolved. Describe the steps necessary and also suggest other options/alternatives.

Architectural Modeling

1. Analyzing different cs577 projects. Since those projects are not identical the goal of this analysis is different from just identifying mismatches. Possible goals could be:
 - In what aspects of the analysis and design stage are people doing well (without view integration) and in what aspects do people encounter difficulties while developing (and integrating) their models.
 - Were certain techniques of UML (or ADL(s) if above was done) primarily applied to certain aspects of the problem. Are there patterns recurring in the use of these techniques and in what way model elements are connected? Which techniques were used to decompose functionality? Or address performance issues, time dependencies, etc.
 - Also answer questions like: What did they do well/poor? Did they decompose their architecture in a functional way or object-oriented way? Were some views favored over others? Which techniques were most/least helpful in understanding the architecture? What is missing in the architecture to understand the problem/solution?
2. Analyzing different Customer 1st Bank/Telecommunications projects to identify different architectural solutions. Did they use C2 (oder other ADLs) in a similar manner. Are the system decompositions similar in that their components and connectors represent same/similar meaning (see also item 2).
3. Use C2/ADL to model (a-part-of) UML. This is basically the reverse of the C2-to-UML technique. This can and should be done 1) in model form using some formal technique and 2)

using examples (e.g. the cs577 or Customer 1st Bank examples). Also investigate multiple transformation issues. E.g. Model C2 architecture in UML and then convert UML architecture back to C2. Are the resulting C2 architectures equivalent? What is different? Also address issues such as which UML/ADL issues are not addressed in the other view(s) and vice versa.

It is not the goal of this project to just solve an example problem in yet another way. What we also would like to see is that you describe how and why you are doing something one way and not another. If you find yourself at a dead end then do not just scratch that out but instead reason why you landed there and how you plan to get out of there.

Furthermore, also try to be as general as possible (or generalize some of your ideas) so that they may be applicable for other projects as well.