0. Purpose

This document describes an example of a particular software engineering technique and accompanying tool that have been reported in selected conference proceedings but that have not been empirically evaluated. The purpose of the exercise is to come up with a feasible way of empirically evaluating the approach using the tools covered in the seminar.

1. Research to be evaluated

Software engineering processes produce series of related models describing the designed system from different viewpoints, each emphasizing a particular concern or standpoint of a stakeholder. As the models describe the same system, they share common model elements. We define a correspondence relationship as a relationship between model elements representing the same semantic concept. The correspondence relationship is needed for reasoning about models, their commonalities and differences, detecting and reconciling the possible inconsistencies, and for integrating models.

However, performing such tasks manually is tedious and error-prone. To provide tool support for exploiting correspondence relationship, the concept of UML set operations is presented. The UML set operations—variants of union, intersection and difference—can be used for merging and slicing of models, and for visually differencing them. As an example, consider the two example class diagrams shown in Figure 1. Intuitively, classes Contract, Client and Firm seem to be corresponding, as do the association between Contract and Client, and the generalization between Client and Firm. A union of the two models is shown on the right-hand side, with commonalities and differences visualized using different colors. Figure 2 shows another example of applying the operations on two UML class diagrams with corresponding classes and relationships highlighted in the merged diagram.

Figure 1. Example of comparing class diagrams\(^1\).

While it is usually assumed that such correspondence relationship is explicitly available in some form, such as repository identifiers, this assumption is not always valid. The models can originate from different sources like different development teams or third party developers, possibly using different modeling tools. Moreover, it is often desirable to keep individual models separate, especially at early stages of design, as modeling tools do not facilitate having several, possibly inconsistent versions of the same model element simultaneously. Finally, it is increasingly common for individual models to be produced automatically by tools. To address this shortcoming, an approach for automatic inference of inter-model correspondence is presented in by Selonen and Kettunen\(^2\).

---

\(^1\) For our purposes, inconsistencies between models can mean inconsistent states among corresponding elements (e.g. inconsistent multiplicities in association ends, abstract vs. concrete class), or inconsistencies between models (e.g. missing relationship, dependency vs. association, differently named elements).

\(^2\) Selonen P. and Kettunen M., "Metamodel-Based Inference of Inter-Model Correspondence", Proc. of CSMR'07, Amsterdam, Netherland, March 2007, pp.71-80. IEEE CS.
Examples exploiting the correspondence relationship include comparison of class diagrams, comparison of activity diagrams, and monitoring differences in evolving platform architecture. Other suggested usage scenarios include model merging, composing and decomposing models, and supporting model comprehension. The operations were originally designed and implemented to be a part of a larger model processing framework that would allow construction of series of general model transformations to perform different model processing tasks.

2. Evaluation criteria

A major problem for evaluating the research is that the research question, terminology and concepts, and stated contributions are not precisely defined but rather vague and ambiguous. To abbreviate, the term approach is used to denote the correspondence inference and the set operations. Two major claims and their rationale can be summarized as follows:

1. Using the approach leads to increased model comprehension. By having the tools to generate new transient, and partial views of the system, and by having the capability of reasoning about the commonalities and differences of models, the designer can improve her understanding and gain a more complete view of the system.

   **Question 1.** How can increased model comprehension be evaluated?

2. Using the approach improves integration and reconciliation of models. Integration of models is an important activity with incremental and iterative model development. Merging models by hand is, however, time consuming and tedious work. This is addressed by tool support to detect inconsistencies and perform the merging.

   **Question 2.** How can improved integration and reconciliation of models be evaluated?

In addition to these high-level claims, there are several smaller criteria to be considered:

- Evaluating the quality of the inferred correspondence relationship. How complete the derivation is, i.e. how big a portion of the implicit dependencies are recovered? How correct the correspondence derivation scheme is, i.e. how many false negatives are suggested?
- Evaluating the inconsistency detection. How many inconsistencies are detected?

3. Designing experiment

According to Wohlin et al (2000), experiment design should include the following constituents:

1. **Definition.** State the hypothesis clearly. The objectives and goals of the experiment must be defined. State the following constituents:
   a. object of the study (what is studied),
   b. purpose (what is the intention),
   c. quality focus (which effect is studied),
   d. perspective (whose view), and
   e. context (where is the study conducted).

2. **Planning.**

3. **Operation.**

4. **Analysis and interpretation.**

5. **Presentation and package.**

Some things to take into account when arranging the experiments include:
- Selecting the right group of subjects, based on their modeling expertise and experience
- Selecting reasonable set of test tasks
- Making sure that the target models are sensible and realistic
- Selecting the right measures to evaluate the approach
- Assessing the generalizability, coverage and how realistic the test cases are
- Assessing how much the usability and user interface of the tools affect the results

**Question 3.** Based on this outline, what would the initial design of the experiment look like?

**Question 4.** What would the internal and external threats to validity comprise?

### 4. Concluding remarks

Designing such an experiment is not a straightforward task, but obviously requires lots of thinking and preplanning. Hopefully the seminar session will stir some discussion and provide useful feedback for attacking these kinds of research evaluation problems!

---

**Figure 1.** Example of merging two class diagrams and visualizing their commonalities and differences (taken from Selonen and Kettunen [SeK07], Figures 6, 7 and 8)