

# Transformation of behavioral models based on compositions of sequence diagrams.\*

Jacques Klein and Noel Plouzeau

Irisa, Campus de Beaulieu, 35042 Rennes, France

This paper shows a behavior composition technique, based on sequence diagrams mergings, that allows a unique, well defined and well controlled result model even in the case of multiple weavings (i.e. weaving more than two fragments). We distinguish the composition of models at a low level of abstraction from the composition at a high level of abstraction. The technique is illustrated by an observer pattern specification. Composing behavioral elements (which can be seen as aspects) at model level may play an important role within both MDA and aspect based software processes.

## 1 Principles of behavioral models compositions

Our models composition comports three steps. First, we introduce a two-level decomposition of UML sequence diagrams: *basic* sequence diagrams (bSDs), and *high-level* sequence diagrams (HSDs). The bSDs only allows sendings and receptions of messages, while more elaborated scenarios (with loop, alt, ...) can be depicted using HSDs. This decomposition simplifies weaving and insertion of behavioral aspects. Second, we propose an operator which allows a merging of two bSDs. This operator uses a third bSD (called SD interface) that indicates the overlap between the two operands. In a last step, we present a high-level composition, where the work of the designer will be to specify if a SD have to inserted or woven.

## 2 Weaving of Observer Pattern

The usefulness of a such composition can be showed by the weaving of the well-known Observer Pattern in the scenario of a clock depicted by Figure 1. In this example, we are only interested in the effects of the method *incSecond()*. As proposed in previous section, we split the scenario of the clock in *basic* sequence diagrams. Consequently, weaving the observer pattern depicted by the sequence diagram *observer* in Figure 1 is simply an insertion. Splitting the initial scenario and performing an insertion has the positive effect that the behavioral aspect is still contained in a single sequence diagram after the weaving; therefore the initial scenario will be almost unchanged.

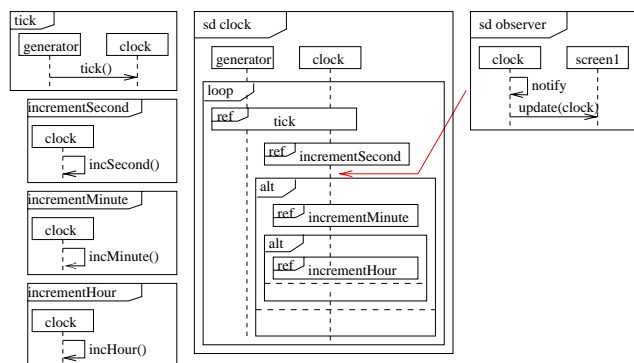


Fig. 1. Scenario of a clock

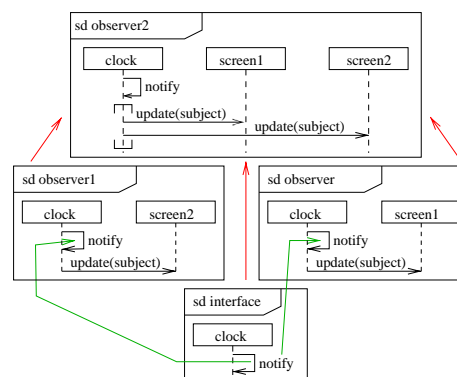


Fig. 2. To add an observer

Let us now add a new observer to our model. If we proceed again by insertion of the sequence diagram that defines the observer pattern, we obtain a scenario with two *notify* messages. This is not the expected behavior. Here, we have to merge SDs. The merge should produce a new behavior that contains both SDs, but does not duplicate their common parts that are identified by the interface. This merging is showed in Figure 2. By recursion, a scenario for any number of observers can be easily obtained with this composition.

\* This work has been partially supported by the ARTIST project (IST-2001-34820).