Weaving behavioural models

Jacques Klein, Jean-Marc Jézéquel, Noël Plouzeau

Weaving Behavioural Models

Jacques Klein, Jean-Marc Jézéquel, Noël Plouzeau
IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France
{jklein, jezequel, nplouzeau}@irisa.fr

1. Introduction

Languages for aspect-oriented programming (AOP) [5], such as AspectJ, are now popular, and the concepts used by the AOP community such as join points, pointcuts and advices are well-known. At the same time, in recent years, the aspect oriented software development (AOSD) approach has been developing itself beyond the programming activity. More particularly, the *Early Aspects Initiative* [9] advocates the management of crosscutting properties, i.e. aspects, at the early development stages of requirements engineering and architecture design to identify the impact of aspects as soon as possible. Some composition operators of aspects exist for these development stages [1][2], but they do not closely match standard AOP concepts (pointcuts, advices…). In this paper, we propose an automatic way for weaving behavioural aspects given as scenarios.

2. Scenarios and Behavioural Aspects

Scenario languages have been the subject of significant interest during the last decade. They are mainly used to describe behaviours of distributed systems at an abstract level or to capture requirements in early development stages with a clear, graphical, and intuitive representation. More precisely, we use scenarios expressed as Message Sequence charts (MSC) [4], which propose two levels of specifications. At the lowest level, basic MSCs (bMSCs) describe simple communication patterns between entities of the system. However, bMSCs alone do not have a sufficient expressive power: they can only define finite behaviours, without real alternatives. For this reason, MSCs have been extended with High-level MSCs, a higher level of specification. HMSCs allow the composition of bMSCs with composition operators such as sequence, alternative and loop as illustrated in figure 1. Roughly speaking, HMSCs are a kind of bMSC automata. Figure 1 shows a classic example of an account creation followed by a log in. More precisely: a user of the system must first enrol with the system. Once enrolled, he must log on to the system for each session. The alternative behaviours are also depicted: an enrolment can be rejected because the data are wrong. The log in can be rejected because the password is wrong. Finally, the id can be wrong, and then the user can enrol or try to log in again.

Let us note that, UML 2.0 Sequence Diagrams [8] are largely inspired by MSCs. (A comparison between UML2.0 and MSC-2000 is given in [3]).

A whole and consistent scenario of a system can be difficult to specify because several behaviours are the result of crosscutting requirements which are relatively independent of the scenario of the system (called nominal scenario). One of the principles of AOSD is to have a separation between the aspects and the nominal system. So, we will simply represent behavioural aspects by a set of independent scenarios, the problem being now to know where and how weave the aspects into the nominal scenario.

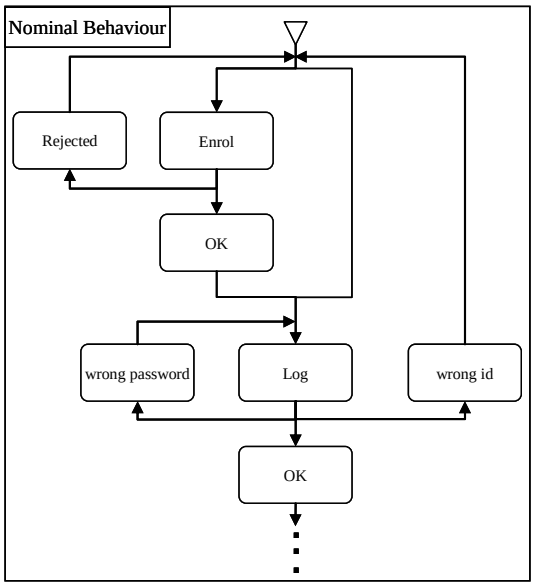

Figure 1 : An nominal scenario.

3. The Weaving Process

As in AspectJ but at the modelling level, we use the concepts of pointcuts and advices in our weaving process. Indeed, our aspects are composed of two parts. One part is the pointcut which matches specified patterns in the nominal scenario where the second part of the aspect, the advice, is woven. More precisely, the advice specifies the behaviour

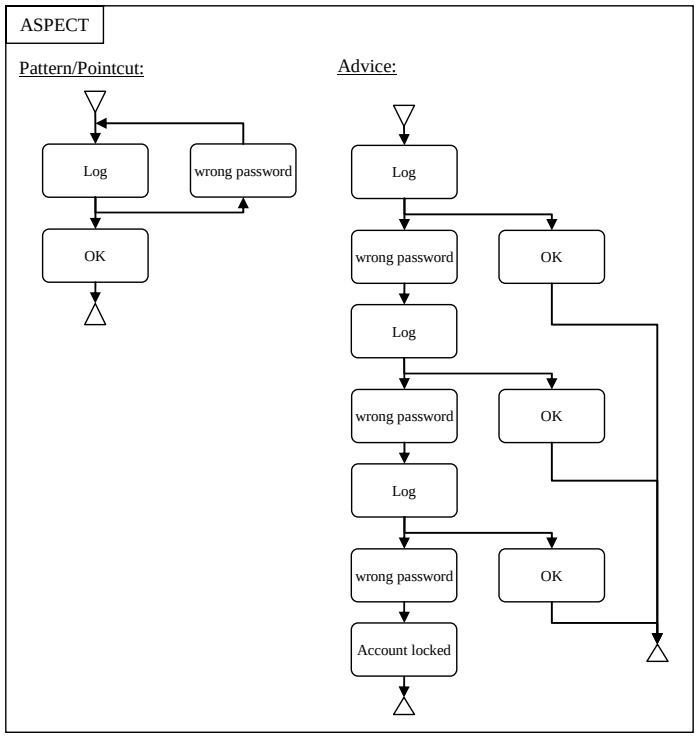

Figure 2: An aspect of security

of the aspect. The pointcuts and the advices are HMSCs (we can consider that the bMSCs of the HMSC pointcut play the role of join points). The figure 2 shows an example of behavioural aspect.

This process introduces two important formal problems. Firstly, the HMSC pointcut has to be identified within the nominal HMSC. If we consider the semantic of partial orders of HMSCs, knowing whether the behaviour represented by the HMSC pointcut is present into the nominal HMSC is undecidable [7]. For this reason, the search for the pointcut will only be done on the name of the bMSCs. The problem thus becomes determining whether an automaton is included into another automaton. Secondly, the advice has to be woven into the nominal HMSC. We propose to use the operator of fibred product, introduced in [6], which is a kind of synchronous product of two automata, but where a third automaton is used to specify the synchronisation. The two operands will be the nominal HMSC and the advice HMSC, while the HMSC pointcut will play the role of synchronizer.

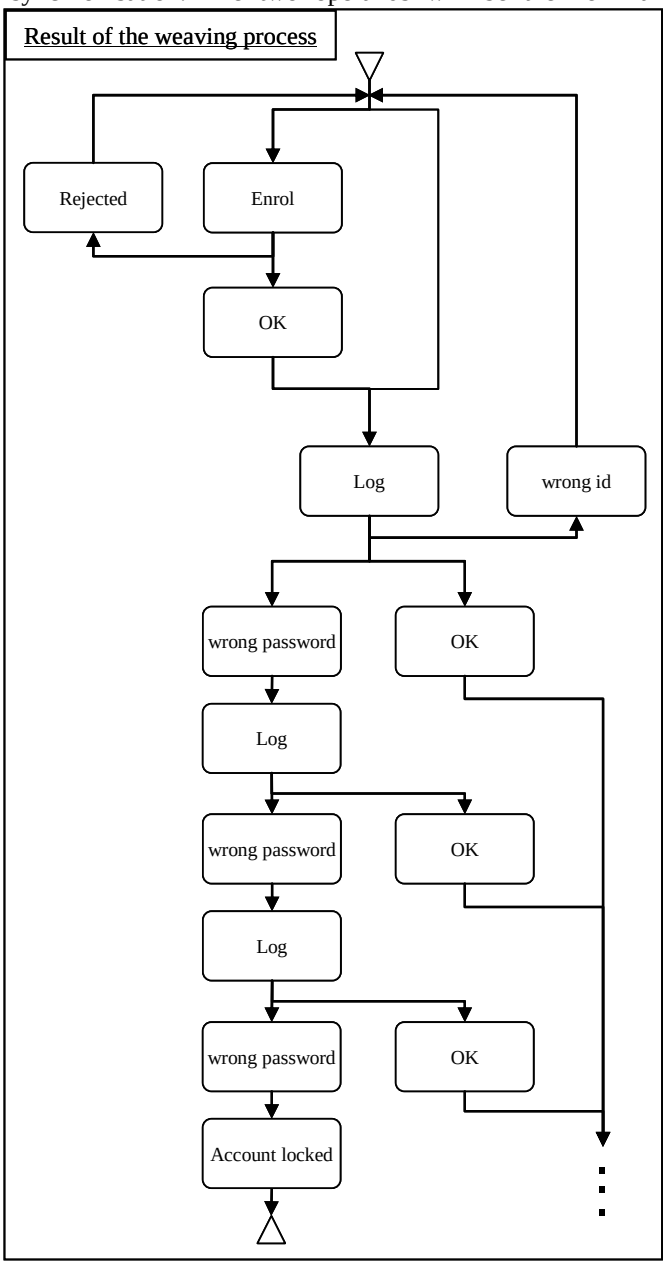

Figure 3: The result of the weaving process.

4. Applying the concept to a case study

The depicted behaviour in figure 1 allows an infinite number of attempts to enter the password, and of course, this behaviour is a godsend for a hacker. To solve this problem, an aspect of security, which allows three attempts only, can be woven. The advice of such aspect is depicted in figure 2, as the pointcut which represents the pattern to detect in the nominal HMSC. The resulting HMSC, depicted in figure 3, is automatically obtained after applying the fibred product between the nominal HMSC and the advice, and where the pointcut is the synchronizer.

References

[1] J. Araujo, J Whittle, and D.-K. Kim. Modeling and composing scenario-based requirements with aspects. *In Proc. of the 12th IEEE International Requirements Engineering Conference*. Japan. September 2004.

[2] S. Clarke and R. J. Walker. Composition patterns: An approach to designing reusable aspects. *In International Conference on Software Engineering*, 2001.

[3] O. Haugen. Comparing UML 2.0 interactions and MSC-2000. *In Proceedings of SAM 2004: SDL and MSC fourth International Workshop*. LNCS 3319, 2004.

[4] ITU-TS. ITU-TS Recommendation Z.120: Message Sequence Chart (MSC). ITU-TS, September 1999.

[5] G. Kiczales, J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming, *Proceedings of ECOOP*, 1997.

[6] J. Klein, B. Caillaud, and L. Hélouët. Merging scenarios. *In 9th International Workshop on Formal Methods for Industrial Critical Systems (FMICS)*, Austria, sep 2004.

[7] A. Muscholl, D. Peled and Z. Su. Deciding properties of message sequence charts. *In Proc. Of FoSSaCS'98*. Portugal. LNCS 1378. 1998.

[8] Object Management Group. Unified modeling language specification version 2.0: Superstructure. Technical Report pct/03-08-02, OMG, 2003.

[9] http://www.early-aspects.net/