

## Session 3 Discussion

Recorded by Martina Maggio, Lund University, Sweden

Shinichi Honiden — Dynamic Software Evolution - Approaches and Issues

Software evolution implies that the software adapts to requirement changes. Facebook system is updated every day during ordinary operation. Motivating example: online shopping system. The first evolution is adding identification function with id and password and the second evolution is adding two-factor authentication functions. After the first evolution in the goal model there are added parts and also for the sequence diagrams there is something added.

How to implement dynamic evolution?

Use of Javassist, a class library to provide functional reflection. Dynamic evolution uses reflection, program can rewrite themselves at runtime. Class can be replaced to change things at runtime. Reflection is the only technique that allows a program to change itself. In terms of the location of changes, reflection is the only technique that can change things everywhere in the program.

Continuous delivery. One should think about before evolution, after evolution and during evolution. Many users, which should seamlessly execute despite the evolution. What happens to the users that are logging in when the two factor authentication process is added.

How to express the behavior of the specification of the dynamic evolution? Model checking.

Model checking would be promising to verify the evolution behavior. Maude: algebraic specification language which supports reflection and model checking. But this does not scale.

---

Yasayuki Tahara — Towards Effective Management of Dynamic Software Evolution

Case study with the shopping website is an adaptation of (Chen et al 2014, Qian et al 2014).

as contexts, existing data structures).