

Session 10 Discussion

Recorded by Kenji Tei, NII, Japan

The Aftermath of Mystery Flight MH370: What Can Adaptive Software Engineers Do?,
Yijun Yu, The Open University, UK

To speculate causes of accidents of flights, much data is required. He showed a new concept "Internet of Flying Things" where data about flight is collected at runtime to identify symptoms of accidents in an early phase. However, bandwidth is not enough to send all raw sensor data from flights. Basic idea is that send knowledge instead of raw data. Our self-adaptive software engineering people use many kinds of models representing knowledge. These models and techniques will be useful. Challenge is how can knowledge used in MAPE-K loops be elicited, simulated, verified, and explained to ensure performance, privacy, security, and trust. We need control with the right-level of knowledge, and we need knowledge for explanation. Currently, pilots exchange knowledge with grand operators through human-to-human communications. If we do not trust pilots, such automation is needed. However, this approach will introduce new vulnerabilities in flights. How should we deal with that? Can we trust the system?

Software Self-Adaptivity Measurement based on Requirements Models,
Zhi Jin, Peking University, China

RE assumes that properties and constraints in the environment can be fixed in design time, but the environment and user goals may change dynamically. A certain approach is introducing a controller (MAPE-K loops) that detects changes and modify software system at runtime. How do we get specification of a controller, and how do we make it better? She introduced a view-based approach. We need to model different aspects of the outer world; environment, situations, and contexts. Controller should capture context changes and select appropriate requirements and architecture. How does the controller synchronize models used in different levels of E,R,and S, at runtime? How does it assure correctness of awareness and adaptation? Take-home message is that RE2016 is held at Beijing. Do not miss it.

Modularity for Uncertainties in Adaptive Software Systems,
Naoyasu Ubayashi, Kyushu University, Japan

How do we cope with uncertainty, in particular known-unknowns, in development of software components. Among alternative designs and implementations, developers select one. However, the other designs and implementations may be valuable for some users. He introduced modularization techniques to cope with such a development time uncertainty, called "Archface-U". Archface-U modularized alternative or optional implementations. It also supports type checking. Discussions were about difference with late binding mechanism supported by other languages and about verification of such a program including non-deterministic behavior.