Notes from seminar topic on "Architecture-based adaptivity"

Presenter: Amir Takerkordi
Time: 1. Lecture on the 12th of February
Note taker: Morten Lindeberg

Presentation

Amir's presentation was based on the article "*An Architecture-Based Approach to Self-Adaptive Software*" by Orezy et. al. The main purpose of the article was to examine the fundamental role of software architecture in self-adaptive systems.

A sample scenario with a fleet of unmanned aircrafts attacking an enemy airbase was given. In the scenario, the unmanned aircraft was first programmed to attack the airbase, as if it was not guarded. As the aircrafts was approaching the enemy airbase, they found that the base in fact was guarded by SAM surface to air guards. The attacking aircrafts then adapted to their changing perception off the enemy airbase, by creating a different and more suitable attacking formatting better fit to take out the SAM missiles as well.

Further on, Amir presented core self-adaptation methodology from the article, by presenting a definition of what self-adaptive software is, and also different degrees of adaptivity ranging from conditional expressions to AI-based learning. A set of questions that a developer of a self-adaptive software system should answer, was also given. E.g., type of autonomy, cost-effectiveness, and adaptation frequencies. Amir then presented a central figure from the article; figure 2, which show the lifecycles of "Adaptation management", and "Evolution management." The circles involved tasks that Amir described further. He pointed out these lifecycles as perhaps the main contribution of the article, and stated interesting points to be the methodology around open-closed, and cost-effective adaptivity, and as well the classification of self-adaptivity. Finally, he stated that the article was presented at a very abstract level, with no details of actual implementation.

Questions and Discussion

Frank first asked a question on how the article can be used. Amir answered that it was a very abstract article, meaning it clearly lacked details. For instance, the mentioning of the two systems "C2" and "Weaver", where the article lacked explanation on how their infrastructure were related. Tonje shared her thought claiming that the article gave a good introduction to the topic, something Amir agreed on, claiming its main purpose perhaps was to draw the big picture.

As a second question, Roman asked what was missing in the paper. For instance, it did not consider churn rates, and how to scale in different settings. As a response, Amir pointed out that the architectural model did not consider testing as part of the evaluation.

A third question from Frank was about the degrees of self-adaptivity, and what algorithm / degree to choose in different settings. In response Tommy stated that this should be chosen on behalf of time aspects. For instance AI based learning (which is the most extreme degree of self-adaptivity) would consume a lot of computation, which in time-constrained domains would not be feasible.

The discussion evolved back into considering the usefulness and the quality of the article. Eli stated that the article was not presenting a clear image of the state-of-the art, rather just an overview. This since the paper doesn't present any results or evaluation of existing results. This she remarked as typical for this kind of paper. Amir questioned the contribution of the article. To wrap up, Frank stated that the article could purpose as an overview, where the circles presenting the lifecycles could serve as a reference framework. Especially the tasks that were presented in the lifecycles were mentioned as important contributions. This although the paper, written in 1999, is now outdated in terms much has happened since then.

Anh asked wetter other approaches existed in the literature. By comparing this with the approach in the article it would be easier to identify how valid the approach in the article is. In his response, Frank mentioned, "Aspect weaving", and also existing complex approaches in the field of AI.

Aida stated that an aspect that was lacking in the article was the expected application lifecycle, which clearly has something to do with how it should adapt to changes. Morten (being me, the note taker) asked how the approach of today's commercial software, performing self-updates by periodically logging into a server, relates to the approach presented in the article. As an answer, both Frank and Christian pointed out the higher degree of human interaction. The initiator of such an update is clearly not the program running at the end-user, rather the manufacturer posting such an update at the server.

Conclusion

The article presents a good overview of the methodology involved in the process of developing self-adaptive software. As pointed out by Tommy, the article is somehow outdated, and describes something's which already are available today. Although the article is old, it is still valid and useful since it introduces and describes the tasks involved in developing self-adaptive software.