Workshop on the Maintenance of Aspect Oriented Systems

Mario Luca Bernardi
Giuseppe A. Di Lucca

Department of Engineering
RCOST – Research Centre on Software Technology
University of Sannio, Benevento, Italy

mlbernard/dilucca@unisannio.it

Motivations and Goals

The Aspect Oriented (AO) Programming paradigm is spreading out more and more when developing software systems. Aspect Oriented Programming (AOP) allows to easily and better modularize and manage the crosscutting concerns by means of program units called “aspects” [Kic97]. The code related to crosscutting concerns, encapsulated in aspects, is “woven” into the code of the traditional program units (e.g. classes in OO code) to build the overall system.

While there are several approaches to reengineer and evolve ‘traditional’ systems towards AO ones, there is still very few about the maintenance of AO systems.

The maintenance of an AO system may be more difficult than a traditional one.

This is mainly because the system resulting after weaving presents relationships among aspects and traditional modules that can heavily modify the structure and behaviour of the basic system (i.e. the system with no weaving).

Thus, due to the heavy and intrusive impact that AOP constructs have on the code they belong, it could be very difficult to evaluate the impact that a change has on the whole system as well as it could be very difficult to evaluate any side/ripple effect in the system the change could introduce.

The maintenance of AO systems will be one of next challenges of software engineering. The software engineers should be provided with adequate methods, techniques and tools to successfully face and overcome the new problems and difficulties that AOP constructs introduce for maintenance tasks.

Some main questions rise at this aim:

- are the to-day methods/techniques/tools to maintain ‘traditional’ software systems (e.g. Object Oriented systems) adequate to maintain AOP systems too?
- which and in which way the AOP constructs can make hard maintenance task?
- how the today maintenance approaches can be adapted to allow them to take into account the new characteristics of AOP?
- do we need to identify “best practices” for AOP paradigm that help to design and implement more maintainable software systems?

Just as an example, traditional code representation models usually used to support the maintenance tasks might not be adequate to fully support the analysis of AO systems to fully comprehend them, due to the specific features characterizing AOP and the effect they have on the code they belong.

Thus, new approaches, or the adaptation of the existing ones, taking into account the specific features of AOP have to be defined and used to effectively identify maintenance problems and to perform maintenance tasks of AO system.

Based on these issues, the proposed workshop has the main aim of:

- making more aware the software maintenance community of the (new) difficulties related to AO system maintenance;
- presenting and discussing proposals about the problems raising in the maintenance of AO systems, and ways to effectively address these issues;
- acting as a forum for the promotion of joint research and experimental studies about the maintenance of AO systems.

Main issues
The proposals and the related discussion should address the following main issues (not forming an exhaustive list):

a) **AO code analysis:** ‘traditional’ code analysis techniques need to be adequate and adapted to the peculiar features of AOP. Proposals and discussion would highlight what and how is to change for static and dynamic code analysis to be able to cover the AOP specific features (such implicit invocation and introductions).

b) **AO system models:** new type of system models, or the adaptation of the existing ones, are needed to consider and represent aspects also highlighting the relationships and interactions existing among aspects and traditional software components. The discussion and the proposals would hence address the definition of such new AO representation models but also techniques to use such models in order to identify and to analyze new properties, relationships and interaction introduced by aspects.

c) **AO code representation forms:** the traditional graph-based representation of source code (e.g., Control flow graph, dependence graph, call graph, data flow graph) are no more fully adequate to represent the AOP code. Proposals and discussion would address how ‘traditional’ representation forms should be modified and adapted to take into account the way AO constructs can modify the control and data flow, as well as the static code structure by mean of pointcuts, joinpoints, advices, intertype declaration.

d) **AOP specific quality models and metrics:** the new AOP constructs introduces coupling and cohesion among components that are difficult to identify since are not based on imperative invocation. A key issue is to identify the ways aspects interacts with the base system components and among themselves: such interactions may affect the effort and the quality of a maintenance intervention. Specific metrics and quality models are hence needed to evaluate the quality of an AOP system and to allow the estimation of maintenance effort.

e) **Aspect mining:** the migration of traditional systems toward AO ones is usually based on the identification of code components making up crosscutting concerns to encapsulate into aspects. Several approaches have been proposed at this aim in the last years. Novel proposals welcome as well as discussion about the effectiveness of existing approaches.

The aspect mining techniques are of key importance with respect to maintenance: the quality of crosscutting identification and the techniques to migrate such crosscutting towards aspect oriented paradigm greatly affect the maintainability of the resulting system. Hence the importance to also discuss how aspect mining techniques produce crosscutting models that support the evolution towards well design and maintainable AOP software system.

A discussion on some of these issues was started at the working session ‘Comprehending Aspect-Oriented Programs: Challenges and Open Issues’ held at the International Conference on Program Comprehension 2007 (ICPC 07) and this workshop has also the scope to take again it and to analyze what new contributions have been provided about them in the meanwhile.

**Workshop organization**

The workshop will be organized and held as a half-day one. The workshop aims to involve people both from academia and industry interested to present and debate on the topics, to collect insights useful to plan studies, researches and experiments by creating collaborations and networks among them to develop and deep specific themes of the discussed issues.

A call for participation will be issued to take on speakers and participants, as well as some speakers are directly invited to present and discuss about some specific themes. Researchers and practitioners will be invited to submit short (not more 4 pages) or position papers addressing the above mentioned research issues presenting novel proposals or experiences addressing them. The submitted papers will be reviewed by the organizers and a selected group of reviewers. A limited set (six-eight) of papers will be selected to be shortly presented (not more than 10 minutes) at the beginning of the workshop. Each presentation would be based on provocative statements or ideas related to any of the workshop issues, as well proposals for researches to jointly carry out with other participants.

The aim is to have presentations proposing challenging ideas providing some solutions to the listed issues and to stimulate the discussion, as well as to favour networking within participants, rather than presenting results.

Each presentation will be followed by a short discussion. The session organizers will collect a list of debated issues, presented proposals, and questions. After all the presentations and related discussions, the collected list is proposed to participants to start a deeper discussion during which further ideas and comments are collected. Working groups focused on specific issues could be organized to carry on the discussion.

The presentations and the results from the discussion will be collected into a set of e-papers published on the workshop website.