Foreword: 2nd Workshop requirements@run.time

Nelly Bencomo
INRIA Paris - Rocquencourt
Le Chesnay, Franceaine de Voluceau, 78153
Le Chesnay, France
nelly@acm.org

Emmanuel Letier, Anthony Finkelstein
Department of Computer Science
University College London, UK
London, UK
(e.letier,a.finkelstein)@cs.ucl.ac.uk

Jon Whittle, Kriss Welsh
Lancaster University
School of Comput. & Comm.
Lancaster, UK
(whittle,welshk)@comp.lancs.ac.uk

***Abstract*—The 2nd edition of the Workshop requirements@run.time was held at the 19th International Conference on Requirements Engineering (RE 2011) in the city of Trento, Italy on the 30th of August 2011. It was organized by Nelly Bencomo, Emmanuel Letier, Jon Whittle, Anthony Finkelstein, and Kris Welsh. This foreword presents a digest of the discussions and presentations that took place during the workshop.**

***Index Terms*—Requirements; reflection; run-time; self-adaptation;**

I. Introduction

The requirements@run.time workshop series provides a forum to explore a radical challenge to the traditional view of requirements models as static, slowly-evolving and purely design-time entities. The second edition of the workshop successfully attracted at least eighteen (18) people from seven (7) countries. Nine (9) papers were submitted. Seven (7) papers were accepted for publication in these proceedings. Every submitted paper was reviewed by at least three program committee members. The workshop aims were to

1) provide a state-of-the-research assessment to guide research in the area;
2) integrate and combine research ideas from RE, monitoring, computational reflection, model-driven engineering (including models@run.time) and autonomic, self-* systems;
3) devise a research agenda for the achievement of requirements-aware systems and stimulate the creation of a network of researchers in the area; and
4) plan and promote further events on these topics.

Next, a summary of the presentations and discussions during the workshop is presented.

II. SESSION SUMMARIES

In the opening presentation, Bencomo outlined the for-mat of the workshop. She then summarized the objectives of the workshop. Following the kick-off of the workshop, the paper presentation sessions followed. Paper presentations were divided into the three (3) paper sessions. Appointed discussants encouraged the discussion after the presentations of papers. The presented papers are summarized as follows:

SESSION 1: Software Evolution and Fuzzy logic

1) "*Requirements models at run-time to Support consistent system Evolutions*", presented by Mori, proposes a framework for supporting runtime evolution of a sys-tem/application that has context-dependent requirements. It takes a feature-oriented approach in which the system configuration is viewed as a set of features. The correct-ness/consistency criteria that the framework embodies are related to feature interactions. In essence the approach can be seen as a dynamic form of a software product line.
2) "*Dealing with Softgoals at Runtime: A Fuzzy Logic Approach*", presented by Serrano, proposes an approach for dealing with softgoals at runtime. The approach is based upon an analysis of how human agents actually go about analyzing softgoals in practice. The paper describes their propagation simulator which combines propagation rules with ideas from fuzzy logic. Based upon this simulator, the authors then describe a reasoning engine to analyze the softgoals to select those plans to satisfy these goals at runtime.

SESSION 2: Software Adaptation

- "*Using NFR and Context to Deal with Adaptability in Business Process Models*", presented by Castro, describes an NFR based method for matching requirements at run time. The process is illustrated with a case study of adaptation of a safety critical hazard management system.
- "*From Awareness Requirements to Adaptive Systems: a Control-Theoretic Approach*", a position paper presented by Silva, aims at building adaptive systems based on Awareness Requirements and "system identification". The authors model adaptive systems as feedback control systems, where the output of the target system is used as the input for the control system. This paper also presents a software development process, from the requirements to the built system.
- 3. "*Reasoning About Adaptive Requirements for Self-Adaptive Systems at Runtime*", presented by Qureshi, is a vision paper which focuses on the reasoning issue. The paper presents the point of view of the authors about what is needed from the system to do the reasoning. The paper reviews the state of the art, and in particular contributions that they and their colleagues have made to it. It then discusses what will be needed in practice, the challenges, including those of performance.

SESSION 3: Software Adaptation and Safety Require-

ments

1) "*Requirements-Driven Adaptation: Compliance, Context, Uncertainty, and Systems*", by Chopra, aims to provoke a discussion on the way we formulate and model requirements for self-adaptive systems. The paper frames the discussion in terms of the ease of proving (or disproving) compliance, the way in which contexts under which alternate requirements (or solution strategies) are adopted are specified, the level of uncertainty in the environment it is feasible to accommodate, and adaptation in multi-agent systems.
2) "*Enforcing Safety Requirements for Industrial Automation Systems at Runtime*", by Moser, describes a me-thod for eliciting safety requirements using Boilerplates. The Boilerplate requirements are transformed into if-then-else rules for run-time enforcement.

III. SESSION SUMMARIES

Each session concluded with a wrap-up discussion that pulled together the themes to emerge. These discussions led to a list of issues that we felt merited more research. For the bulk of the afternoon session, we split into two groups. Each group chose a subset of issues to emerge form the paper sessions that they wanted to discuss.

A. Group A

This group considered the issues: a. *What do we mean by "new"? Inferred by the system? Injected by interactions by systems?* b. *What are the differences between unforeseen vs fore-seen, and evolution vs adaptation?* c. *What are contexts in requirements? What are their roles in requirements at runtime?* The discussions initially focused on questions a and b as they are close. Later they found that c is in fact related to a and b as well. The observations agreed during the discussions are as follows: (1) Motivated by "*delaying design decisions to the run-time*", requirements at runtime are tolerating a boundary that may be relaxed initially to freely explore design alternatives, but as the requirement of adaptation rather than evolution, they must confine to the boundary again after the free exploration. The evolution, on the other hand, may allow the boundary to shift to a brand new situation that does not require the original boundary to be recovered. Delaying the design decisions increases the freedom of choices in general thus are preferable. (2) Contexts of self-adaptive systems are defined with different focuses for different approaches. For examples, when being asked "*what triggers an adaptation*?", some consider generally context changes, however it can be considered as well the negotiation of soft goals, the awareness requirements, and any change to the ontological knowledge. When generalizing these different triggering conditions a specific definition of context arised: "Contexts are anything that influence the decision making". Considering delaying design decisions is the purpose of self-adaptive systems, if that is successful, more contexts will be accumulated before the decisions, which may help the decisions to be better informed. The variation of different decision support systems, i.e., planning logic, would not change the fundamental problem of context-awareness. Finally, contexts are not everything because one does not really need to obtain all information before the decisions, only those changes that are relevant to the decision makings will be considered in the contexts. Monitoring such contexts thus becomes effective for the self-adaptive systems, in which system identifications partition the system boundary such that only few indicators is needed for making decisions.

B. Group B

This group chose the following issues: The group started by considering the question of *how to demonstrate the dependability of adaptive systems and whether run-time representation of requirements could help for this*. Traditional requirements engineering techniques assume a stable context, the motivation for requirements at runtime is to deal with changing contexts so that decisions that are traditionally made at design time can be postponed to runtime when the context is know better. The group continued discussing *the distinction between foreseen and unforeseen contexts*. What is meant by an unforeseen context is not always clear. A system may fail due to exceptional events that were not entirely unforeseen by the system designers (the possibility of these events may have been indentified) but handling of these events may have been ignored due to cost or tradeoffs with other goals. The problem is to identify the boundary of the system (what events should be considered, which ones will be ignored) but sometimes defining this boundary upfront is impossible and that's were new unforeseen requirements and contexts come from. They moved on to consider whether there are already in-dustrial applications that at least in part rely on models of their requirements at run-time and what we could learn from those. The area of artificial intelligence and robotics rely heavily on run-time models of the system environment and goals, for example through the use of Belief-Desire-Intention (BDI) architectures. However, the elaboration of BDI system relies on many domain assumption and goals that remain implicit during the construction of the model. Failing to document these assumptions and goals also prevent a systematic validation of the BDI system against them. Furthermore, some of these goals and assumptions can become invalid during the system lifetime. So we would need a process to manage these goals and assumptions and related them to the BDI model. They continued by asking themselves what are the most important questions we should work on and whether there are topics we should stop working on. The first opinion is that requirements engineers should stop doing research on adaptive systems with fixed condition-action rules because the problems of defining and acting on those rules are solved and it doesn't address the issue of adaption to unforeseen context. Instead, we should be working on techniques to manage the evolution of such condition-action rules at run-time, i.e. we should consider evolving adaptation policy. A second opinion is that we should stop merely consider-ing adaptive devices (e.g. mobile phones) and that instead we should devote more

effort to studying adaptation in the context of socio-technical systems. We should also stop working on toy, unrealistic examples (where the adaptation scenarios are not convincing) and work more on real problems. However, validation of requirements engineering techniques for adaptive systems is difficult and time-consuming because it usually requires building and demonstrating full systems. Finally, they concluded researchers should stop working on the assumption of closed systems. Systems are open without known boundaries. For example, if you consider a desktop application such as a word processor, for engineers developing this application, it could be viewed as an application with well-defined inputs and outputs, but the actual goals of its users and the context in which it is used are varied and unbounded. Every system is connected to other systems, and there are no clear boundaries between them.

IV. FINAL REMARKS

Final Remarks: It is interesting to note that discussions of the different groups converged to similar topics and even recommendations; as for example the relevance for context information and boundaries of systems. A general wrap-up discussion was held at the very end of the afternoon. The workshop was closed with a friendly ”thank you” from the organizers to all participants for a fruitful workshop.

Acknowledgments. No workshop is successful by the efforts of only a few people. We would also like to thank the members of the program committee who acted as anonymous reviewers and provided valuable feedback to the authors. We also thank Yijun for leading the writing of the summary for group A. Last but not least, we thank the authors for their interesting submissions and for helping us making this workshop possible.