ISSN 2186-7437

NII Shonan Meeting Report

No. 2012-3

Engineering Autonomic Systems (EASy)

Arosha Bandara
Shinichi Honiden
Yijun Yu

May 14–17, 2012

National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-Ku, Tokyo, Japan

Engineering Autonomic Systems (EASy)

Organizers:
Arosha Bandara (The Open University, U.K.)
Shinichi Honiden (NII, Japan)
Yijun Yu (The Open University, U.K.)

May 14–17, 2012

1 Overview of the Meeting

Increasing complexity of software systems poses a number of challenges for software engineers, IT service managers and end-users. In 2003, Kephart and Chess published their vision of autonomic computing, which aimed to address some of the challenges of software complexity. The essence of this vision was to create systems that would be able to adapt to their operating environment in a manner analogous to the autonomic nervous system, allowing human administrators and users to concentrate on setting the longer-term high-level goals for the system rather than the operational minutiae required to keep it running on a day to day basis. The autonomic paradigm brought together existing research in the areas of control systems, adaptive networking and context-aware computing which focussed on the challenges of developing the underlying technical frameworks that could enable autonomic operation. At the same time new areas of research, such as Business Driven IT Management, have developed to investigate ways in which users can configure autonomic systems in ways that meet the higher-level objectives of organisations.

As a result of these research efforts, several approaches have been proposed to address different aspects of the autonomic computing challenge from policy-based systems and biologically-inspired computing at the architecture level to techniques for analysing autonomic systems and mapping business requirements into specifications for autonomic behaviour. However, numerous issues remain, including the following:

- How do we engineer software for autonomic systems?
- What are the usability issues of autonomic systems? How do we ensure effective interaction with complex software systems that have autonomic components?
- How will deployed autonomic systems evolve? How can we ensure their evolution happens in a systematic way?
- Autonomic systems can be just one component of much larger (and more complex), software intensive socio-technical systems - how do we prevent failures in these complex socio-technical systems?

The purpose of this workshop was to bring together leading researchers from a diverse range of disciplines to discuss the latest research in the area of self-adaptive (autonomic) systems, and explore new ideas, positions, opinions, problems and solutions that could advance the state of the art in this area.

2 Overview of Talks

The short talks of each participant were organised into five sessions, which was followed by a panel discussion encompassing issues that arose from all the presentations in each session. The details of the division of talks across the sessions is presented in Table 1.

Although the keynote and five sessions are not standalone, they allow the panelled discussions after each session to be more focused. Every talk is given maximal 15 minutes, although some speakers did not need the full allocation of time to complete their presentations. Questions were allowed during the sessions for individual talks, while general questions related to the whole panel are deferred to the panelled discussions followed by the individual talks in the session. Here we summarise the discussed questions by panels, which reveals a list of emergent topics during the discussions.

Keynote

Motivated by ubiquitous computing and service computing, Carlo Ghezzi uses Zave & Jackson's notion of separating machines and world, context as the key for analysis of dependability and changeability problems. He categorises the changing contexts relevant to self-adaptive software as changes of the environment and of the system, while contexts of the system are classified into usage context. Changes at maintenance time versus runtime blurs the boundary of both, therefore to compute the reliability MTTF, it is important to consider probabilities as part of the model and to consider changes on part of the model through model checking. These contexts form the basis of constructing very interesting quantitative model checking problems such as PRISM, JMT and queuing network simulations, which turns out to be useful to predicting failures of requirements, i.e., the reachability of absorbing states.

There are also some exciting runtime V&V results about parameterising the behaviour models, and instead of relying on manual assumptions and guarantees regarding the kind of modularisation approach, using probabalistic model checkers to predict the possible failures. The key is to represent the contexts as structures, which can be processed in polynomial time through the Floyd attribute-based parsing techniques in programming language domain. This is identified as a possible connection between software engineering and programming language research.

Panel 1. General Problems

Distinction between Adaptation and Evolution: Mylopoulos' presentation pointed out that there is one key difference between adaptation and evolution. Drawing on an analogy from biological systems: individual adaptation does not change the blueprint of the system, or the class of the species, while

Session	Title	Speaker
Keynote	The challenges of (self-)adaptive software	Carlo Ghezzi, *DEI-Politecnico di Milano, Italy*
Session 1: General Problems	An Architectural approach to Self-Managed Adaptive Systems: a status report	Jeff Kramer, *Imperial College London, UK*
	Context-Aware Self-Adaptive Software Intensive Systems – How to Build Smarter Systems for a Smarter Planet	Hausi A. Muller, *University of Victoria, Canada*
	Requirements-driven adaptive control	John Mylopoulos, *University of Trento, Italy*
Session 2: Application Areas	Adaptive Security and Privacy	Bashar Nuseibeh, *The Open University, UK & Lero, Ireland*
	An asset-centric approach to engineering adaptive security	*Liliana Pasquale, Lero, Ireland*
	Autonomic Role and Mission Allocation Framework for Wireless Sensor Networks	Kenji Tei, *National Institute of Informatics, Japan*
Session 3: Representation and Transformations	The Role of Models@run.time in the Engineering of Autonomic Systems	*Nelly Bencomo, INRIA, France*
	Lifelong Management of Quantitative Requirements: From Models to Run-time Adaptation	Giordano Tamburrelli, *Politecnico di Milano, Italy*
	Self-adjusting Computation and Bidirectional Transformation	Zhenjiang Hu, *National Institute of Informatics, Japan*
Session 4: Middleware and Implementation	A Middleware for Self-organising Pervasive Systems	Luciano Baresi, *Politecnico di Milano, Italy*
	Dynamic Synthesis of Mediators to Support Interoperability in Autonomic Systems	Amel Bennaceur, *INRIA, France*
	Invariant Twin Peaks – Fixed Points for Adapting to Changes	Yijun Yu, *The Open University, UK*
Session 5: Construction	A Goal Model Normalization Process for Constructing Self-adaptive Systems	Shinichi Honiden, *National Institute of Informatics, Japan*
	Towards Dynamic Evolution in Self-adaptive Systems Based on Dynamic Updating of Control Loops	Hiroyuki Nakagawa, *National Institute of Informatics, Japan*
	Driving Elaboration of Environment Models using Controller Synthesis	Sebastian Uchitel, *University of Buenos Aires, Argentina and Imperial College London, UK*

Table 1: Overview of Workshop Presentations

evolution is the opposite. The blueprint for software systems could be the requirements, if so, the requirements change for evolving systems, while remain the same for adaptation systems. Of course it is possible that this analogy doesn't hole because biological systems and software systems are different. However, it is also suggested that the environment and system overlaps in terms of the contexts. Further, it is emphasised the uncertainty in adaptive systems, e.g., whether or not a system is hierarchical, could be reconfigured to reflect changes although participants also highlighted the need to consider different types of uncertainty.

Probability versus fuzziness in quantifying the models: The discussion raises the issue of whether treating probability quantification and fuzzy quantification is compatible. Furthermore, there is an open issue regarding where the information learnt from and the cost of doing so. Ghezzi suggested that it is the choice of analyst how accurate the simulation or analysis is required and how much uncertainty one would live with. He also referred to the talks of Tamburrelli and Baresi to elaborate these points. There is also discussion regarding the meaning of Mean-Time-To-Failure (MTTF) and whether it is possible to consider frequency of failure as a replacement of the probability of failures. Under the assumptions that the system is running for the time so far, such distinction is blurred.

Awareness requirements: This refers to requirements at one or more meta-levels higher than the "vanilla" requirements. The concept is from Mylopoulos' talk that links to the related concepts such as "relaxed" requirements (Bencomo) and "adaptive requirements" (Ghezzi) schools of thought. This raised a related problem of how to mix the different requirements of meta-levels to reason together.

Goal/Plan/Component layers + Runtime V&V: Some discussions about the crosscutting dimensions to self-adaptive or autonomic systems. Kramer's reference architectural model is the basis, with the runtime V&V (e.g., Ghezzi's keynote talk) Muller suggests to be yet another layer (for analysis at least) that may crosscut various dimensions.

Panel 2. Applications

Assets are valuable contexts: Pasquale's talk raises a question about the concept of assets, whether they shall be the 1st class citizen in analysing adaptive security requirements? It is explained that the value interpretation of the assets is the key to quantify the models for Bayesian analysis.

Security/Privacy/Energy/Quality awareness and the need for exemplar case studies: Nuseibeh noted that these models need to have human in the loop and Yu points out the possible relation between privacy awareness to the general awareness requirements. The challenge is whether it is possible to mix meta-level requirements with the functional requirements to reason cohesively. Perhaps the "operationalisation" of quality requirements such as security and privacy provides a bridge to do this. The "energy-efficiency" concerns of

Tei's wireless sensor network belong to the general category of non-functional requirements catalogs too. Therefore, we need to seek some abstraction to consider quality awareness. The challenge is again whether specific application domains have something specific to be modelled to fit the purposes. The discussion highlights the importance of having exemplar case studies per area of quality requirements so that different approaches can be compared with each other.

Tradeoffs: The discussion of several quality requirements prompted the observation that tradeoffs must be a common problem to consider. There was consensus on this point.

Panel 3. Representations

Known vs. unknown: Bencomo's talk brought up the issues concerning 'unknown knowns' (which Nuseibeh refers to as tacit knowledge) and 'unknown unknowns' (which Kramer considers impossible to reason about). According to Bencomo, unknown unknowns are the unforeseeable future that might be learnt from retrospective investigations. Emergent behaviour, as Muller described, is likely the kind of unknown unknowns at the time of development. However, as Kramer points out, at development time there is no way to deal with unknown unknowns. He later cites the 7th motto from Ludwig Wittgenstein "Whereof one cannot speak, thereof one must be silent" to suggest it is better leave out these concerns. However, the subsequent discussion posited that at runtime, with the aid of learning, unknown unknown could become known known. Therefore challenge is about how to design adaptive systems that are capable of learning emergent properties about their environment and behaviour.

Self-adjusting computation and incremental solutions: Hu's convex hull example of the self-adjusting computing illustrated the point of incremental evolution of solutions to address drifting requirements. It was observed that finding minimally changed solutions for evolving systems seems to fit quite well with this notion. Bidirectional transformation was considered to be a useful technique for handling change propagation problems for adaptive systems.

Historical information such as trends and rates: Referring to Tamburrelli's talk about probabilistic model checker, the question was raised about whether it is possible to bring in historical information when analysing adaptive systems. Related to this is the question whether such model checkers deliver control synthesis or not. However, there was no definitive answer to either of these questions. The need for handling such stateful representations in stateless service computing seems to be unnecessary, however in general adaptive systems, it can be considered to be a challenge.

Panel 4. Middleware

Baresi's talk provided a few simulations of how effective distributed middleware can reduce the bandwidth of communication between multiple agents for adaptive services. The new middleware A3 is an ongoing effort which hopefully will

be available for comparison to other middlewares. This is recognised to be an important step to benchmark them against adaptivity.

Benacour's talk presented a comprehensive solution to interoperable service computing by generating the mediator service between different services. Such mediators takes into account the semantics of service interfaces in terms of workflow scenarios.

Yu's talk focussed on generalising the invariant traceability between requirements and architecture to the runtime such that the vision of invariant twin peaks is possible.

Panel 5. Construction

Exemplar case study: The video game simulator presented by Honiden and Nakagawa was observed to have many common features with example systems from the robotics domain (c.f. Kramer and Bencomo's work). It was agreed that, even if it is not the ultimate benchmark, robotics by simulation, real deployment or conceptual illustrations are definitely a good showcase of many self-adaptive mechanisms.

Controller vs Feedback Loops: Uchitel's talk is largely a roadmap for future work of controller synthesis that search-based approach might help at runtime to compose the specification of components with that of the environment. The controller synthesis is in terms of the notion of behaviour of the machine in Jackson's sense. The formalisation he introduces is an interesting extension to Jackson's simpler formula. Deriving the behaviour models out of the interfaces, domains and requirement properties seem to be a viable technique. The question is how to effectively narrow the search space at runtime.

3 New findings

The objectives of the workshop were two fold:

1. Bring together researchers to discuss the topic broadly and explore joint interests deeply;
2. Seek tangible results that matter to the community, through group activities together.

The 1st objective was refined into 16 individual talks and 5 panel discussions. The 2nd objective was achieved through a group activity which was described to the participants as follows:

1. You shall vote for one and only one topic, but can propose new topic if you want
2. During the topic merging phase, you can change your mind
3. Once decided, you shall participate in one and only one topic group for discussion.
4. The discussion shall reflect the interests of all participants of the meeting

5. After your topic is fully discussed, one member of the group shall present the idea in no more than 5 minutes.

6. The presentation shall be shared with all the participants in the end through the organisers

It was agreed that the 3-layer, goal-plan-component reference architecture for self-management systems (Kramer and Magee, 2007) is well-accepted, therefore the discussions did not focus on this aspect of adaptive systems . Instead, it was considered more interesting to focus on different dimensions of such systems and a candidate list of such topics was prepared by the organisers before the workshop:

1. Distinction between Adaptation and Evolution
2. Learning Uncertainty
3. Runtime models
4. Application domains
5. Exemplar case studies
6. Feedback control loops and control theory
7. Synthesis versus planning
8. Context- and situation-awareness
9. Non-functional requirements and tradeoffs
10. Efficiency concerning moving online computation to offline
11. Social dimensions
12. Variability at runtime, aka space of alternatives

The participants declared interests to fit into these topics and after some discussions of the individuals, the participants 'self-organised' into three discussion groups:

Topics	Subscribers
Learning Uncertainty	Ghezzi, Nuseibeh, Bencomo, Tamburrelli, Baresi, Tei
Feedback loops and control theory	Uchitel, Honiden, Kramer, Hu, Benacour, Nakagawa
Awareness	Yu, Muller, Mylopoulos, Pasquale

After each group had it's detailed discussions, a summary was presented in a plenary session. Instead of listing all the bullets, here we just summarise them into three paragraphs.

Learning uncertainty. Uncertainty is the key motivation for self-adaptive systems - when in doubt, plan for graceful failures and it is important to know the limits that a system can tolerate to avoid compromising system integrity and reliability.

Feedback loops are the key. Comparing to traditional controller systems, autonomic or self-adaptive systems must have at least one feedback loop. Both the structure and behaviour model for the runtime system need to be monitored, analysed, planned, and executed. Yet different autonomic systems may require different configurations of feedback loops and the utility/transfer function would need to be known beforehand.

Awareness. Context-, situation-, and requirements-awareness are related. Situation awareness links the context-awareness and requirements-awareness at runtime. Dependencies among the situations can be exploited to reduce the runtime overhead and improve the tradeoffs. Indicators or requirements satisfaction shall be defined such that monitoring and tradeoffs are achievable at runtime.

4 Summary and Lessons Learnt

At the conclusion of the workshop, the organisers received compliments from participants regarding the diversity of topics and dynamic nature of the discussions, reflecting the 'self-adaptive systems' theme of the workshop. It was recognised that autonomic systems as a topic was somewhat narrower than the participants' interests, whereas adaptive systems seem to cover most people's interests. The span of two and half day workshop was not long enough to have in-depth discussions of each topic, yet the high-level research highlights are discussed in full. If one would organise such a workshop again, it was felt that the inclusion wider disciplines (e.g., HCI, machine learning, etc) would be more useful. However, given this area is new, most researchers tackle the problem from their own perspective. Therefore it is natural to have both overlaps and differences of research interests and future challenges.