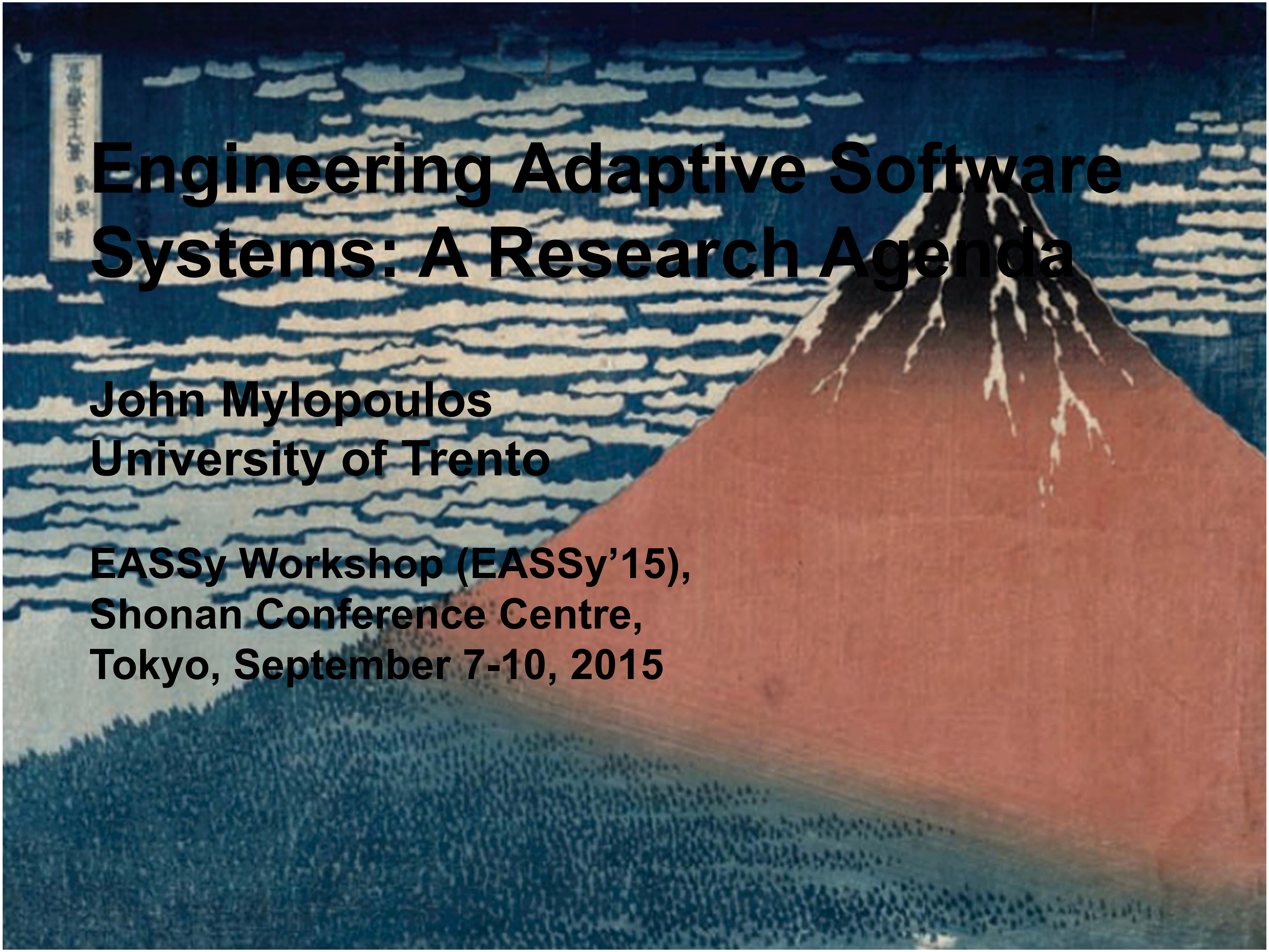


Engineering Adaptive Software Systems: A Research Agenda

John Mylopoulos
University of Trento

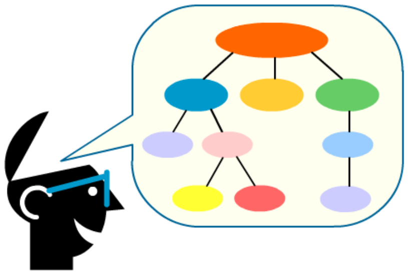
EASSy Workshop (EASSy'15),
Shonan Conference Centre,
Tokyo, September 7-10, 2015

Katsushika Hokusai, 1830-33

Abstract

Adaptive software systems need to be capable of multiple behaviours for fulfilling their requirements, so that if one behaviour fails, the system can adapt to an alternative. The most important question then in engineering such systems is: how do we design and implement systems that are capable of fulfilling their requirements in multiple ways? What are the concepts in terms of which such systems are conceived, designed and implemented? What are the adaptation mechanisms through which an adaptive system monitors its behaviour, determines root causes for failing requirements and selects a suitable adaptation?

The presentation reviews answers we have given to some of these questions in the PhD theses of Vitor Souza and Kostas Angelopoulos (on-going), also some of the open questions that constitute our current research agenda.

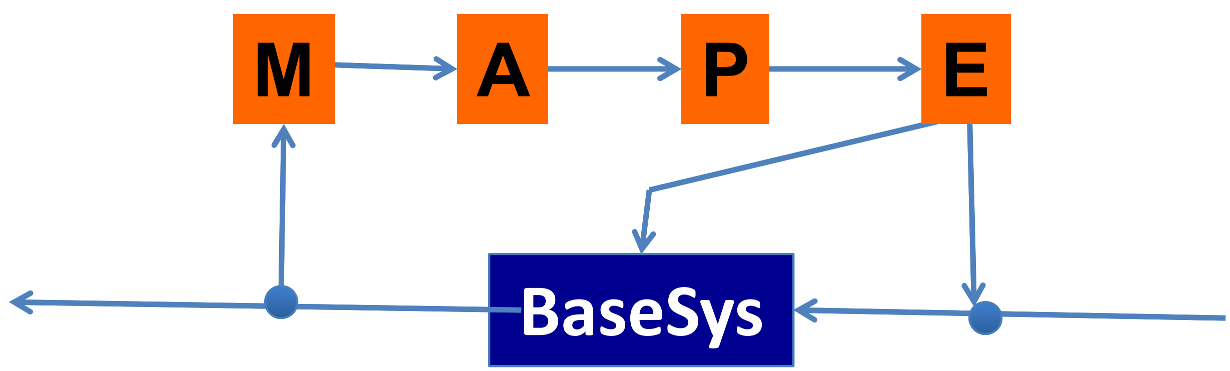


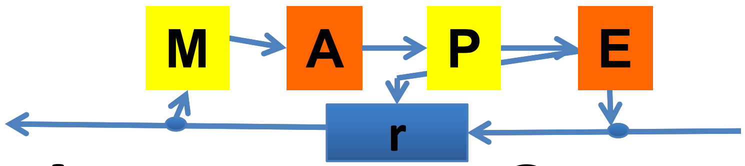
Engineering Adaptive Software Systems

- ***All*** Engineering disciplines are about concepts, tools and techniques for building systems of some sort.
- Our task then is to come up with such concepts, tools and techniques for building adaptive software systems (ADSSs).
- So, for each item on our agenda, we want to discuss what are the new concepts, tools and techniques.

What is special about ADSSs?

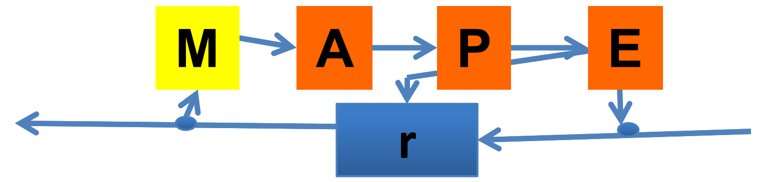
- More specifically, why can't we just use vanilla SE techniques?
- Baseline assumption: They are special, have a distinctive architecture that "separates concerns" between the base system and the adaptation mechanism (feedback/MAPE loop).





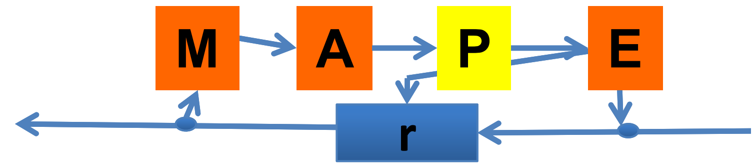
Q1: What requirements lead to a MAPE?

- Some answers:
  - ✓ Awareness requirements [Souza11]
    "Requirement R will not fail >3 times/mo"
  - ✓ Evolution requirements [Souza12]
    "If R fails >3 times/mo, change it to R-"
  - ✓ Adaptation requirements [Angelopoulos14]
    "Adaptations should not affect non-failing requirements"
  - ✓ Contextual requirements [Ali10]
    "R is a requirement only in context C"
- For each such requirement, we need a MAPE loop that monitors-analyzes-plans-executes.
- In all cases, we need tools and techniques for dealing with each proposed new class of requirements.



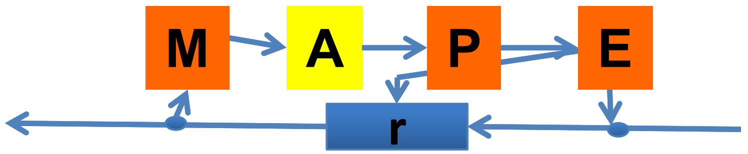
Q2: What do runtime reqs look like?

- Some answers:
  - ✓ They consist of hierarchies of goal and their instances capturing state, history and behaviour [Dalpiaz13].
  - ✓ Fuzzy runtime requirements models [Baresi10];
  - ✓ Runtime requirements models for reflection [Bencomo10];
  - ✓ …
- This is a critical issue because monitoring is rendered intractable and non-scalable for an over-expressive runtime requirements modeling language.



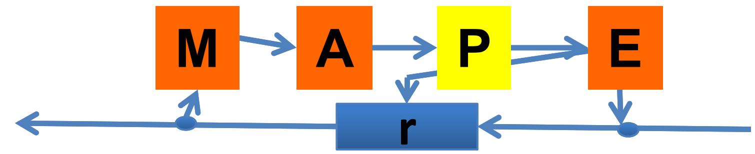
Q3: What failures trigger adaptation?

- Some answers:
  - ✓ Failures of requirements [Souza11]
    “Requirement R will not fail >3 times/mo”
  - ✓ Faults of the system (bugs/hardware breakdowns/…) [Wang07]
  - ✓ Changes of context [Ali10] ???
  - ✓ …
- A good answer here would account for many of the situations where adaptation is needed.



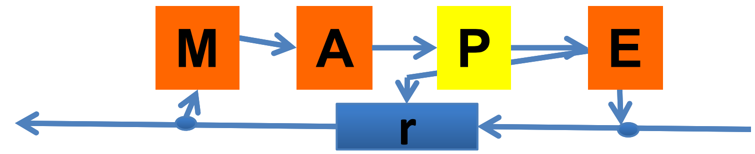
Q4: Diagnosing the problem

- A failure is often a symptom not the problem; e.g., if meeting scheduling is failing more than 3 times/mo (symptom), this may be due to too few meeting rooms, or overbooked participants.
- We need to be able to get to the "root causes" of a failure and decide on an adaptation accordingly.
- Some answers:
  - ✓ Use AI diagnostic techniques [Wang07]
  - ✓ … (many others)
- Diagnosis constitutes a form of abductive reasoning and is therefore intractable. In Wang's work the tool she built proved to be scalable.



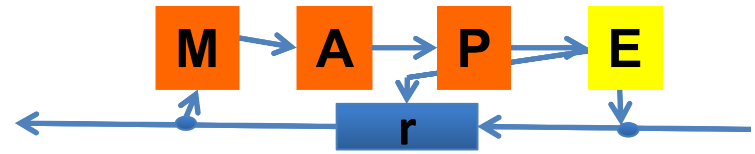
Q5: Systems with large adaptation spaces

- An adaptation space includes all possible adaptations supported by a given ADSS. The larger it is, the better!
- For example, in Vitor's thesis he used ***requirements*** (goal) models extended with control variables to define an adaptation space for an ADSS [Souza11].
- Rainbow, on the other hand, defines an adaptation space in terms of what changes can be made to the ADSS ***architecture***, e.g., add a server if performance deteriorates.
- Possible further answers: [Angelopoulos15], see presentation this afternoon.
- This issue bears striking resemblance to the problem of designing large variability spaces for product families.



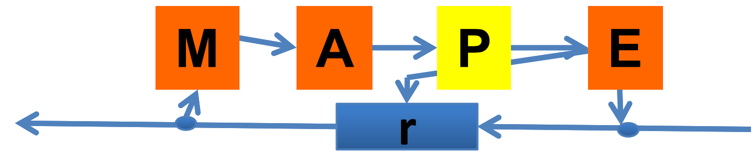
Q6: Dealing with multiple failures

- An adaptation that deals with one problem may interfere with adaptations that deal with others.
- Possible answers:
  - ✓ Rules, as in "If R1 and R2 are failing do A" [Garlan06]
  - ✓ Qualitative reasoning [Angelopoulos14]
  - ✓ …
- Using if-then rules can lead to an exponential number of rules. Qualitative reasoning does better, relying on adaptation requirements that define policies such as "If an adaptation A for problem P interferes with another problem P', don't use it".



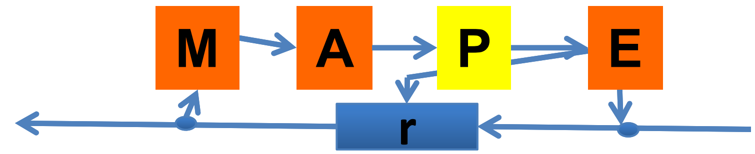
Q7: When can you reconfigure the system?

- Remember, we are talking about a running system that needs to be changed to a new configuration and with new values for its control variables.
- Possible answers:
  - ✓ Kramer et al [Kramer90]
  - ✓ Ghezzi et al [XiaoxingMa11]
  - ✓ …
- /* I'm not very familiar with this one …☹…



Q8: Optimizations for adaptation reqs

- In general, there are many adaptations that meet all the constraints at-hand, so we need additional constraints.
- Some requirements call for optimizing some objective function(s), e.g., "***Minimize*** meeting management costs while restoring R within 2mo", where R := "Meeting scheduling won't fail >3 times/mo".
- Now, you can't just assign 100 new rooms for meetings, which will solve the problem immediately but at great cost.
- The problem here is that we don't know how to do SAT-based and optimization-based reasoning together.
- Possible answers:
  - ✓ Use latest results from the Automated Reasoning community, maybe [Sebastiani15] ...
  - ✓ ???



Q9: The identification problem for ADSSs

- In order to choose among possible adaptations, you need to know the impact each one of them will have on the failing requirement(s).
- For example, to restore “Meeting scheduling won’t fail >3 times/mo”, we’d like to know by how much we’ll reduce failure rates for each added meeting room.
- Possible answers:
  - ✓ Guestimate qualitative “differential relations” [Souza11b]
  - ✓ Use case-based reasoning [Qian14].
  - ✓ Learn the relations over time.
  - ✓ …

Q10: Full adaptation[1]

- With the exception of evolution requirements, all the research questions discussed so far assume and/or adopt an “invariant requirements” assumption.
- Can we develop techniques that allow requirements to evolve due to failures in ways unanticipated at design time?
- A (very) preliminary answer: [Sabatucci15] proposes to use AI planning and domain knowledge to propose new requirements at run-time (best paper award, SEAMS 2015).

---

[1] See Jeff Kramer’s excellent keynote talk at SEAMS 2015.

Conclusions

- I have tried to summarize some of the major research questions we encountered in our research on ADSSs over the past decade.
- Unlike Mathematics where a solution to an open problem “closes” the problem, in Engineering many solutions are welcome until one is found that is good enough for practice.
- There is room for more solutions for all ten of the questions discussed here …

References

- [Ali10] Ali R., Dalpiaz F., Giorgini P., “A Goal-Based Framework for Contextual Requirements Modeling and Analysis”, *Requirements Engineering 15(4)*, Springer-Verlag, 2010, 439-458.
- [Angelopoulos14] Angelopoulos K., Souza V., Mylopoulos J., “Dealing with Multiple Failures in Zanshin: A Control-Theoretic Approach”, 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Hyderabad, May 2014.
- [Angelopoulos15] Angelopoulos K., Silva Souza V., Mylopoulos J., “Capturing Variability in Adaptation Spaces: A Three-Peaks Approach”, 34th International Conference on Conceptual Modelling (ER’15), Stockholm, October 2015.
- [Baresi10] Baresi L., Pasquale L., Spoletini P., “Fuzzy Goals for Requirements-Driven Adaptation”, 18th IEEE International Requirements Engineering Conference, Sydney, September 2010.
- [Bencomo10] Bencomo N., Whittle J., Sawyer P., Finkelstein A., Letier A., "Requirements Reflection: Requirements as Runtime Entities" 32nd ACM/IEEE International Conference on Software Engineering, Capetown South Africa, May 2010, 199-202.
- [Dalpiaz13] Dalpiaz F., Borgida A., Horkoff J., Mylopoulos J., “Runtime Goal Models”, 7th IEEE International Conference on Research Challenges in Information Science (RCIS’13), Paris, May 2013.

References (cont'd)

- [Garlan04] Garlan D., Shang-Wen Cheng, An-Cheng Huang, Schmerl B., Steenkiste P., "Architecture-Based Self-Adaptation with Reusable Infrastructure", *IEEE Computer*, Oct. 2004.
- [Garlan06] Shang-Wen Cheng, Garlan D., Schmerl B., "Architecture-Based Self-Adaptation in the Presence of Multiple Objectives", 2nd Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Shanghai, May 2006.
- [Kramer90] Kramer J., McGee J., "The Evolving Philosophers' Problem: Dynamic Change Management" *IEEE Transactions on Software Engineering 16(11)*, 1990, 1293-1306.
- [Sabatucci15] Sabatucci L., Cossentino M., "From Means-Ends Analysis to Proactive Means-Ends Reasoning", 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Florence, May 2015.
- [Sebastiani15] R. Sebastiani R., Tomasi S., "Optimization Modulo Theories with Linear Rational Costs", *ACM Transactions on Computational Logics*, 16(2), (to appear).
- [Souza11] Souza V., Lapouchnian A., Robinson W., Mylopoulos J., "Awareness Requirements for Adaptive Systems", 6th ICSE International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'11), Waikiki Honolulu, May 2011.

References (cont'd)

- [Souza11b] Souza V., Lapouchnian A., Mylopoulos J., "System Identification for Adaptive Software Systems: A Requirements Engineering Perspective", 30th International Conference on Conceptual Modelling (ER'11), Brussels, November 2011, 346-361.
- [Souza12] Souza V., Lapouchnian A., and Mylopoulos J., "(Requirement) Evolution Requirements for Adaptive Systems", 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS'12), Zürich, June 2012.
- [Wang07] Wang, Y., McIlraith, S., Yu, Y., Mylopoulos, J., "An Automated Approach for Monitoring and Diagnosing Requirements", 22nd IEEE/ACM International Conference on Automated Software Engineering (ASE'07), Atlanta, November 2007, 293-302.
- [XiaoxingMa11] Xiaoxing Ma, Baresi L., Ghezzi C., Panzica La Manna V., Jian Lu, "Version-Consistent Dynamic Reconfiguration of Component-Based Distributed Systems" 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering, Szeged Hungary, September 2011.