Welcome to SEENG 480B / CSC 485A / CSC 586A Self-Adaptive and Self-Managing Systems

Dr. Hausi A. Müller
and Lorena Castañeda
Department of Computer Science
University of Victoria

http://courses.seng.uvic.ca/courses/2015/summer/seng/480a
http://courses.seng.uvic.ca/courses/2015/summer/csc/485a
http://courses.seng.uvic.ca/courses/2015/summer/csc/586a

Announcements

- Midterm II
  - Thu, July 16 in class
- A4
  - Posted by Tuesday, July 14
  - Due Friday, July 31
- A3
  - Due Friday, July 10
  - July 13 Part 2 demos
    before and after class
    Sign up for demos (!)
- A2 grading questions
  - Ron Desmarais
  - Mon, July 13
    4-5 pm in ECS 415
- Grad project
  - Posted
  - Due Friday, July 24
  - Presentations Mon, July 27 and Thu, July 30
  - All students are expected to assess presentation as part of course participation mark

July Calendar

- July 9—MRAC and MIAC class
- July 10—A3 due
- July 13—MART class continued and A3 P2 demos
- July 16—Midterm II in class
- July 20/23—Characterizing SAS Problems
- July 24 Grad Presentation Slides due
- July 27/30—Grad Presentations
  - Non-presenters evaluate presentations
- July 31 A4 due

Demo details - July 13, 2015

ECS 660 [9:30 – 11:30]
The demos start at 10:00, the room is available to students from 9:45 to get the set up ready.

ECS 418 [12:15 – 12:55]
The demos are 5 min long. there is 5 min extra between demos to allow groups' set up given the restrictions of the space.

Email submit@rigiresearch.com to take a time slot, these will be assigned in order of arrival

NOTE: Students with classes in the morning will have preference for section 4 which is during the class time. Please be considerate about this restriction to your fellow classmates and take a morning time slot if you don't have class (or other impediment) in the morning

Place	Section	Time	No. Groups
ECS 660	1A	10:00 - 10:05	[1]
	1B	10:05 - 10:10	[1]
	1C	10:10 - 10:15	[1]
	1D	10:15 - 10:20	[1]
	2A	10:25 - 10:30	[1]
	2B	10:30 - 10:35	[1]
	2C	10:35 - 10:40	[1]
	2D	10:40 - 10:45	[1]
	3A	10:50 - 10:55	[2]
	3B	10:55 - 11:00	[2]
	3C	11:00 - 11:05	[2]
	3D	11:05 - 11:10	[2]
ECS 418	4A	12:15 - 12:20	[1]
	4B	12:25 - 12:30	[1]
	4C	12:35 - 12:40	[1]
	4D	12:45 - 12:50	[1]
	4E	12:50 - 12:55	[1]

Assignment 3 Demos on Monday, July 13

Graduate Student Research Paper Presentations

- Brun, Y., Di Marzo Serugendo, G., Gacek, C. Giese, H. Kienle, H.M., Litoiu, M., Müller, H.M., Pezzè, M., Shaw, M.: Engineering Self-Adaptive Systems through Feedback Loops. Software Engineering for Self-Adaptive Systems, pp. 48-70 (2009) — **Presentation by Simar Arora Khushboo Gandhi: July 27**
- Garlan, D., Cheng, S.-W., Huang, A.-C., Schmerl, B., Steenkiste, P.: Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure. IEEE Computer 37(10):46-54 (2004) — **Presentation by Stephan Heinemann and Waseem Ullah: July 27**
- Oreizy, P., Medvidovic, N., Taylor, R.N.: Runtime Software Adaptation: Framework, Approaches, and Styles. In: ACM/IEEE International Conference on Software Engineering (ICSE 2008), pp. 899-910 (2008) — **Presentation by Sumit Kadyan and Adithya Rathakrishnan: July 27**
- Kramer, J., Magee, J.: Self-Managed Systems: An Architectural Challenge. In: ACM /IEEE International Conference on Software Engineering 2007 Future of Software Engineering (ICSE), pp. 259-268 (2007) — **Presentation by Ernest Aaron and Harshit Jain : July 27**

Graduate Student Research Paper Presentations

- Aksanli, J. Venkatesh, L.Z., Tajana R.: Utilizing Green Energy Prediction to Schedule Mixed Batch and Service Jobs in Data Centers. In: Proceedings 4th Workshop on Power-Aware Computing and System (HotPower 2011), Article 5 (2011) — **Presentation by Junnan Lu and Francis Harrison: July 30**
- Ebrahimi, S., Villegas, N.M., Müller, H.A., Thomo, A.: SmarterDeals: a context-aware deal recommendation system based on the SmarterContext engine. CASCON 2012: 116-130 (2012) — **Presentation by Carlene Lebeuf and Maria Ferman: July 30**
- Villegas, N.M., Müller, H.A., Tamura, G., Duchien, L., Casallas, R.: A framework for evaluating quality-driven self-adaptive software systems. In: Proc. 6th Int. Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2011), pp. 80-89 (2011) — **Presentation by Parminder Kaur and Navpreet Kaur: July 30**
- Villegas, N.M., G. Tamura, H.A. Müller, L. Duchien, and R. Casallas, DYNAMICO: A reference model for governing control objectives and context relevance in self-adaptive software systems, in: R. de Lemos, H. Giese, H.A. Müller, and M. Shaw (Eds.), Software Engineering for Self-Adaptive Systems, LNCS 7475, Dagstuhl Seminar 10431. Springer, pp. 265-293 (2013) — **Presentation by Arturo Reyes Lopez and Babak Tootoonchi,: July 30**

Guidelines for Grad Student Presentations

- Format of presentation
  - Presentation 10 mins
  - Q&A 5 mins
  - Practice talk (!!)
  - Practice of the best of all instructors
- Slides
  - High quality and polished
  - Submit slides by July 24 to instructor for approval
  - Submit final slides 1 day after presentation for posting on website
- Talk outline
  - Motivation
  - Problem
  - Approach
  - Contributions of the paper
  - Relation to what we learned in the course so far
- Assessment
  - All students have to fill out an evaluation form
  - Counts towards class participation

Presentation Assessment

Evaluator's name:		
Graduate students:		
Quality of presentation		
Did I learn something? Did the presentation stimulate my interest?	5	
Do I know now what the paper is all about?	5	
Does the presenter know the subject well?	5	
Presentation style: main points reiterated; positive attitude; excited about the subject.	5	
How did the presenter perform in the Q&A session?	5	
Subtotal	**25**	
Other comments		

July 27 and July 30 CSC 586A Presentations

Midterm II

Thu, July 16 in class

- All materials presented in class including Mon, July 13
  - Before and after Midterm I
  - More questions from after Midterm I
  - All on-line lecture notes
- Study sample Midterm II questions carefully

- Format
  - Same format as Midterm I
  - Crib sheet in the form of a paper
  - Argue convincingly
  - Define terms
  - Essay questions

No cheating

Crib Sheet for Midterm II

- **Crib sheet**: a concise set of notes for quick reference

  H.A. Müller and N.M. Villegas: Runtime Evolution of Highly Dynamic Software, in *Evolving Software Systems,* T. Mens, A. Serebrenik, and A. Cleve (Eds.), Springer, pp. 229-264 (2014)

  http://link.springer.com/chapter/10.1007%2F978-3-642-45398-4_8

- Summarizes a significant part of this course
- You will have access to a hard copy during Midterm II
- Contains answers to selected Midterm II questions

Topics
Autonomic Computing

- Autonomic manager
- MAPE-K loop
- Monitoring
- Analysis
- Symptoms
- Planning
- Policies
  - Action
  - Goal
  - Utility-function
- Sensing
- Actuating
- Knowledge bases for AC
- ACRA
- Manageability interfaces
- Models at runtime
- MART
- Uncertainty

Topics
Control loops

- Types of feedback: positive, negative, bipolar
- Hellerstein feedback loop model
- Controller
- Managed element, process, plant
- Disturbance input
- Noise input
- Transducer
- Reference model
- Simulation model
- Model identification
- MIAC
- MRAC
- PID controller

Interesting Potential Midterm II Questions

- Design a concrete and viable
  - action policy
  - goal policy
  - utility-function policy
  - Design a Green utility-function policy
  - How can cost be integrated into a utility-function?
- PID controllers
- Explain the notion of adaptive control
  - MRAC architecture
  - MIAC architecture
  - How do they relate?
  - How do they relate to ACRA?

Interesting Potential Midterm II Questions

- What is the difference between anticipated and un-anticipated adaptation?
- What is the difference between fully autonomous systems and human-in-the-loop systems?
- What is the difference between design-time and run-time adaptation?
- What are self-* properties?
- What are requirements at runtime?
- What are models at runtime (MART)?
- What is runtime V&V?

Interesting Potential Midterm II Questions

- What aspects of the environment should a self-adaptive system monitor?
  - The system cannot monitor everything in the environment
  - What aspects of the environment are truly relevant?
- How should a self-adaptive system react if it detects changes in the environment?
  - Maintain high-level goals
  - Relax non-critical goals to allow the system a degree of flexibility
  - Goal trade-off analysis

Course Requirements

Unit	Undergrads Weight	Grads Weight	Remarks
A1	12%	9%	Due Fri, May 29, 2015
A2	12%	9%	Due Fri, June 19, 2015
A3	12%	9%	Due Fri, July 10, 2015
A4	12%	9%	Due Fri, July 31, 2015
Grad Project		12%	Due Sat, July 25, 2015
Participation and presentation	7%	7%	Only graduate students are required to give a presentation towards the end of the course.
Midterm 1	20%	20%	June 4, 2015 in class. Closed books, closed notes, no phones, no computers, no calculators, no gadgets.
Midterm 2	25%	25%	July 16, 2015 in class. Closed books, closed notes, no phones, no computers, no calculators, no gadgets.
Total	100%	100%	Have a great course!

- All materials discussed in class are required for the midterm examinations
- Completing all midterms and assignments is required to pass the course
- Passing the midterms is not absolutely required to pass the course, but of course highly recommended

Reading Assignments

- “Models@run.time” Blair et al. 2009 http://dx.doi.org/10.1109/MC.2009.326
- “Models@run.time to Support Dynamic Adaptation” Morin et al. 2009 http://dx.doi.org/10.1109/MC.2009.327
- “The role of models@run.time in supporting on-the-fly interoperability” Bencomo et al. 2013 http://link.springer.com/article/10.1007%2Fs00607-012-0224-x
- “Living with uncertainty in the age of runtime models” Holger Giese et al. 2014 http://link.springer.com/chapter/10.1007%2F978-3-319-08915-7_3

Other reading material

- Models@run.time Foundations, Applications, and Roadma
Editors: Bencomo, N., France, R.B., Cheng, B.H.C., Aßman
http://www.springer.com/us/book/9783319089140

Model at runtime - MART

(models@run.time, runtime models)

Models at runtime (MART)

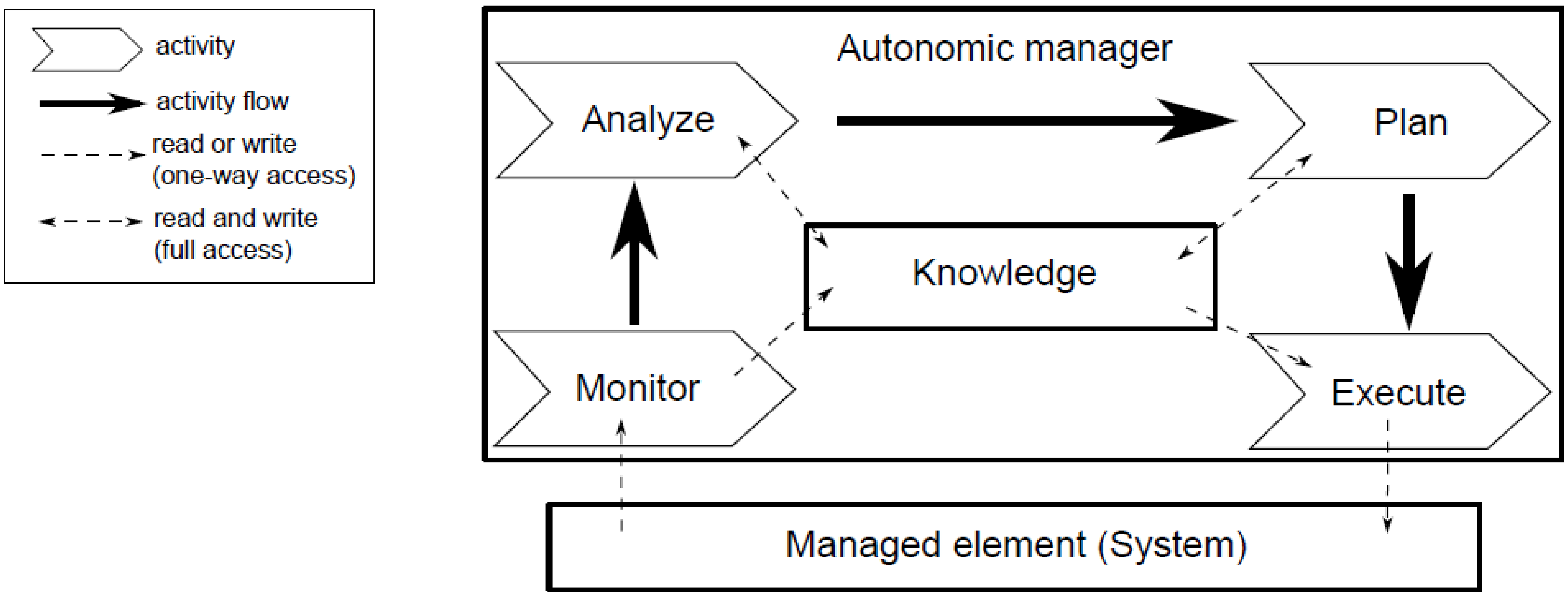
- represent the system’s complete environment (possibly more than one MART for a system), up-to-date information (i.e., context, users, and requirements)
- is accessible at runtime by the system, available in the form of software artefacts
- the system must be causally connected, are implemented to support runtime events
- manipulable and capable to evolve during execution time

Uncertainty

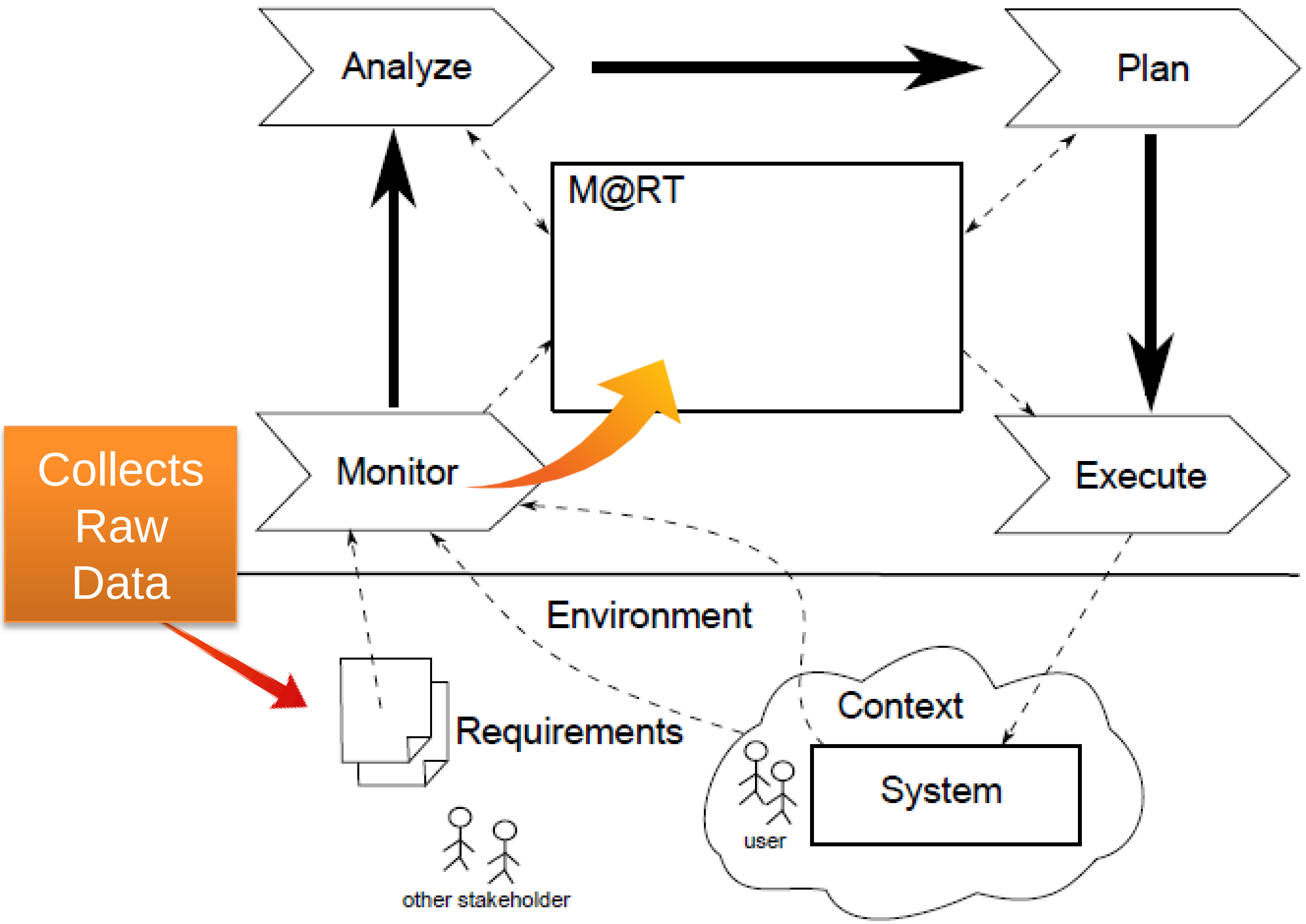
Uncertainty within a model is the difference between the amount of information about the original and the information that the model could, in theory, represent about the original at a certain instant in the system lifetime [Giese 2014]

- MART deal with the uncertainty of the context which makes predictions a real challenge
- Dynamic models increase the level of uncertainty over time because of the “possible” continuous updates in order to reflect changes in the original.

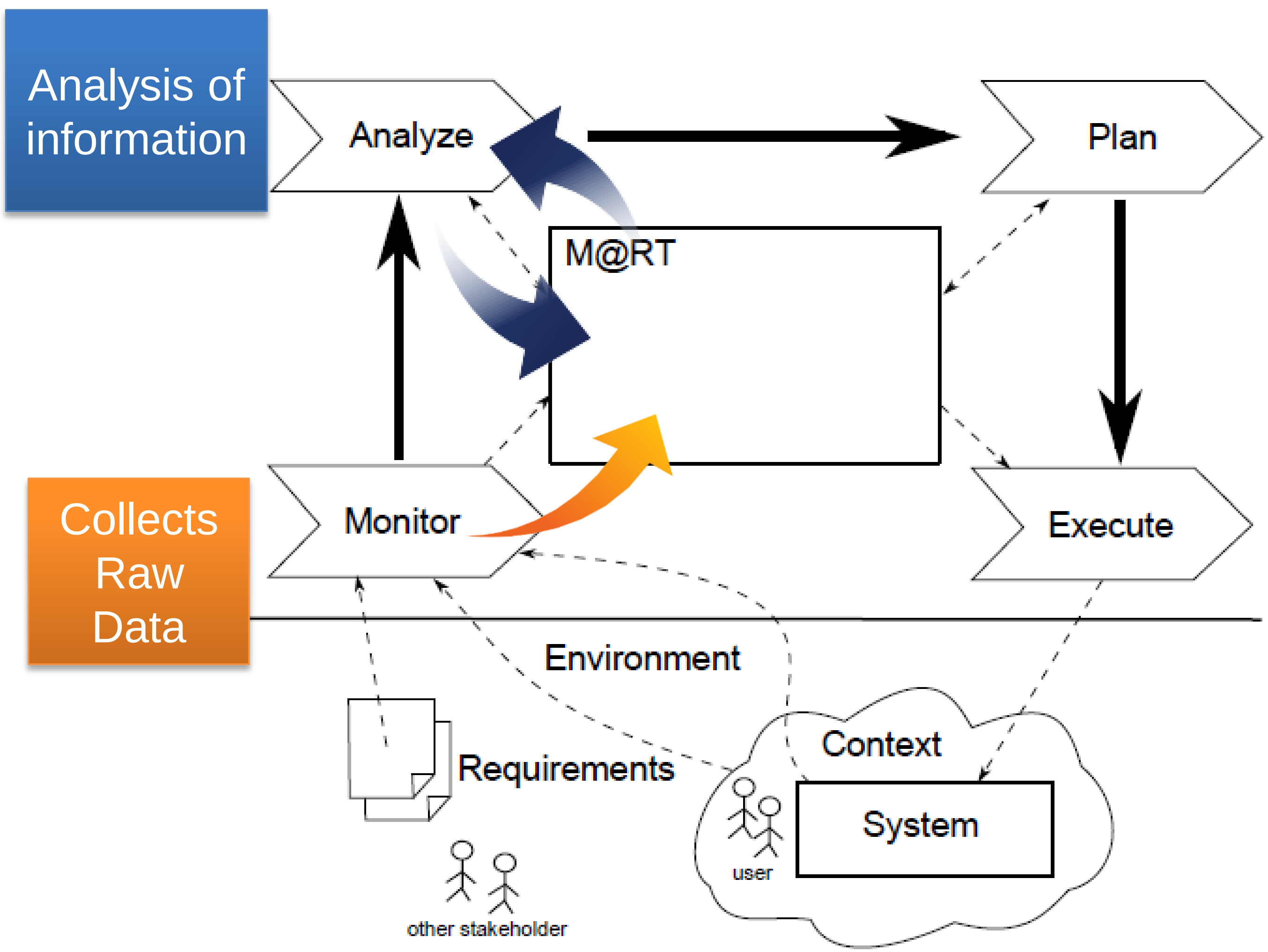
MAPE-K feedback loop



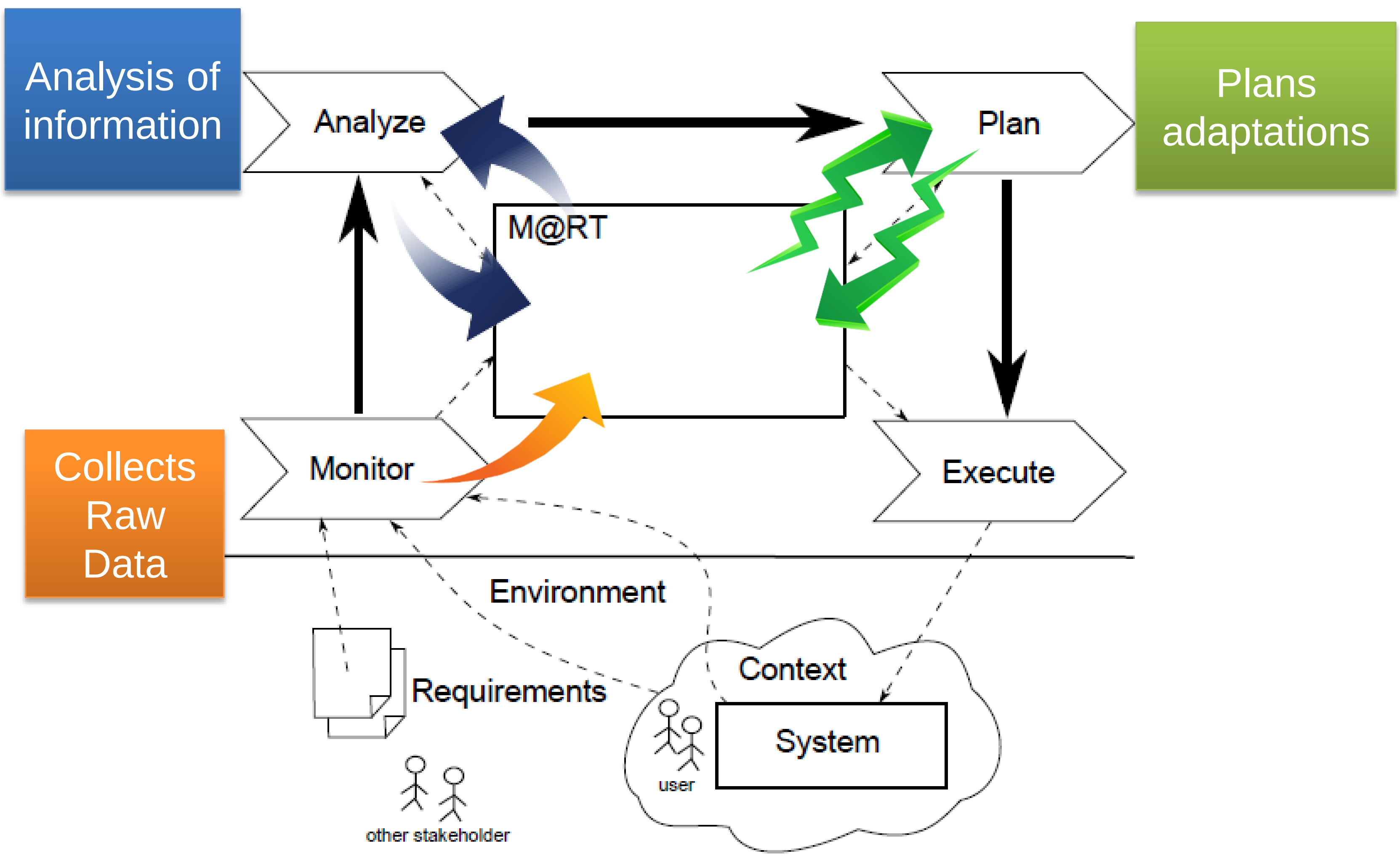
Extended MAPE-K



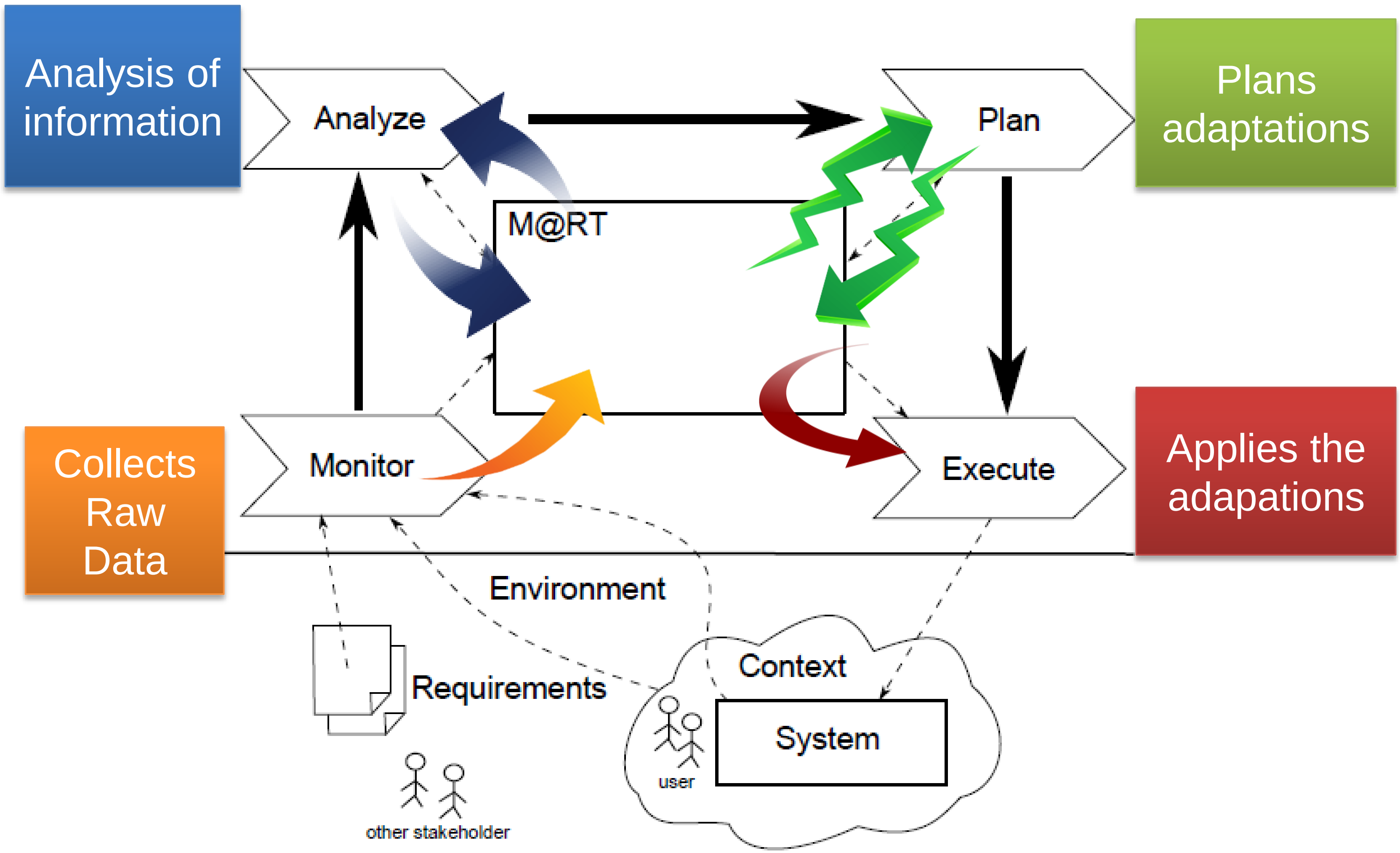
Extended MAPE-K



Extended MAPE-K



Extended MAPE-K



Uncertainty

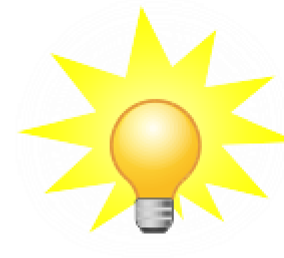
Uncertainty within a model is the difference between the amount of information about the original and the information that the model could, in theory, represent about the original at a certain instant in the system lifetime [Giese 2014]

- MART deal with the uncertainty of the context which makes predictions a real challenge
- Dynamic models increase the level of uncertainty over time because of the “possible” continuous updates in order to reflect changes in the original.

Epistemic uncertainty

- ***The development-time models do not reflect the system or the context at execution time*** ***[Giese 2014]***
- The design-time requirements do not reflect the real needs. The requirements were ambiguous
- The requirements (expectations) and the resulting software quality are in conflict
- Changes that might occur between development-time and deployment

Let's have a discussion

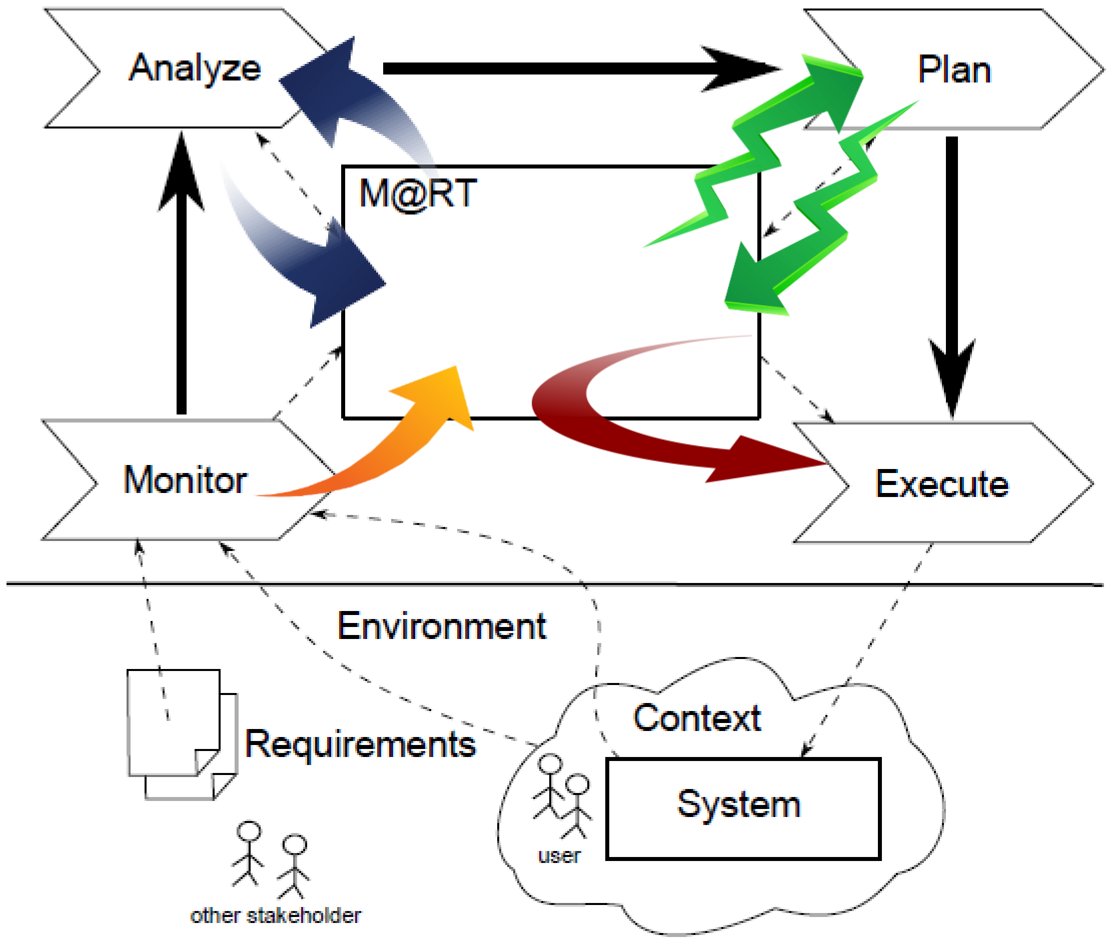


What are the problems of uncertainty in self-adaptive systems?

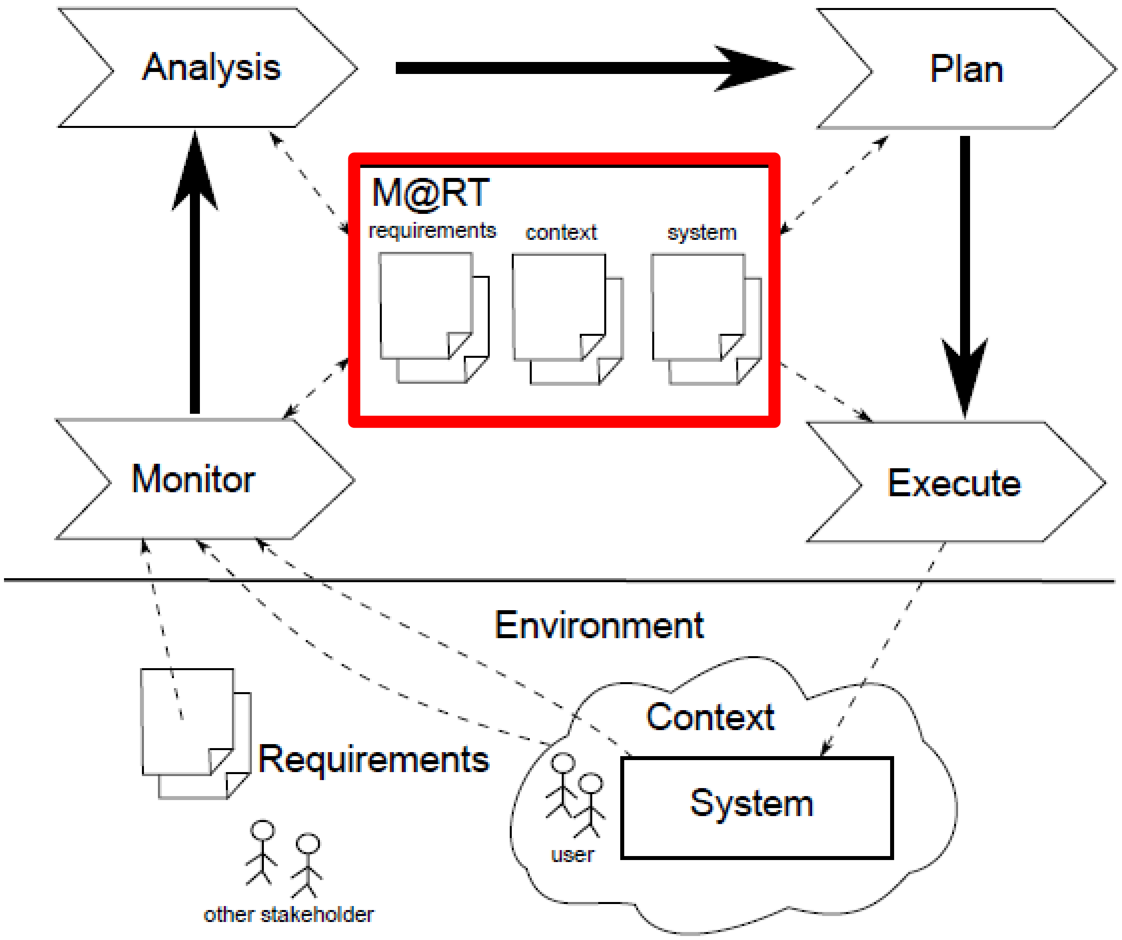
Do these capabilities help? How?

- Context-awareness
- Situation-awareness
- Requirements-awareness

MAPE-K to handle uncertainty?

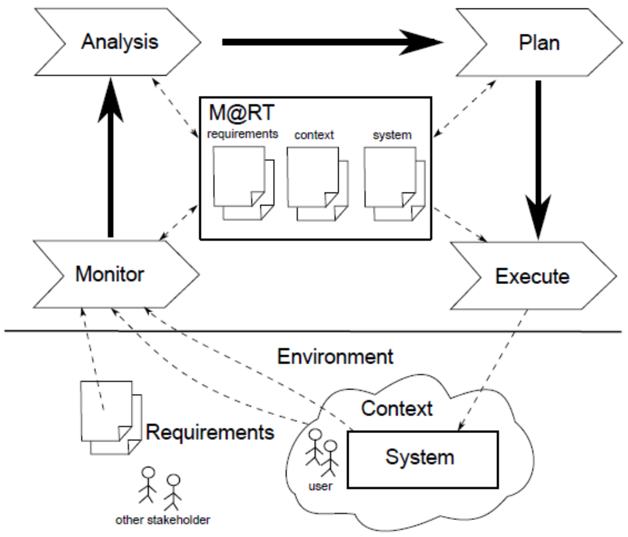


MART at different levels



Monitor

- Measures raw data
- Update the corresponding MART



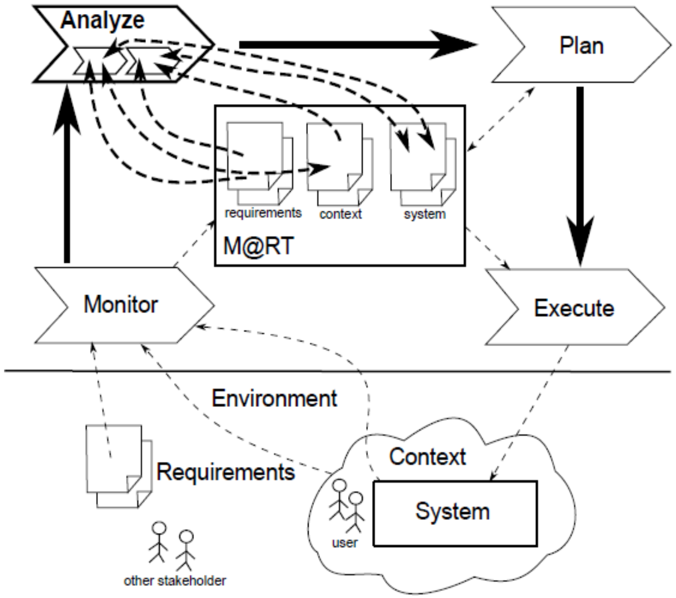
Uncertainty

- By updating constantly the information

Sensors are limited to the accuracy and precision of their measurements

Analyze

- Interprets data from the monitor and the MART in order to analyze new and old information
- Verifies that the monitored information satisfies the requirements



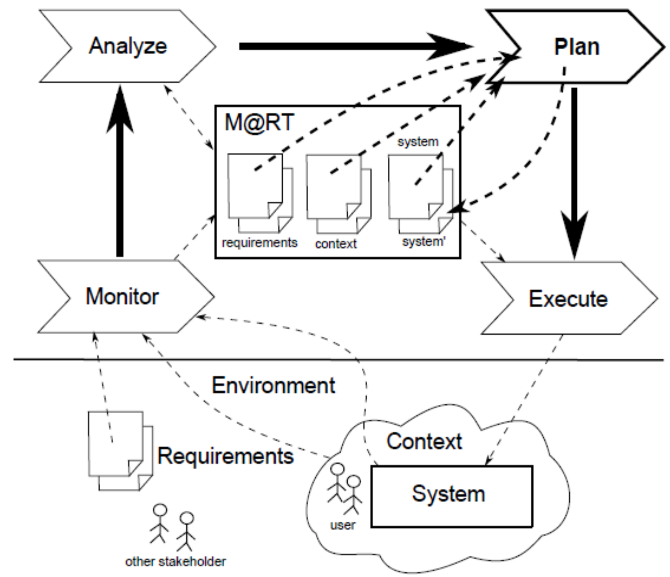
Uncertainty

- By using strategies such as observing, learning and updating

Analysis and diagnosis can be ambiguous and imprecise.

Plan

- Reads the runtime models enriched by the analysis and performs some reasoning to identify the best adaptations for the running system
- Records the system changes in the corresponding MART



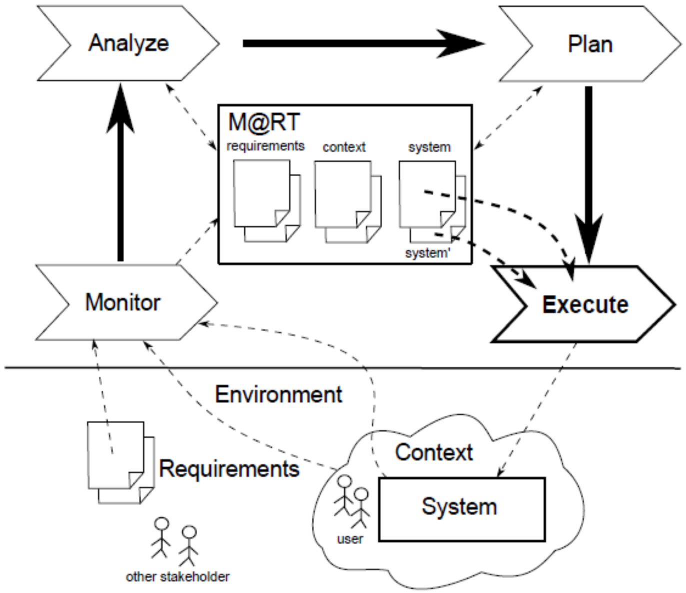
<u>Uncertainty</u>

- The plan takes the form of a prediction of the future state of the system

The precision and accuracy of the prediction depends of the uncertainty in the MART

Execute

- Applies the set of changes stored in the MART updated by the Planner
- Acts as the causal connection between the MART and the running system



<u>Uncertainty</u>

- Applying changes requires an additional loop to verify inconsistencies → Time is an issue

The execution can not give any guarantees that the MART will be in complete sync with the system

Some research questions

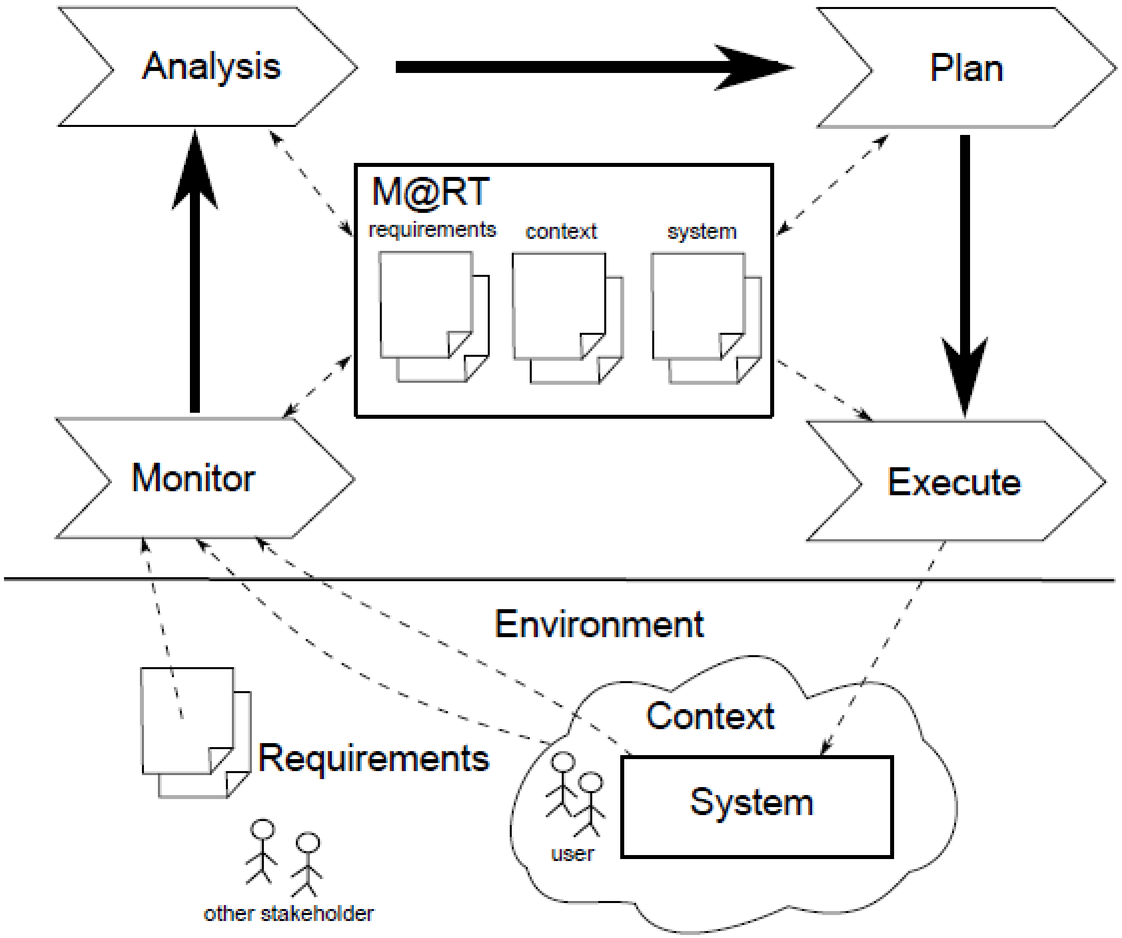
[Giese 2014]

- How can we determine the imprecision caused by temporal constrains / delays?
- Does the MAPE-K also need to adapt?
- How do MART represent what to monitor and how to do it?
- Does the perspective of “what is relevant” for the MART need to adapt at execution time?
- Should the criteria for decision-making in the analyser adapt itself?
- How does the planner handles strategies when uncertainty exist?
- Do temporal delays create an inconsistent view of the MART?
- How are MART affected by external influences outside the MAPE-K loop?

Summary

Models at runtime (MART)

- ✓ Represent the system's up-to-date information
- ✓ Is accessible at runtime
- ✓ The system must be causally connected
- ✓ Manipulable and capable to evolve during execution time



Demo Section 4

ECS 418	4A	12:15 - 12:20	[1]: G11
	4B	12:25 - 12:30	[1]: G5
	4C	12:35 - 12:40	[1]: G12
	4D	12:45 - 12:50	[1]: G9
	4E	12:50 - 12:55	[1]: G14