

# Self-adaptive Service Based Applications: **Challenges in Requirements Engineering**



Anna Perini

Center for Information Technology CIT-Irst, FBK  
Trento, Italy

*Keynote at  
16/05/12*



# Google's Driverless Car & Car2X



radar,  
GPS,  
a rotating roof sensor  
video camera

video camera

radar

[http://money.cnn.com/2012/05/10/autos/google-driverless-cars-safety/?source=cnn\\_bin](http://money.cnn.com/2012/05/10/autos/google-driverless-cars-safety/?source=cnn_bin)

**P2P cars comm.**

**Google map**

odometer

**Traffic management  
comm.**

**Traffic info**

**Passenger services**

**Green Driving**

"Integrating services into car environment" W. Wahlster 2012

- Google's Driverless Car; Car2X
- Travel Domain (e.g. Travel Companion)

...

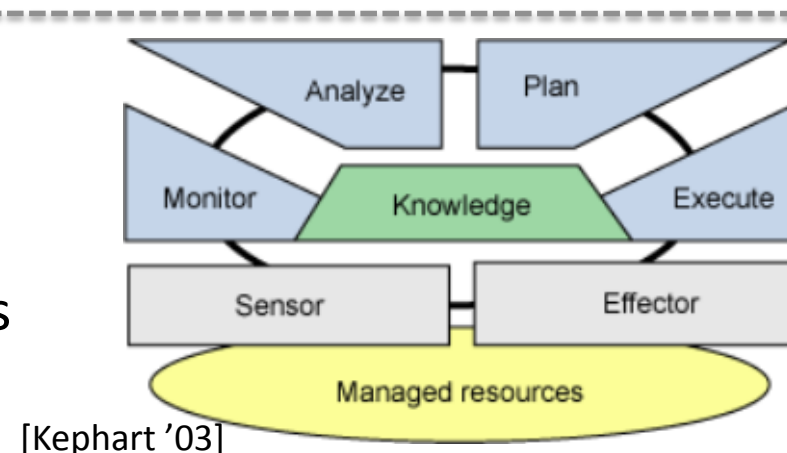
Examples of Self-Adaptive Service-Based Applications (SBA) which rest on multiple services, made available over the Internet by **different** service providers, which are **dynamically** discovered, selected and composed **at run-time**

- By nature SBA are:
  - **Distributed** systems, they run over the Internet
  - Not under direct **control** of their developers, services can be **used**, but they are **owned by third parties**
  - Live in an **open world**, external events can create unforeseen situations
  - **Dynamic**, since services used in a composition can be discovered and selected dynamically at run-time
- With self-adaptivity mechanisms we make them :
  - **Context-aware**, that is able to assess: availability of needed operational resource (e.g. bandwidth), availability of component services ... and the state of real objects they manipulate
  - Able to **prevent failure** and to enact **reconciliation** or **compensation** mechanisms if service qualities are degrading
  - Accessible seamlessly through a **variety of devices** (e.g. smart phones, laptop), by **heterogeneous end-users**, with different skills and changing needs and preferences on the go

- “Self-adaptive software **evaluates** its own behavior and **changes** behavior when the evaluation indicates that it is not accomplishing what the software is intended to do, or when better functionality or performance is possible ... This implies that the software has **multiple** ways of accomplishing its purpose, and has **enough knowledge of its construction** to make effective changes at runtime.”

Definition reported as DARPA Broad Agency Announcement on Self Adaptive Software (BAA-98-12) [Laddaga 1997], cited in following publications by Robertson&Laddaga

- Realized as Monitor-Adapt internal control loop in SBA
- And more generally in the MAPE-K architecture of autonomic elements proposed in autonomic computing



[Kephart '03]

- Middleware for enabling context aware, self-adaptive SBA (e.g. MUSIC, DiVA European projects)
- A considerable amount of work on technology, methods and standards for engineering self-adaptive SBA has been developed (e.g. ref. S-Cube network results: [s-cube-network.eu](http://s-cube-network.eu)):
  - Service infrastructure (SOC and grid computing)
  - Service composition and coordination
    - Context-aware automated retrieval and composition of services based on AI techniques (e.g. AI planning [Bertoli et al. 2010])
    - Verification and cross-cutting monitoring&adaptation (e.g. [Kazhamiakin et al. 2010])
  - Business Process Management
  - Methods for Analysis, Design, Development, Quality Assurance of services, Testing ...
    - E.g. SOA paradigm; life cycles models, specification languages

That is the machinery for engineering self-adaptive SBA seem well defined and consolidating ...



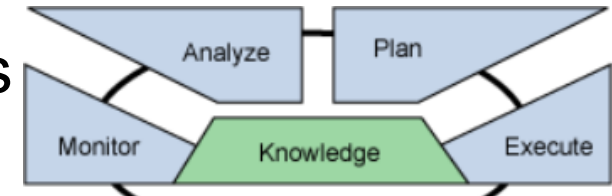
- Service Based Applications
  - Features
  - Self-adaptivity
- Requirements Engineering
  - Research challenges
  - Review of some recent work
- Conclusion & Open issues

- RE is conventionally defined as a systematic process aiming to
  - understand the problem domain (e.g stakeholder goals and strategic dependencies; domain assumptions)
  - identify the purpose for the system-to-be
  - build a specification for the system-to-be that is able to satisfy the goals of its users
- It is usually performed at the outset of the whole development process, at Design-Time
- Underlying assumption: “closed world”, i.e. static models are acceptable approximations
- How should we interpret these statements in the case of distributed, open and highly-dynamic systems, as self adaptive SBA?
- Is it needed an RE at Run-Time? ... and how should it be conceived?
  - Questions analyzed in the last 4-5 years

[Cheng et al, Daghtul 2008]; [Di Nitto et al. 2008]; [Sawyer Et al. RE'10]

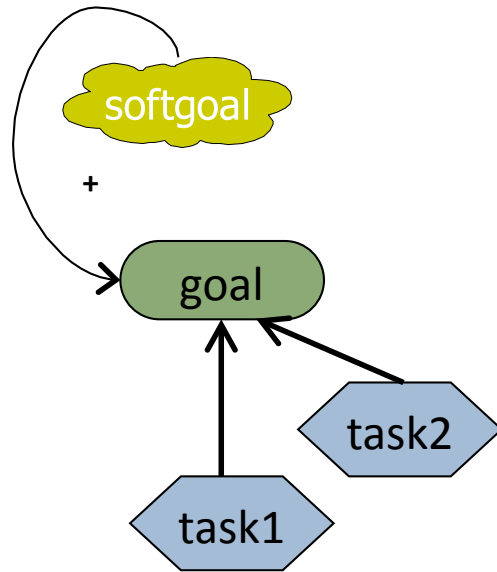


- Among main challenges:
  - Run-time representation of requirements
    - Imply: “alive requirements” and sort of Run-Time RE ( ≠ Design-Time RE?)
  - Uncertainty and unpredictable changes
    - i.e. limits of the knowledge available at Design-Time vs. limits of expressiveness of the languages used at DT
  - End-user centrality
    - From user types to personalization at individual level



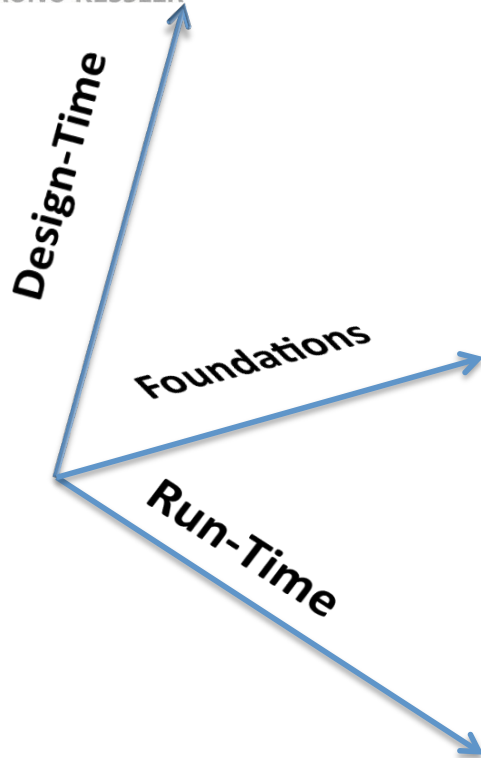
- Recent work address them ... I'll try to pick up essential features ...

- Recent work address self-adaptivity, among them:
  - Tropos4AS [Morandini Et al.2008-2011]
  - Adaptive Req. (ARML+CARE) [Qureshi Et al.2009-2012]
  - FLAGS [Pasquale Et al. 2010-2011]
  - Awareness Req. (Zanshin) [Souza Et al.2010-2012]
- Build on seminal work, including among the others requirements monitoring and context aware systems
  - E.g. from Fickas and Feather 2005 and following work, e.g [Robinson'06], on RE for DAS [Berry'05 ]; on context-aware systems (e.g.[Salifu'07])



- Capture **objectives** of stakeholders as **goals** of the system
- Decompose goals into smaller, more concrete ones (AND/OR decomposition)
- Goals are *operationalized* into a set of specifications (*tasks*) for the system-to-be, which satisfy the requirements (goals), given the domain assumption
- Identify **alternative** ways to achieve goals (variability design)
  - “Goal refinement generates **a space of possible specifications** and the requirements problem amounts to finding those that satisfy R” [Mylopoulos’12]
- Several GO approaches (e.g. i\*, KAOS, ...), with additional entities, e.g. Actor; dependencies, etc.
- **Techne** [Jureta Et al. RE10] adds Preferences & priorities; Quality constraints
  - a solution consists of a specification that satisfies all mandatory goals and a maximal consistent subset of preferred ones ... an **optimization** problem

# A View on RE for Self-adaptivity

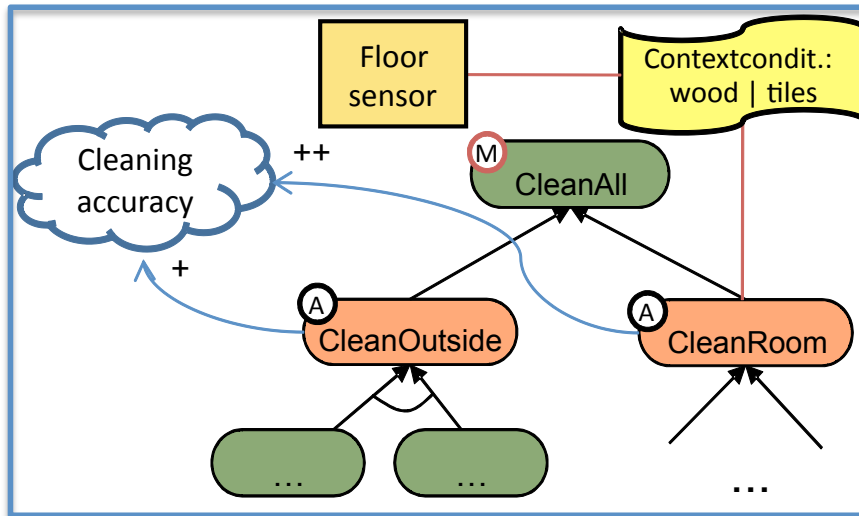


- Requirements Engineering perspectives:
  - a) Design-Time
    - An **ontology** for requirements to represent elicited information; **modeling primitives**
    - Analysis (e.g. variability, consistency, goal satisfaction)
  - b) Run-Time
    - Monitoring requirements
    - Reconfiguring for requirements satisfaction
    - Requirements Change management
  - c) Foundations
    - Notion of requirements compliance
    - Requirements problem formulation / underlying framework

- Service Based Applications
  - Features
  - Self-adaptivity
- Requirements Engineering
  - Research challenges
  - Review of some recent work
    - [Tropos4AS](#)
    - Adaptive Req. ([ARML+CARE](#))
    - [FLAGS](#)
    - Awareness Req. ([Zanshin](#))
- Conclusion & Open issues

# Tropos4AS [Morandini: SEAMS08, ASE08, AAMAS09]

Design-time

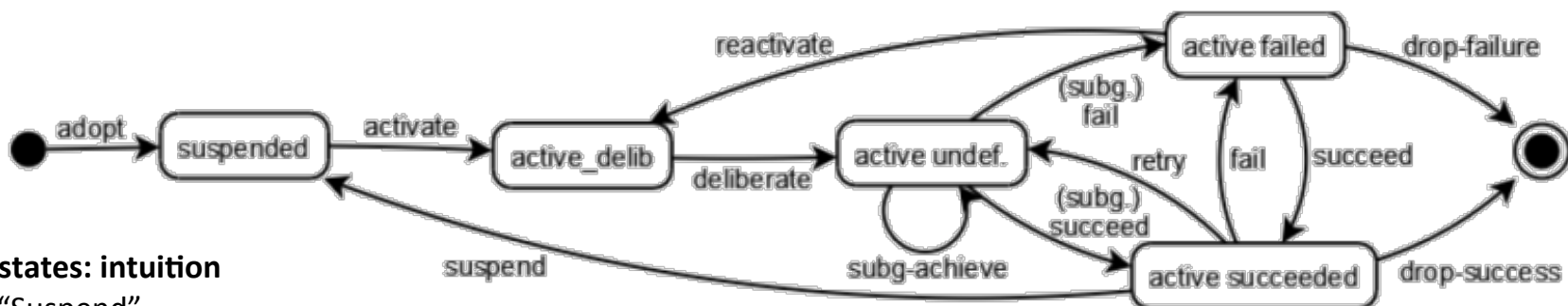


Tropos4AS goal model for a Cleaner-Agent

- Requirements specifications represented as **goal models** (Tropos foundation, {G, S, P, R});
- + explicit modeling of **goal types** (achieve, maintain, perform), **environment** and **failure**;
- + Relations between environment and conditions for goal creation, achievement, failure
- Variability design supports analysis for adaptation to **prevent failure** (i.e. expliciting alternative behaviours for goal satisfaction)

- 
- JADEX implementation of the goal model with all the alternatives
  - Operational semantics defined by transition rules

Run-time



## Goal states: intuition

- "Suspend"
- "Active, deliberate": find applicable subgoals
- "Active, undefined": subgoal achievement taking place
- "Active, succ./failed": "provisional" success/failure state.

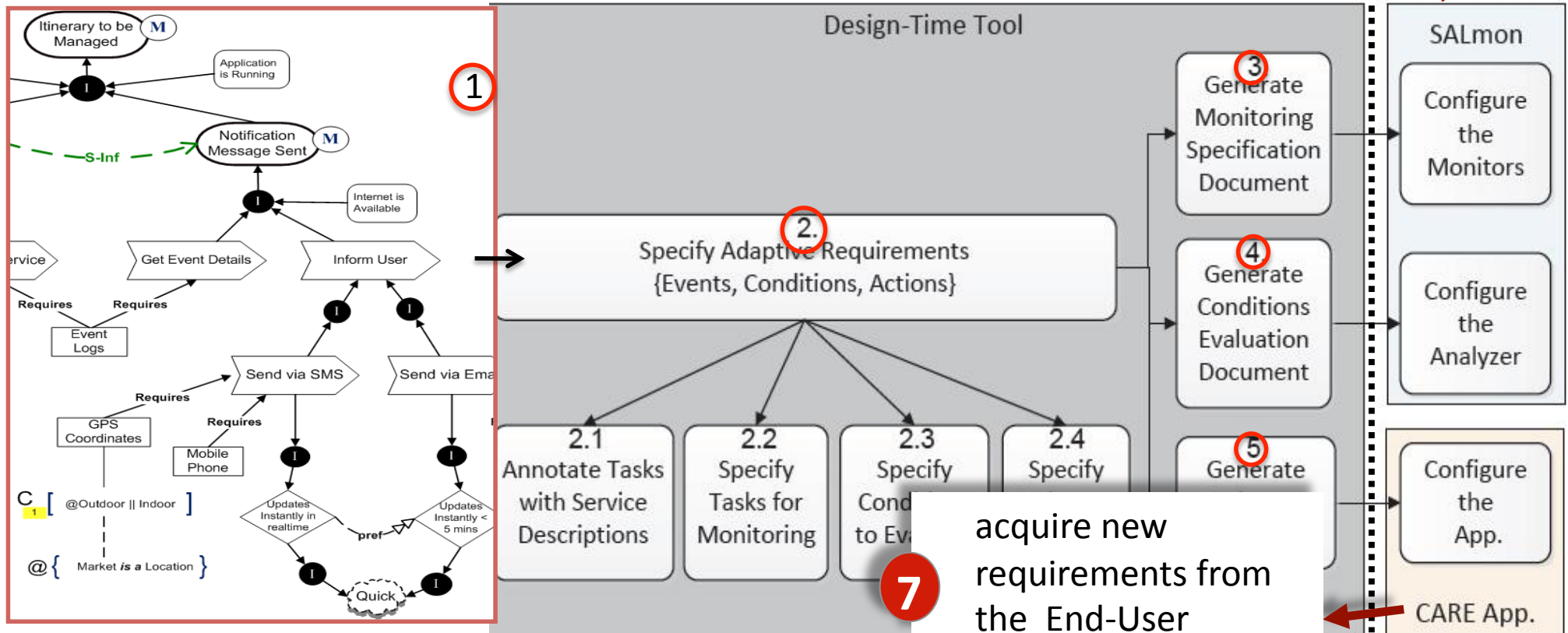
# ARML+CARE [Qureshi: SEAMS08, CAISE10, RE10, REFSQ12]

Design-time

- **Requirements** specifications represented as **goal models** (CORE ontology revisited, Techne foundation, {G, T, Q, D, **Context**, R; **relegate**, **influence** rels});
- **Adaptive reqs**: “A confirmation message for booking is generated **as soon** as the booking has been processed and sent to the user on her **current device** (e.g. PDA) by seamlessly observing: the **user’s context** (Profile, Location, Device) and QoS attributes until the message is delivered in a correct form for the device i.e. PDA, or an alternative notification is performed.”

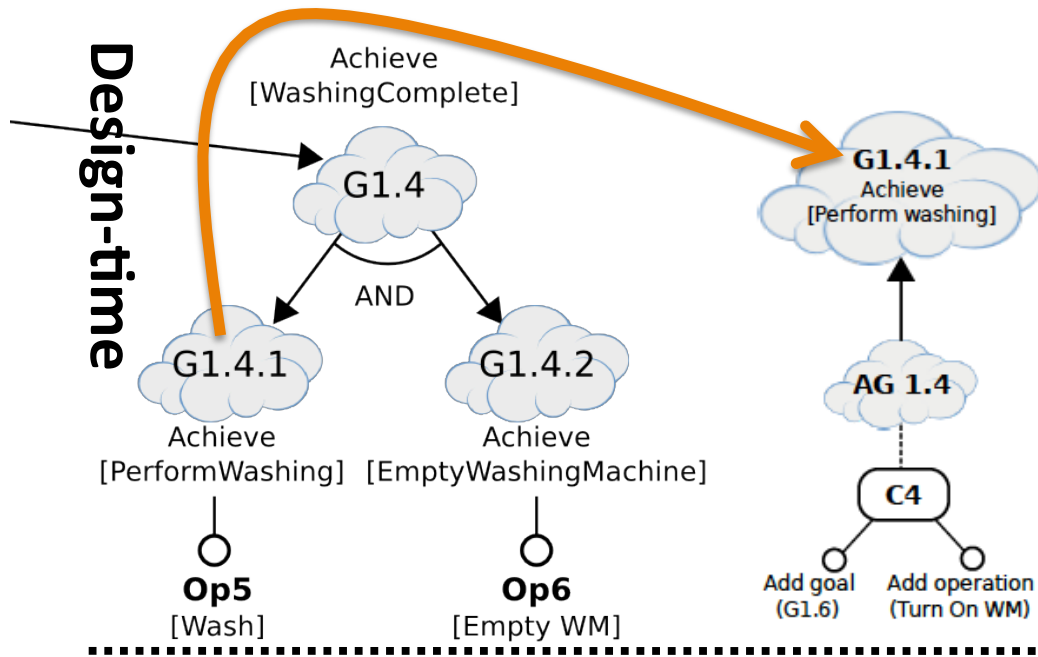
Run-time

Monitor env., identify context, adapt by selecting the suitable variant

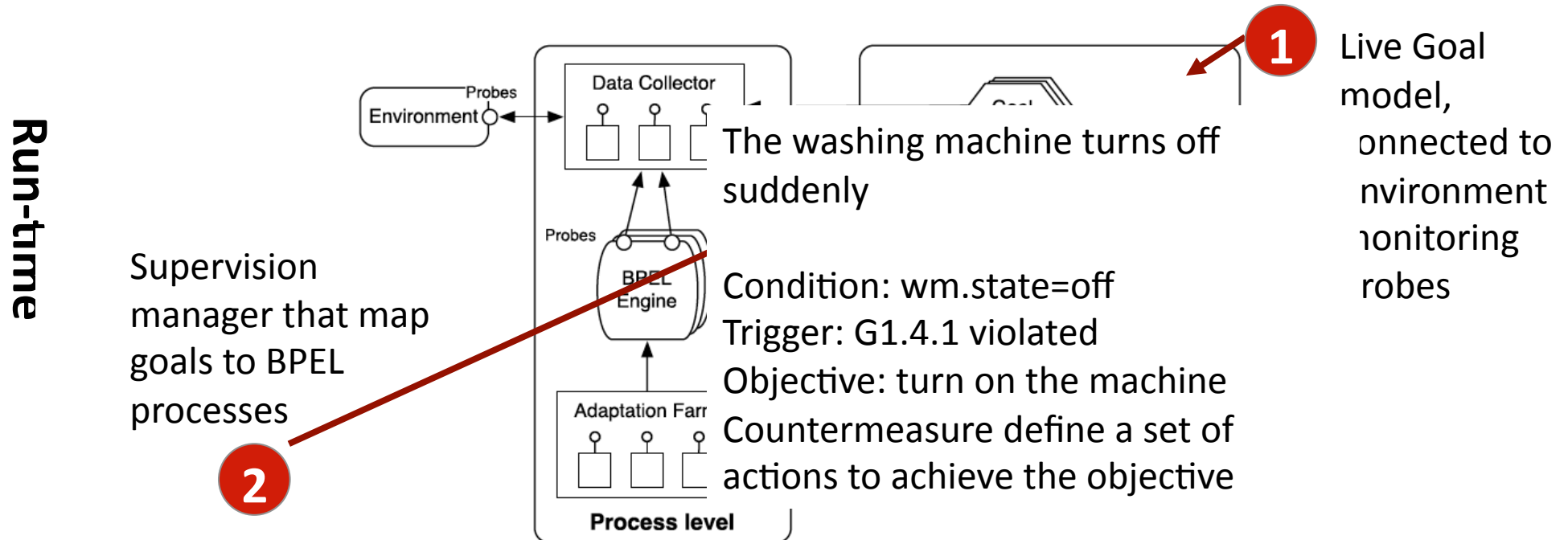




# FLAGS [Pasquale: SEAMS'10, RE'10]

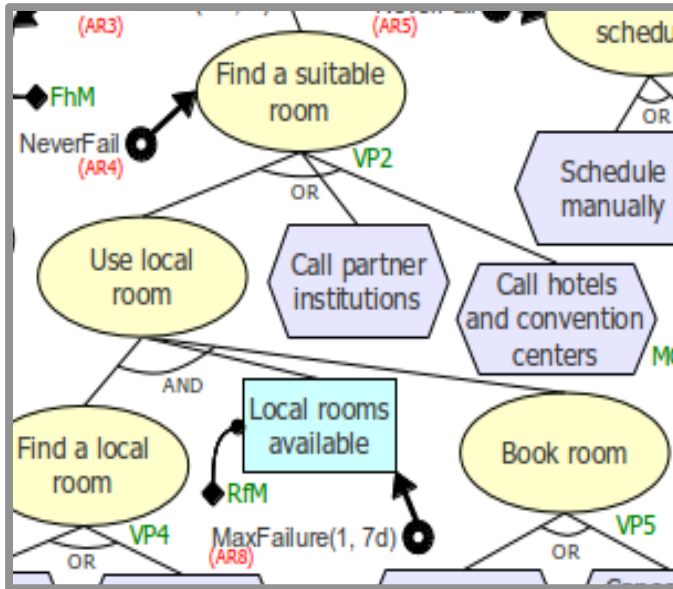


- **Requirements** specifications represented as **goal models** (KAOS foundation, with LTL formalization);
- + **Adaptive Goal**, which specifies adaptation countermeasure that are associated with events that trigger its execution at runtime;
- + **Fuzzy Goal**, which coexist with clearly defined goals
- + **Goal satisfaction** fuzzyfying temporal operators, i.e. “always”, become “almost always



# Zanshin [Souza: SEAMS 2011, ER 2011]

# Design-time



- **Requirements** specifications represented as **goal models** (Techné foundation, {G, T, Q, D});
- **Awareness reqs**: “Goal 'Find a suitable room' should never fail / should have 90% success” (AR4);
- **Parameters** for reconfiguration:
  - OR-refinements / variation points (VP2);
  - Control variables (RfM = Rooms for Meetings);
  - Differential relations:  $\Delta(\text{AR4} / \text{RfM}) > 0$ .

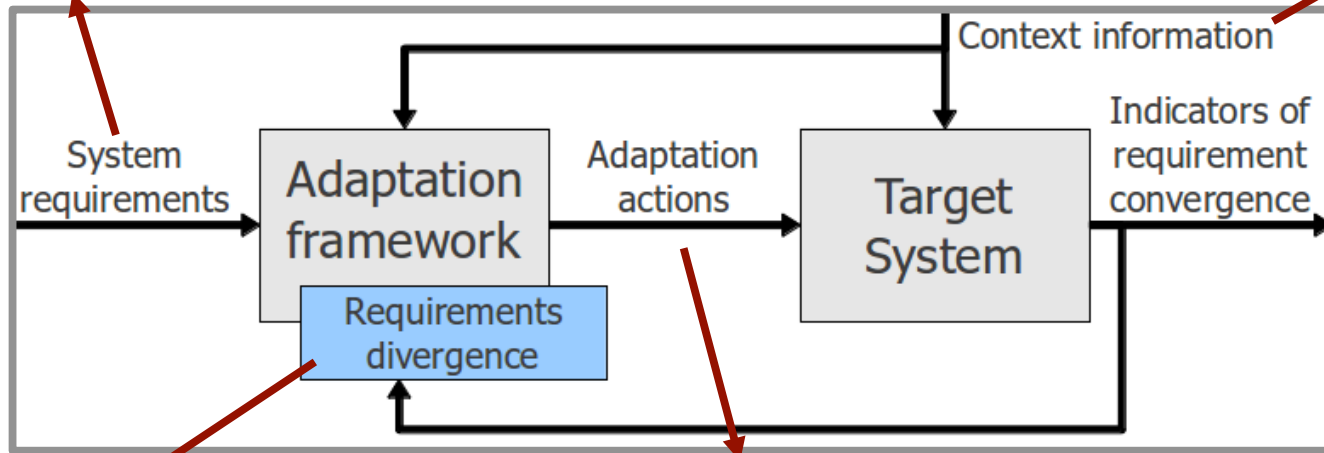
# 1

Users should always find a suitable room (Awareness)

## Is the meeting important?

[Lapouchnian & Mylopoulos, ER 2009]

## Run-time



# 2

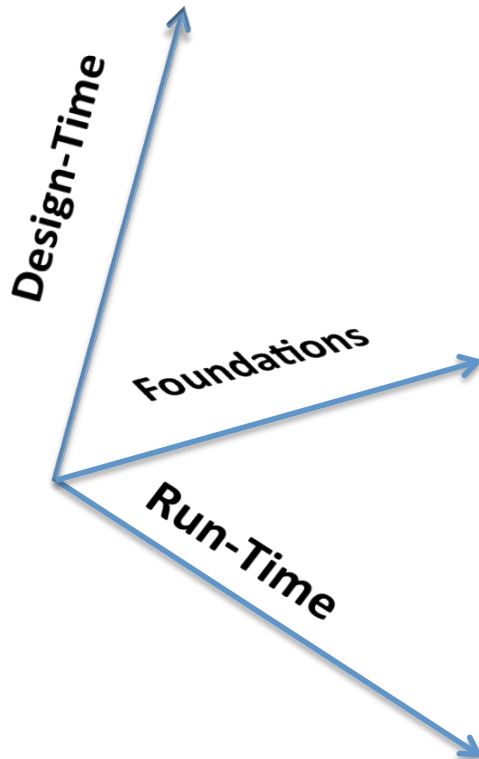
No rooms available!  
(Aw. Req. failure)

3

Reconfigure: arrange a new meeting room? Look for a room elsewhere (partners, hotels)?

## Challenges

- Alive Reqs
- Uncertainty/Unforeseen
- User-centrality

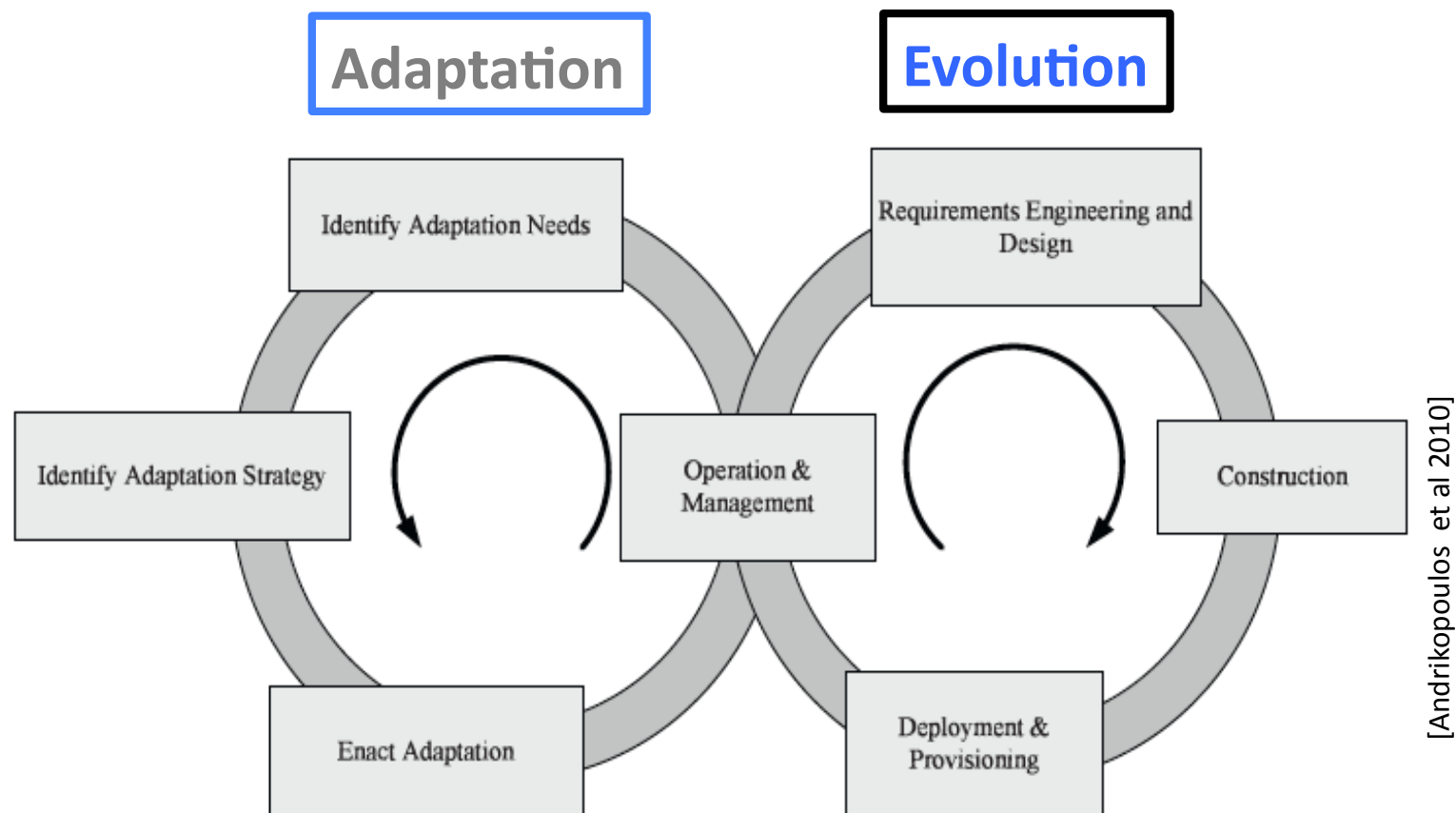


- All considered work address **Design-Time RE** proposing suitable concepts and analysis technique to **design for Run-Time monitoring and adaptation**
- DT uncertainty is managed combining **variability design** (variability in context and system behavior) with **monitoring specs.**; by **relaxing goal satisfaction criteria** (e.g. using fuzzy logics, qualitative reasoning, delegate operator)
- RT representation of requirements is provided either as Java **ECA rules** (e.g. CARE, Zanshin) or as JADEX **executable goal-model** (Tropos4AS)
- End-user centrality is only **partially** addressed through modelling user context, but individual end-user involvement is still poorly addressed (CARE)

- Service Based Applications
  - Features
  - Self-adaptivity
- Requirements Engineering
  - Research challenges
  - Review of some recent work
    - Tropos4AS
    - Adaptive Req. (ARML+CARE)
    - FLAGS
    - Awareness Req. (Zanshin)
- Conclusion & Open issues

# Adaptation or Evolution?

- Adaptation and Evolution are two simultaneous and connected processes (double lifecycle); [S-Cube Book at [s-cube-network.eu](http://s-cube-network.eu)]



- Self-adaptation

- Design-Time: Analysing problem variants and expliciting corresponding alternative solutions elements, with evaluation criteria for their selection (Variability design+ relaxed compliance analysis)
- Run-Time: representation of the whole requirements solution space; monitoring and adapt according to revised notion of requirement compliance (e.g. fuzzy criteria; qualitative reasoning)
- This seems fine for anticipated changes and in case not-exhaustive scenarios for future changes can be identified at DT (i.e. level 3-uncertainty)

- Self-Evolution

- So far offline evolution, human in the loop
- If new requirements emerge, the problem space change so both problem and solution spaces co-evolve (addressed as “Fluidity of design” in [Jarke Et al. ‘11], addressed as Dynamic RE Problem in [Qureshi Et al. Caise’11], but needs end-user involvement)

Requirements Problem dynamic formulation [Qureshi Et al. Caise'11], based on a revision of CORE ontology for RE [Jureta Et al. Re'08]

- **K**: Domain Assumption
  - e.g. Seats Available
- **G**: Goals, Soft Goals
  - e.g. Travel for Business, Convenience
- **Q**: Quality Constraint
  - e.g. Booking done in < 5 Screens
- **A**: Attitudes: Optional and Preferred Requirements
  - e.g. Book an Aisle Seat and Flight must not transit through USA (due to Visa Requirements)
- **T**: Tasks (specification to meet Goals under K and Q)
  - e.g. Book Flight, Schedule a Meeting

**Depend on** Context (C)

e.g. Location) Airport ^ Scheduled flight is cancelled today

**Refer to** Resource (R)

e.g. Mobile, laptop

i.e. a Candidate Solution to the requirements problem in C exists if:

$K_C, T_C, R_C \mid \sim G_C, Q_C, A_C$

Available resources R are needed to realize CS in C

CS are compared using preferences and is-optional relations



- Possible lines for further investigation:
  - Suitability of the framework to investigate (co-) evolution ?
  - End-user (explicit/implicit) on-line feedback to support both dynamic adaptation and evolution ?
  - Pro-active adaptation ?

# Acknowledgements

- A special thank to Cu D. Nguyen (CIT, FBK Trento), Vitor Souza (DISI-UNITN), Antonio Bucchiarone and Annapaola Marconi (SOA-CIT,FBK) for enlightening discussions and to the colleagues of the SE group for useful suggestions

# THANK YOU

Questions?

- [Bertoli et al. AI.2010] Piergiorgio Bertoli, Marco Pistore, Paolo Traverso: Automated composition of Web services via planning in asynchronous domains. *Artif. Intell.* 174(3-4): 316-361 (2010)
- [Berry et al. REFSQ 2005] D. M. Berry, B. H. Cheng, and J. Zhang. The four levels of requirements engineering for and in dynamic adaptive systems. In *REFSQ05*, pages 95–100, 2005
- [Bucchiarone et al 2011] Stephen Lane, Antonio Bucchiarone, Ita Richardson: SOAdapt: A process reference model for developing adaptable service-based applications. *Information & Software Technology* 54(3): 299-316 (2012)
- [Cheng et al, Dagstuhl 2008] Betty H.C. Cheng, Rogerio de Lemos, Holger Giese, Paola Inverardi, and Jeff Magee. *Software Engineering for Self-Adaptive Systems: A Research Roadmap*, pages 1–26. Springer-Verlag, Inc. 5525 edition, 2009.
- [Di Nitto et al. 2008] Elisabetta Di Nitto, Carlo Ghezzi, Andreas Metzger, Mike P. Papazoglou, Klaus Pohl: A journey to highly dynamic, self-adaptive service-based applications. *Autom. Softw. Eng.* 15(3-4): 313-341 (2008)
- [Fickas et al 1995] M. Feather and S. Fickas. Requirements Monitoring in Dynamic Environment. In *Int. Conf. on Requirements Engineering*, 1995
- [Jarke Et al. 2010] M. Jarke, P. Loucopoulos, K. Lyytinen, J. Mylopoulos, and W. N. Robinson, “The brave new world of design requirements: Four key principles,” in *CAiSE*, ser. *Lecture Notes in Computer Science*, B. Pernici, Ed., vol. 6051. Springer, 2010, pp. 470–482.
- [Jureta Et al. RE10] Ivan J. Jureta, Alex Borgida, Neil A. Ernst, and John Mylopoulos. Techne: Towards a new generation of requirements modeling languages with goals, preferences, and inconsistency handling. In *18th IEEE Int. Requirements Eng. Conf.*, pages 115–124, Sydney, Australia, 2010.
- [Kazhamiakin et al 2010] in *S-Cube Book, Service Research Challenges and Solutions for the Future Internet. S-Cube - Towards Engineering, Managing and Adapting Service-Based Systems*. Springer 2010
- [Kephart '03] Jeffrey O. Kephart and David M. Chess. The vision of autonomic computing. *IEEE Computer*, 36(1):41–50, 2003
- [Mylopoulos 2012] Talk at UNITN, Trento, Title: “The Requirements Problem Revisited”
- [Papazoglou Et Al. 2011] Mike P. Papazoglou, Vasilios Andrikopoulos, Salima Benbernou Managing Evolving Services, May/ June 2011, *IEEE SOFTWARE*
- Robinson'06] William Robinson. A Roadmap for Comprehensive Requirements Monitoring. *Computer*, 43(5):64–72, 2009.
- [Salifu'07] M. Salifu, Yijun Yu, and B. Nuseibeh. Specifying monitoring and switching problems in context. *15th IEEE International Requirements Engineering Conference (RE '07)*, pages 211–220, Oct. 2007.

[Wahlster '12] "Towards Software-Defined Networked Cars as Social Agents". The 7th International Conference on Agents and Artificial Intelligence, ICAART 2012 Keynote Lecture

[Whittle Et al. RE'10] Jon Whittle, Pete Sawyer, Nelly Bencomo, Betty Cheng, and Jean-Michel Bruel. Relax: a language to address uncertainty in self-adaptive systems requirement. Requirements Engineering, 15:177–196, 2010. Wahlster 2012

## **TROPOS4AS [Morandini: SEAMS08, ASE08, AAMAS09]]**

[Morandini: AAMAS09] Mirko Morandini, Loris Penserini and Anna Perini. Operational Semantics of Goal Models in Adaptive Agents. Proc. of the 8th Int. Conf on Autonomous Agents and Multi-Agent Systems (AAMAS'09), Budapest, Hungary, May 2009, pp. 129-136.

[Morandini: SEAMS08] Mirko Morandini, Loris Penserini and Anna Perini. Towards Goal-Oriented Development of Self-Adaptive Systems. In Proc. of the Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS) at ICSE08, Leipzig, Germany, May 2008, ACM, pp. 9-16

[Morandini: ASE08] Mirko Morandini, Loris Penserini, Anna Perini. Automated Mapping from Goal Models to Self-Adaptive Systems. In Proc. of the 23<sup>rd</sup> Int. Conf. on Automated Software Engineering (ASE 2008), demo session, L'Aquila, Italy, September 2008, IEEE, pp. 485-486.

## **ARML+CARE [Qureshi: SEAMS08, CAISE10, RE10, REFSQ12]**

[Qureshi et al. 2010] Nauman A Qureshi and Anna Perini. Continuous adaptive requirements engineering: An architecture for self-adaptive service-based applications. *Int. Workshop on Requirements@Run.Time in Conjunction with 18th Int. IEEE Requirements Engineering Conference, RE'10, Sydney, Australia, 2010.*

[Qureshi et al. 2011] Nauman A. Qureshi, Ivan J. Jureta, Anna Perini, Requirements Engineering for Self Adaptive Systems: Core Ontology and Problem Statement , Lecture Notes in Computer Science, CAiSE, Berlin / Heidelberg, Springer, vol. 6741, 2011, pp. 33-47

[Qureshi et al. 2012] Nauman A. Qureshi, Ivan Jureta, Anna Perini, Towards a Requirements Modeling Language for Self-Adaptive Systems, Lecture Notes in Computer Science, in Springer, REFSQ, vol. 7195, 2012, pp. 263-279

## **FLAGS [Pasquale: SEAMS'10, RE'10]**

- [Baresi Et al. SEAMS10] Luciano Baresi and Liliana Pasquale. Live goals for adaptive service compositions. In SEAMS '10: Proceedings of the 2010 CSE Workshop on Software Engineering for Adaptive and Self- Managing Systems, pages 114–123, New York, NY, USA, 2010. ACM
- [Baresi Et al. RE10] Luciano Baresi, Liliana Pasquale, and Paola Spoletini. Fuzzy goals for requirements-driven adaptation. In 18th IEEE Int. Requirements Eng. Conf., pages 125–134, Sydney, Australia, 2010

## **AWARENESS Req. [Souza: SEAMS 2011, ER 2011]**

- [Souza: SEAMS 2011] Souza, V. E. S., Lapouchnian, A., Robinson, W. N., & Mylopoulos, J. (2011a). Awareness Requirements for Adaptive Systems. Proc. of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (pp. 60-69). Honolulu, HI, USA: ACM. doi:10.1145/1988008.1988018
- [Souza ER 2011] Souza, V. E. S., Lapouchnian, A., & Mylopoulos, J. (2011b). System Identification for Adaptive Software Systems: a Requirements Engineering Perspective. In M. Jeusfeld, L. Delcambre, & T.-W. Ling (Eds.), Conceptual Modeling -- ER 2011 (Vol. 6998, pp. 346-361). Springer. doi:10.1007/978-3-642-24606-7\_26

## **OTHER PROJECTS/ NETWORKS**

- IST-Music. Context-aware self-adaptive platform for mobile applications. [http:// ist-music.berlios.de/site/](http://ist-music.berlios.de/site/), 2010
- Project DIVA. Dynamic variability in complex, adaptive systems. <http://www.ict-diva.eu/>, 2011
- S-Cube, the European Network of Excellence in Software Services and Systems, (2008-2011) <http://www.s-cube-network.eu/>