”

PRIN 2007: D-ASAP project

Roma “Tor Vergata” Unit

People@RomaTorVergata

Name	**Effort** (m/y)
Vincenzo Grassi	6 + 6
Salvatore Tucci	6 + 6
Francesco Lo Presti	6 + 6
Valeria Cardellini	6 + 6
Emiliano Casalicchio	6 + 6
Stefano Iannucci	
Luca Silvestri	

”

Our involvement in D-ASAP WP's

- WP 1 , WP 2 , WP 5
  - (+ WP0 and WP6)

- general framework : runtime adaptable systems

- our focus is on the *Service Oriented* (SOA) domain
  - architectural features
    - applications built using loosely coupled services
    - dynamic *discovery*, *selection* and *composition*
      - » different services with same functionality and different QoS/cost
  - contractually specified QoS/cost requirements: *Service Level Agreement* (SLA)
    - *provider* role, *requester* role
    - both roles in case of composite service

Outline

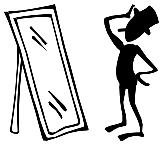
- (this talk)
  - an exploration of the runtime adaptation problem space
    - focused on the SOA domain
  - MOSES : our approach to runtime adaptation of SOA systems
    - methodology
  - ongoing work

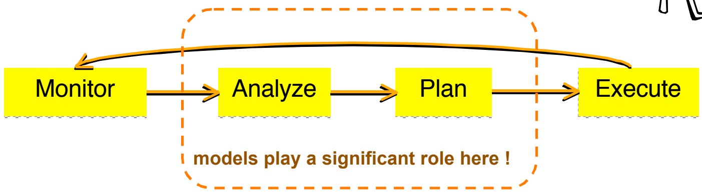
- (next talk)
  - MOSES: architecture and implementation status

”

Model-based runtime self-adaptation

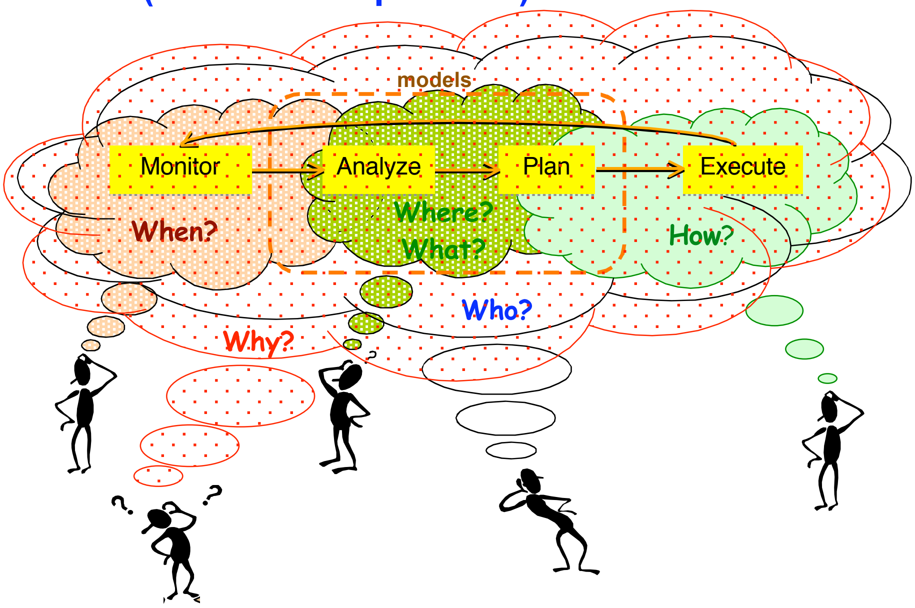
- self-adaptive systems based on one or more closed loops
  - feedback from both the system *self* and *context*





- which models ?
  - targeted to specific issues
    - the *why, when, where, what, how, who* questions
  - tailored for particular *domains* (and *regions* within them)
    - the case of the *service-oriented* (SOA) domain

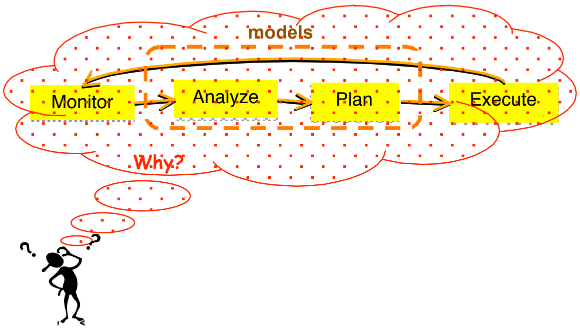
(domain independent) basic issues



"

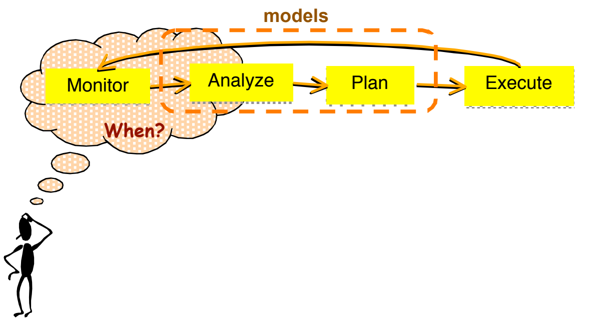
(possible) answers for the SOA domain

- why?
  - fulfillment of functional requirements
  - fulfillment of non functional requirements (QoS, cost)
    - stated in SLAs
      - average values
      - higher moments, percentiles



(possible) answers for the SOA domain

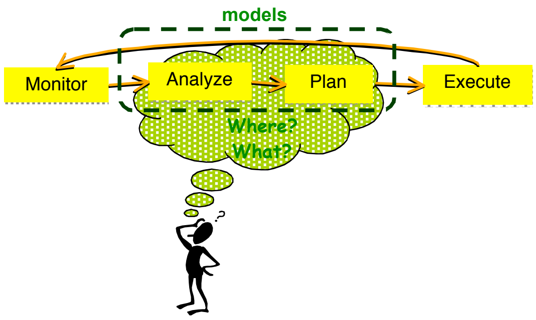
- when?
  - reactive approaches, upon detecting :
    - service availability/unavailability
    - delivered QoS level variation
    - load variations
    - …
  - proactive approaches
    - model-based predictions
    - …



”

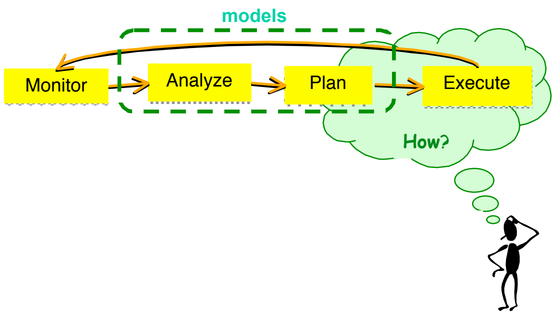
(possible) answers for the SOA domain

- where? / what?
  - level
    - “fringes” of the service composition
      - » implementation of the required functionalities (*abstract tasks*)
    - whole service composition
      - » modification of the overall workflow
  - scope
    - granularity
      - » single request
      - » flow of requests
    - no. of systems
      - » single system
      - » multiple systems



(possible) answers for the SOA domain

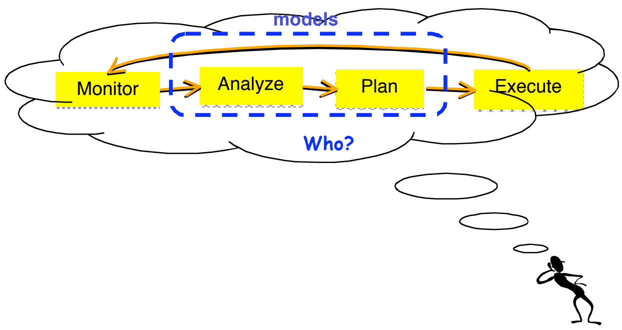
- how?
  - “fringe-oriented” approaches
    - service tuning
    - (re-)selection of single services
    - (re-)selection of coordination patterns
    - …
  - “whole composition” approaches
    - compensating workflows
    - workflow replanning
    - …



”

(possible) answers for the SOA domain

- who?
  - centralized coordination
    - single authority (service broker)
  - distributed coordination
    - multiple cooperating authorities
    - multiple non cooperating authorities
      - » non cooperative game



a “region” in the SOA domain

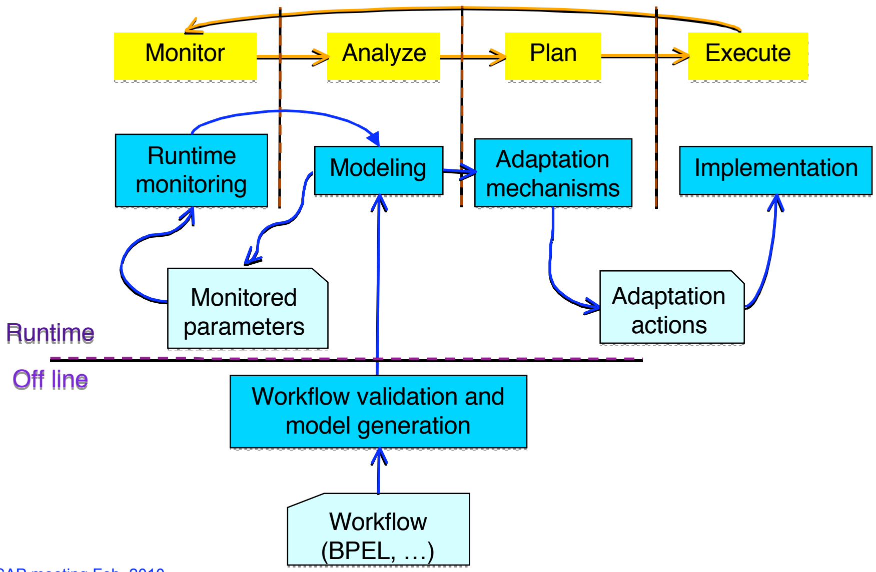
- why?
  - fulfillment of functional requirements
  - fulfillment of non functional requirements (QoS, cost)
    - stated in SLAs
      - » average values
      - » higher moments, percentiles
- when?
  - *reactive* approaches, upon detecting :
    - service availability/unavailability
    - delivered QoS level variation
    - load variations
    - …
  - *proactive* approaches
    - model-based predictions
- how?
  - “fringe-oriented” approaches
    - service tuning
    - (re-)selection of single services
    - (re-)selection of coordination patterns
    - …
  - “whole composition” approaches
    - compensating workflows
    - workflow replanning
    - …

- where? / what?
  - level
    - “fringes” of the service composition
      - » implementation of the required functionalities (*abstract tasks*)
    - whole service composition
      - » modification of the overall workflow
  - scope
    - granularity
      - » single request
      - » flow of requests
    - no. of systems
      - » single system
      - » multiple systems
- who?
  - centralized coordination
    - single authority (service broker)
  - distributed coordination
    - multiple cooperating authorities
    - multiple non cooperating authorities
      - » non cooperative game

”

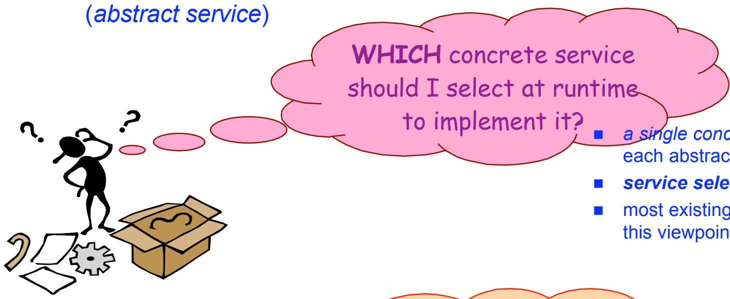
The MOSES framework

- MOSES: *MOdel-based SElf-adaptation of SOA systems*

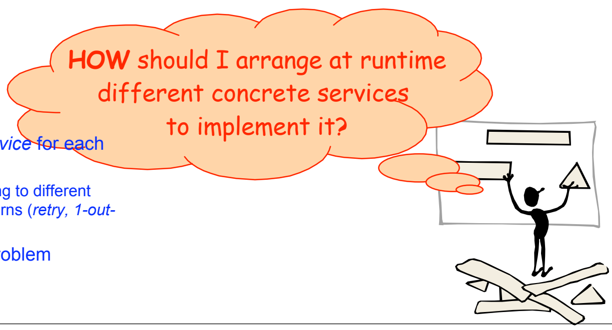


How to adapt at runtime a SOA system

- two viewpoints for the implementation of each required functionality (*abstract service*)



- *a single concrete service* for each abstract service
- ***service selection*** problem
- most existing approaches adopt this viewpoint

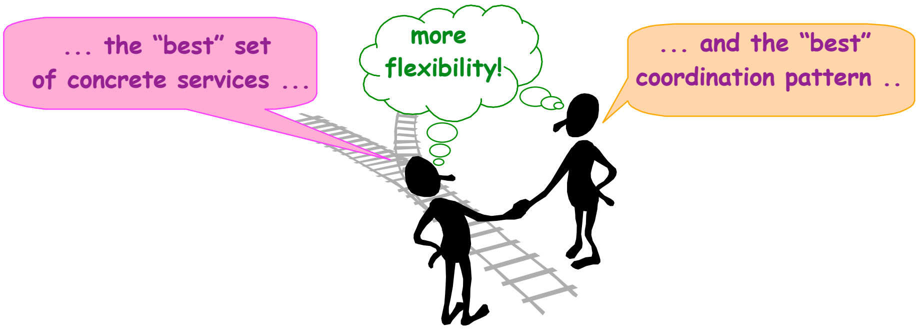


- *a set of concrete service* for each abstract service
  - arranged according to different coordination patterns (*retry, 1-out-of-N*, …)
- ***pattern selection*** problem

”

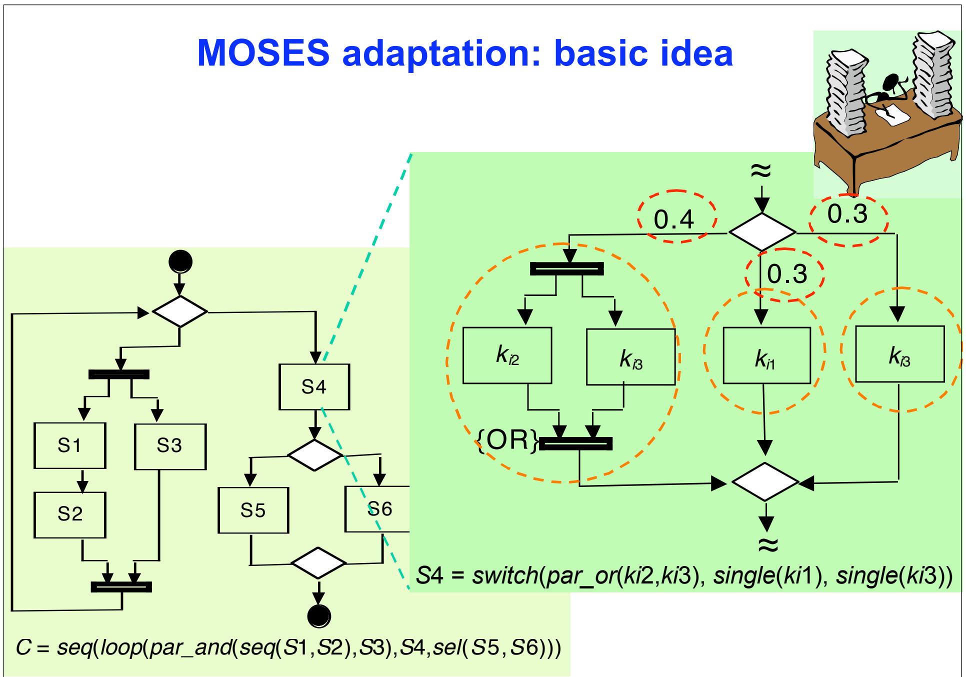
the MOSES approach

- based on both viewpoints
- for each abstract service, we determine at runtime …



- … to fulfill the QoS/cost requirements of different (classes of) users
- concerning the **average value** of QoS/cost attributes …
  - response time, dependability, cost (others could be added)
- … and the overall **flow** of service requests for each user

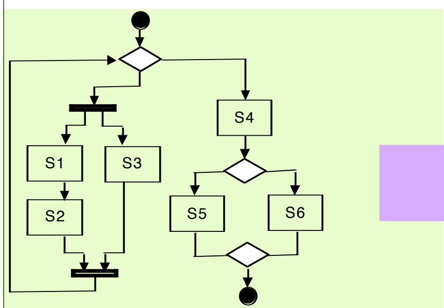
MOSES adaptation: basic idea



”

MOSES adaptation: runtime model

- we build at runtime a linear programming optimization model
  - model parameters based on SLA parameters and usage profile parameters
    - usage profile: V_i^u: number of requests to *Si* for each client *u* request
      - » updated by monitoring activity
  - the solution drives the system implementation
  - new solution calculated at each environment change
    - user profile change
    - arrival/departure of users
    - component services change



C = *seq*(*loop*(*par_and*(*seq*(*S*1,*S*2),*S*3),*S*4,*sel*(*S*5,*S*6)))

Constrained multi-criteria optimization problem

Minimize $F(\mathbf{x})$

subject to $Q^{\alpha}(\mathbf{x}) \leq Q^{\alpha}_{max}$ } QoS constraints

$Q^{\beta}(\mathbf{x}) \geq Q^{\beta}_{min}$

$\mathbf{x} \in A$ } Functional constraints

MOSES adaptation: runtime model

- just a taste of the underlying mathematics …

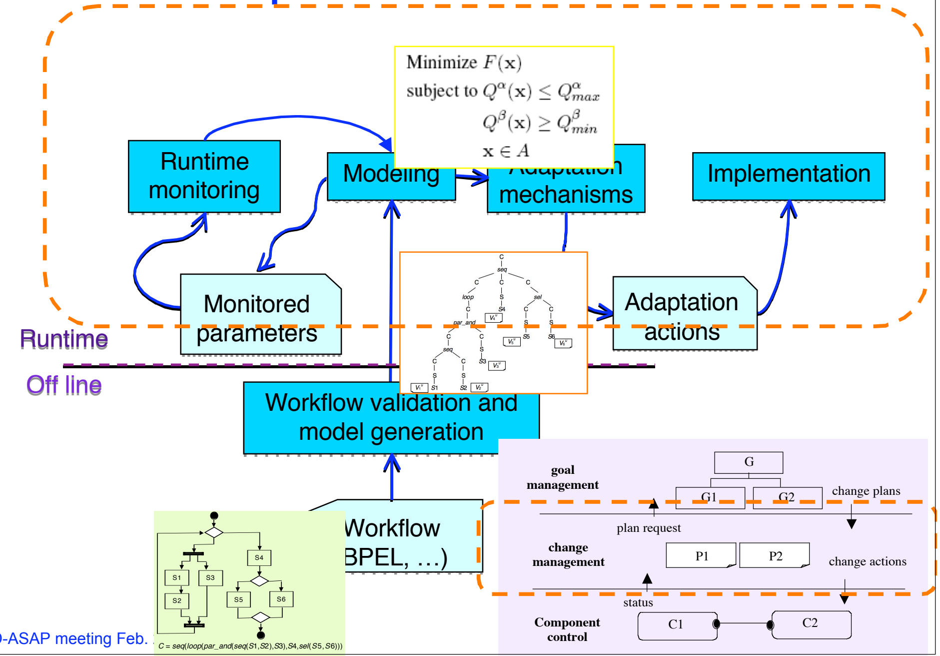
$$\max F(\boldsymbol{x}) = w_r \frac{R_{\max} - R(\boldsymbol{x})}{R_{\max} - R_{\min}} + w_d \frac{D(\boldsymbol{x}) - D_{\min}}{D_{\max} - D_{\min}} + w_c \frac{C_{\max} - C(\boldsymbol{x})}{C_{\max} - C_{\min}}$$

subject to:

$$C^u(\boldsymbol{x}) \leq C_{\max}{}^u, \quad u \in U$$
$$D^u(\boldsymbol{x}) \geq D_{\min}{}^u, \quad u \in U$$
$$R^u(\boldsymbol{x}) \leq R_{\max}{}^u, \quad u \in U$$
$$R_{l'}{}^u(\boldsymbol{x}) \leq R_l{}^u(\boldsymbol{x}), \quad l' \in d(l),\ l \in \Pi,\ u \in U$$
$$R_l{}^u(\boldsymbol{x}) = \sum_{i \mathrm{p}_{dd} l} \frac{V_h^u}{V_i^u} \sum_{J \in \Im i} x_{iJ}^u\, R(Si;J)$$
$$+ \sum_{h \in \Pi, h \mathrm{p}_{dd} l} \frac{V_h^u}{V_l^u} R_h{}^u(\boldsymbol{x}) \quad l \notin \Pi,\ u \in U$$
$$\sum_{u \in U} x_{iJ}{}^u Vi^u L^u \leq L_{ij}, \quad 1 \leq i \leq m,\ J \in \Im i$$
$$x_{iJ}{}^u \geq 0, \ \sum_{J \in \Im} x_{iJ}{}^u = 1, \quad 1 \leq i \leq m,\ u \in U$$

”

MOSES adaptation: functional architecture



MOSES : present and future

- present
  - a methodology for QoS-driven reactive adaptation of SOA systems (just presented)
    - flow based
    - multiple QoS classes
    - stateless/stateful services
    - SLA : average values of QoS attributes
  - a simulator
    - for validation purposes
  - a real prototype (next presentation)
    - running, but not yet experimented
    - modular
- ongoing work
  - tighter SLA : percentile-based
  - monitoring and proactive adaptation
  - admission control
  - distributed (and cooperative) MOSES
  - real world experiments
  - prototype implementation : continuous update

”

collaborations

- Poli MI 
- CNR PI

publications

(1) E. Casalicchio, D.A. Menascé, V. Dubey, L. Silvestri, “Optimal service selection heuristics in service oriented architectures”, Proc. of 3rd Int’l **Workshop on Advanced Architectures and Algorithms for Internet DElivery and Applications** (AAA-IDEA 2009), Las Palmas de Gran Canaria, Spain, Nov. 2009.

(2) Grassi V., Mirandola R., Randazzo E., “Model-Driven Assessment of QoS-Aware Self-Adaptation” in **Software Engineering for Self-Adaptive Systems** (Betty H.C. Cheng, Rogerio de Lemos, Holger Giese, Paola Inverardi, Jeff Magee eds.), LNCS 5525, Springer-Verlag (2009), pp. 201-222

(3) V. Cardellini, E. Casalicchio, V. Grassi, F. Lo Presti, R. Mirandola “Towards QoS aware self-adaptation in service oriented systems” in **Architecting Dependable Systems 6** (R. de Lemos, J.-C. Fabre, C. Gacek, F. Gadducci, M. H. ter Beek eds.), LNCS 5835, Springer-Verlag (2010)

(4) V. Cardellini, E. Casalicchio, V. Grassi, F. Lo Presti, R. Mirandola “QoS-driven Runtime Adaptation of Service Oriented Architectures” **Joint 12th European Software Engineering Conference and 17th ACM SIGSOFT Symposium on the Foundations of Software Engineering (ESEC-FSE 2009),** August 26-28, 2009, Amsterdam, The Netherlands, pp. 131-140

(5) Valeria Cardellini, Emiliano Casalicchio, Vincenzo Grassi, Francesco Lo Presti, Raffaela Mirandola “A Scalable Approach to QoS-Aware Self-Adaption in Service-oriented Architectures” **Proc. of QShine 2009, The Sixth International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness**, Las Palmas de Gran Canaria, Spain, 23-25 November 2009.

(6) S. Casolari, M. Colajanni, F. Lo Presti "Runtime state change detector of computer system resources under non stationary conditions", **Proc 17th IEEE.ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems**, London, Sept. 21-23, 2009.

(7) Daniel A. Menasce, Emiliano Casalicchio, Vinod Dubey, On optimal service selection in Service Oriented Architectures, **Performance Evaluation**, In Press, Corrected Proof, Available online 17 July 2009, DOI: 10.1016/j.peva.2009.07.001.