deep se dependable evolvable pervasive software engineering

Self-Adaptation of Service based Systems based on Cost/Quality Attributes Tradeoffs

Raffaela Mirandola ‡ Pasqualina Potena §

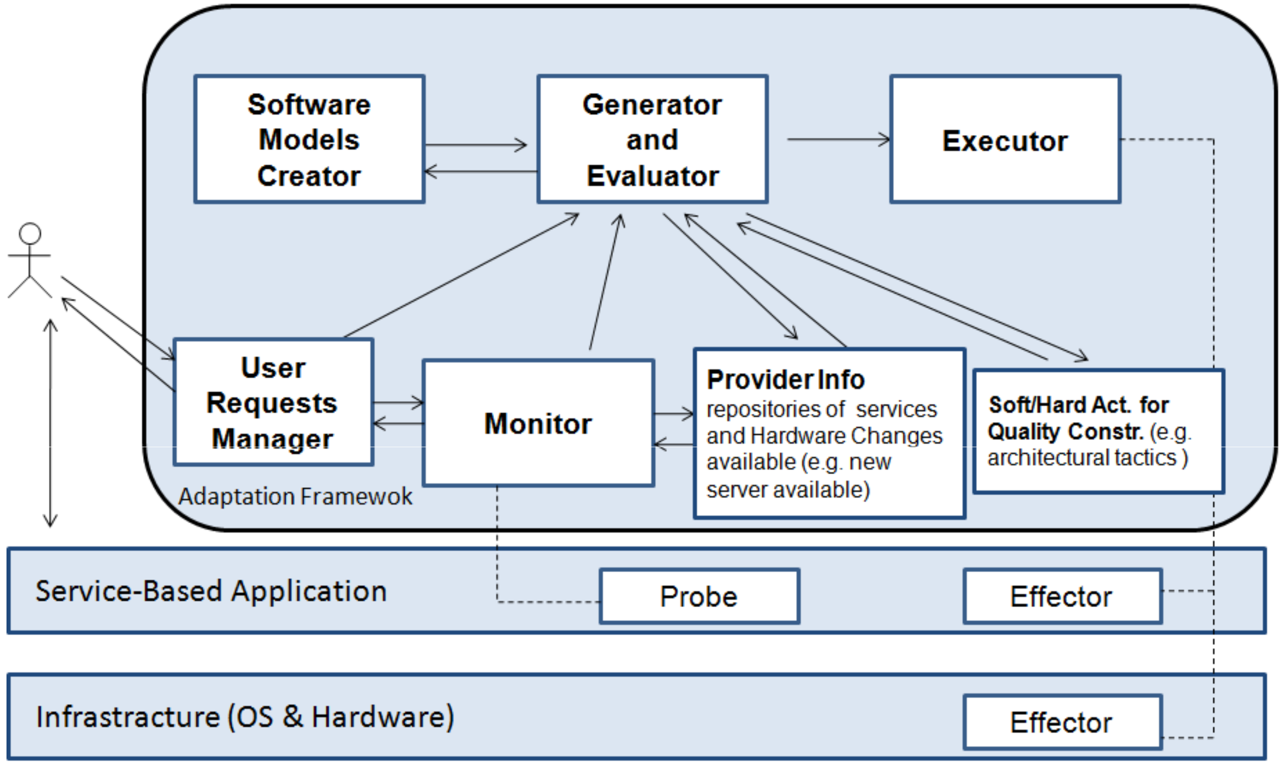
‡ Politecnico di Milano, Italy

§ Universita' dell'Aquila, Italy

An application should be self-adaptive in order to automatically and autonomously adapt its behavior for several reasons, such as

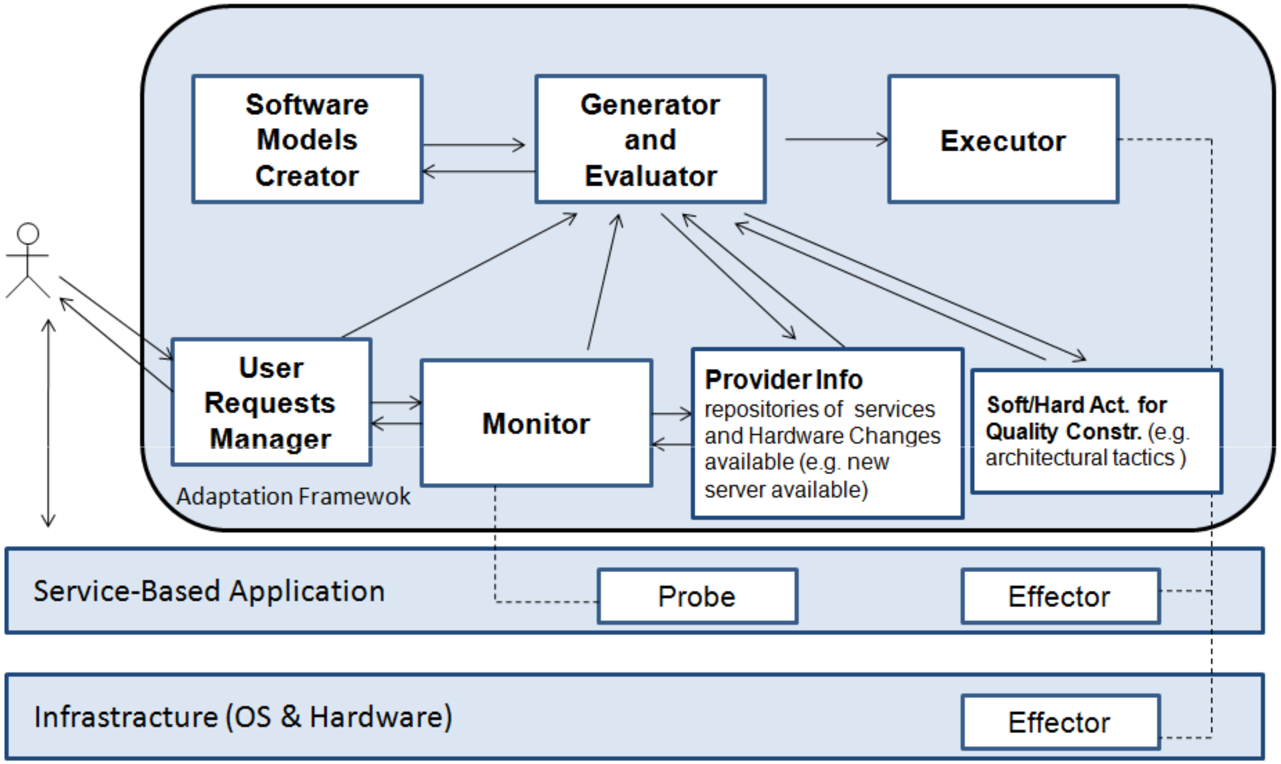
- service evolution (e.g. a new version may be available)
- hardware volatility (e.g. network quality changes)
- varying users demands with new requirements (e.g. a new functionality or a different level of quality of service)

Objective of the paper : A Framework,…



…, based on an optimization model (generated and solved by the *Generator and Evaluator* module), dynamically adapts a system (by **changing software and hardware features** by means of the *Executor* module) while minimizing the **adaptation costs** and guaranteeing a required level of the **system qualities**.

Objective of the paper : A Framework



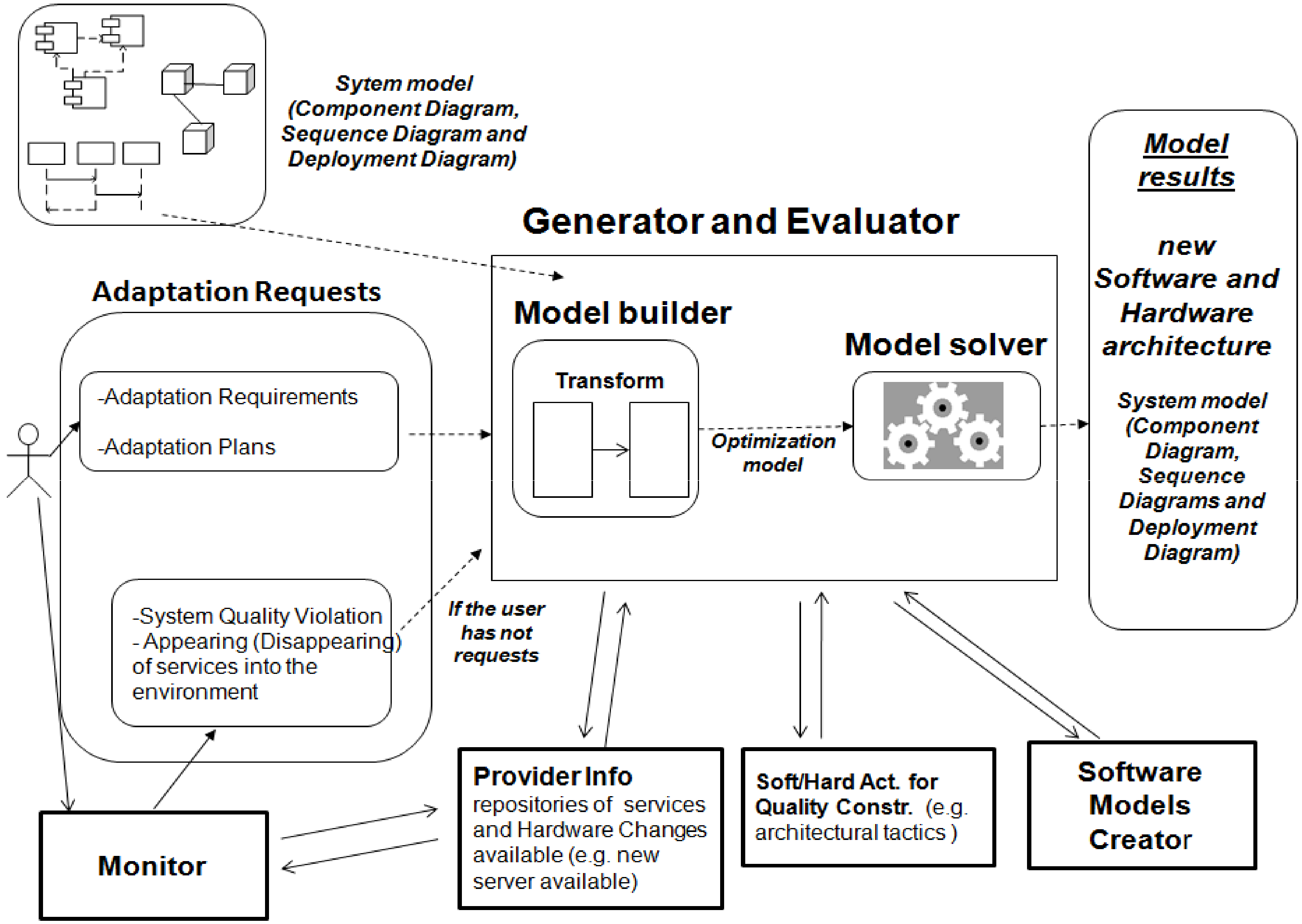
Adaptation actions can be triggered both by an user (using the *User Requests Manager* module that can also interact with the *Monitor*) and/or automatically by the framework itself (after it receives alerts from the *Monitor* monitoring the system and interacting with services repositories by means of the *Provider Info module*).

Presentation roadmap

- Overview of Optimization Model
- Zoom-in into state of the art
- Conclusion

Presentation roadmap

- Overview of Optimization Model
- Zoom-in into state of the art
- Conclusion


Sytem model (Component Diagram, Sequence Diagram and Deployment Diagram)
Generator and Evaluator
Adaptation Requests
-Adaptation Requirements
-Adaptation Plans
-System Quality Violation
- Appearing (Disappearing) of services into the environment
Model builder
Transform
Optimization model
Model solver
If the user has not requests
Model results
new Software and Hardware architecture
System model (Component Diagram, Sequence Diagrams and Deployment Diagram)
Monitor
Provider Info
repositories of services and Hardware Changes available (e.g. new server available)
Soft/Hard Act. for Quality Constr. (e.g. architectural tactics)
Software Models Creator

Overview of Optimization Model

Let us assume to dispose of a deployed system composed by software services and that for each (not) active service offered by the system we dispose of an UML SD.

Getting User Requests – An user adaptation scenario is a set of new requirements…

(i) a **new functional requirement**, i.e. introducing a new external service/modifying the dynamics of an existing external service

(ii) a **new non-functional requirement**, i.e. requiring a value threshold for a new system quality/modifying the value threshold for an already required system quality.

Getting Alerts by the Monitor -

(i) the **violation of a non-functional requirement**

(ii) the **appearing/disappearing of a service** in the environment.

To satisfy either the new requirements required by the user or exploit the alerts raised by the *Monitor* some adaptation actions have to be performed.

The user defines ***adaptation plans*** for each requirement, whereas the framework itself defines ***adaptation plans*** when it gets alerts about the violation of non-functional constraints.

An ***adaptation plan*** is a set of actions modifying the static and dynamic structure of the software architecture and the hardware architecture to (exploit a certain alert) address a certain requirement.

Software adaptation actions:

- Introducing new software services
- Replacing existing service instances with functionally equivalent ones
- Modifying the interactions between software services in a certain external service

Hardware adaptation actions:

- Defining the deployment of software service on hardware nodes
- Introducing hardware resources
- Modifying hardware resources
- Modifying the interactions between hardware nodes

In addition, we leave to the solver the possibility to choose additional service replacement actions and hardware actions that have not been embedded in any selected plan.

Requirement ID	Adaptation Plan ID	Adaptation Plan Description
req_1	ap_{11}	Replacing s_3 with its first instance AND Replacing s_4 with its second instance
	ap_{12}	Adding a new service $news_2$
req_2	ap_{21}	Replacing s_2 with its first instance
	ap_{22}	Adding a new service $news_1$ AND Replacing s_5 with its first intance
	ap_{23}	Adding a new service $news_2$

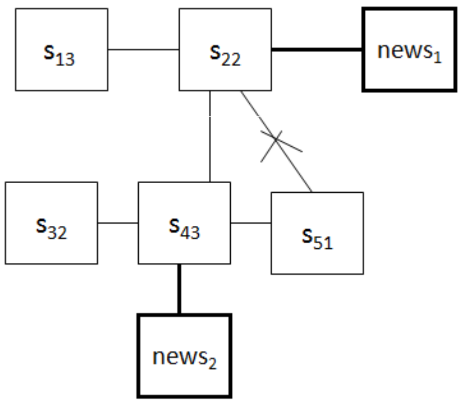
Example of Output of the Model

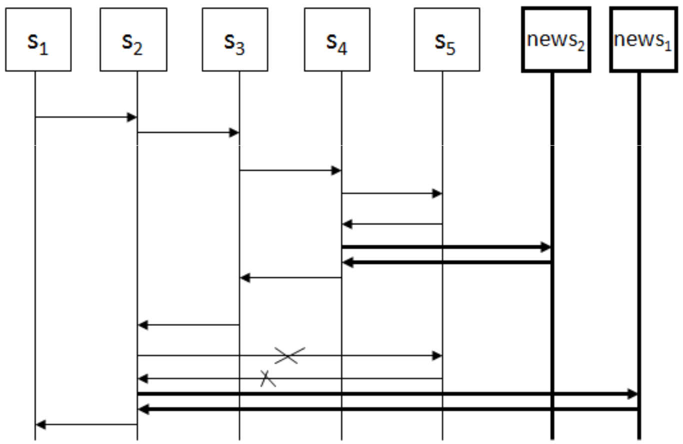
Requirement ID	*Adaptation Plan ID*	*Adaptation Plan Description*
req_1	*ap_{11}*	*Replacing s_3 with its first instance AND Replacing s_4 with its second instance*
	ap_{12}	*Adding a new service $news_2$*
req_2	*ap_{21}*	*Replacing s_2 with its first instance*
	ap_{22}	*Adding a new service $news_1$ AND Replacing s_5 with its first intance*
	ap_{23}	*Adding a new service $news_2$*

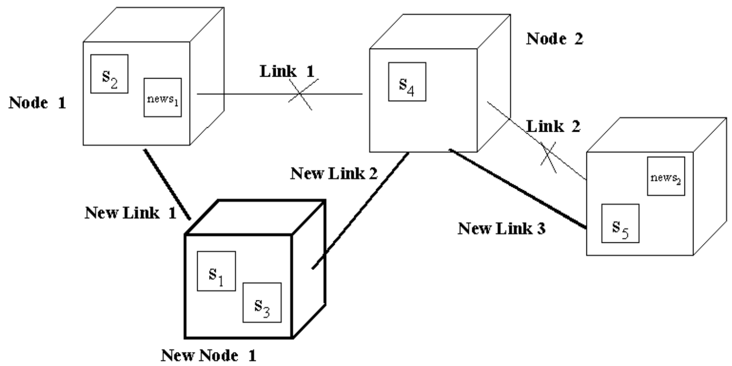
Example of Output of the Model

Requirement ID	Adaptation Plan ID	Adaptation Plan Description
req_1	ap_{11}	Replacing s_3 with its first instance AND Replacing s_4 with its second instance
	ap_{12}	Adding a new service $news_2$
req_2	ap_{21}	Replacing s_2 with its first instance
	ap_{22}	Adding a new service $news_1$ AND Replacing s_5 with its first intance
	ap_{23}	Adding a new service $news_2$

Example of Output of the Model







OPTIMIZATION MODEL

$$\min Cost(\theta_1^0, \ldots, \theta_n^0, \overline{\theta}_1^0, \ldots, \overline{\theta}_{|NewS|}^0)$$

$$\sum_{k \in K} pexec_k \cdot G_{kq}(\theta_{k1}^q, \ldots, \theta_{kn}^q, \overline{\theta}_{k1}^q, \ldots, \overline{\theta}_{k|NewS|}^q) \geq \Theta^q$$

....

where:

n number of existing services

NewS number of new services

pexec$_k$ probability that the *k*-th system functionality will be invoked

Θ^q threshold value required for the *q*-th quality attribute

θ^0_i ($\overline{\theta^0}_i$) cost of the existing/new service

θ^q_{ki} ($\overline{\theta^q}_{ki}$) value of the *q*-th quality attribute of the *i*-th existing (new) service

OPTIMIZATION MODEL

$$\min Cost(\theta_1^0,\ldots,\theta_n^0,\overline{\theta}_1^0,\ldots,\overline{\theta}_{|NewS|}^0)$$

$$\sum_{k\in K} pexec_k \cdot G_{kq}(\theta_{k1}^q,\ldots,\theta_{kn}^q,\overline{\theta}_{k1}^q,\ldots,\overline{\theta}_{k|NewS|}^q) \geq \Theta^q$$

....

n number of existing services

NewS number of new services

$pexec_k$ probability that the *k*-th system functionality will be invoked

Θ^q threshold value required for the *q*-th quality attribute

An example of Cost function

$$Cost = \sum_{i=1}^{n} \sum_{j=1}^{|Aval_i|} c_{ij} x_{ij} + \sum_{h=1}^{|NewS|} \overline{c_h} z_h$$

c_{ij} cost of the *j*-th instance available for the service *i*

$\overline{c_h}$ cost to adopt the *h*-th new service into the system

Optimization Model

$$\min Cost(\theta_1^0,\ldots,\theta_n^0,\overline{\theta}_1^0,\ldots,\overline{\theta}_{|NewS|}^0)$$

$$\sum_{k\in K} pexec_k \cdot G_{kq}(\theta_{k1}^q,\ldots,\theta_{kn}^q,\overline{\theta}_{k1}^q,\ldots,\overline{\theta}_{k|NewS|}^q) \geq \Theta^q$$

....

n number of existing services

NewS number of new services

$pexec_k$ probability that the *k*-th system functionality will be invoked

Θ^q threshold value required for the *q*-th quality attribute

q-th quality attribute of the *i*-th existing service

$$\theta_{ki}^q = \sum_{j=1}^{|Aval_i|} x_{ij}\Gamma_{kq}\left(\lambda^q{}_{i1},\ldots,\lambda^q{}_{iu},\lambda'^q{}_{i1},\ldots,\lambda'^q{}_{iv},\Lambda^q{}_{ij1},\ldots,\Lambda^q{}_{ijw}\right)$$

Γ_{kq} function that predicts the *q*-th quality attribute
u number of software architecture observable parameters
v number of hardware observable parameters
w number of parameters expressing the specific features of the service implementation
bp_{ki} number of busy periods that the service *i* shows in the SD *k*

$VBP_p(i,k)$ variation of the number of busy periods of the *i*-th existing service in the *k*-th external service.

for example, the number of busy periods…

$$\lambda^q{}_{i1} = bp_{ki} + \sum_{r=1}^{m}\sum_{p\in AP_r} VBP_p(i,k)\cdot y_{rp}$$

deep se

OPTIMIZATION MODEL

$$\min Cost(\theta_1^0,\dots,\theta_n^0,\overline{\theta}_1^0,\dots,\overline{\theta}_{|NewS|}^0)$$

$$\sum_{k\in K} pexec_k \cdot G_{kq}(\theta_{k1}^q,\dots,\theta_{kn}^q,\overline{\theta}_{k1}^q,\dots,\overline{\theta}_{k|NewS|}^q) \geq \Theta^q$$

....

n number of existing services

NewS number of new services

pexec$_k$ probability that the *k*-th system functionality will be invoked

Θ^q threshold value required for the *q*-th quality attribute

q-th quality attribute of the *h*-th new service

$$\overline{\theta}_{kh}^q = z_h \overline{\Gamma}_{kq}(\lambda_{h1}^q,\dots,\lambda_{hu}^q,\lambda'^q_{h1},\dots,\lambda'^q_{hv},\overline{\Lambda}_{h1}^q,\dots,\overline{\Lambda}_{hw}^q)$$

$\overline{\Gamma}_{kq}$ function that predicts the *q*-th quality attribute
u number of software architecture observable parameters
v number of hardware observable parameters
w number of parameters expressing the specific features of the service implementation
BPp(h, k) number of busy periods that the chosen adaptation plans suggest for the *h*-th new service within the *k*-th external service

for example, the number of busy periods...

$$\lambda^q{}_{h1} = \sum_{r=1}^{m} \sum_{p\in AP_r} BP_p(h,k)\cdot y_{rp}$$

OPTIMIZATION MODEL

$$\min Cost(\theta_1^0, \ldots, \theta_n^0, \overline{\theta}_1^0, \ldots, \overline{\theta}_{|NewS|}^0)$$

$$\sum_{k \in K} pexec_k \cdot G_{kq}(\theta_{k1}^q, \ldots, \theta_{kn}^q, \overline{\theta}_{k1}^q, \ldots, \overline{\theta}_{k|NewS|}^q) \geq \Theta^q$$

....

n number of existing services

NewS number of new services

pexec$_k$ probability that the *k*-th system functionality will be invoked

Θ^q threshold value required for the *q*-th quality attribute

An example of system quality : the System Reliability

$$G_{kq} = \prod_{i=1}^{n} \left(\sum_{j=1}^{|Aval_i|} x_{ij} (1 - \Lambda^q{}_{ij1})^{\lambda^q{}_{i1}} \right) \cdot \prod_{h=1}^{|NewS|} (1 - \overline{\Lambda}^q{}_{h1} z_h)^{\lambda^q{}_{h1}}$$

$\Lambda^q{}_{ij1}$ probability of failure on demand of the *j*-th instance available for the service *i*

$\overline{\Lambda}^q{}_{h1}$ probability of failure of the new service *h*

$$\lambda^q{}_{i1} = bp_{ki} + \sum_{r=1}^{m} \sum_{p \in AP_r} VBP_p(i,k) \cdot y_{rp}$$

$$\lambda^q{}_{h1} = \sum_{r=1}^{m} \sum_{p \in AP_r} BP_p(h,k) \cdot y_{rp}$$

Presentation roadmap

- Overview of Optimization Model
- Zoom-in into state of the art
- Conclusion

Existing Approaches:

✓ Usually the frameworks adapt a system only after a user triggers a request.

- ➢ They support service selection with respect to a composition defined by an user (e.g. the VRESCo runtime environment [Rosenberg09])

- ➢ They choose the service composition, which they have generated together with a finite set of other candidates, that better fulfill the required quality (e.g. [Chiu09], [Ibrahim09])

✓ The spontaneous service composition is typically not supported.

In [Ibrahim1-09] the spontaneous service selection and the extension of system functionality is supported.

[Rosenberg09] F. Rosenberg, P. Celikovic, A. Michlmayr, P. Leitner, and S. Dustdar. An End-to-End Approach for QoS-Aware Service Composition. In EDOC, pages 151–160, 2009

[Chiu09] D. Chiu, S. Deshpande, G. Agrawal, and R. Li. A Dynamic Approach toward QoS-Aware Service Workflow Composition. In ICWS, pages 655–662, 2009.

[Ibrahim09] N. Ibrahim and F. L. Mouël. A Survey on Service Composition Middleware in Pervasive Environments. CoRR, abs/0909.2183, 2009.

[Ibrahim1-09] N. Ibrahim, F. L. Mouël and S. Frénot. MySIM: a spontaneous service integration middleware for pervasive environments. In ICPS '09: Proceedings of the 2009 International conference on Pervasive services, pages 1–10, 2009

With respect to existing approaches, our approach…

- is the first one (to the best of our knowledge) introducing a dynamic self-adaptive framework supporting both the software (including both static and dynamic models) and hardware architecture adaptation using an optimization model
- is general and does not rely on specific architectural style, development process or service application domain
- can facilitate the work of a maintainer that does not have to insert as input value architectures satisfying all changes required

Presentation roadmap

- Overview of Optimization Model
- Zoom-in into state of the art
- Conclusion

We have introduced a ***framework***, based on an optimization model, that dynamically adapts both the software and hardware features of a service based system while minimizing the ***adaptation costs*** and guaranteeing a required level of the ***system qualities***.

Adaptation actions can be triggered both by an user request and/or automatically after the runtime violation of system quality constraints, or the appearing/disappearing of services into the environment.

We intend to…

• specialize our framework by enhancing it for guaranteeing specific properties of a service application domain. We are implementing a prototype to apply our approach on realistic examples

• investigate the evaluation of dependencies among requirements

• investigate the use of other quality constraints

• take into account dependencies betweeen quality attributes and dependencies between services

• solve problems due, for example, to the model solution too large computation time using the meta-heuristic techniques combinated with the simulation techniques

• introducing dependencies between adaptation plans

….