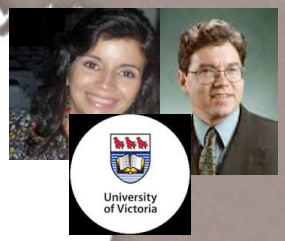
QoS Contract Preservation through Dynamic Reconfiguration: A Formal Semantics Approach

Gabriel Tamura, Rubby Casallas, Anthony Cleve, Laurence Duchien



QoS Contract Preservation through Dynamic Reconfiguration : a Formal Semantics Approach

The Quest for Quality-of- Service in Software Services

Software Services:

- In everyday life
- Massively accessed
- Users increasingly depend on them

... but also require increasing quality of service guarantees!

QoS Software Contracts

Quality-of-Service Challenges in SaaS - Cloud Computing

A QoS Contract Example

Controlling the Black Friday Syndrome: QoS contracts must be preserved

at run-time (challenge 1)

How critical is to preserve this QoS contract?

(i.e., Can the SaaS/IaaS cope with the next Black Friday's unforeseeable load?)

QoS Contracts Are Out There:
The Amazon EC2's SLA* based on availability for SaaS/IaaS

	Context Condition	QoS Service Level
Annual Uptime≥99.95%	Periods of 5 minutes (« Region Unavailable » state) along a Year	No guarantees
(based on cumulative RUs)	Outside the « Region Unavailable » state	100% Availability

Drops below 99.95% grant you a 10% refund of your bill

Amazon reserves the right to modify this SLA at anytime

QoS Contract Preservation Must be Reliable

QoS contracts must be preserved autonomously and **reliably**

(challenge 2)

case of contract unfulfillment

*http://www.bloomberg.com/news/2011-04-21/amazon-com-says-some-web-services-for-businesses-not-available.html

... QoS Contracts are a Reality and a Necessity

Software services must satisfy them:

- At run-time (continuously)
  - Autonomously
    - Reliably

Presentation Outline

1. Overview
2. Background
3. Contributions
4. Validation
5. Conclusions & Future Work

Overview

QoS-CARE: A Comprehensive Solution to Preserve QoS Contracts Reliably at Run-time

- From a formal foundation
- ... to a concrete software architecture and implementation
- and its validation:
  - Theoretically, on the formal model properties to guarantee its reliability
  - Experimentally, on its practical feasibility

However (at the beginning):

- How to exploit formal models in software engineering (to preserve QoS contracts)?
- Is the result going to be feasible in practical terms?

The Story of QoS-CARE:
A Theory for Reliable Preservation of QoS Contracts

An example of the use of Formal Models as foundations for Software Engineering for SAS

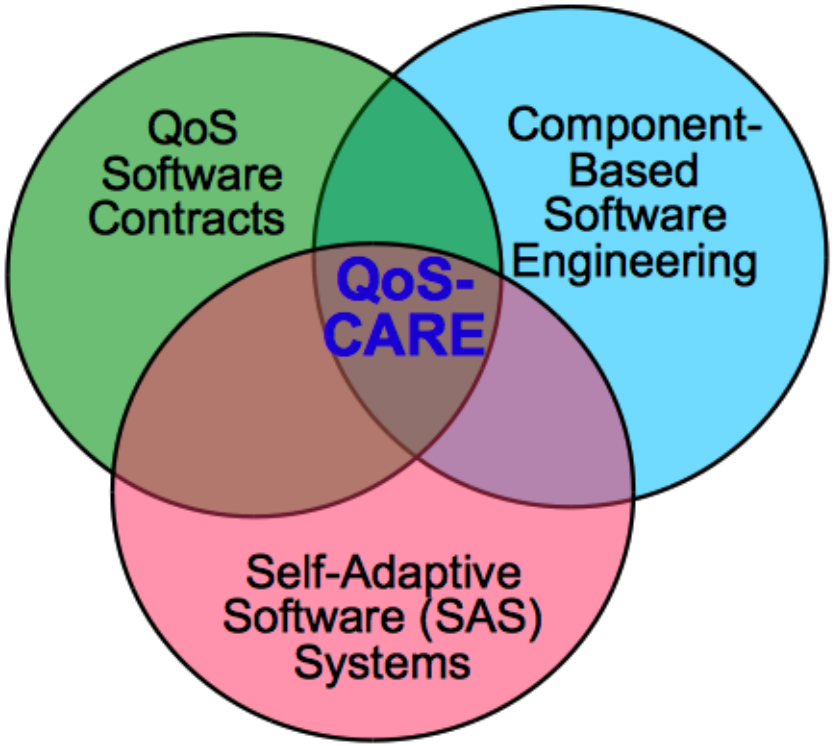
2. Background

Finding the path to follow...

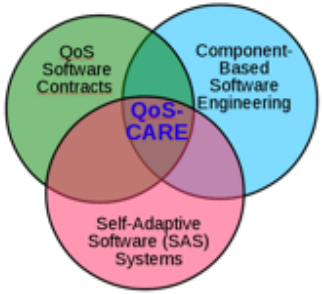
Departure Points (1): Background Research Areas

The main research question:

How to preserve QoS contracts fulfillment in component-based systems through dynamic reconfiguration reliably and autonomously ?



Departure Points (2): Secondary Research Questions



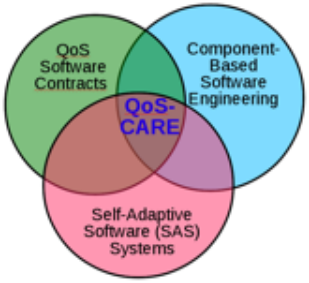
Q1: How to specify QoS contracts?

- QoS Contract Specifications [Frølund and Koistinen, 1998; Keller and Ludwig, 2003]
- Languages and models [Becker, 2008; Jureta et al., 2009]

Q2: What is the QoS contract semantics?

- QoS Contract Models [Meyer, 1992; Beugnard et al., 1999; ISO, 2001; Collet et al., 2005; Comuzzi and Pernici, 2009]
- Formal semantics [Braga et al., 2009; Cansado et al., 2010]

Departure Points (3): Secondary Research Questions

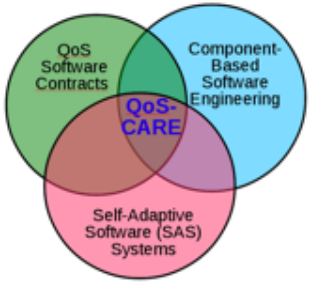


Q3: What components need to be reconfigured to preserve a QoS contract?
How?
Into what?
At run-time!

- CBSE Foundations [Szyperski, 1998; Bachmann et al., 2000; Heineman and Councill, 2001; Papazoglou et al., 2007]
- CBSE models and frameworks [Bruneton et al., 2006; Beisiegel et al., 2007a; Bures, 2007; OMG, 2009; Seinturier et al., 2009]

- SAS Foundations [Shaw, 1994; Oreizy et al., 1999; Kephart and Chess, 2003; Hellerstein et al., 2004; IBM, 2006; Kramer and Magee, 2007; Müller et al., 2008; Cheng et al., 2009a; de Lemos et al., 2011]
- SAS Approaches [Garlan et al., 2003; Litoiu et al., 2005; Solomon et al., 2007; Weyns et al., 2010; Parra et al., 2011; Bentancour et al., 2011]

Departure Points (4): Secondary Research Questions



Q4: How to manage context uncertainty?

- Approaches [Murray et al., 2003; Goldsby and Cheng, 2008; Lin et al., 2009]

Q5: How to evaluate self-adaptive software?

- Approaches [Candea et al., 2004; Garlan et al., 2004; Salehie and Tahvildari, 2009; Cardellini et al., 2009; Cheng et al., 2009b; Léger et al., 2010]

3. Contributions

What are formal models useful for?

Finding the appropriate “handles” to grab the problem

"Give me a place to stand, and I shall move the Earth with a lever"

(Archimedes)

General Contributions

1. Adaptation Properties

2. A Formal Model for QoS Contracts-Preserving Reliable Reconfiguration

3. QoS-CARE Architecture, Implementation, and Validation

Adaptation Properties
Specific Contributions

- Evaluation Framework
  - Characterizing dimensions for SAS systems
  - Identification of quality factors

- Catalog of adaptation properties: means for
  - Standard evaluation
  - Verifying and guaranteeing self-adaptation goals
  - Determining the reliability of QoS-CARE

>50 analyzed papers 20 characterized papers 6 dimensions 37 classification options	4 SASO properties - Stability - Accuracy - Settling time - Small overshoot	9 adaptation properties	4 quality attributes 12 quality factors

General Contribution 1:
Adaptation Properties

How to determine if a solution (e.g., QoS-CARE) is acceptable?

Adaptation properties are fundamental to certify, compare and combine SAS approaches

No cle
assessm
methods
self-adap

eparation of
erns in the
essment

A Framework for Evaluating Quality-Driven Self-Adaptive Software Systems.
N. Villegas, H. Müller, G. Tamura, L. Duchien, R. Casallas. SEAMS 2011.

Adaptation Properties (partial view)

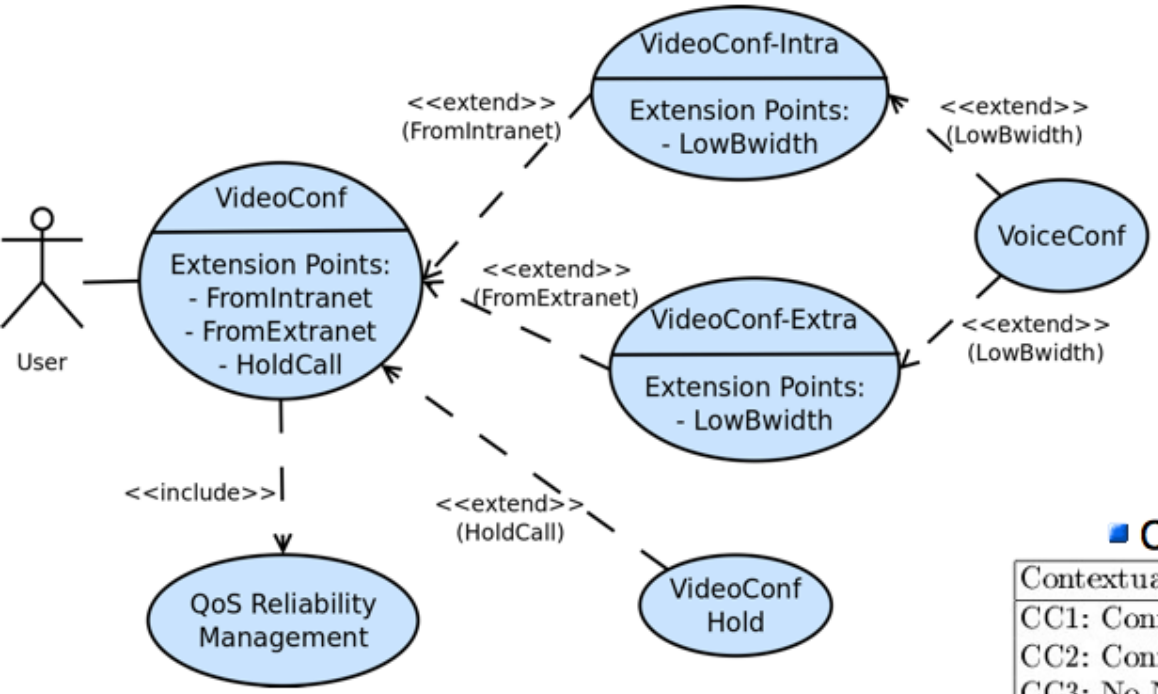
Adaptation Property	Quality Attributes	
Stability	Performance	Latency Throughput Capacity
	Dependability	Safety Integrity
	Security	Integrity
Accuracy	Performance	Latency Throughput Capacity
Settling Time	Performance	Latency Throughput
Small Overshoot	Performance	Latency Throughput Capacity
Robustness	Dependability	Availability Reliability
	Safety	Interact. Complex. Coupling Strength
Termination	Dependability	Reliability Integrity
Consistency	Dependability	Maintainability Integrity
Scalability	Performance	
Security	Security	Confidentiality Integrity Availability

SCA integrity constraints

General Contribution 2:
A Formal Model for QoS Contracts-Preserving Reliable Reconfiguration

Example Scenario

Reliable Mobile Videoconference System (RVCS):



Represented in a QoS Contract:

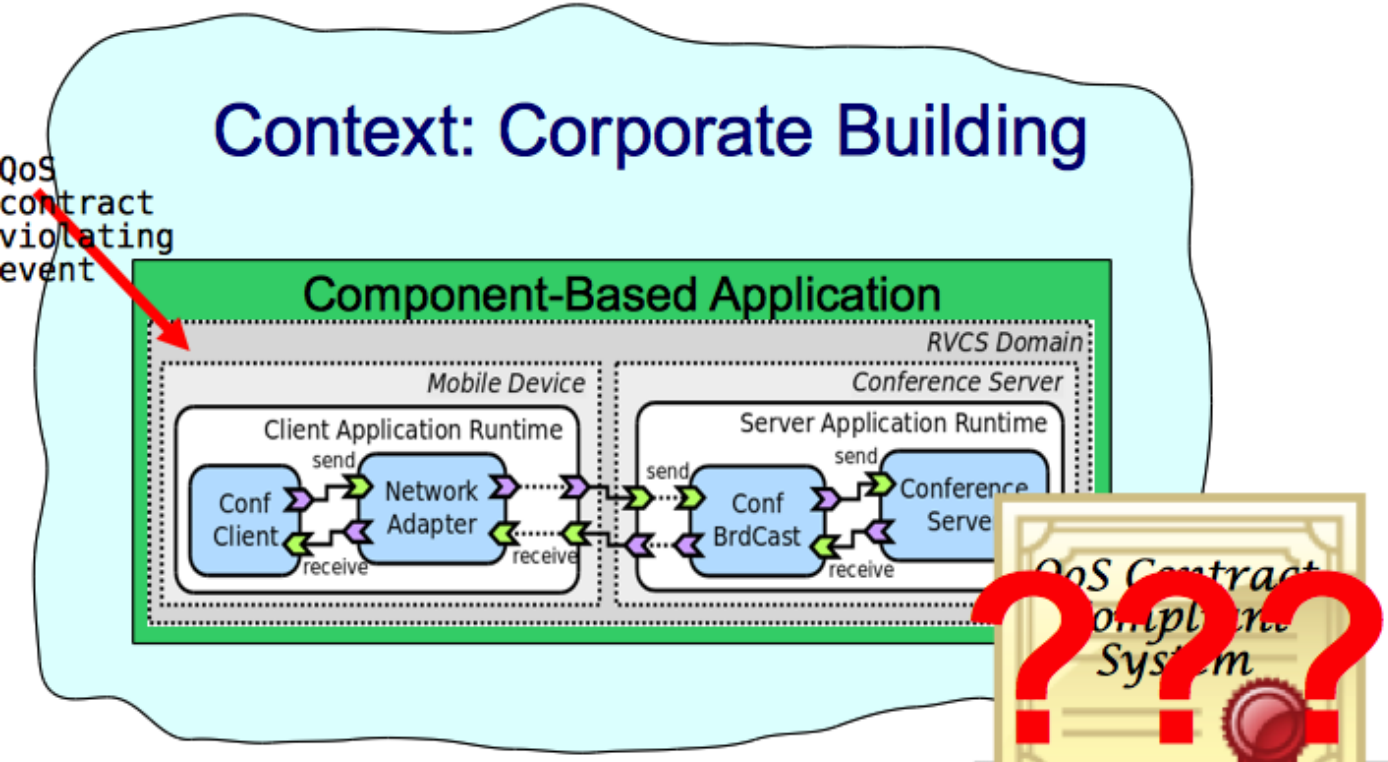
- Confidentiality: access point

Contextual Condition	Service Level Objective
CC1: Connection from Intranet	Clear Channel
CC2: Connection from Extranet	Confidential Channel
CC3: No Network Connection	Call on Hold

- Availability: bandwidth

Contextual Condition	Service Level Objective
CC4: $BandWidth \leq 12$	Call on Hold
CC5: $12 < BandWidth \leq 128$	Voice Call
CC6: $128 < BandWidth$	Voice and Video Call

The Component-Based Application Subject to a QoS Contract



Events (e.g):

- Change user location (access point, bandwidth, ...)
- Season change
- Calendar date (conference start,...)

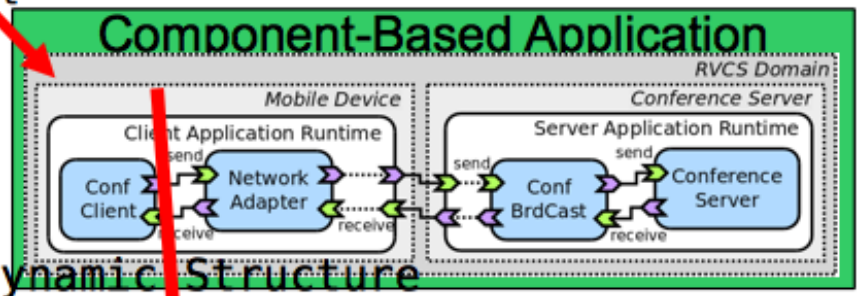
C	Service Level Objective
CC1: Connection [illegible] Intranet	Clear Channel
CC2: Connection from Extranet	Confidential Channel
CC3: No Network Connection	Call on Hold

Contextual Condition	Service Level Objective
CC4: $BandWidth \leq 12$	Call on Hold
CC5: $12 < BandWidth \leq 128$	Voice Call
CC6: $128 < BandWidth$	Voice and Video Call

How to Preserve the Contract under the New Context Condition?

Violating event: move to extranet

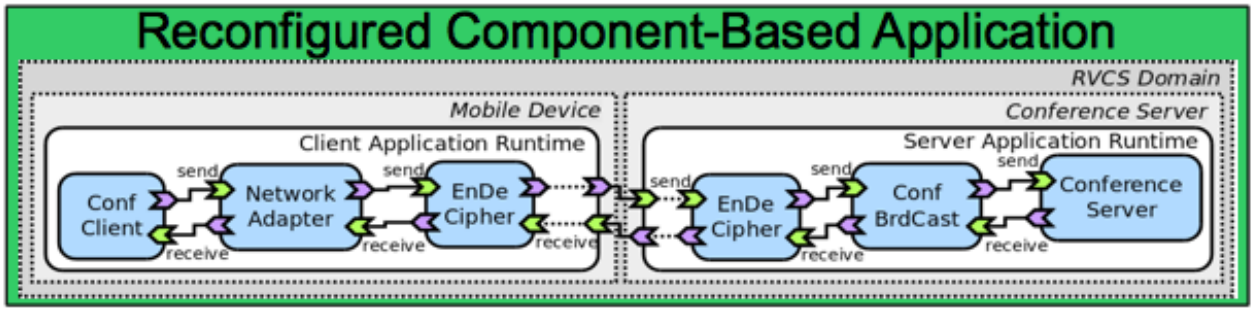
Context: intranet



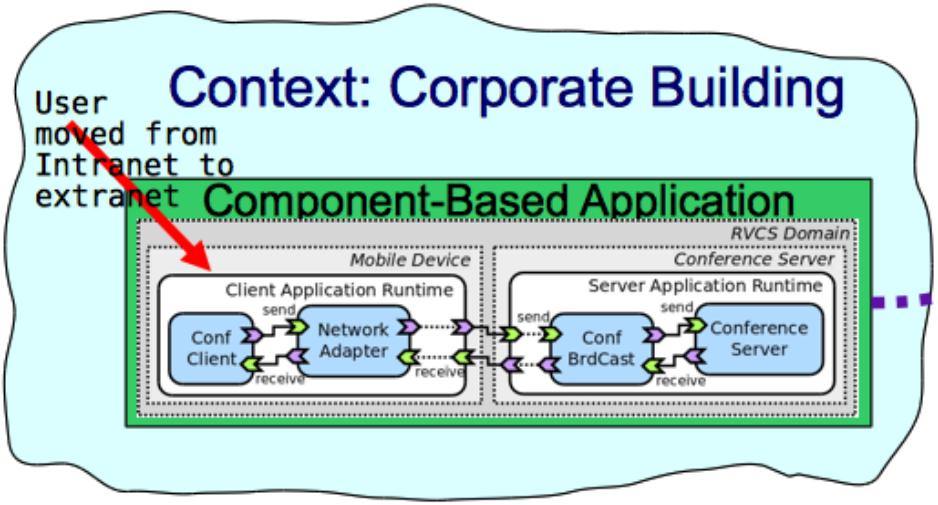
Dynamic Structure Reconfiguration

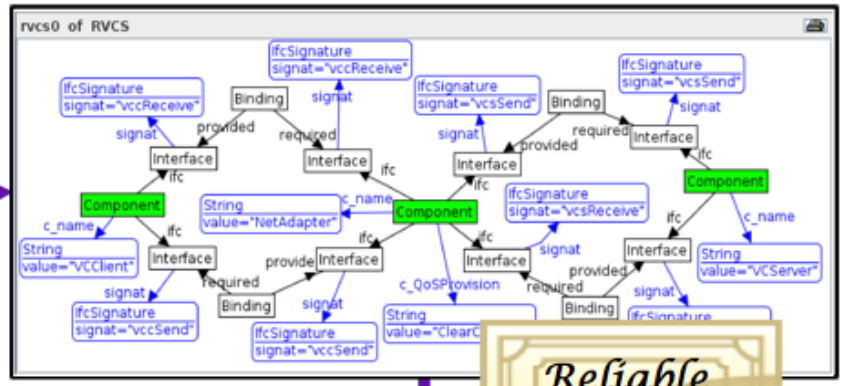
The secure-proxy design pattern

Context: extranet

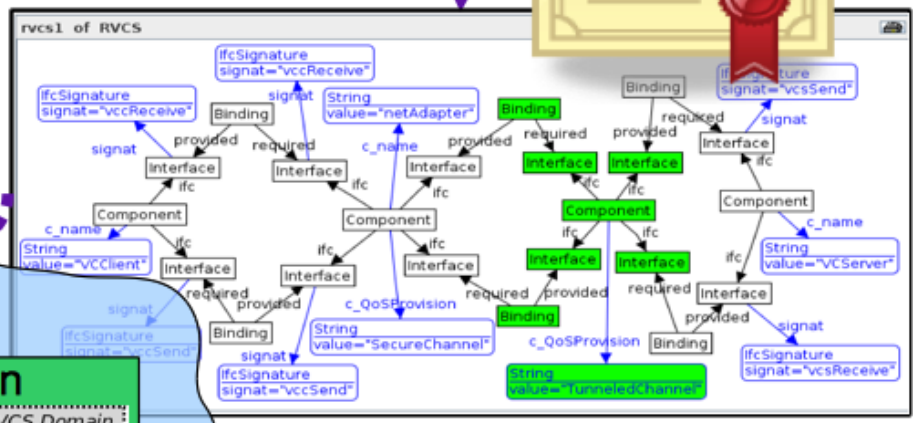


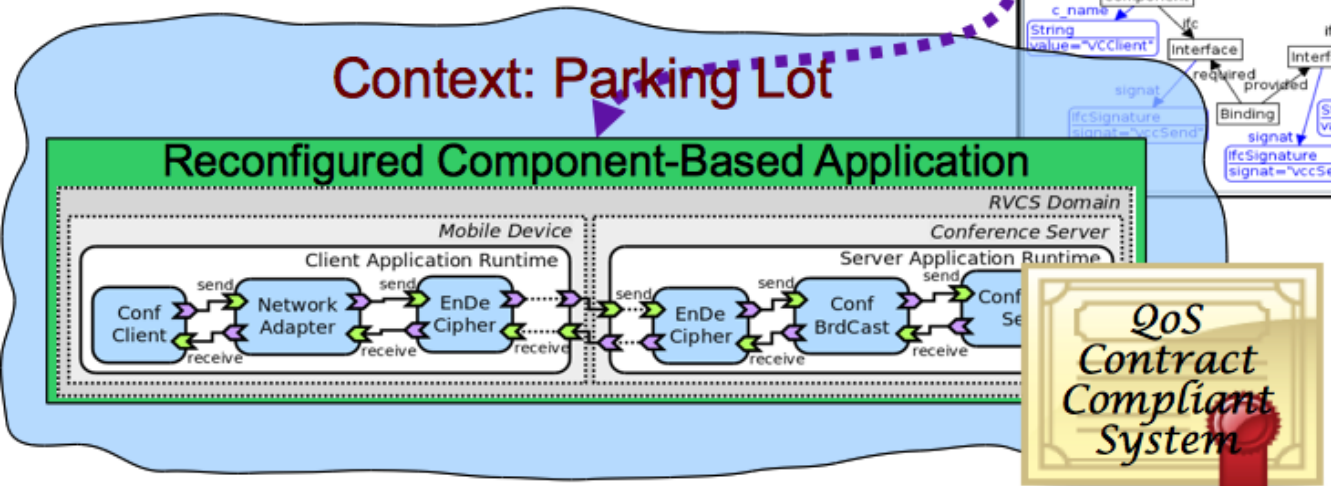
QoS-CARE Approach: TAGTS





Formal Properties





Formal Models
Specific Contributions

I. Formal Models (e-graphs) for:

1. Component-Based Software Structure

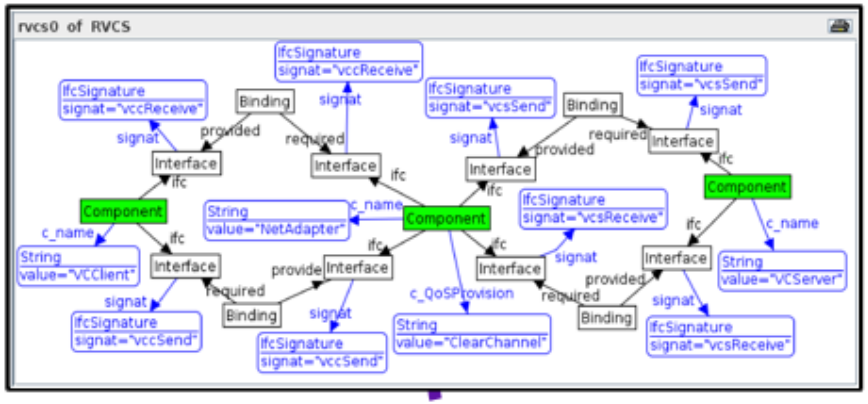
2. QoS Contracts Structure

3. Reconfiguration Rules
Design patterns encoded in LHSs and RHSs

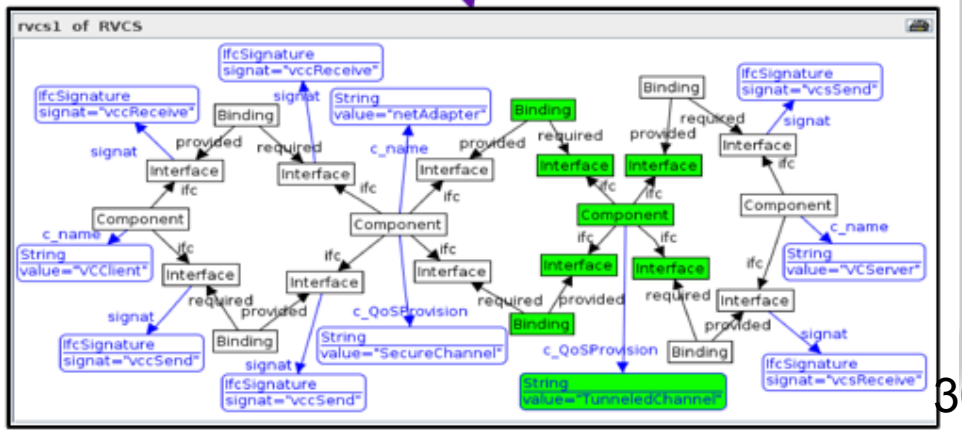
QoS Contract-Aware Reconfiguration of Component Architectures Using E-Graphs.
G. Tamura, A. Cleve, R. Casallas and L. Duchien. FACS'2010.

Formal Models
Specific Contributions

II. Reconfiguration System (TAGTS Extension)



Graph pattern-matching and transformation



Reconfiguration rules:

- Parameterized in the contract
- Provided by the user (software evolution architect)
- design patterns @ runtime

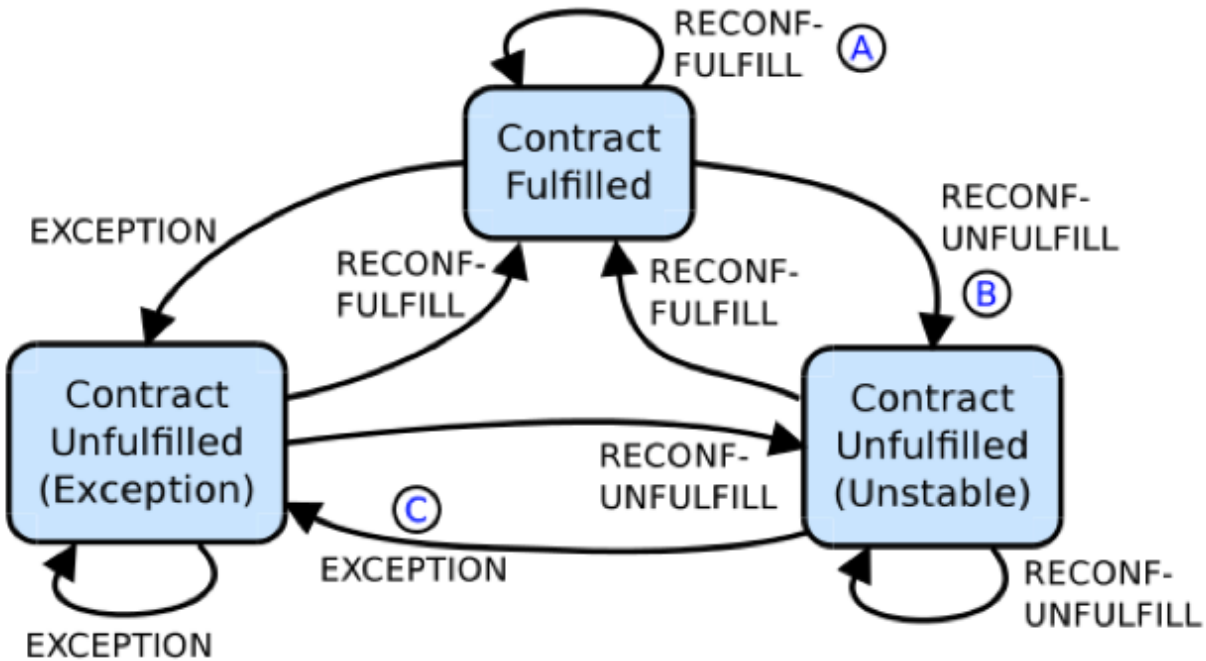
Is TAGTS+ enough?

Dealing with Context Uncertainty

Formal Models: Specific Contributions

III. QoS Contract Semantics (FSM extension)

- QoS contract states of fulfillment: enriched with unfulfillment states



This semantics provide

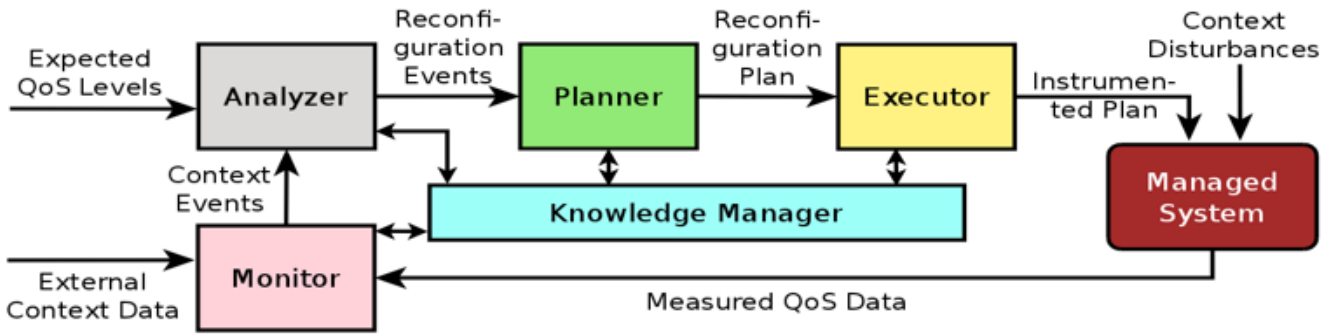
- Autonomous QoS Contract Preservation
- Uncertainty Managemer

General Contribution 3: QoS-CARE Architecture, Implementation, and Validation

How to implement the Formal Model?

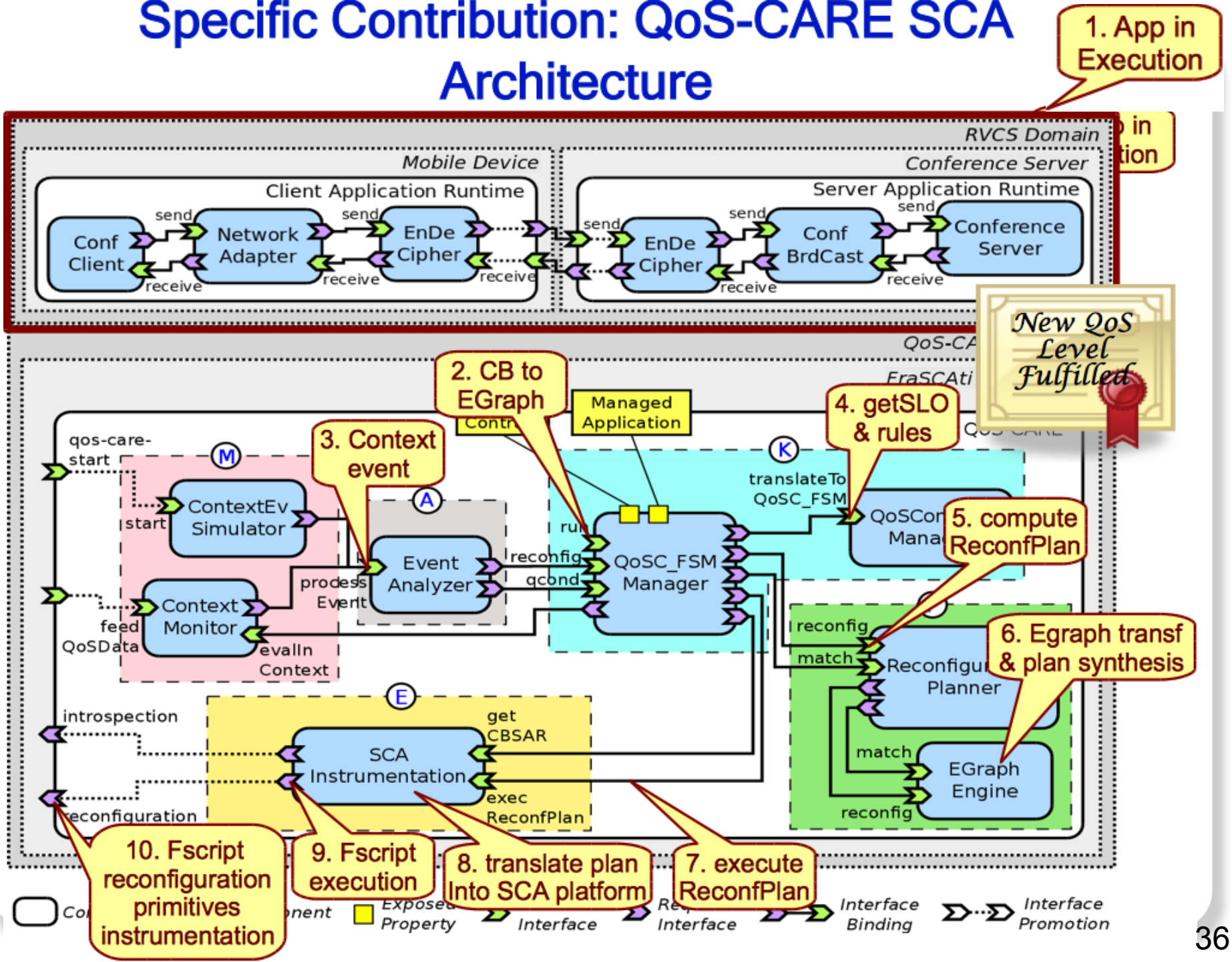
Maintaining the Formal Model Properties
while
Sticking to SEfSAS Design Principles

Specific Contribution: The Formal Model in the Unified MAPE-K/Feedback Loop



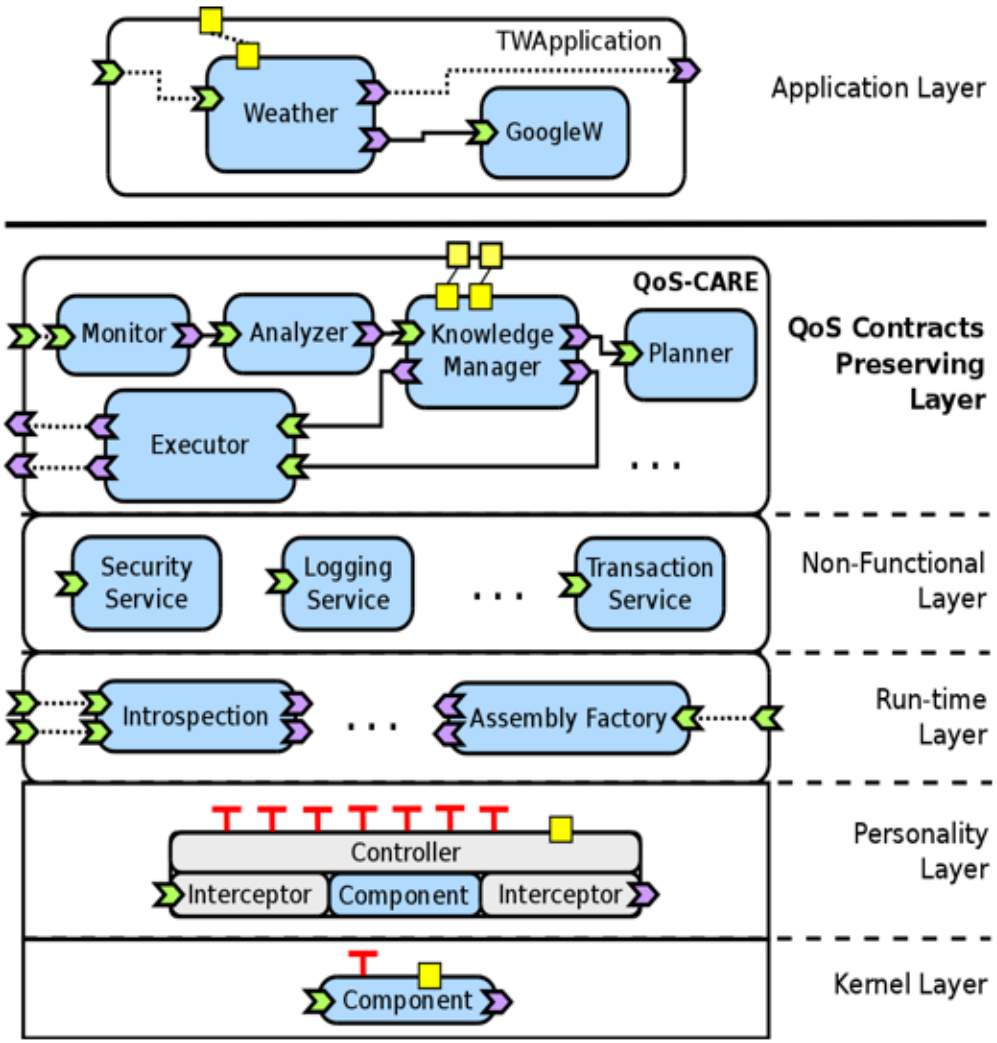
QoS-CARE Reconfiguration Loop Element	*QoSC_FSM* Formal Model Operations
Monitor	$-evalInContext : PREDICATE \times \Delta^U \rightarrow \{F, U\}$
Analyzer	$-$ (uses) $match : CBSRR \times CBSAR \rightarrow Boolean$ $-$ (uses) $evalInContext : PREDICATE \times \Delta^U \rightarrow \{F, U\}$
Planner	$-reconfig : CBSAR \times \mathcal{P}(\Gamma) \rightarrow Boolean$ $-match : CBSRR \times CBSAR \rightarrow Boolean$
Executor	$-getCBSAR : \rightarrow CBSAR$ $-execReconfPlan : INSTRUCTION^* \rightarrow Boolean$
Reconfiguration Knowledge Manager	$-\Psi : STATES \rightarrow PREDICATE$ $-\kappa : STATES \rightarrow PREDICATE$ $-\eta : STATES \rightarrow \mathcal{P}(\Sigma)$ $-\rho : STATES \rightarrow \mathcal{P}(\Gamma)$ $-\pi : STATES \rightarrow QoSProperty$

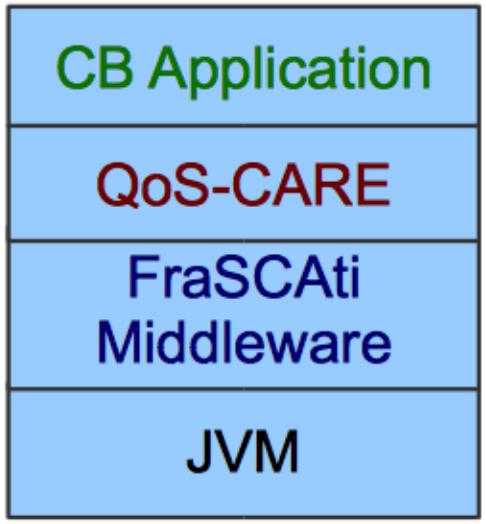
Specific Contribution: QoS-CARE SCA Architecture



Specific Contribution: QoS-CARE in the SCA Stack

An SCA layer for dynamic reconfiguration in FraSCAti





4. Validation

Formal Models Re-evaluated

Not A Solution for Everything, yet A Good Decision

Specific Contribution: QoS-CARE Properties V&V

Reconfiguration Independence and Separation of Concerns in QoS-CARE:

- Re-usable
- Comparable
- Maintainable

Towards Practical Runtime Verification and Validation of Self-Adaptive Software Systems. G. Tamura, N. Villegas, H. Müller, J. Sousa, B. Becker, M. Pezzè, G. Karsai, et al. SEfSAS Book, 2012.

DYNAMICO: A Reference Model for Governing Control Objectives and Context Relevance in Self-Adaptive Software Systems. N. Villegas, H. Müller, G. Tamura, L. Duchien, R. Casallas. SEfSAS Book, 2012.

Validation Scenarios

Determining the practical feasibility of QoS-CARE

QoS-CARE Applied to Preserve QoS Contracts

(Executed in FraSCAti)

1. A mobile videoconference system (RVCS)
QoS contract: availability and confidentiality

2. A simple Web mashup application (Twitter / Weather)
QoS contract: service readiness

QoS-CARE Benchmark (Overhead and Settling-time)

Benchmark Results
Confirm the
QoS-CARE Practical Feasibility

[b] Rule for changing to a "confident channel" from a "clear channel" configuration.
[c] Instrumented over a ~4Mbps, 11 hops Internet connection.

5. Conclusions & Future Work

Conclusions

Self-adaptive software properties and metrics: to standardize SAS assessment (compare, improve, combine approaches)

Formal model for QoS contracts preservation through self-reconfiguration providing verifiable properties:

- Termination
- Atomicity
- SCA structural conformance
- Robustness
- Settling-time

Architecture and implementation: platform-independent SCA layer for dynamic reconfiguration

Reliability reconfiguration properties: validated formally and experimentally

Benchmark results: practical feasibility (settling-time, overhead), re-usability

Future Work

Leverage design-patterns @ run-time

Improve robustness to context uncertainty

Guarantee additional adaptation properties

Thank You!

Publications

International Journals

- Gabriel Tamura; Rubby Casallas; Anthony Cleve; Laurence Duchien. QoS Contract Preservation through Dynamic Reconfiguration: A Formal Semantics Approach, Science of Computer Programming, 94(3):307-332, Elsevier, 2014.

Book chapters

- Gabriel Tamura, Norha M. Villegas, Hausi A. Müller, João P. Sousa, Basil Becker, Mauro Pezzè, Gabor Karsai, Serge Mankovskii, Wilhelm Schäfer, Ladan Tahvildari, and Kenny Wong. Towards Practical Runtime Verification and Validation of Self-Adaptive Software Systems. In: Software Engineering for Self-Adaptive Systems 2, LNCS Vol. 7475. 2013.
- Norha M. Villegas, Gabriel Tamura, Hausi A. Müller, Laurence Duchien, and Rubby Casallas. DYNAMICO: A Reference Model for Governing Control Objectives and Context Relevance in Self-Adaptive Software Systems. In: Software Engineering for Self-Adaptive Systems 2, , LNCS Vol. 7475. 2013..
- Rogerio de Lemos, Holger Giese, Hausi Müller, Mary Shaw, Jesper Andersson, Luciano Baresi, Basil Becker, Nelly Bencomo, Yuriy Brun, Bojan Cukic, Ron Desmarais, Schahram Dustdar, Gregor Engels, Kurt Geihs, Gabriel Tamura, *et al*. Software Engineering for Self-Adaptive Systems: A Second Research Roadmap. In: Software Engineering for Self-Adaptive Systems 2, LNCS Vol. 7475. 2013..

International Symposiums

- Norha Villegas, Hausi Müller, Gabriel Tamura, Laurence Duchien, and Rubby Casallas. A Framework for Evaluating Quality-Driven Self-Adaptive Software Systems. Procs. of 6th Intl. Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2011.

International Workshops

- Gabriel Tamura, Rubby Casallas, Anthony Cleve, and Laurence Duchien. QoS Contract-Aware Reconfiguration of Component Architectures Using E-Graphs. Procs. of 7th Intl. Workshop on Formal Aspects of Component Software, 2010. Selected also for publication on the LNCS Vol. 6921 – Formal Aspects of Component Software

Thesis

- Gabriel Tamura QoS-CARE: A Reliable System for Preserving QoS Contracts through Dynamic Reconfiguration, Ph.D. dissertation, University of Lille 1 - Science and Technology, and Universidad de Los Andes, 2012.