Towards Accurate Failure Prediction for the Proactive Adaptation of Service-oriented Systems

Andreas Metzger

ASAS@ESEC 2001
Szeged, Hungary
September 2011

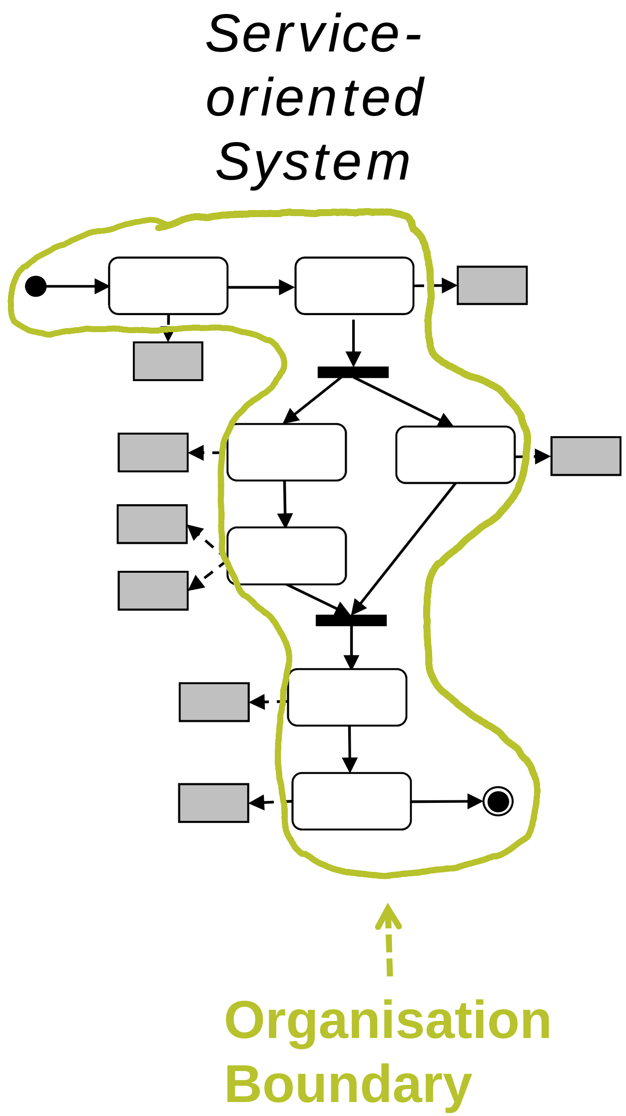
AGENDA

- Need for Proactive Adaptation
- Online Failure Prediction and Accuracy
- Experimental Assessment of Existing Techniques
- Observations & Future Directions

Service-oriented Systems

About *[Di Nitto et al. 2008]*

- Software services **separate**
  - ownership, maintenance and operation
  - from use of software
- Service users: **no need to acquire, deploy and run** software
  - Access the functionality of software from remote through service interface
- Services **take concept of ownership to extreme**
  - Software is fully executed and managed by 3rd parties
  - *Cf. COTS:* where "only" development, quality assurance, and maintenance is under control of third parties



Service-oriented Systems

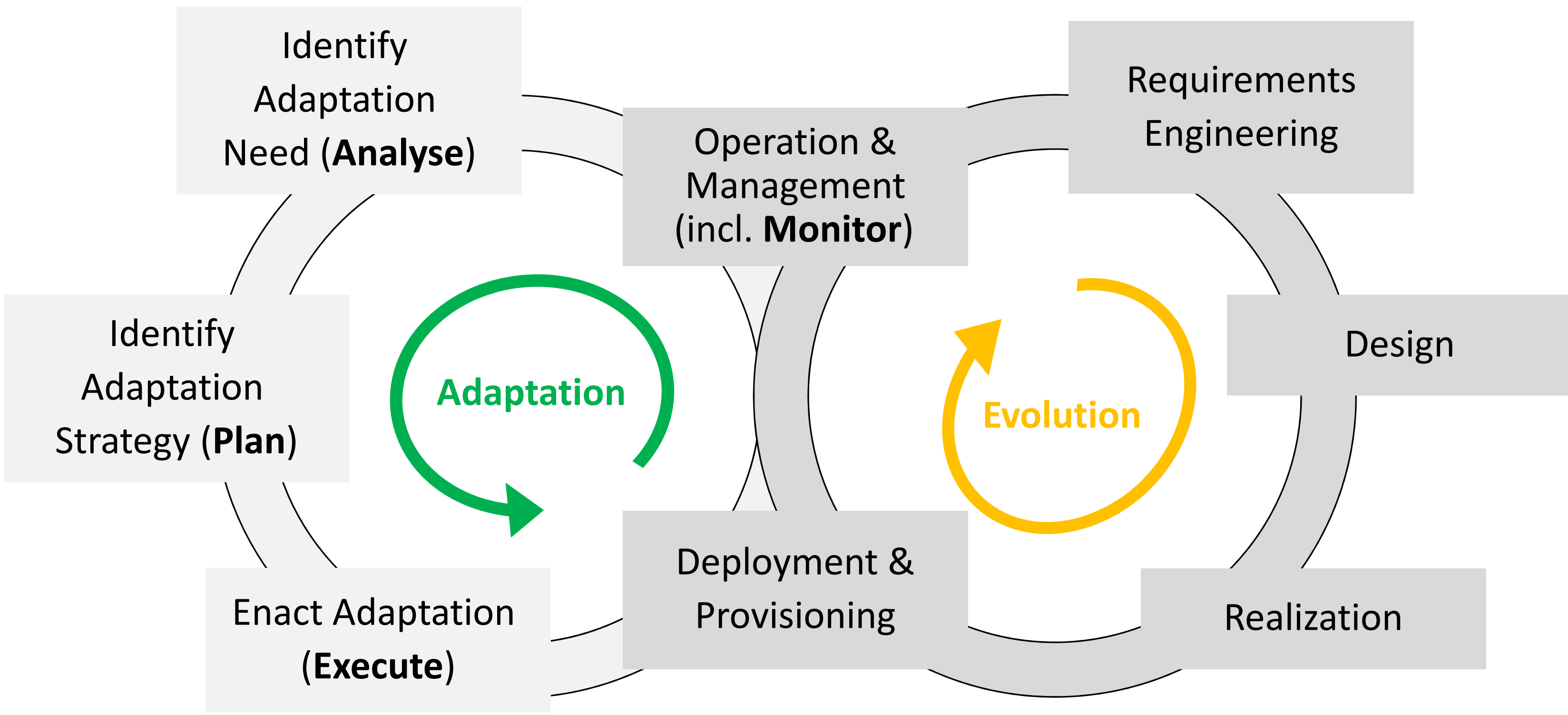
Need for Adaptation

- **Highly dynamic changes due to**
  - 3rd party services, multitude of service providers, …
  - evolution of requirements, user types, …
  - change in end-user devices, network connectivity, …
- **Difference from traditional software systems**
  - Unprecedented level of change
  - No guarantee that 3rd party service fulfils its contract (SLA)
  - Hard to assess behaviour of infrastructure (Internet) at design time

Service-oriented Systems

Need for Adaptation

S-Cube Service Life-Cycle Model



Run-time („MAPE“ loop)

Design time

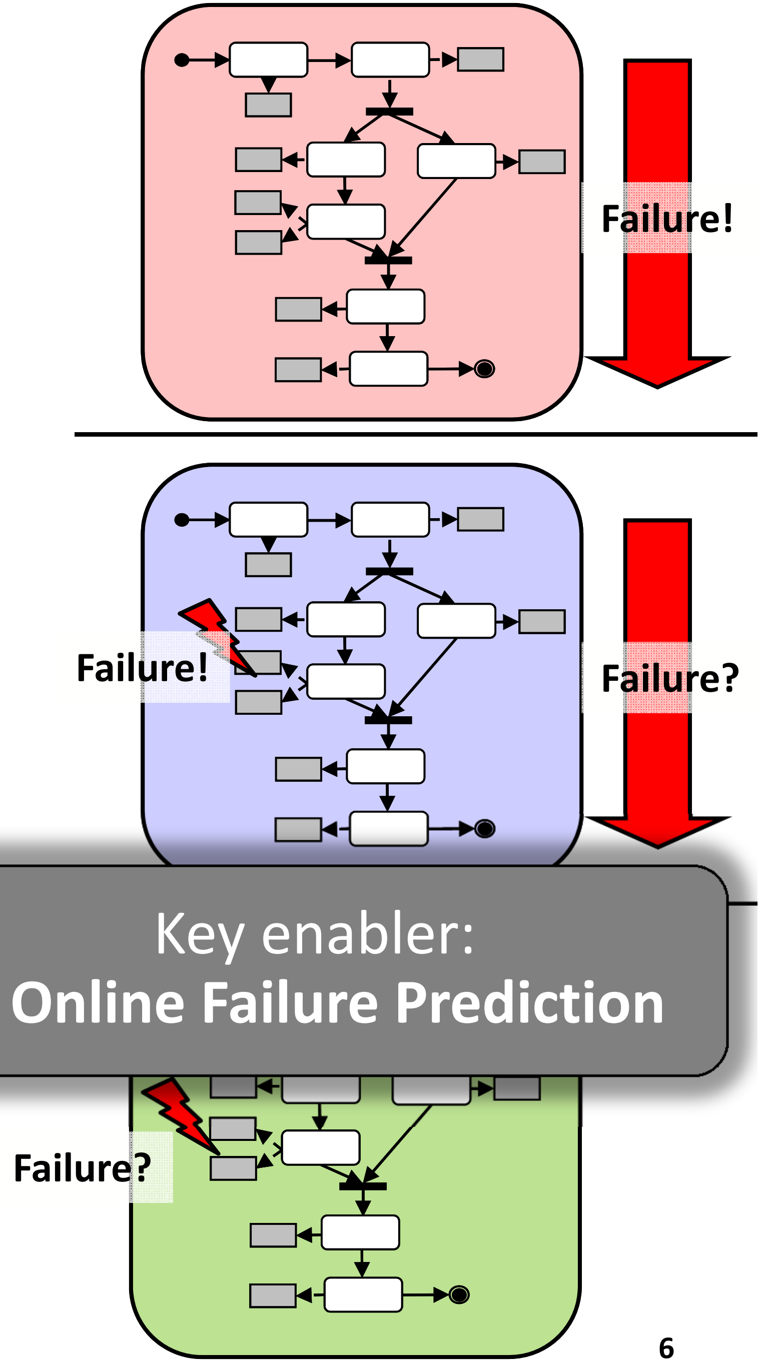
Types of Adaptation

Types of Adaptation (general differences)

- **Reactive Adaptation**
  - **Repair/compensate external failure visible to the end-user**
- **Preventive Adaptation**
  - A local failure (deviation) occurs

  → Will it lead to an external failure?
  - <u>If “yes”</u>: **Repair/compensate local failure (deviation) to prevent external failure**
- **Proactive Adaptation**

  → Is local failure /deviation imminent (but did not occur)?
  - <u>If “yes”</u>: **Modify system before local failure (deviation) actually occurs**



Key enabler:
Online Failure Prediction

The Ruhr Institute for Software Technology

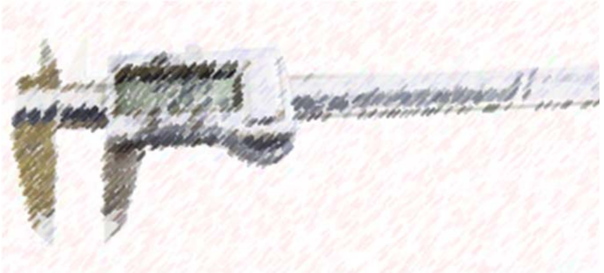
AGENDA

- Need for Proactive Adaptation
- Online Failure Prediction and Accuracy
- Experimental Assessment of Existing Techniques
- Observations & Future Directions

Need for Accuracy

Requirements on Online Failure Prediction

- **Prediction must be efficient**
  - Time available for prediction and repairs/changes is limited
  - If prediction is too slow, not enough time to adapt



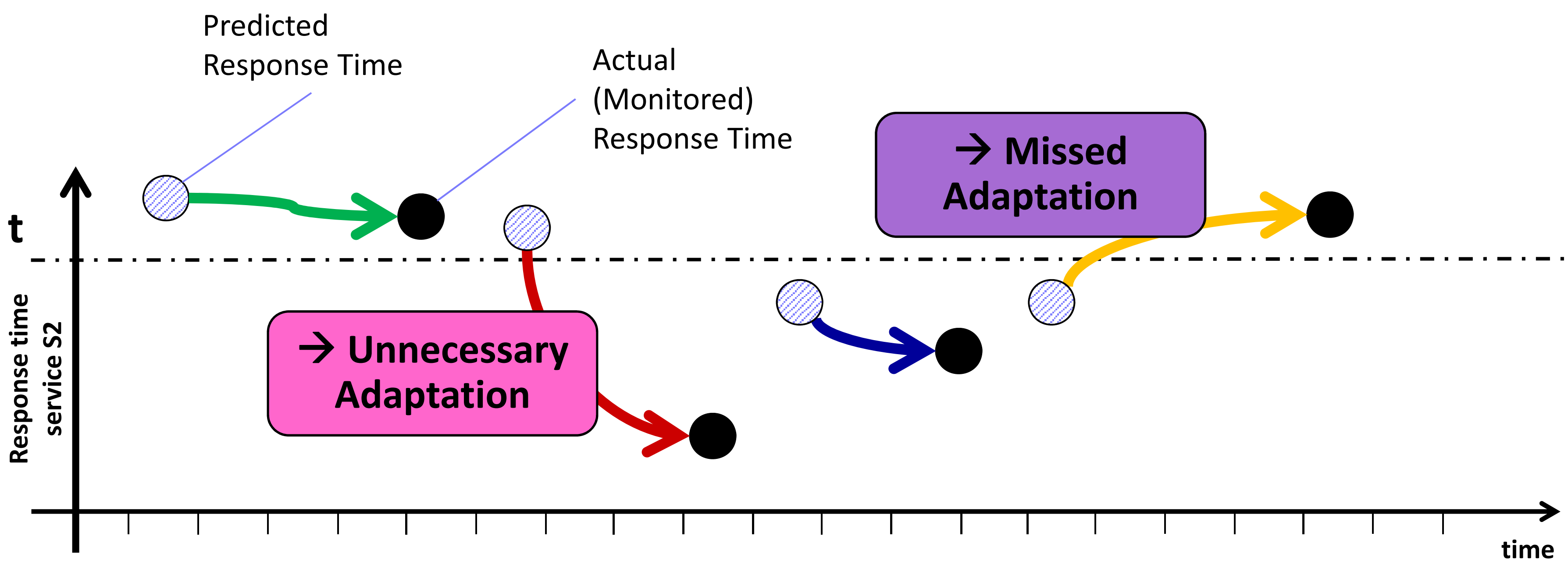
- **Prediction must be accurate**
  - **Unnecessary adaptations** can lead to
    - **higher costs** *(e.g., use of expensive alternatives)*
    - **delays** (possibly leaving less time to address real faults)
    - **follow-up failures** (*e.g., if alternative service has severe bugs*)
  - **Missed proactive adaptation opportunities** diminish the benefit of proactive adaptation
    (*e.g., because reactive compensation actions are needed*)

Measuring Accuracy

Contingency Table Metrics

(see [Salfner et al. 2010])

	Actual Failure	Actual Non-Failure
Predicted Failure	True Pos.	False Pos.
Predicted Non-Failure	False Neg.	True Neg.



Measuring Accuracy

Some Contingency Table Metrics *(see [Salfner et al. 2010])*

Precision:

$$p = \frac{TP}{TP + FP}$$

How many of the predicted failures were actual failures?

Higher *p* → less unnecessary adaptations

Negative Predictive Value:

$$v = \frac{TN}{TN + FN}$$

How many of the predicted non-failures were actual non-failures?

Higher *v* → less missed adaptations

Recall (True Positive Rate):

$$r = \frac{TP}{TP + FN}$$

How many of the actual failures have been correctly predicted as failures?

Higher *r* → less missed adaptations

Specificity (True Negative Rate):

$$s = \frac{TN}{TN + FP}$$

How many of the actual non-failures have been correctly predicted as non-failures?

Higher *s* → less unnecessary adaptations

Measuring Accuracy

Other Metrics

Accuracy

$$a = \frac{TP + TN}{TP + TN + FP + FN}$$

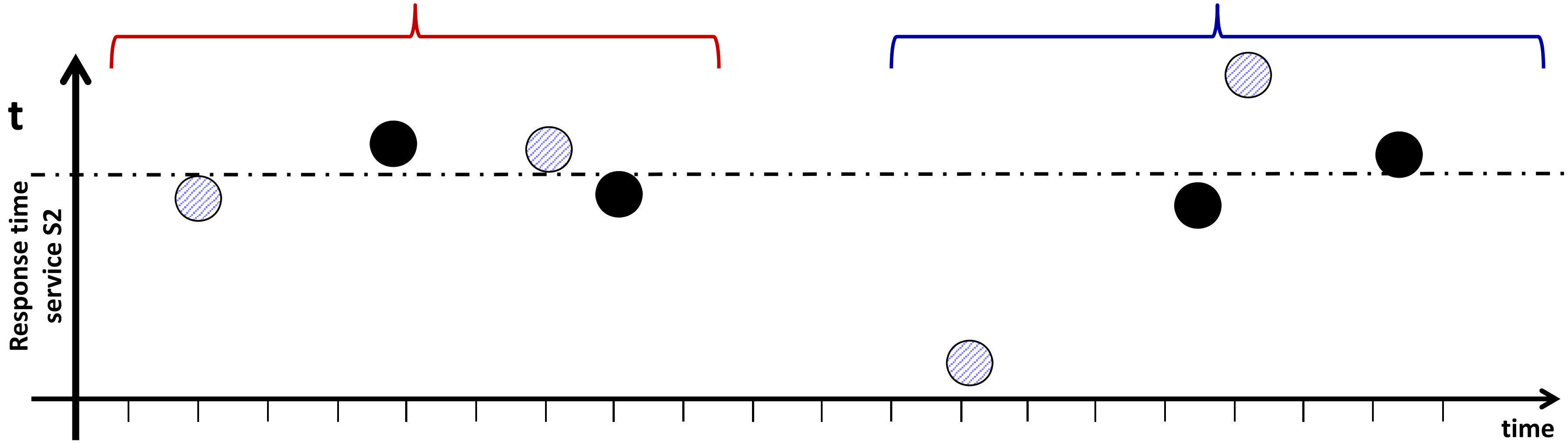
How many predictions were correct?

- Actual failures usually are rare
  → prediction that always predicts "non-failure" can achieve high *a*

Prediction Error

- Does not reveal accuracy of prediction in terms of SLA violation *(also see [Cavallo et al. 2010])*

Small error, but wrong prediction of violation ⇔ **Large error, but correct prediction of violation**



Caveat: Contingency table metrics influenced by the threshold value of SLA violation

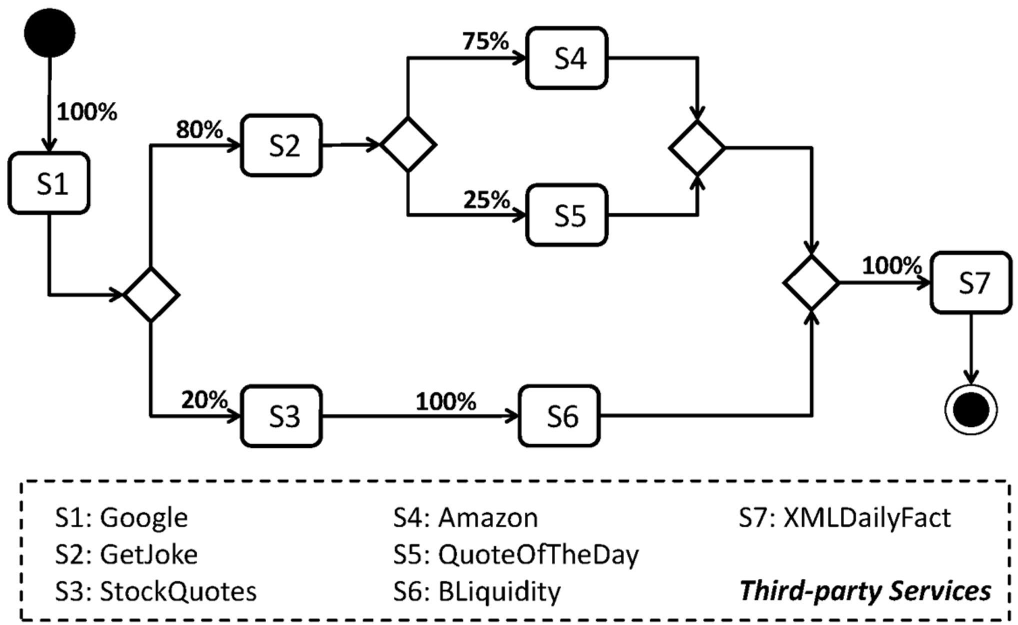
AGENDA

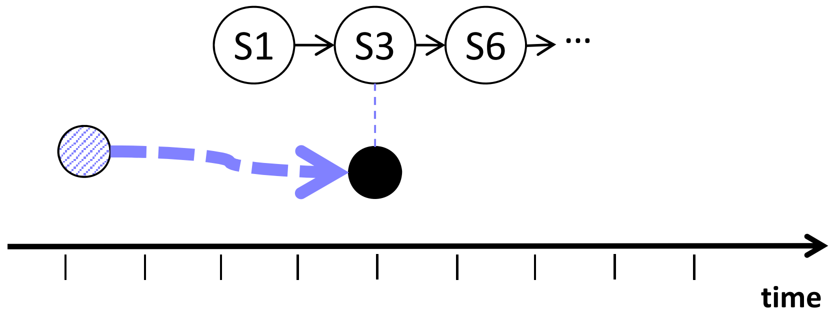
- Need for Proactive Adaptation
- Online Failure Prediction and Accuracy
- Experimental Assessment of Existing Techniques
- Observations & Future Directions

Experimental Assessment

Experimental Setup

- Prototypical implementation of different prediction techniques
- Simulation of example service-oriented system (100 runs, with 100 running systems each)
- (Post-mortem) monitoring data from real services (2000 data points per service; QoS = performance measured each hour) *[Cavallo et al. 2010]*
- Measuring contingency table metrics (for S1 and S3)
  - Predicted based on ”actual” execution of the SBA





Experimental Assessment

Prediction Techniques

- **Time Series**
  - **Arithmetic average:**

$$\widehat{m}_t = \frac{1}{n}\sum_{i=1}^{n} m_{t-i}$$

  - Past data points: ***n* = 10**

  - **Exponential smoothing:**

$$\widehat{m}_t = \alpha \cdot m_{t-1} + (1-\alpha) \cdot \widehat{m}_{t-1}$$

  - Weight: α **= .3**

Experimental Assessment

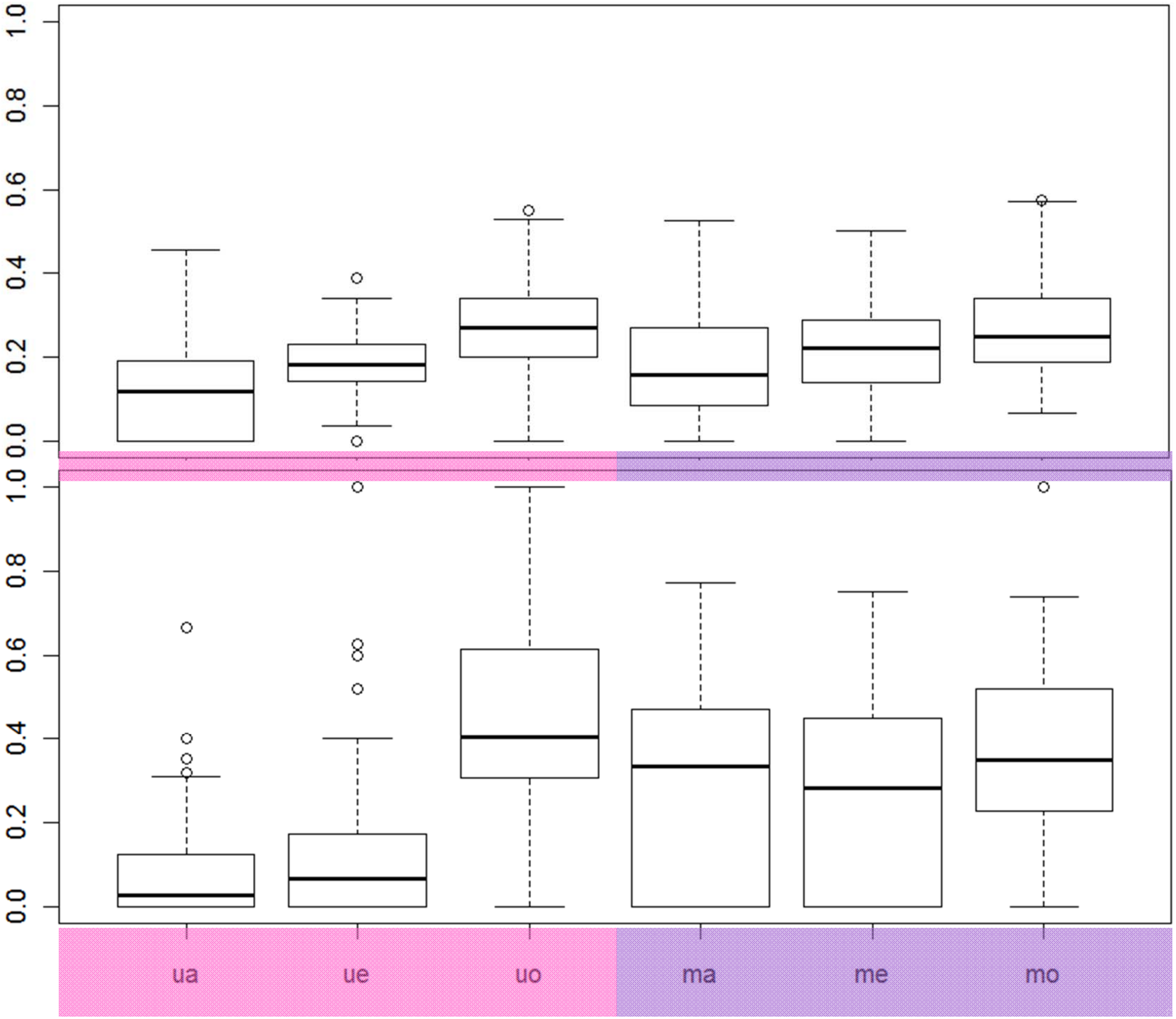
Prediction Techniques

- **Online Testing:**
  - **Observation:** Monitoring is “observational”/“passive”

  → May not lead to “timely” coverage of service
  (which thus might diminish predictions)
  - **Our solution: PROSA** *[Sammodi et al. 2011]*
    - Systematically test services in parallel to normal use and operation *[Bertolino 2007, Hielscher et al. 2008]*
    - **Approach:** “Inverse” usage-based test of services
      - If service has seldom been used in a given time period dedicated online tests are performed to collect additional evidence for quality of the service
      - Feed testing and monitoring results into prediction model (here: **arithmetic average, *n* = 1**)
      - Maximum 3 tests within 10 hours

Experimental Assessment

Prediction Models – Results



S1
("lots of monitoring data")

S3

$u = p \cdot s$
$m = r \cdot v$

AGENDA

- Need for Proactive Adaptation
- Online Failure Prediction and Accuracy
- Experimental Assessment of Existing Techniques
- Observations & Future Directions

Future Directions

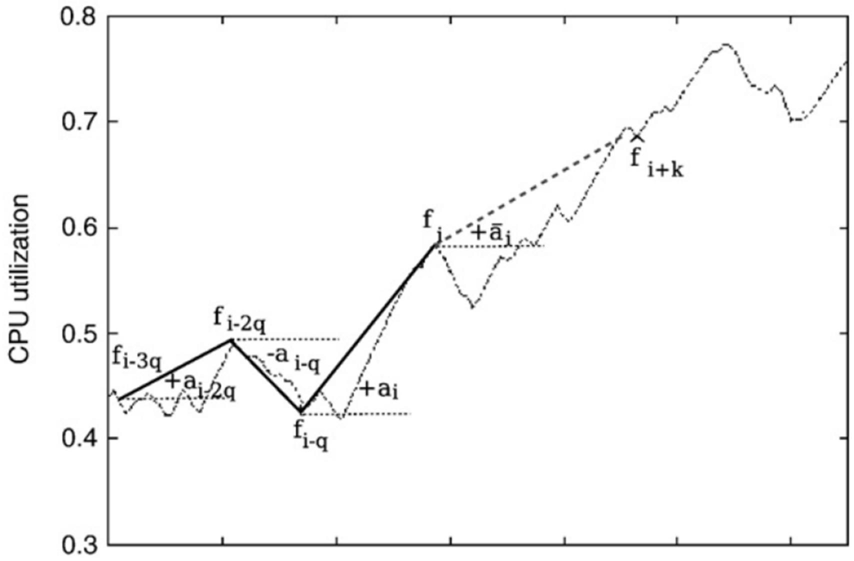
Experimental Observations

- Accuracy of prediction may depend on many **factors**, like
  - **Prediction model**
    - *Caveat:* Only "time series" predictors used in experiments (alternatives: function approx., system models, classifiers, …)
    - *Caveat:* Data set used might tweak observations
      → we are currently working on more realistic benchmarks
    - *NB:* Results do not seem to improve for ARIMA *(cf. [Cavallo et al. 2010])*
  - **Usage setting**
    - *E.g., usage patterns impact on number of monitoring data available*
    - Prediction models may quickly become "obsolete" in a dynamic setting
  - **Time since last adaptation**
    - Prediction models may lead to low accuracy while being retrained
- **Accuracy assessment** is done **"post-mortem"**

Future Directions

Solution Idea 1: Adaptive Prediction Models

- **Example: Infrastructure load prediction** *(e.g., [Casolari & Colajanni 2009])*
  - Adaptive prediction model (considering the trend of the "load" in addition)



  - **Open:** Possible to apply to services / service-oriented systems?

Future Directions

Solution Idea 2: Online accuracy assessment

- **Run-time computation of prediction error** *(e.g., [Leitner et al. 2011])*
  - Compare predictions with actual outcomes, i.e., difference between predicted value and actual value
  - **But:** Prediction error not enough to assess accuracy for proactive adaptation (see above)

- **Run-time determination of confidence intervals** *(e.g., [Dinda 2002, Metzger et al. 2010])*
  - In addition to point prediction determine range of prediction values with confidence interval *(e.g., 95%)*
  - **Again:** Same shortcoming as above

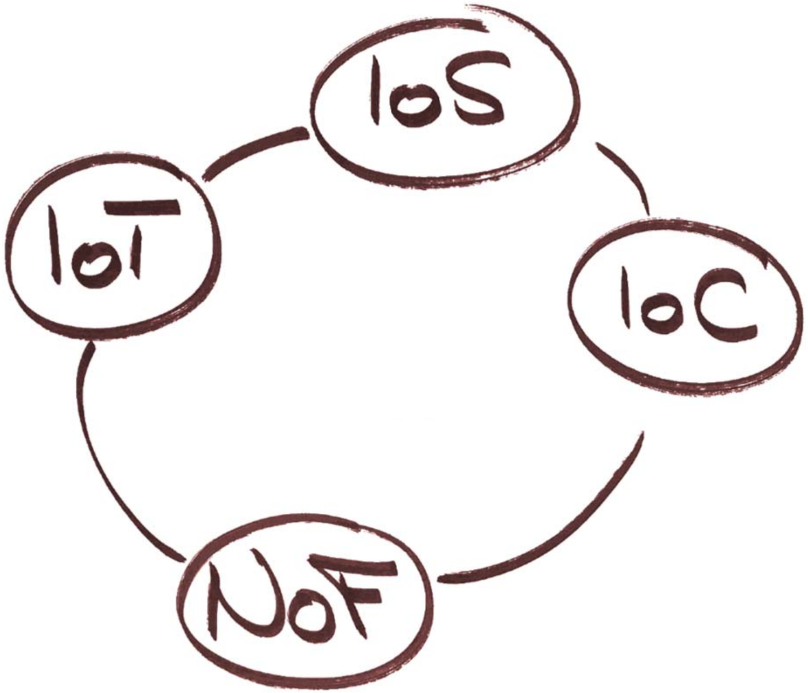
Future Directions

Solution Idea 3: Contextualization of accuracy assessment

- **End-to-end assessment**
  - Understand impact of predicted quality on end-2-end workflow (or parts thereof)
    - Combine with existing techniques such as: machine learning, program analysis, model checking, …
- **Quality of Experience**
  - Assess the perception of quality by the end-user (utility functions)
    - *E.g., 20% deviation might not even be perceived by end-user*
- **Cost Models**
  - Cost of violation may be smaller than penalty, so it may not be a not problem if some of them are missed (small recall is ok)
  - Cost of missed adaptation vs. cost of unnecessary adaptation should be taken into account
    - *E.g., maybe an unnecessary adaptation is not costly / problematic*
  - Cost of applying prediction *(e.g,. Online testing)* vs. benefits

Future Directions

Solution Idea 4: Future Internet *[Metzger et al. 2011, Tselentis et al. 2009]*



Even higher dynamicity of changes
→ More challenges for prediction

But also: **More data for prediction**
→Opportunity for improved prediction techniques

Thank You!

Acknowledgments

Osama Sammodi (Paluno)
Eric Schmieders (Paluno)
Clarissa Marquezan (Paluno)

Danilo Ardagna (Politecnico di Milano)
Manuel Carro (UPM)
Philipp Leitner (TU Vienna)

Members of S-Cube ‘Quality Prediction’ Working Group
http://www.s-cube-network.eu/QP

Funded by the EC’s 7th FP under
Objective 1.2 'Services & Software Architectures, Infrastructures & Engineering‘
http://www.s-cube-network.eu/

References

[Bertolino 2007] A. Bertolino. Software testing research: Achievements, challenges, dreams. In *FOSE 2007*

[Cavallo et al. 2010] B. Cavallo, M. Di Penta, and G. Canfora. An empirical comparison of methods to support QoS-aware service selection. In *PESOS@ICSE 2010*

[Casolari 2009] Sara Casolari, Michele Colajanni. Short-term prediction models for server management in Internet-based contexts. Decision Support Systems 48 (2009) 212–223

[Dinda 2002] P. A. Dinda. Online prediction of the running time of tasks. *Cluster Computing*, 5(3):225–236, 2002.

[DiNitto et al. 2008] E. Di Nitto, C. Ghezzi, A. Metzger, M. P. Papazoglou, and K. Pohl, A journey to highly dynamic, self-adaptive service-based applications, Autom. Softw. Eng., vol. 15, no. 3-4, pp. 313–341, 2008.

[Hielscher et al. 2008] J. Hielscher, R. Kazhamiakin, A. Metzger, and M. Pistore. A framework for proactive self-adaptation of service-based applications based on online testing. In *ServiceWave 2008*

[JRA-1.3.5] O. Sammodi and A. Metzger. Integrated principles, techniques and methodologies for specifying end-to-end quality and negotiating SLAs and for assuring end-to-end quality provision and SLA conformance. Deliverable CD-JRA-1.3.5, S-Cube Consortium, March 2011.

[Leitner et al. 2010] P. Leitner, A. Michlmayr, F. Rosenberg, and S. Dustdar. Monitoring, prediction and prevention of SLA violations in composite services. In *ICWS 2010*

[Metzger et al. 2010] A. Metzger, O. Sammodi, K. Pohl, and M. Rzepka. Towards pro-active adaptation with confidence: Augmenting service monitoring with online testing. In *SEAMS@ICSE 2010*

[Metzger et al. 2011] A. Metzger. C. Marquezan. Future Internet Apps: The next wave of adaptive service-oriented systems? In *ServiceWave 2011*

[Salfner et al. 2010] F. Salfner, M. Lenk, and M. Malek. A survey of online failure prediction methods. *ACM Comput. Surv.*, 42(3), 2010.

[Sammodi et al. 2011] O. Sammodi, A. Metzger, X. Franch, M. Oriol, J. Marco, and K. Pohl. Usage-based online testing for proactive adaptation of service-based applications. In *COMPSAC 2011*

[Tselentis et al. 2009] G. Tselentis, J. Domingue, A. Galis, A. Gavras, and D. Hausheer. *Towards the Future Internet: A European Research Perspective*. IOS Press, 2009.