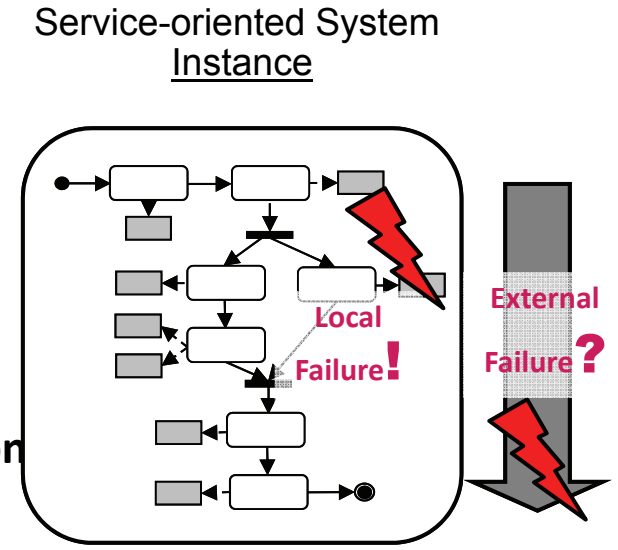
S-Cube Research Challenges on „Quality Prediction“

November 22, 2011

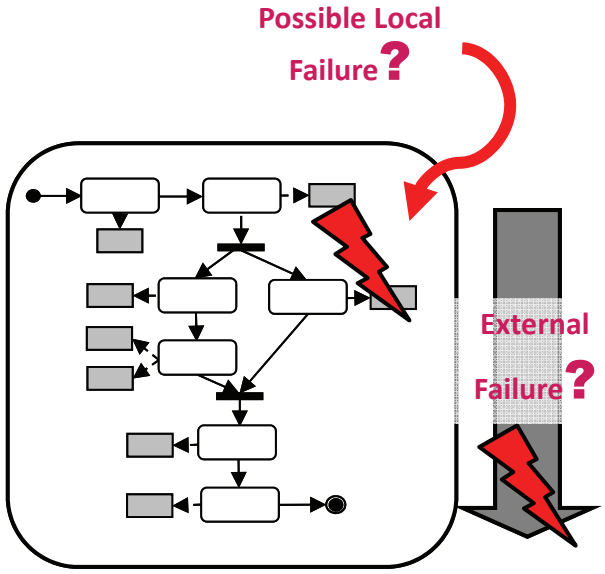
Barcelona, Spain

Usage Settings for Online Failure Prediction

- **Preventive Adaptation**
  - A local failure (deviation) occurs
  - → Will it lead to an external failure?
  - If "yes": **Repair/compensate local failure (deviation) to prevent external failure**



- **Proactive Adaptation**
  - → Is local failure /deviation imminent (but did not occur)?
  - If "yes": **Modify system before local failure (deviation) actually occurs**



Key Challenges for Online Failure Prediction

S-CUBE

RC1 - How to make prediction techniques timely?

- Time available for prediction & repairs/changes is limited
  - Especially considering the need to check adaptation decision during run-time (cf. session 1)
- If prediction is too slow, not enough time to adapt / get decision

RC2 - How to make prediction techniques 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*)

Refined Challenges for Accurate Prediction

S-CUBE

RC1 - How to make prediction techniques timely?

- **RC1.1 – How can be predict the point in time when the predicated failure will impact?**
  This will allow us determining the time-span that we have for adaptation and thus will influence the decision on the type of adaptation strategy to select
- **RC1.2 – How to relate short-time prediction ("online" / adaptation) with long-term prediction (evolution)**
  I.e., how does online prediction relate to the more traditional prediction of system characteristics?

Refined Challenges for Accurate Prediction

S-CUBE

RC2 - How to make prediction techniques accurate?

- **RC2.1 - What are the right metrics for measuring accuracy?**
  E.g., contingency table metrics (precision / recall) vs. error
- **RC2.2 - How to incorporate the notion of Quality of Experience (QoE) considering diverse stakeholders?**
  E.g., accuracy may be different for personal / professional use
- **RC2.4 – What are the relevant "external" factors that are needed to contextualize the predictions?**
  E.g., if fast/easy adaptation mechanisms, less accuracy needed
- **RC2.3 - How to incorporate cost models?**
  E.g., even if wrong prediction, overall costs may be negligible
- **RC2.4 - When to assess accuracy?**
  E.g., post mortem vs. online

3. Proactive Adaptation / Prediction
* Challenges from S-Cube Workshop *

a. Concepts, techniques and tools for timely quality prediction, including predicting the point in time when a failure will impact/occur (cf. RC1) -- **Yagil**

b. Differentiating and correlating short-time predictions ("online" / adaptation) with long-term predictions (evolution) -- **Annapaola**

c. Metrics, techniques and tools for measuring the accuracy of quality predictions (cf. RC2) -- **Andreas**

d. Concepts and factors relevant for contextualizing accuracy, including cost models, QoE, as well as post-mortem vs. online – **Andreas**

e. Fully-automated processes for assurances of service-oriented systems in order to architect resilience against "unknown" situations and for dealing with rare events – **Rogerio**

f. Concepts, models and algorithms for prediction of quality of heterogeneous (real-world and IT) service-oriented systems, integrating proactive Complex-Event-Processing with Quality Prediction for SOA – **Andreas**

g. Strategies and techniques for handling big data (huge number of events / high frequency of events) during prediction; e.g., sampling from millions of data points arriving per second – **Yagil**

h. Quality assurance techniques to prevent "run-time" design decisions / adaptations to lead to inconsistent situations – **Andreas**

i. Techniques for IaaS performance prediction beyond traditional workload prediction **-- Chi-Hung**

© S-Cube