



A Probabilistic Approach for Long-Term B2B Service Compositions

Adrian Klein

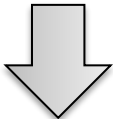
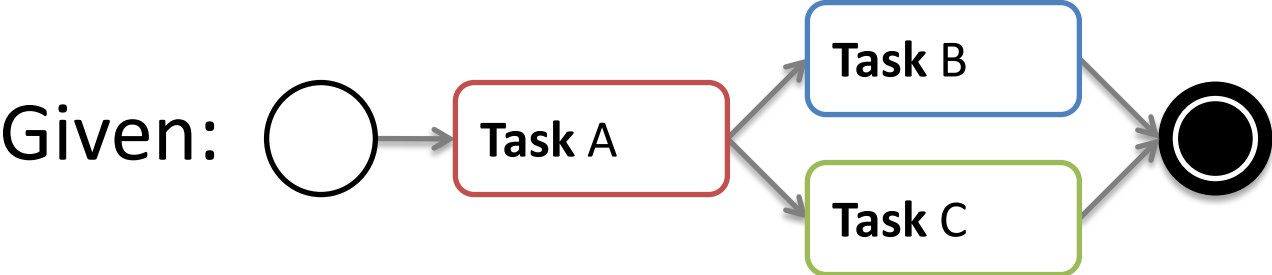
The University of Tokyo

PhD Student (3rd year)

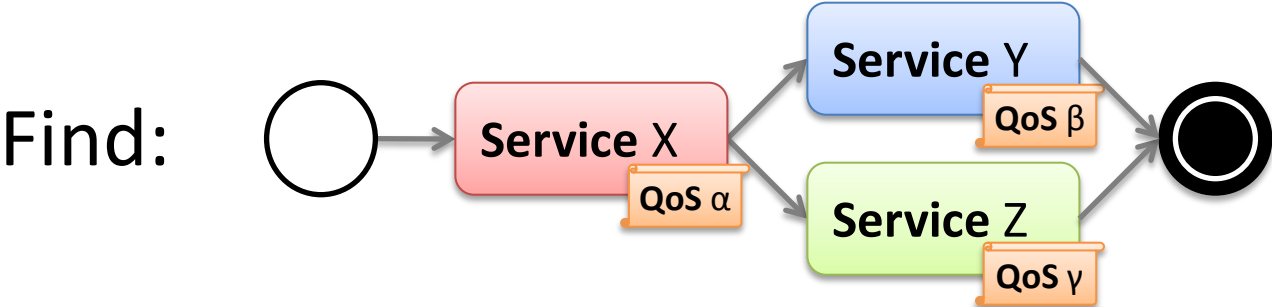
Outline

- 1. Standard Service Composition**
2. B2B Service Composition
3. Probabilistic Approach
4. Evaluation

Standard Service Composition



MINIMIZE $\{0.8 t + 0.2 p\}$
ENSURE $\{t \leq 50\text{ms} \text{ AND } p \leq 5\$\}$



Standard Service Usage

For each Task

- Select **1** Service
 - with **1** Service Level Agreement, e.g. [100ms, 5\$]
- (**Always**) choose the same selected service at execution time

Outline

1. Standard Service Composition
- 2. B2B Service Composition**
3. Probabilistic Approach
4. Evaluation

B2B Service Compositions (1)

- Are

1. **Complex** (e.g. long and nested workflows)
2. Executed **many** times

- Require

1. Effort to **build** workflows
2. Effort to **integrate**
3. Effort to **execute**

=> **Meant** for the **long-term** (many executions)

B2B Service Compositions (2)

- Require a high **reliability**

| Application type | 9s | % |
|-------------------|----|----------|
| Non-critical | 2 | 99% |
| Task critical | 3 | 99.9% |
| Business critical | 4 | 99.99% |
| Mission critical | 5 | 99.999% |
| Safety critical | 6 | 99.9999% |

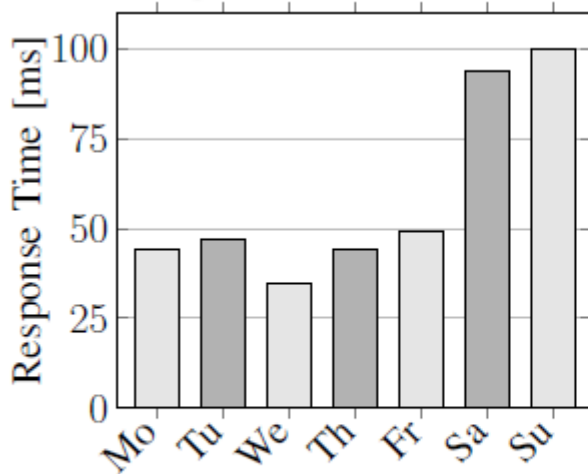
=> **Hard** to achieve for complex workflows

Outline

1. Standard Service Composition
2. B2B Service Composition
- 3. Probabilistic Approach**
4. Evaluation

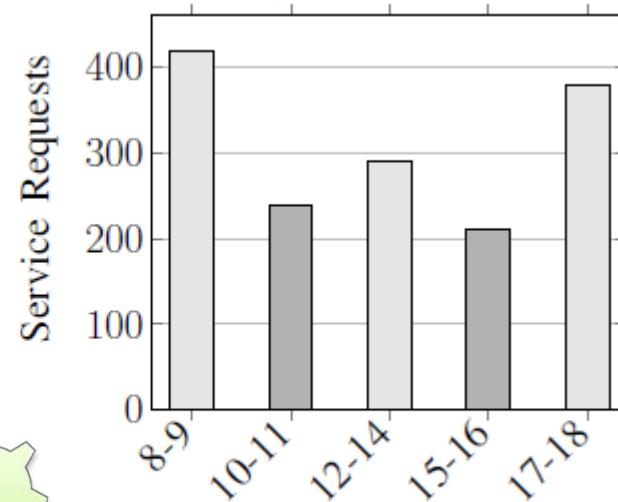
Probabilistic Patterns

How do the QoS of a service change over time?

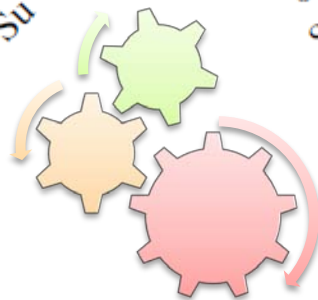


(a) Qos Pattern

When is a service used by a specific user?



(b) Usage Pattern



=> **Compute** expected long-term QoS

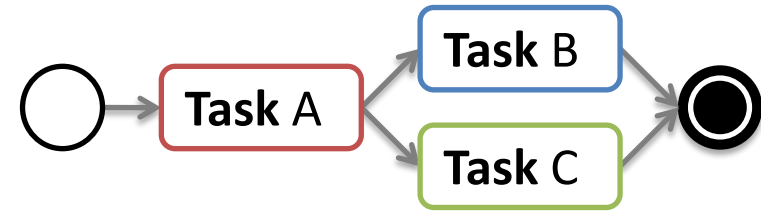
Service Groups

- Select **multiple** services
 - which will be tried to execute **sequentially**
 - until one **succeeds** or all have failed.
- Example

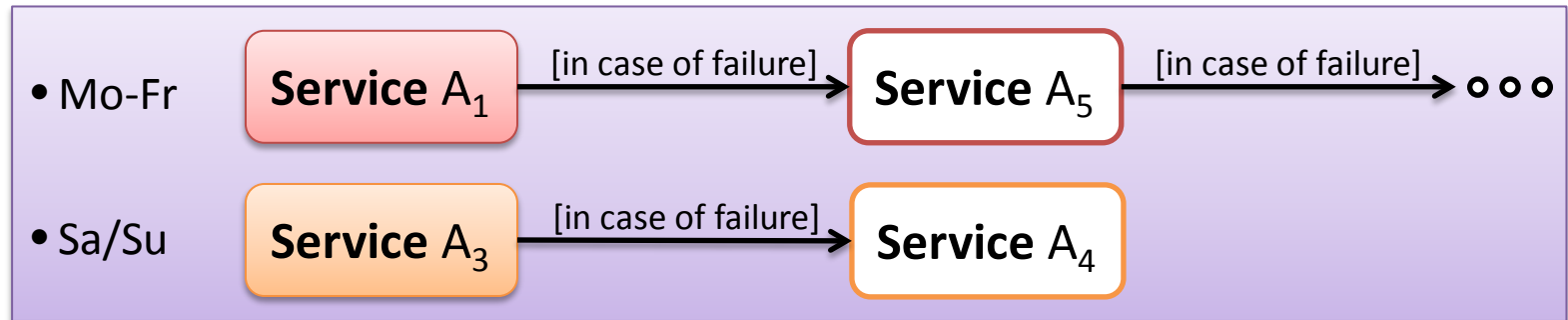


=> Achieve **high reliability**

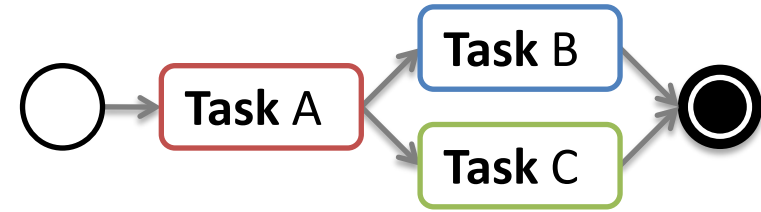
Time-dependent Execution Policy



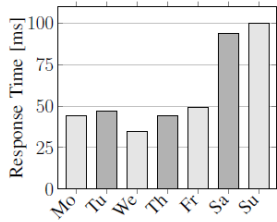
- For each task
 - Set of discrete points in time $\{t_1, t_2, \dots, t_n\}$
 - For each point in time
 - Service group $\{s_1, s_1, \dots, s_n\}$
- E.g. two **service groups** for **Task A**



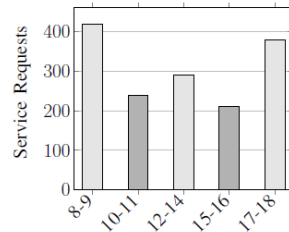
Approach



(Providers') QoS Pattern

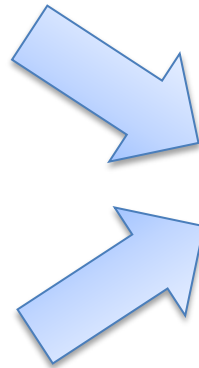


(a) Qos Pattern



(b) Usage Pattern

(Users') QoS Pattern

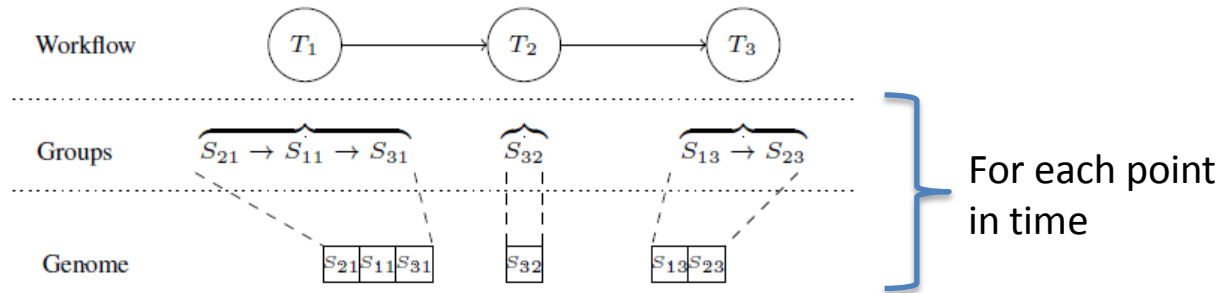


Adaptive GA

Time-dependent Execution Policies

Adaptive GA

- Encoding



- Customized Operators

1. Mutate

(avoid *duplicate* values inside group)

2. Uniform Crossover

(account for *different* parents' group sizes)

3. New **Adapt Operator**

Adapt Operator

Applied after mutate and crossover operators

- 1. Rank** service groups according to Δ of **desired** and **actual reliability** of their task
- 2. Adapt** a fixed ratio of groups
- 3. Increase or decrease group size** by one service depending on Δ

Outline

1. Standard Service Composition
2. B2B Service Composition
3. Probabilistic Approach
- 4. Evaluation**

Evaluation. Algorithms

GA_i

denotes a naive adoption of using *service groups* with a **static group size** that is denoted by the *index*.

GAUP_i

(analog, considering QoS and **Usage Patterns**)

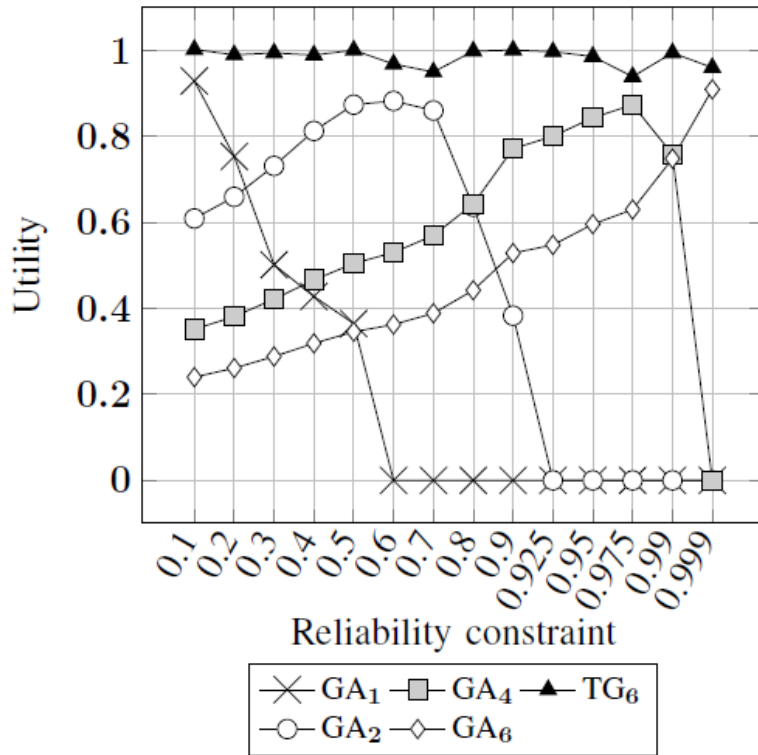
TG₆

denotes our **adaptive Teikou algorithm** which determines the groups sizes *dynamically* up to a maximum size of six services per group.

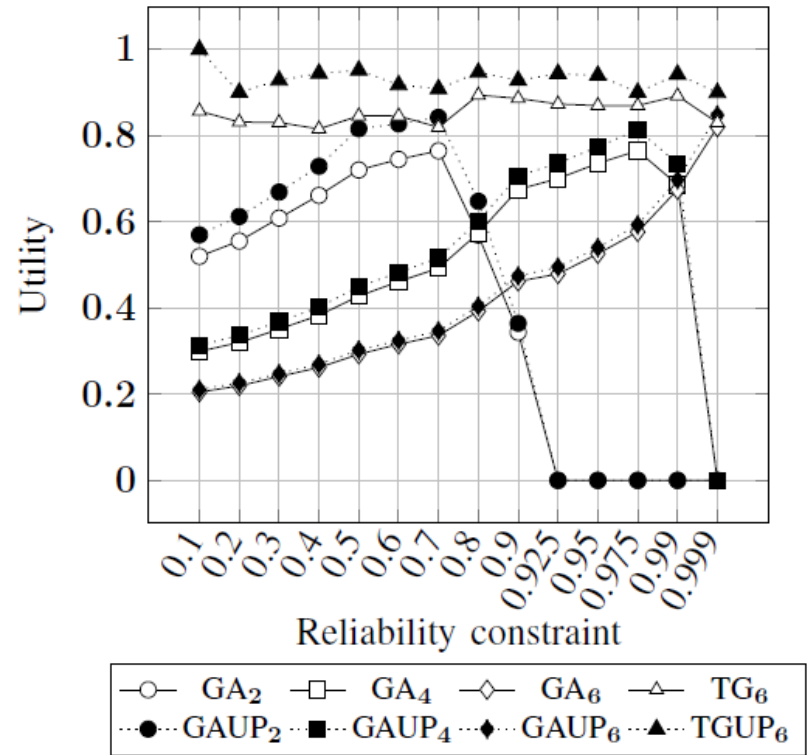
TGUP₆

(analog, considering QoS and **Usage Patterns**)

Utility

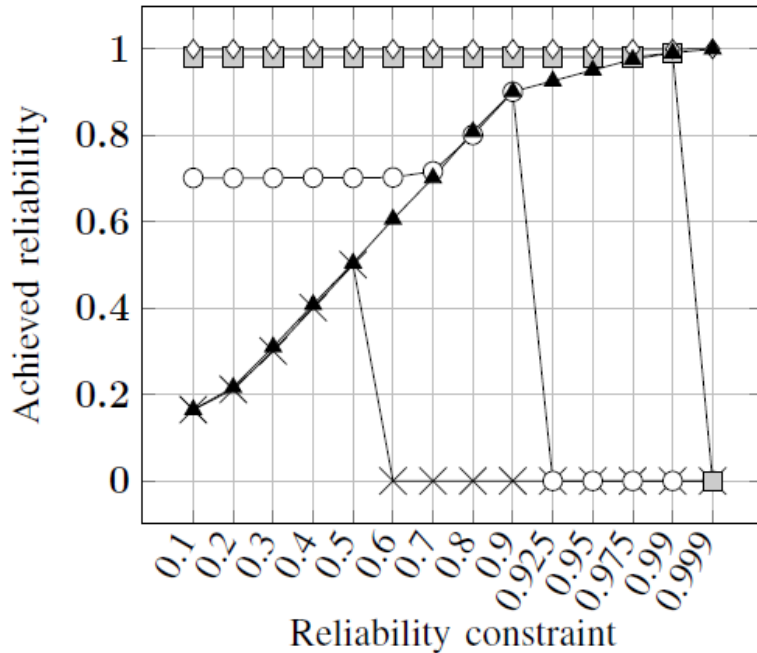


(a) Utility without Usage Patterns

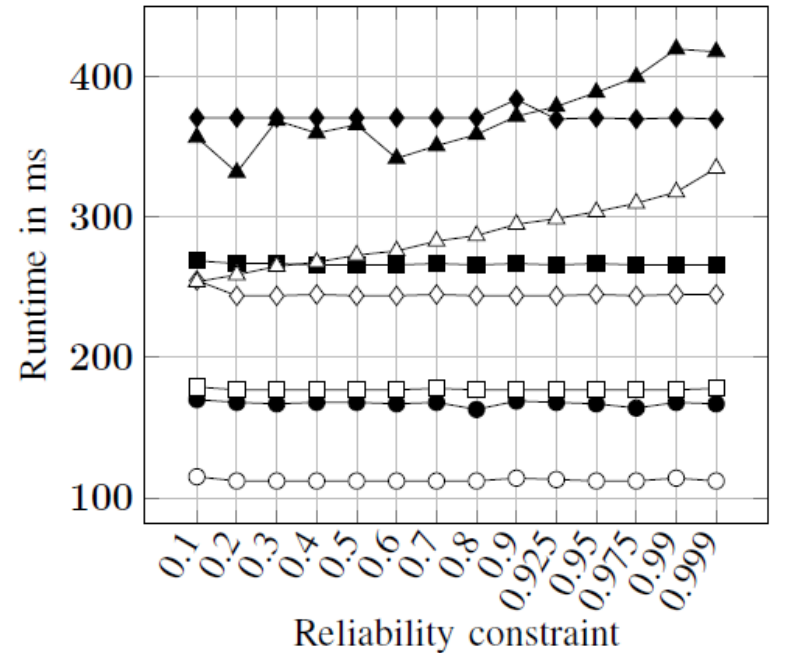


(b) Utility with Usage Patterns

Reliability & Runtime



(a) Reliability



(b) Runtime

Summary

- We addressed the following two concerns of B2B service compositions
 - Long-term QoS
 - High reliability
- Our approach is
 - Adaptive to the reliability constraint
 - Incurs a reasonable overhead



Thank you for your attention!
Looking forward to
your **questions** and **comments**