Exploration of Adaptation Space

- Linking with Efforts in Service-Oriented Computing

Fuyuki Ishikawa

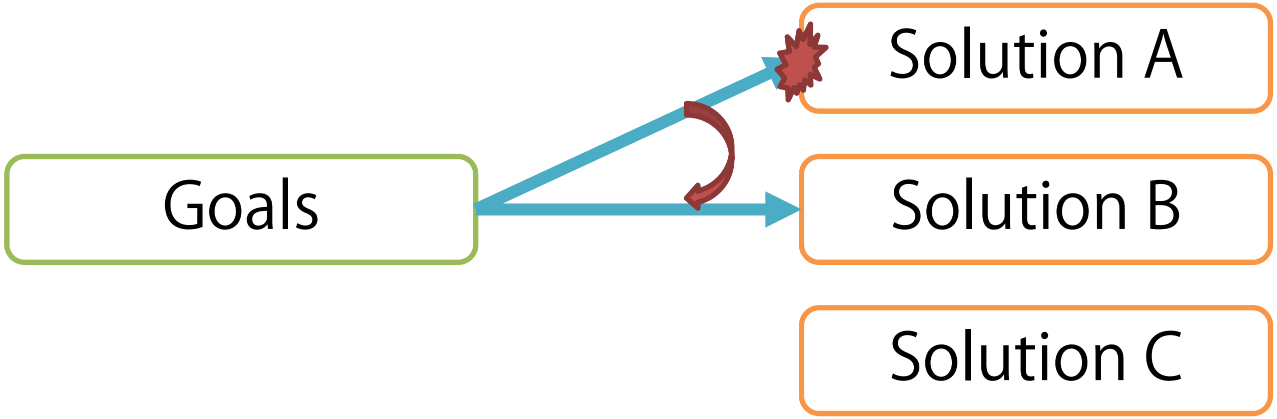
Associate Professor
Digital Content and Media Sciences Research Division
National Institute of Informatics (NII)

NII 大学共同利用機関法人 情報・システム研究機構
国立情報学研究所
National Institute of Informatics

(Generalized) Overview

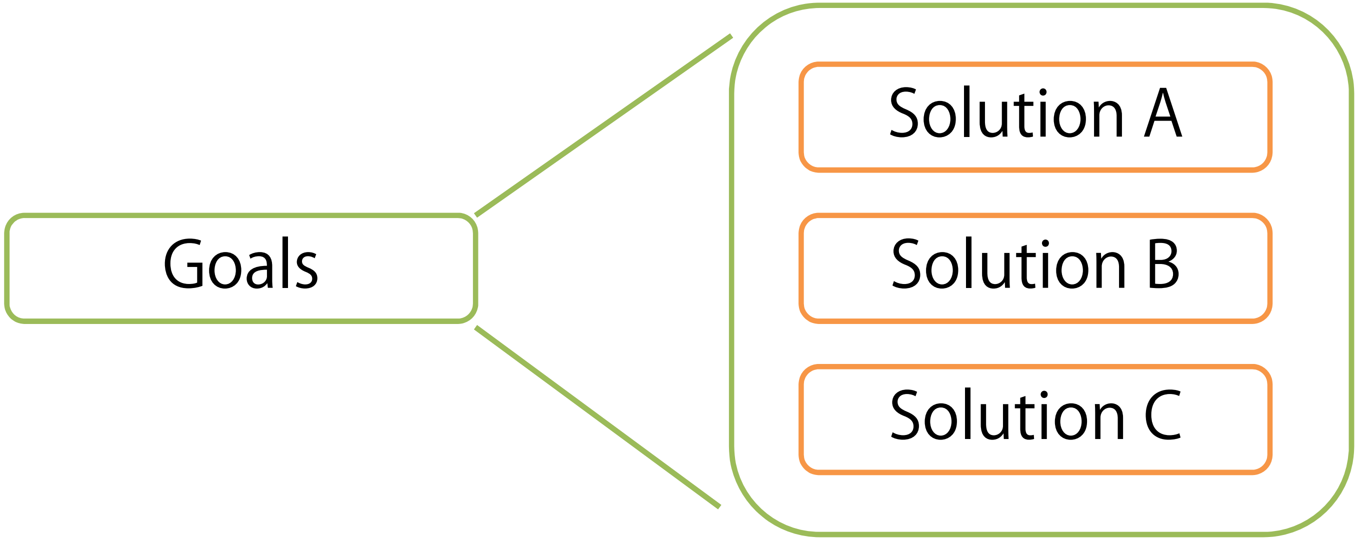
- Adaptation is typically "switching solutions"

  According to changes of the environment, in particular, *changes of availability/effectiveness of the solutions*



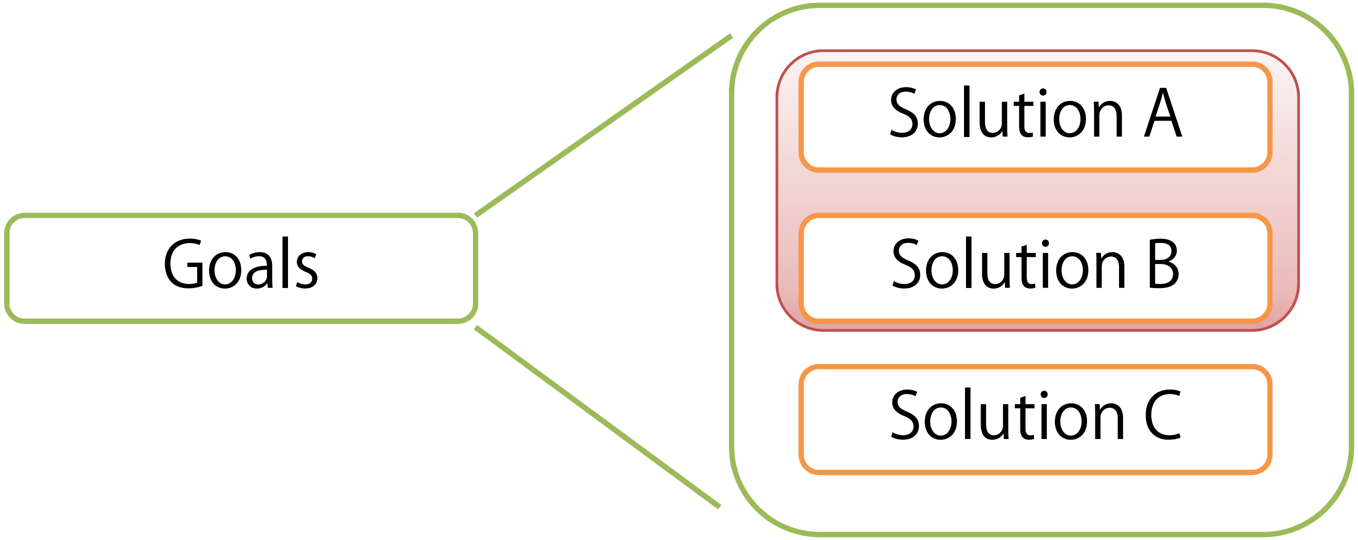
(Generalized) Overview

- *Identification of Solution Space?*
  (at the design level, when solutions are reusable)



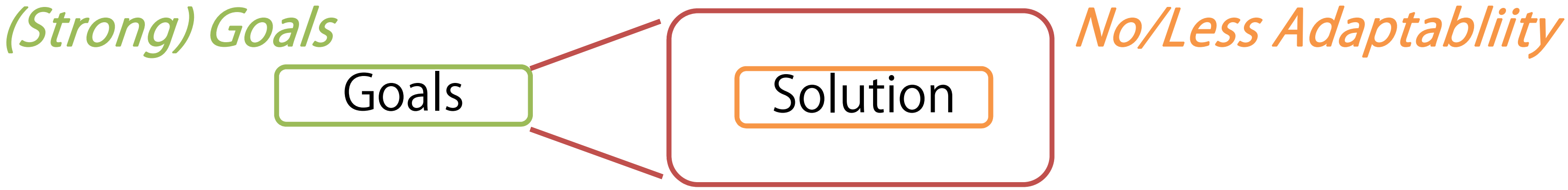
(Generalized) Overview

- *Selection of "Attractive" Adaptation Space?*
  inside the potential solution space
  (at the design level, for efficiency)



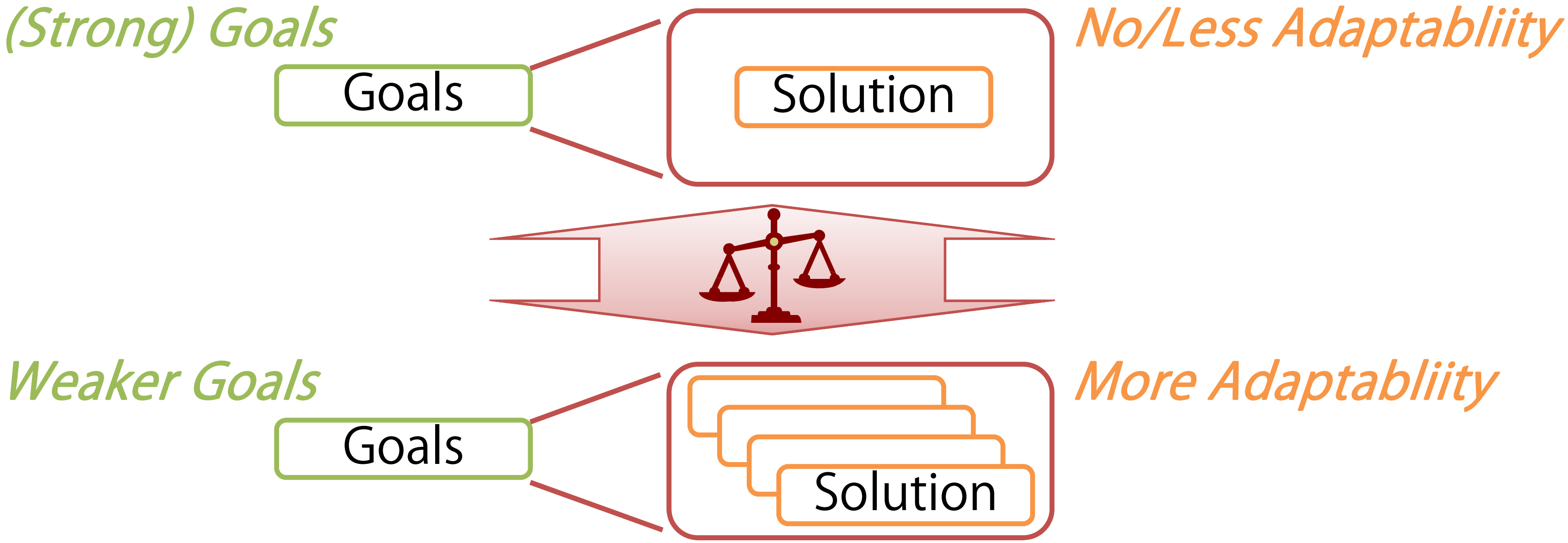
(Generalized) Overview

- *Is adaptability (success rate) always enough?*



(Generalized) Overview

- *Is adaptability (success rate) always enough?*
  Or, should we make a decision with trade-off?



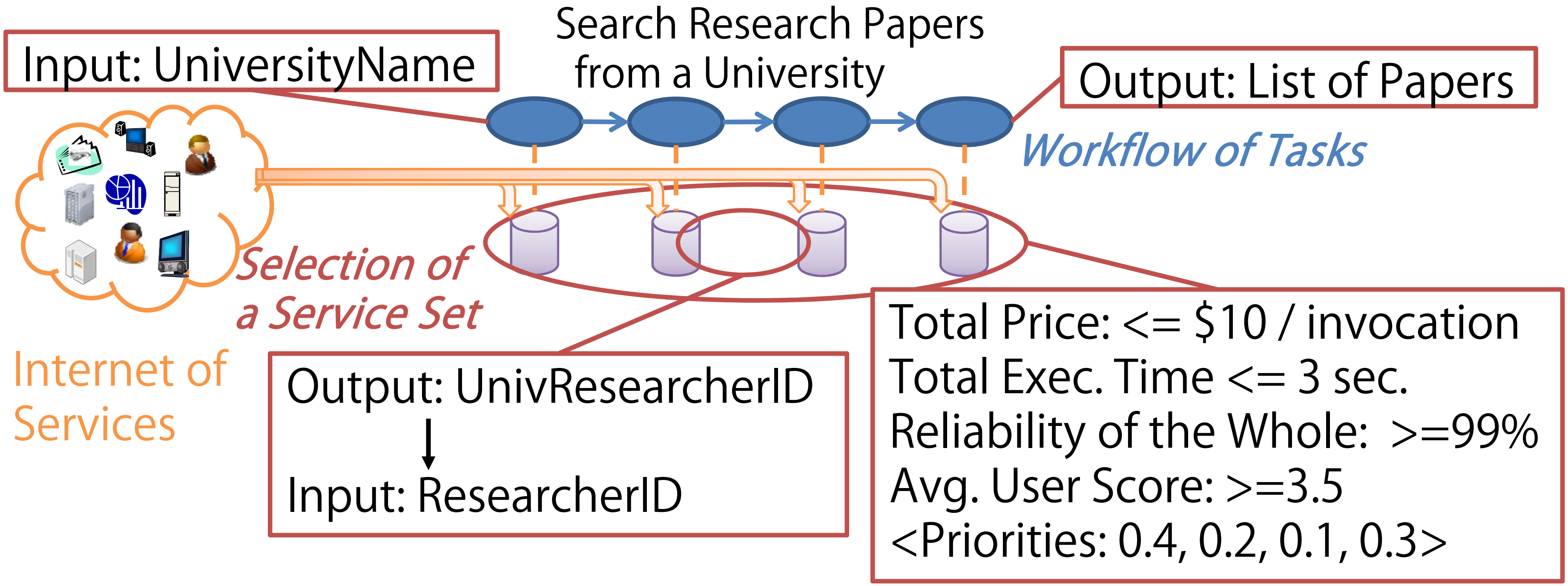
Try to Link…

- With one of our efforts in the area of Service-Oriented Computing
  - One instantiation of self-adaptive systems

With F. Wagner, B. Kloepper, S. Honiden,
Towards ***Robust*** *Service Compositions*
in the Context of Functionally Diverse Services,
WWW 2012

General Service Composition

- Function (input/output/precondition/effect)
  - Goals as the whole, and consistency of the mid-flow
- Quality (various criteria) <u>including success rate</u>
  - Optimize under priorities and <u>global</u> constraints



Input: UniversityName

Search Research Papers from a University

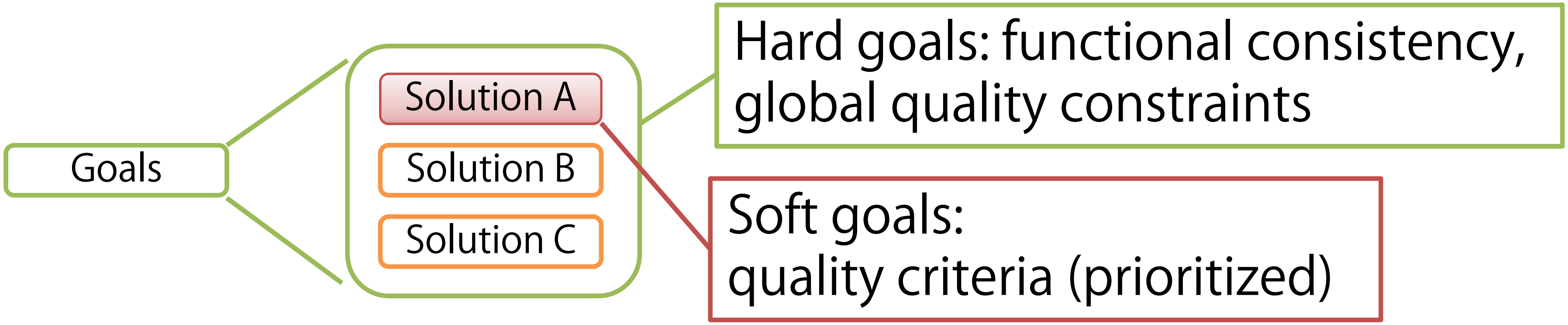
Output: List of Papers

Output: UnivResearcherID
↓
Input: ResearcherID

Total Price: <= $10 / invocation
Total Exec. Time <= 3 sec.
Reliability of the Whole: >=99%
Avg. User Score: >=3.5
<Priorities: 0.4, 0.2, 0.1, 0.3>

General Service Composition

- In summary, the general problem has been:
  - May be solved repeatedly with updated information at runtime (even partially during execution)



Hard goals: functional consistency, global quality constraints

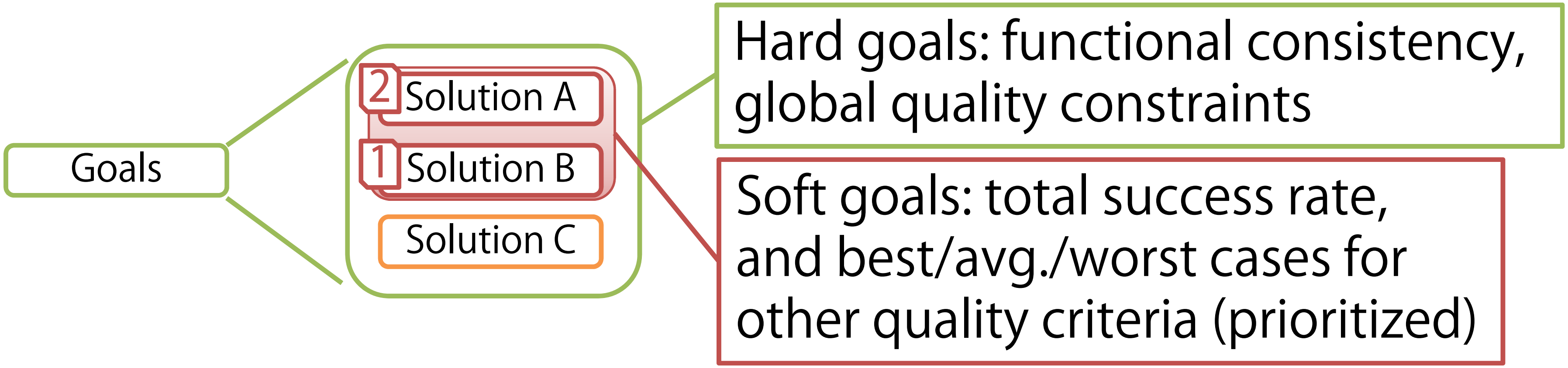
Soft goals:
quality criteria (prioritized)

"Solution" under attention:
service selection for each of the tasks

Our Work on Adaptive Compositions

Analyze alternative services (i.e., potential solution space) at design/deployment-time

➡ *Derive the "best" part to be used at runtime*



Hard goals: functional consistency, global quality constraints

Soft goals: total success rate, and best/avg./worst cases for other quality criteria (prioritized)

- Avoid overhead and miss of optimality in "greedily deriving one best solution, repeatedly" at runtime
- Naturally accompanies analysis of adaptability

[WWW'12]

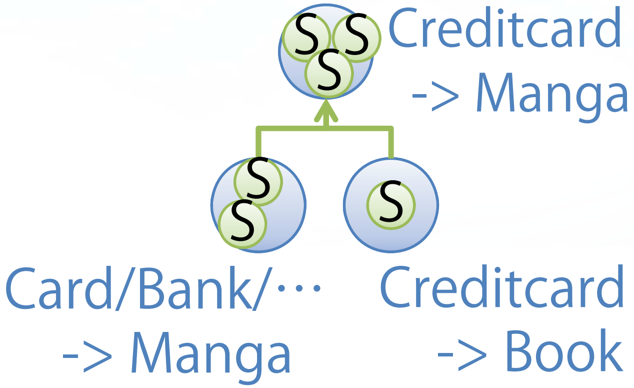
Our Work on Adaptive Compositions

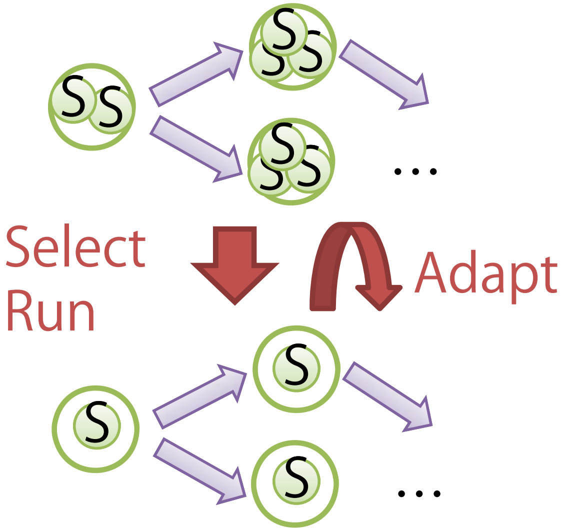
- *Construct "graphs" of service functions*

➔ Descendants are alternatives
  - Less/weaker input/precond
  - More/stronger output/postcond

used to

- Analyze adaptability
- Construct a "loose" plan (an adaptation space)
- (Efficiently check matching between connected services)





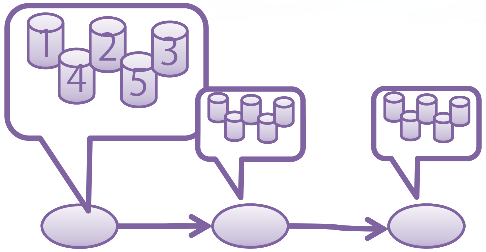
[WWW'12]

Our Work on Adaptive Compositions

- Derive “*an adaptation space to be deployed*“ (for quick adaptations at runtime)
  - In any case, functionally-consistent, and global constraints satisfied
  - “(Near-)Optimal” for given priorities

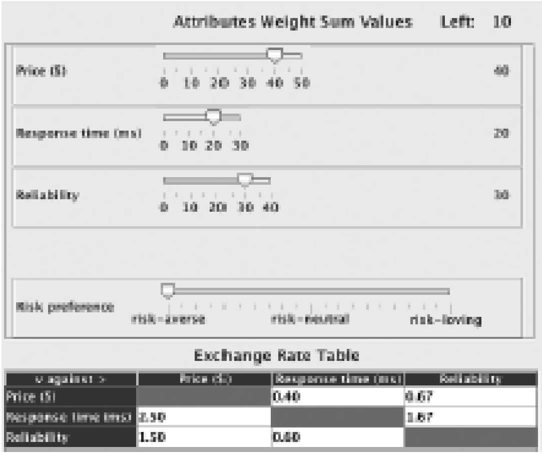
    Total success rate

    Best/avg./worst cases for other quality criteria



By a custom, scalable genetic algorithm for this setting

- Outperforms other methods in quality/scalability (details omitted)



Attributes Weight Sum Values Left: 10

Price ($) 40

Response time (ms) 20

Reliability 30

Risk preference

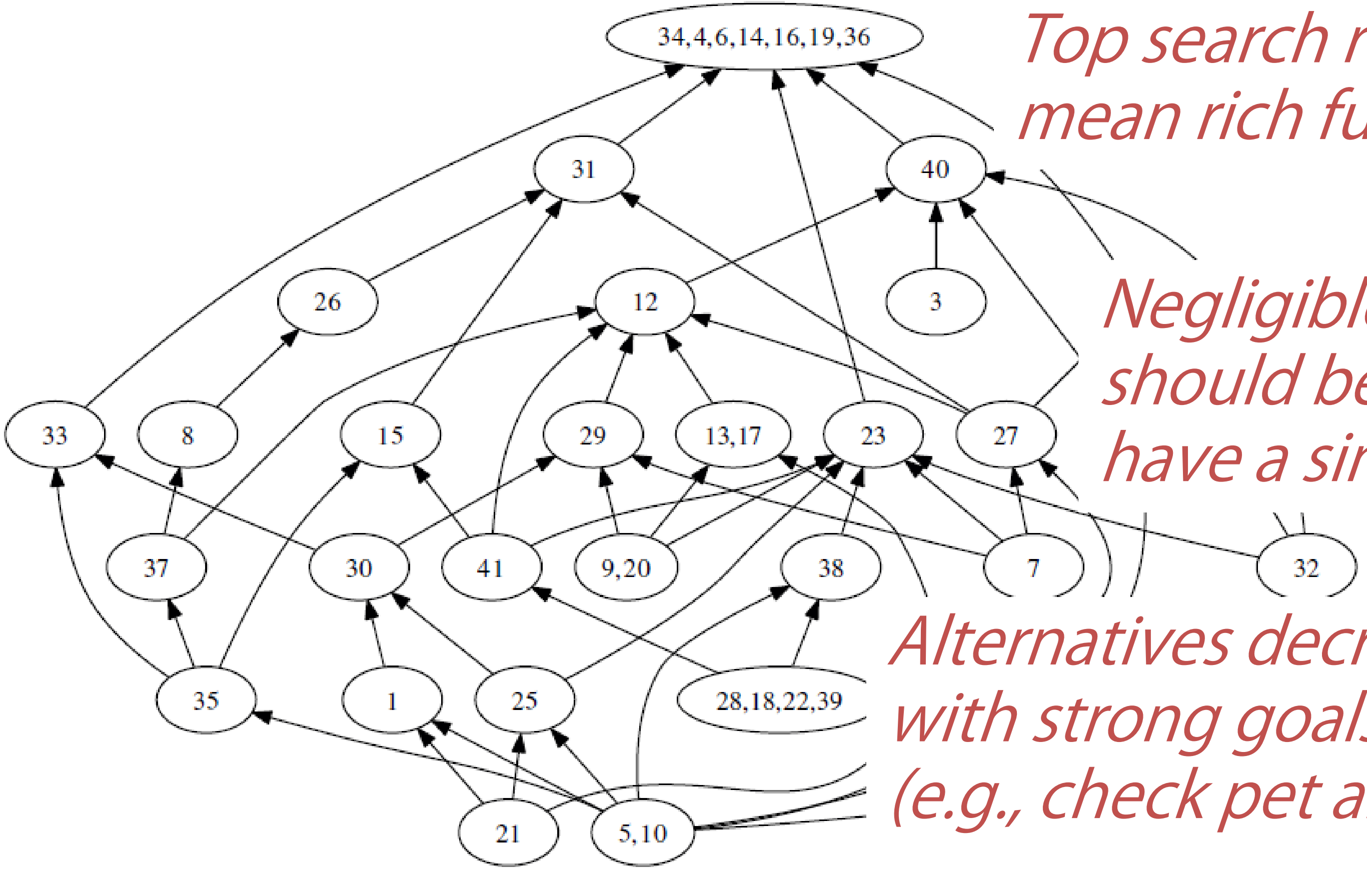
Exchange Rate Table

v against >	Price ($)	Response time (ms)	Reliability
Price ($)		0.40	0.67
Response time (ms)	2.50		1.67
Reliability	1.50	0.60	

[WWW'12]

Appendix: Example

- Hotel search functions (output compatibility) (extracted from top 100 pages of Google search)



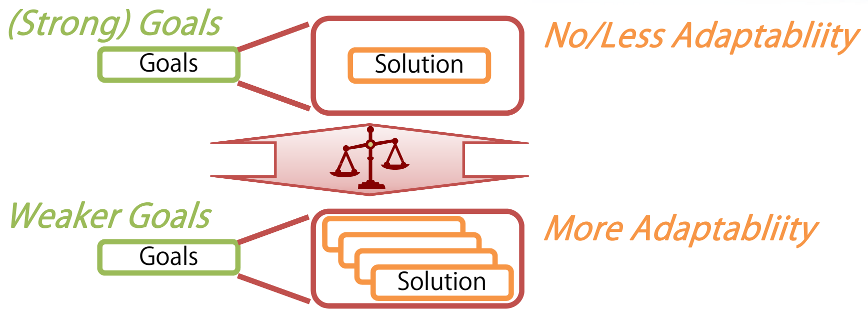
Top search results do not mean rich functions

Negligible differences should be defined to have a simpler graph…

Alternatives decrease with strong goals (e.g., check pet allowance)

Summary and Directions

- *Efforts on adaptive service compositions viewed as exploration of adaptation space*



(Strong) Goals

Goals

Solution

No/Less Adaptabliity

Weaker Goals

Goals

Solution

More Adaptabliity

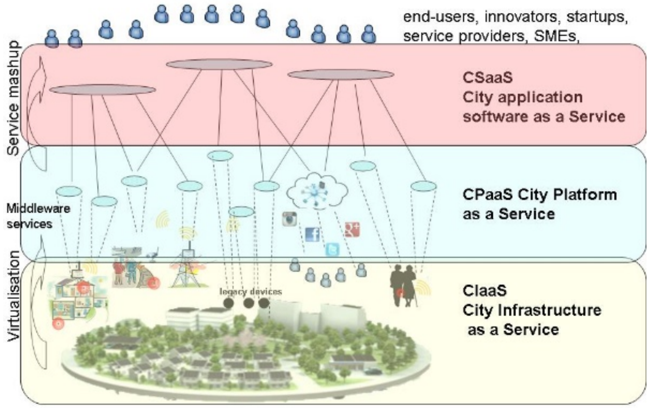
- Ongoing discussions:

  1. @runtime

  2. Use weaker services

  (human intervention?)

- Application case studies: under FP7 EU-Japan Project (IoT/Crowd as-a-Service)



http://clout-project.eu/

Thank you!

f-ishikawa@nii.ac.jp
http://research.nii.ac.jp/~f-ishikawa/en/