A Dynamic Framework for Quality Web Service Discovery

Atousa Pahlevan, Hausi A. Müller
MESOA, September 2010.
Static Service Discovery

Static Service Discovery process involves a service requester interacting with a client interface. The client interface is connected to a matchmaker which selects services based on static criteria. The selected services are represented by Service Interface nodes, each with a static description. The service requester interface is connected to the service provider interfaces, illustrating the service discovery and selection process.
Incorporating Changing Attributes
Motivation
Motivation Cont’d.
Motivation Cont’d.
Motivation Cont’d.
Motivation Cont’d.
Motivation Cont’d.
Research Approach

- Investigating a uniform, extendable, and flexible WS-dynamic attributes representation
- Investigating conceptual model of a domain & context (ontological theory)
- Incorporating service quality attributes
  - machine-readability with computational semantics
  - Develop dynamic-based WS matchmaking algorithms
  - Develop dynamic-based WS selection method/alg.
Service Discovery Proposed Overall Architecture

Static Discovery (SD)

Dynamic Selection (DS)

Web Service Discovery (SDDS)
Service Self-Adaptive Manager

SDDS

DynamicServiceDriven

Publish

ServiceExpiration

proxy

QueryDynamic

StaticDiscovery

getDynamicManager

User

DynamicManager

DynamicSelection

ServicesReturned

Schedule

Standard UDDI

QueryKnowledge

Request

Analyzer

Plan

Ontology Knowledge

Dynamic Repository

getDynamicManager

getDynamicManager
Service Self-Adaptive Manager Cont’d.

DynamicServiceDriven

Mauna Kea

Publish

Autonomic Quality Service Selection

Check visibility

visibility()

Clear view

Mauna Kea

Dynamic Repository

Standard UDDI

Ontology Knowledge

Monitor

Aclator

Soror

Mauna Kea

Mauna Kea
Service Self-Adaptive Manager Cont’d.
Service Self-Adaptive Manager Cont’d.

DynamicServiceDriven

Get(Telescope)

Requestor

Mauna Kea

Autonomic Quality Service Selection

Analyse

Plan

Monitor

Execute

Ontology Knowledge

Actuator

Sensor

Dynamic Repository

Mauna Kea

Standard UDDI

Mauna Kea Paranal
Issues in the SDDS Design

- 3 Components:
  - UDDI Service Discovery
  - User Dynamic Manager
  - Service Dynamic Manager

- Need to work in a synchronous fashion, for databases to maintain integrity
  - The services are tightly coupled

- SDDS design does not enforce synchronization

- System cannot guarantee database consistency
DynamicServiceDriven

Get(Telescope)
Requestor

Delete(Mauna Kea)

Autonomic Quality Service Selection

Analyze
Plan
Monitor
Execute

Ontology Knowledge

Dynamic Repository

Actuator

Sensor

Standard UDDI

Mauna Kea

Delte(Mauna Kea)

Paranal
Solution

• Introduce a synchronization manager
  • Keeps 3 components in lock-step

• Interleaved processing still possible for scalability...
  • $2^{nd}$ and $3^{rd}$ requests can be worked on before $1^{st}$ request is completely dealt with, as long as each step is performed in an atomic fashion.

• The services are loosely-coupled.
SDDS with Synchronization
Contributions

• A sensing technique to acquire the required information from existing service discovery mechanism (UDDI) on demand.

• A means for consumers of a web service to add the constraints to the search operation, reflecting the context of the request.

• A filtering mechanism for using dynamic attributes as a secondary criterion for service selection.

• An enriched method to automatically evaluate the service and consumer’s context in order to identify the selection criteria automatically.
Ongoing Work

- A comprehensive measurement technique to rank the discovered services
- Support more types of context
- A self-adaptive mechanism to evaluate the service and user’s context, and change the service selection mechanism if better options are available.
- RESTful Quality-Aware Web Service Discovery