Service Discovery using Diane Service Descriptions

Ulrich Küster and Birgitta König-Ries
University Jena
Germany

ukuester|koenig@informatik.uni-jena.de
Agenda

- Introduction to DIANE Service Descriptions (DSD)
- Overview of Jena Discovery Solution
- Presentation of Composition Algorithm
What is DIANE and DSD?

- DIANE project: complete efficient automation of discovery and invocation

- Basic Services:
  - atomic, state-less black boxes
  - two-phase choreography
    (n stateless estimation steps, 1 execution step)

- DSD language:
  - lightweight **ontology language** (object oriented)
  - **special constructs** to describe services
  - Keep **efficient matching** in mind
Motivation: Dynamic Service Binding

BPEL based order management process:

- ReceiveOrderFromCustomer
- PrepareMessages
- Invoke: CheckForShipment
- PrepareOrderShipmentMessage
- CallShipmentService
- ProcessResults...
- Flow
- Continue with process...

**semantic request descriptions**

**semantic offer descriptions**

- Binding
- indirect dynamic binding
- (statically bound)
- dynamic discovery and binding
- discovered and bound

**Semantic Middleware**
Service specific elements of DSD (1)

- **Operational elements**
  - to capture world altering **effects**
  - **states**
    (instances of state ontology: Owned, Known, Printed, Shipped, Accessible, ...)

- **Aggregational elements**
  - Moon sells more than one item, Muller transports to a variety of countries
  - Usually describe **sets of effects**
  - Normal semantic: One out of a set of effects is requested / created
Service specific elements of DSD (2)

- **Selecting elements**
  - to support configuration by the requestor (select one out of the offered effects)
  - to inform requestor about produced effect
  - **variables** (input / output)

- **Valuing elements**
  - to express **preferences** of the requestor
  - **fuzzy sets** (the higher the membership, the higher the preference)
  - **strategies** (specify how to i.e. trade-off price versus shipping time, underspecified offers, ...)
  - **unbiased, deterministic, precise matching**
Example: Excerpt from Muller offer description

```
upper muller : Service
  presents
  upperProfile : ServiceProfile
    effect
    conditions
      $pickupEnd > (+,$pickupBegin,<PT90M>)
    Shipped
      fromAddress
      toAddress
      pickup
        $pickupBegin Date Time
        <= <20:00>
        <= nowPlusTwoWorkingDays as XSD_DateTime at xsc
        $pickupEnd Date Time
        >= <07:00>
        >= now as XSD_DateTime at xsc
      shippingTime
      cargo
      shippingTime
      currency
        Double
          amount
            Currency
              == usd
      duration
        beginDateTime
        endDateTime
      physicalEntity
        weight
      weightMeasure
        val
        unit
          Double
            <= 50
          WeightUnit
              == pound
```

Given fuzzy request $r$ and configurable offer $o$ solve the following problem:

a) Compute fuzzy containment value $\text{subset} \in [0, 1]$ of $o$ in $r$
   (How well is the offer contained in the requested effects?)

b) Where possible, configure $o$ such as to maximize subset

Implementation descends through description graphs, fills variables with optimal values, recursively computes subset for each element, combines subset values
Solution Overview

- **Discovery Scenario I**
  - solved all goals
  - rule-based computation delegated to external services (shipping prices, expected shipping times)

- **Discovery II and Simple Composition**
  - solved all but one goal
    - insufficient expressivity for lists (compatibility notebooks – docking stations)
    - solution correct but not complete

⇒ Will focus on composition aspect in this talk
Composition Aspects: Discovery II, Goal C4

```
swsDiscoveryIIIC4 : Service
  upper
  presents
  upperProfile : ServiceProfile
    effects
      Owned
        entity
          Notebook
            (processor mul exp(processor 3)) mul (display mul exp(memory 2))
            hardDisc
            memory
            processor
            display
            Company
              apple
            resY
            resX
            Currency
              usd
            Currency
              double
            Integer
              value
              unit
            Integer
              value
              unit
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
            Integer
              value
              Double
              usd
```
A Three-Step Matchmaking Algorithm

**Step A:**
- **Plug-In Matcher**
  - Effect-to-effect, ignores variables
  - Rummage notebooks and webcams
  - Hawker notebooks and webcams
  - Bargainer notebooks

**Step B:**
- **Service Composer**
  - 1. Compute effect coverages, 2. Restrict variables
  - Coverage 1
    - Rummage notebooks and webcams
    - Hawker notebooks and webcams
  - Coverage 2
    - Bargainer notebooks

**Step C:**
- **Final Matcher**
  - Effect-to-effect, fills variables, local optimization
  - Coverage 1
    - MV 072
    - Rummage notebooks and webcams
  - MV 09
    - Hawker notebooks and webcams
  - MV 08
    - Rummage notebooks and webcams
  - Coverage 2
    - MV 00
    - Hawker notebooks and webcams

**Final Result:**
- In ready-to-deploy options, solution correct but not complete.
A Matching Algorithm for Multiple Effects (1)

- Matching on an effect-to-effect base
- Ignores variables (does not configure offer yet)
- Greatly reduces number of remaining offers (very precise and selective matching)
A Matching Algorithm for Multiple Effects (2)

- Compute all coverages (polynomial complexity)
  - based on combining service, not service configurations
  - 3 services for notebook, 1 for sleeve, 2 for webcam
  - 6 coverages
A Matching Algorithm for Multiple Effects (3)

- Matching on an effect-to-effect base
- Fills variables / configures offer
- Local optimization (correct, but not complete)
Summary

- Able to model all but one goal
- Three-phase matching algorithm to perform composition correctly but not complete

Main directions of future work:
- Implement list support in matchmaking (model missing goal)
- Optimal configuration of composition with multi attribute conditions
Thank you for your attendance!

Questions?

Ulrich Küster
DIANE project (services in ad hoc networks)
http://hnsp.inf-bb.uni-jena.de/DIANE/