



DIANE

An Integrated Approach to Automated Service Discovery, Matchmaking and Composition

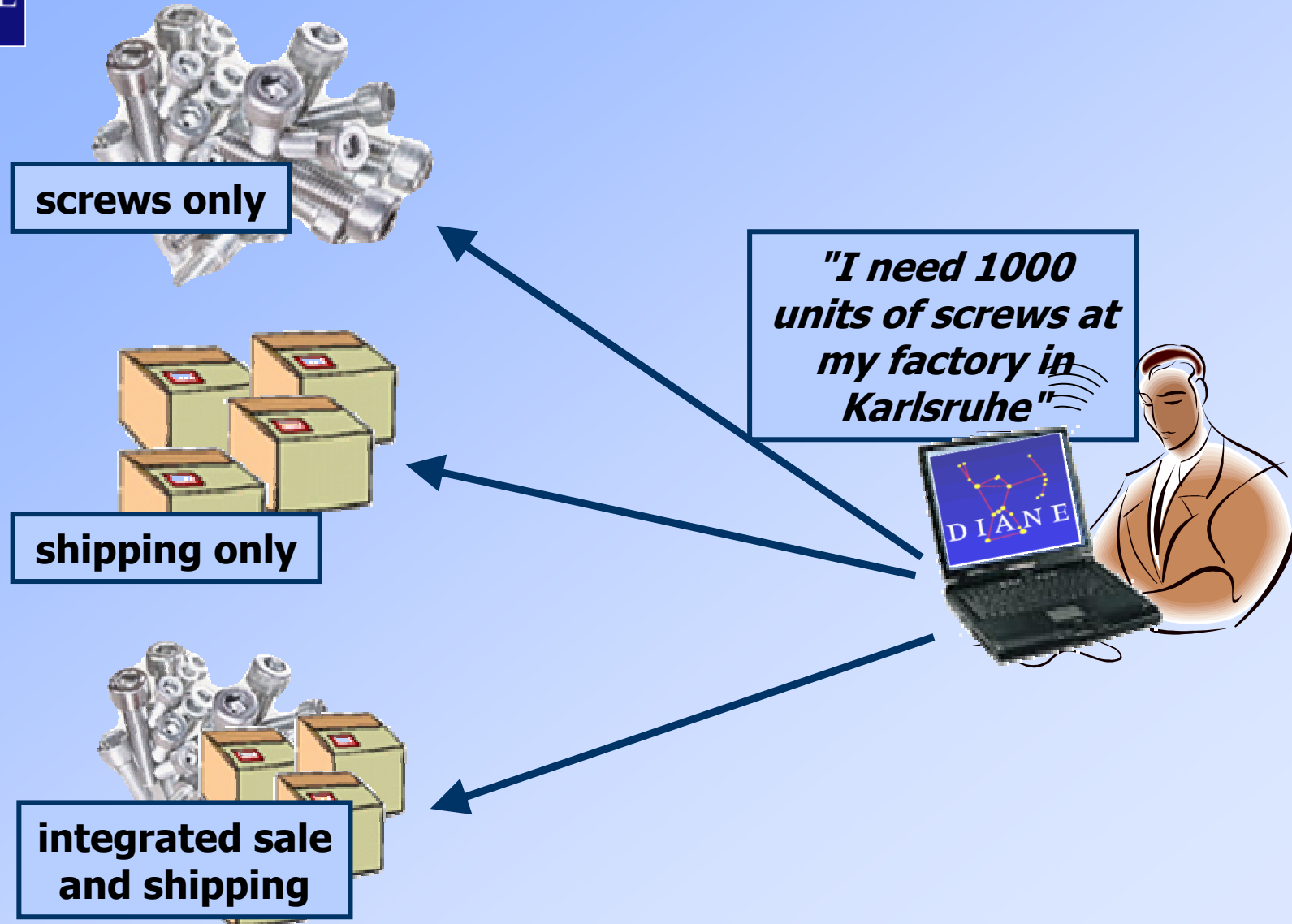
Ulrich Küster¹, Birgitta König-Ries¹,
Mirco Stern² and Michael Klein²

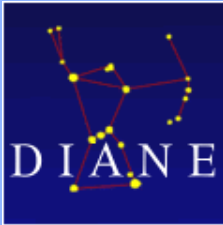
¹University Jena, ²University Karlsruhe, Germany

ukuester|koenig@informatik.uni-jena.de
mirco.stern|kleinm@ipd.uni-karlsruhe.de



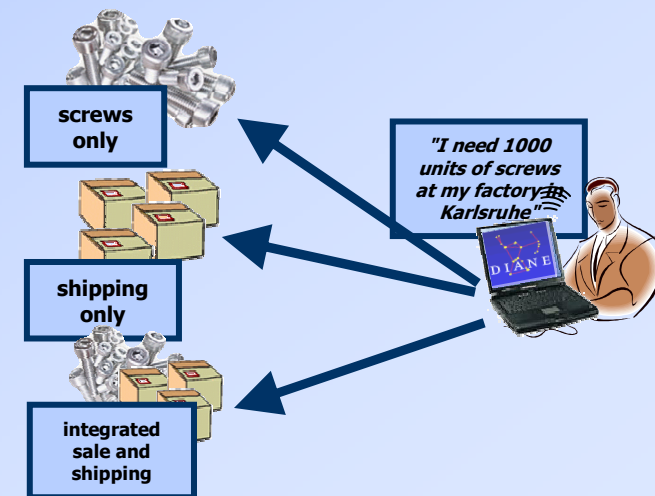
Motivating Example





Motivating Example (cont.)

- For humans trivial, for a matchmaker not!
- Main problems:
 - efficiently determine good configuration
 - ensure constraints on composition (and configuration)
 - shipping departs where screws are purchased
 - specific screw packages must adhere to shippers restriction on weight and size
 - ...
- Need to express such constraints
- Need a matcher that automatically composes services and handles those constraints efficiently





Agenda

- 1. Introduction and Motivation**
- 2. *Background information on DSD
(service descriptions and matchmaking)***
- 3. Adaptation to matching multiple effects**
- 4. Evaluation and Summary**



DIANE Service Descriptions (DSD)

- Goal: Complete and efficient automation of service discovery, matchmaking and invocation
- Layered lightweight ontology language
 - concepts from frame languages, DLs, fuzzy logic and modal logic
 - Specific elements for service descriptions
- more intuitive modelling
- fine-tuned expressivity
- efficient yet precise matchmaking



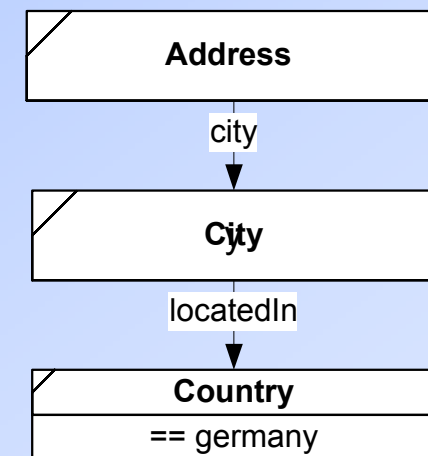
Characteristics of DSD (1)

- **Set-based declarative descriptions**

- Vendors sell thousands of articles
- Shipper provide transportation to a multitude of locations
- ➔ Offer described as set of possible effects

- Requests accept differing services with differing preference
- ➔ Request described as fuzzy set of acceptable effects

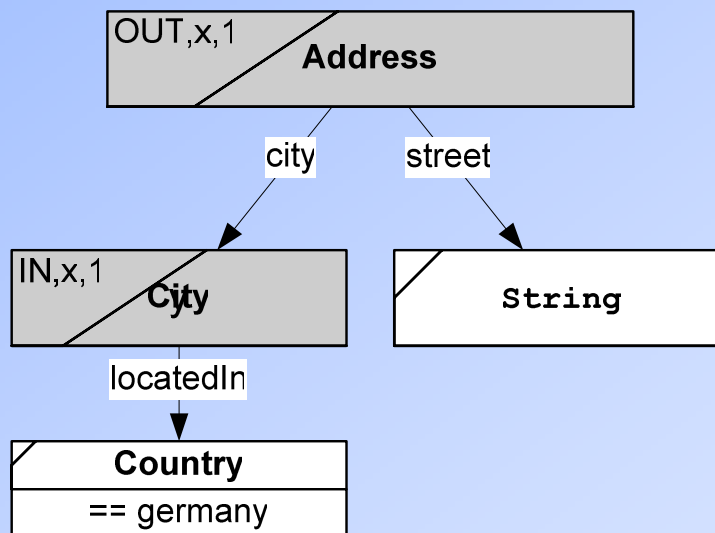
➔ Semantics: One out of the described set of effects is requested / will be created





Characteristics of DSD (2)

- **Configuration by Variables**
 - Link inputs and outputs of a service with concepts in the description
 - Offer-In-Variables (configure offers / provide input)
 - Offer-Out-Variables (provide output)

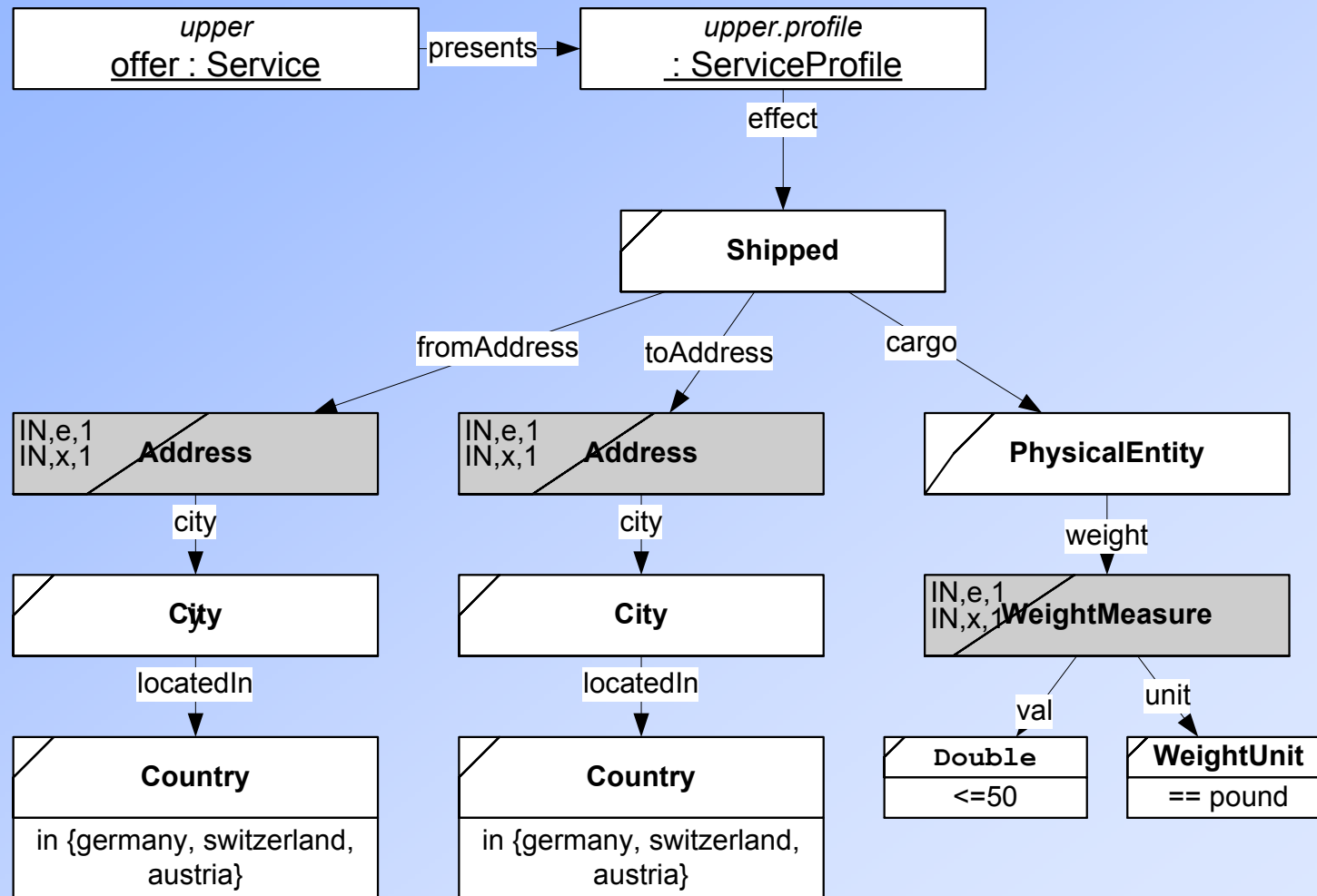


Input: German city

Output: Complete address in that city



An exemplary offer description (excerpts)





Matching DSD-Descriptions

- Match fuzzy request set r with configurable offer set o
 - Compute fuzzy containment value $\mathbf{subset} \in [0, 1]$ of o in r
(How well is the offer contained in the requested effects?)
 - **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 according to strategies

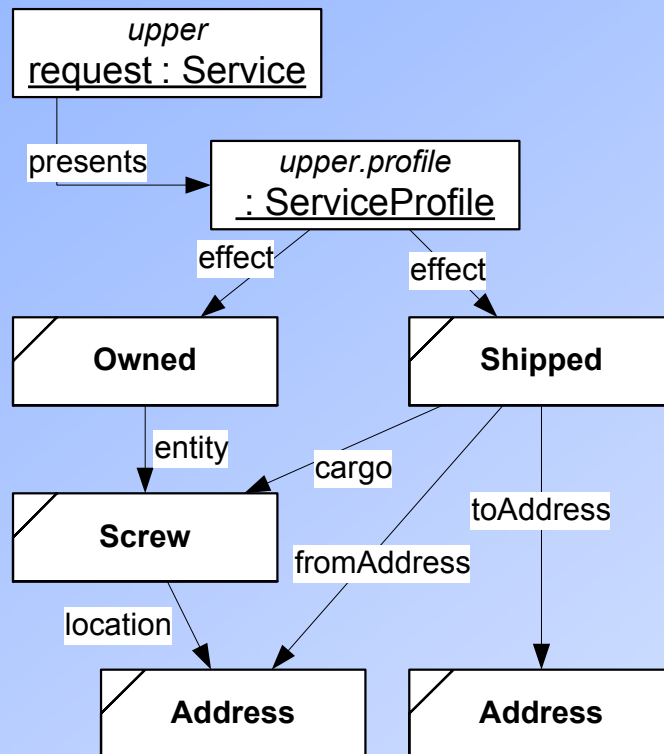


Agenda

- 1. Introduction and Motivation**
- 2. Background information on DSD
(service descriptions and matchmaking)**
- 3. *Adaptation to matching multiple effects***
- 4. Evaluation and Summary**



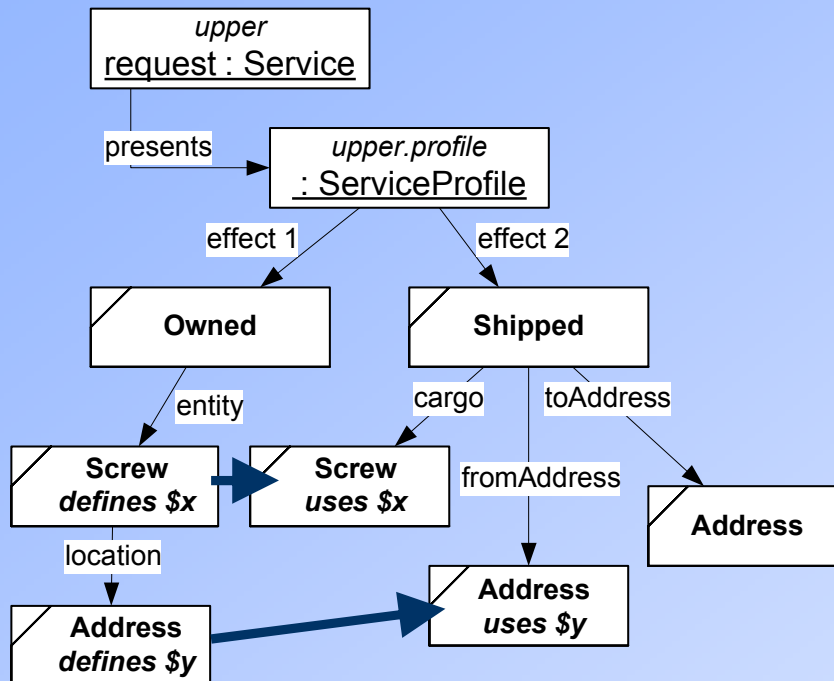
Expressing constraints – intuitive approach



- Intuitive Approach:
Point to the same concept
- Breaks tree structure of descriptions
- Poses problems when configuring an offer
- Need to consider all possible combinations of configurations
→ not efficient



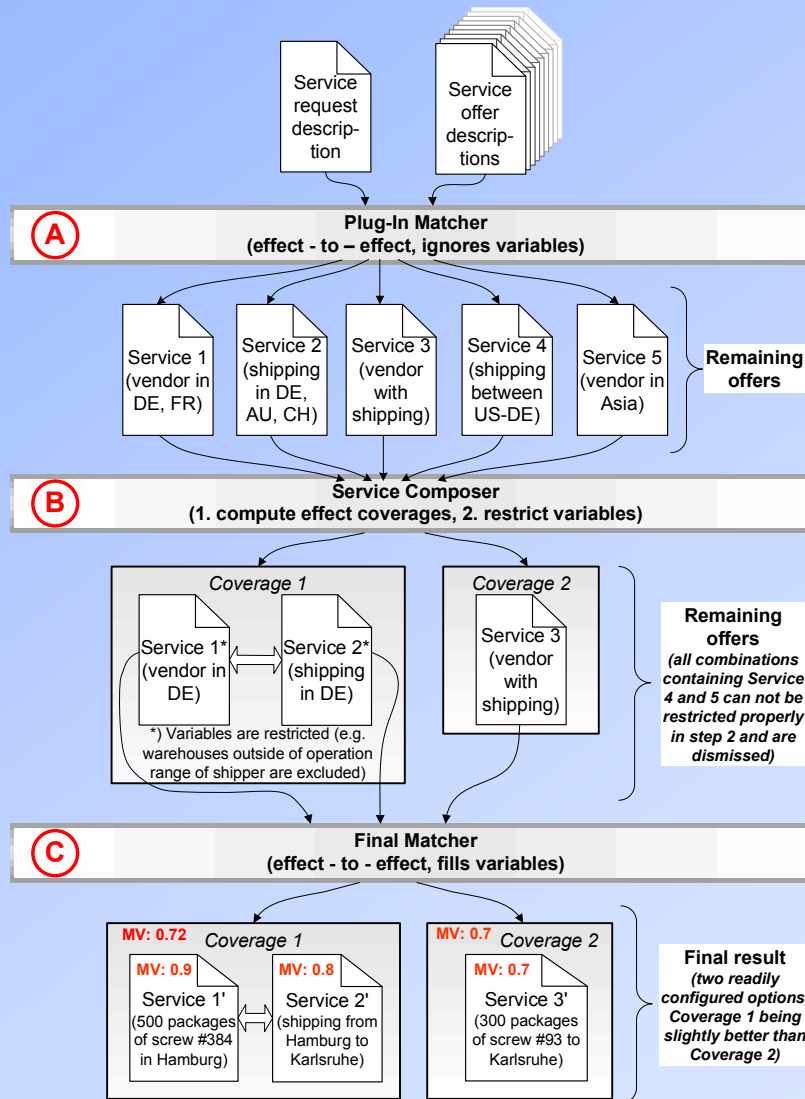
Expressing constraints – Value Propagation



- Effects are ordered
- Configuration will be performed at the first encounter (locally and no more globally optimized)
- Chosen value will be propagated
- Preprocessing prevents fatal configurations (can be done efficiently)

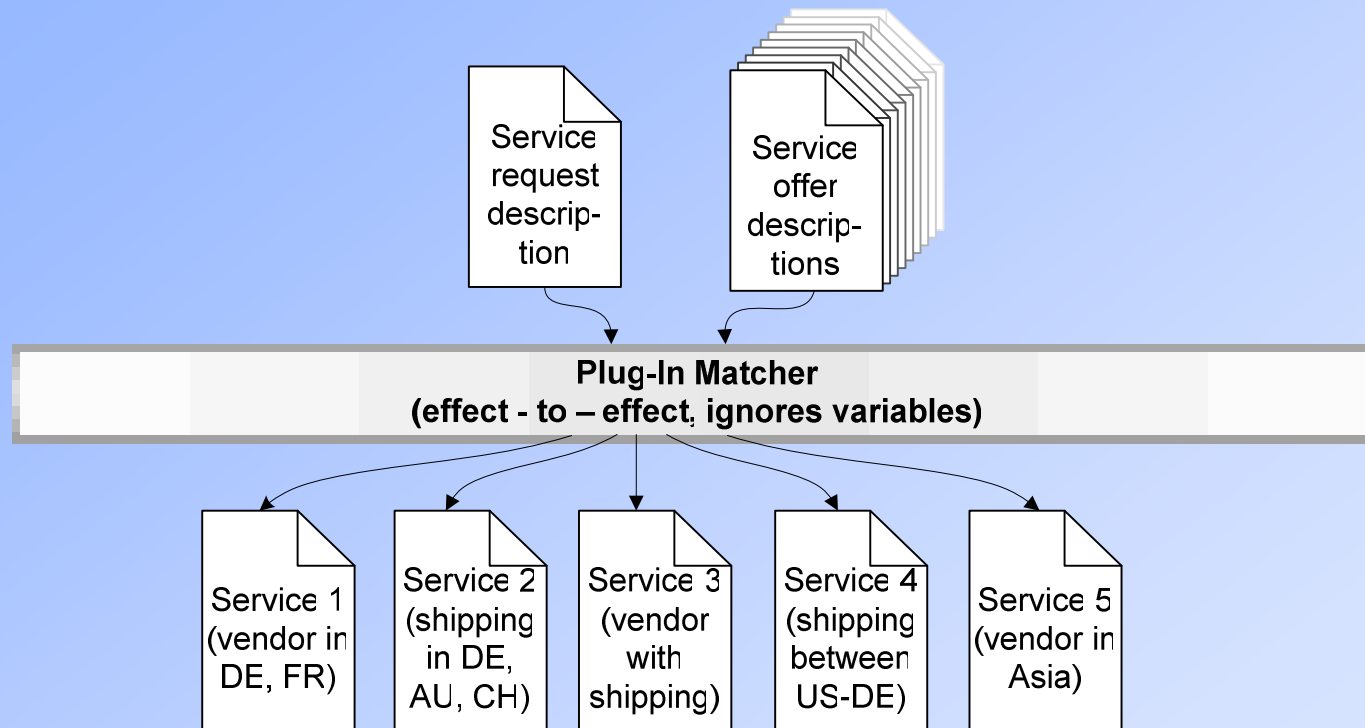


A Three-Step Matchmaking Algorithm



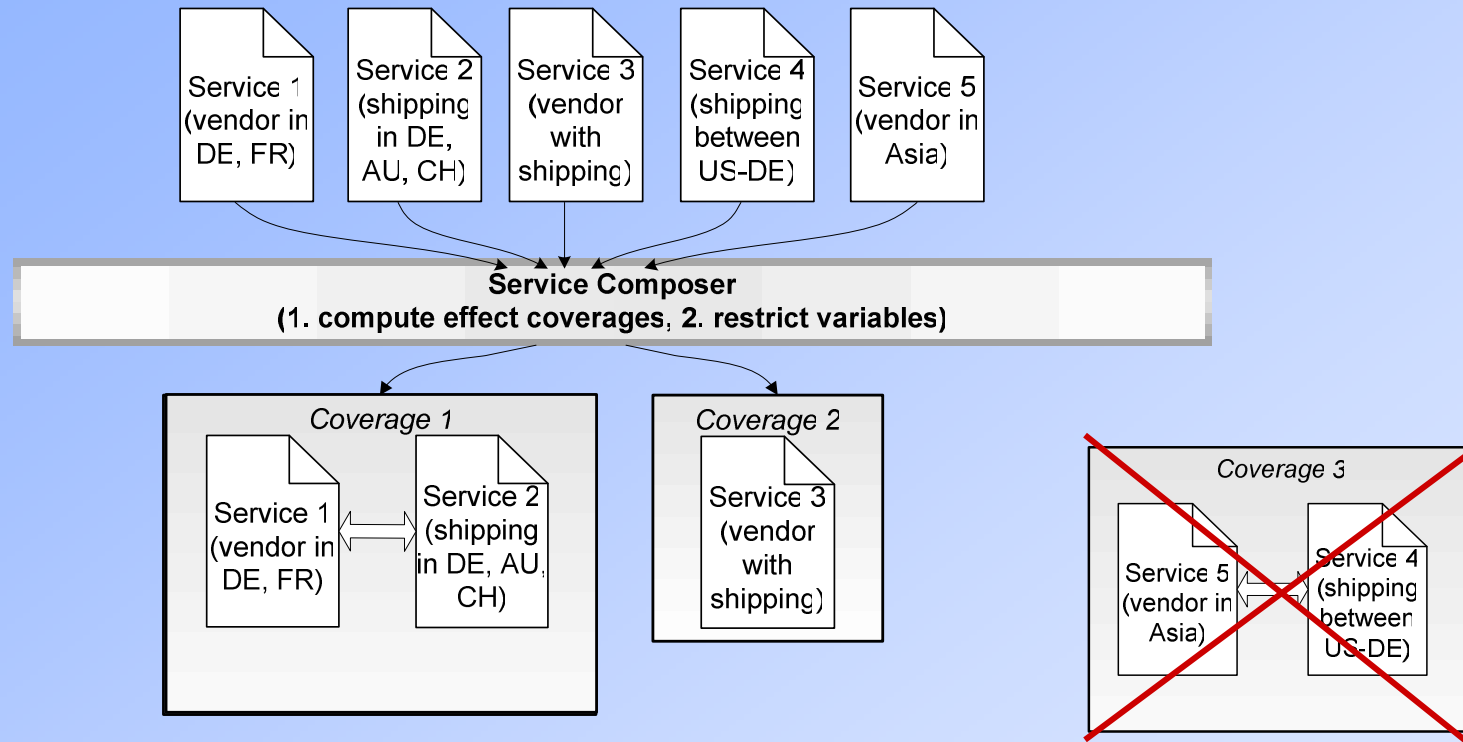


A Matching Algorithm for Multiple Effects (1)



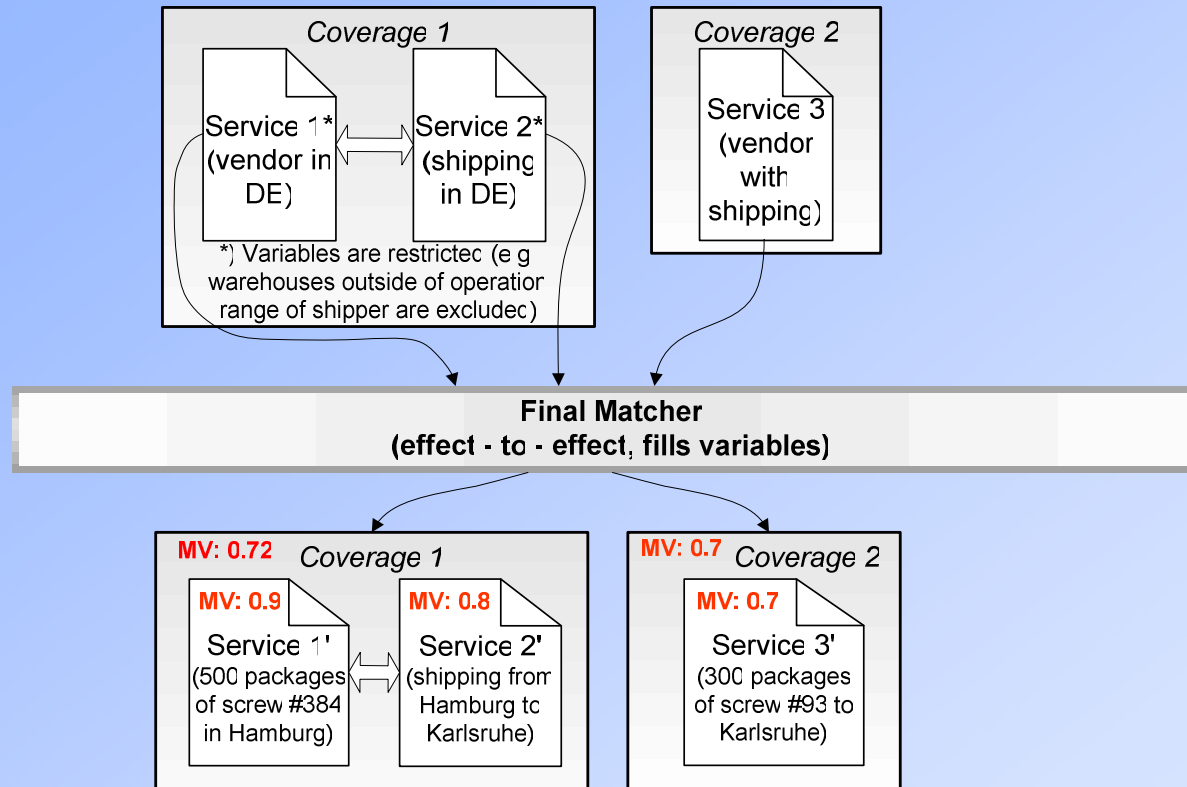
- Matching on an effect-to-effect base
- Ignores variables (does not configure offer yet)
- Greatly reduces number of remaining offers (very precise and selective matching)

A Matching Algorithm for Multiple Effects (2)



- Compute all coverages (polynomial complexity)
- Restrict variables to the cut of linked concepts (linear complexity)
- Filter coverages with empty cut (incompatible services)

A Matching Algorithm for Multiple Effects (3)



- Matching on an effect-to-effect base
- Fills variables / configures offer (includes value propagation)
- Computes overall match value



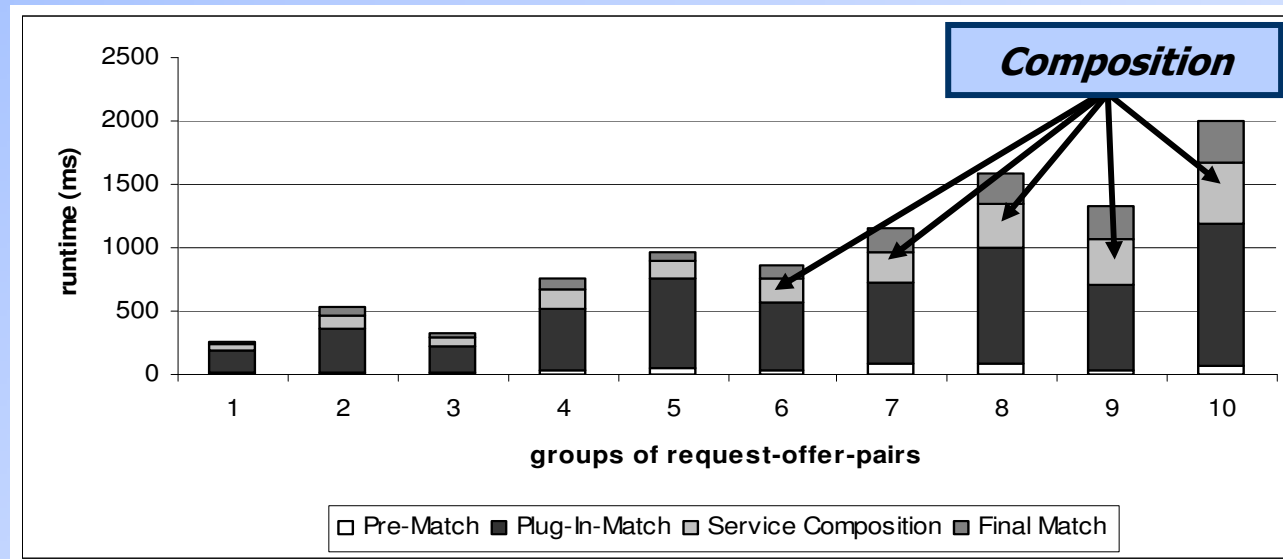
Agenda

- 1. Introduction and Motivation**
- 2. Background information on DSD
(service descriptions and matchmaking)**
- 3. Adaptation to matching multiple effects**
- 4. *Evaluation and Summary***



Experimental Evaluation of Approach

- Matchmaker fully implemented (Java)
 - 11 service offer descriptions at different granularity
 - 11 request descriptions with one to four effects
 - 40 options to create matching compositions
 - Runtime for each match on average less than 1 s
 - Runtime dominated by Plug-In-Match
- ➔ composition doesn't compromise efficient matching





Summary

- Approach integrates automated composition into matchmaking
- Main problems:
 - proper configuration of offers
 - compatibility of used offer configurations
- Addressed problems by
 - extending service description language
 - adapting matching algorithm
 - introducing value propagation semantics
- Implemented and tested approach
 - Working solution for considered case
 - Matching time not dominated by composition



Ongoing Work

- Better (but still efficient) configuration possible using techniques similar to top-k query answering?
- Other composition cases apart from multiple effects (like service chaining to transform inputs and outputs)
- Objective evaluation and certification by SWS-Challenge initiative (www.sws-challenge.org) / W3C SWS-Testbed Incubator Group (www.w3.org/2005/Incubator/swsc/)





Thank you for your attendance!

Questions?

Ulrich Küster

Ulrich.Kuester@uni-jena.de

DIANE project (services in ad hoc networks)

<http://hnsp.inf-bb.uni-jena.de/DIANE/>