WSMO Implementation Workshop 2004

Woogle
meets
Semantic Web Fred

U. Keller, M. Stollberg, D. Fensel

Roadmap ...

- WSMO and Web Service Discovery
- A uniform semantic Modelling Approach
- Discovery Process & WSMO
- Woogle & a Woogle-Prototype
- An Application: Semantic Web Fred
- Conclusions & Future work

Discovery & WSMO

Web Service Discovery

- *Process of identifying web services which can (possibly) help in resolving a user's goal*
- Web Service **vs.** Service
- Requires: **Goal** and **Service specification**

- Currently, we plan to support a wide range of techniques …
  - Require **different efforts** for specification
  - Allow for **different degrees of accuracy**
  - Have **different computational complexity**

Discovery & WSMO (II)

- Three principle approaches …
  - **Syntactical approaches**
    - Keyword-based search, natural language processing techniques, Controlled vocabularies
  - **Lightweight semantic approaches**
    - Ontologies, Action-Object-Modelling, Coarse-grained semantic description of a service
  - **Heavyweight semantic approaches**
    - Describes the service capability in detail, takes in-out relationship into account, Fine-grained web service description

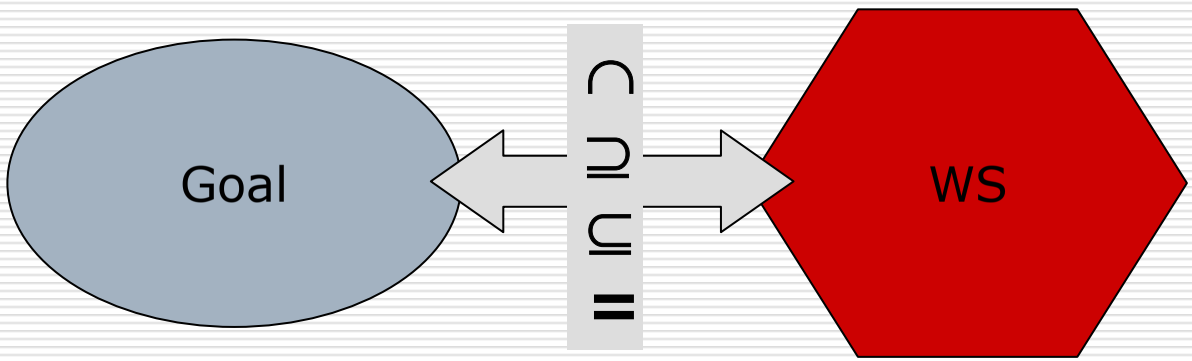
» Here: we are concerned with *semantic-based* approaches!

WS Discovery: Modelling I

- **DL-Style Modelling**
  - The world is understood as sets of things
  - Goal = Set of items ; WS = Set of items
  - Checking a **match** means to establish a certain **relationship** between these **sets**
  - Simple and intuitive way to see the world.

WS Discovery: Modelling Ib

- **Illustration:**
  - ***Intersection***: *G ∩ WS (weak contribution)*
  - ***Subsumes***: *G ⊇ WS (strong contribution)*
  - ***Plugin***: *G ⊆ WS (coverage)*
  - ***Equality***: *G = WS (exact match)*



WS Discovery: Modelling II

- **Extended DL-Style Modelling**
  - The world is understood as sets of things
  - Goal = Set of items ; WS = Set of items
  - Match means presence a certain relationship between these sets
    - Intersection, Plugin, Subsumes, Equality
  - ***Now:*** *The language for defining these sets is very rich (unlike common DLs)*

    goal(x) $\leftrightarrow$ **φ(x)**

    ws(x) $\leftrightarrow$ **ψ(x)**

    where φ, ψ are **arbitrary WSML-FOL** formulas.

    e.g. Plugin-Match: forAll x (goal(x)$\rightarrow$ ws(x))

WS Discovery: Modelling III

- **EDL-style-Modelling with Relations**
  - The world is understood as sets of things
  - Goal = Set of items; WS has inputs *i1, …, iN*
  - WS delivers a **set of items** for **each tuple** of **input values** *i1, …, iN (WS = relation)*
  - Matching means establishing a certain relationship between these sets wrt. (hypoth.) executions of a web service
  - *Now: Match can refer to (hypothetical) execution of a web service*

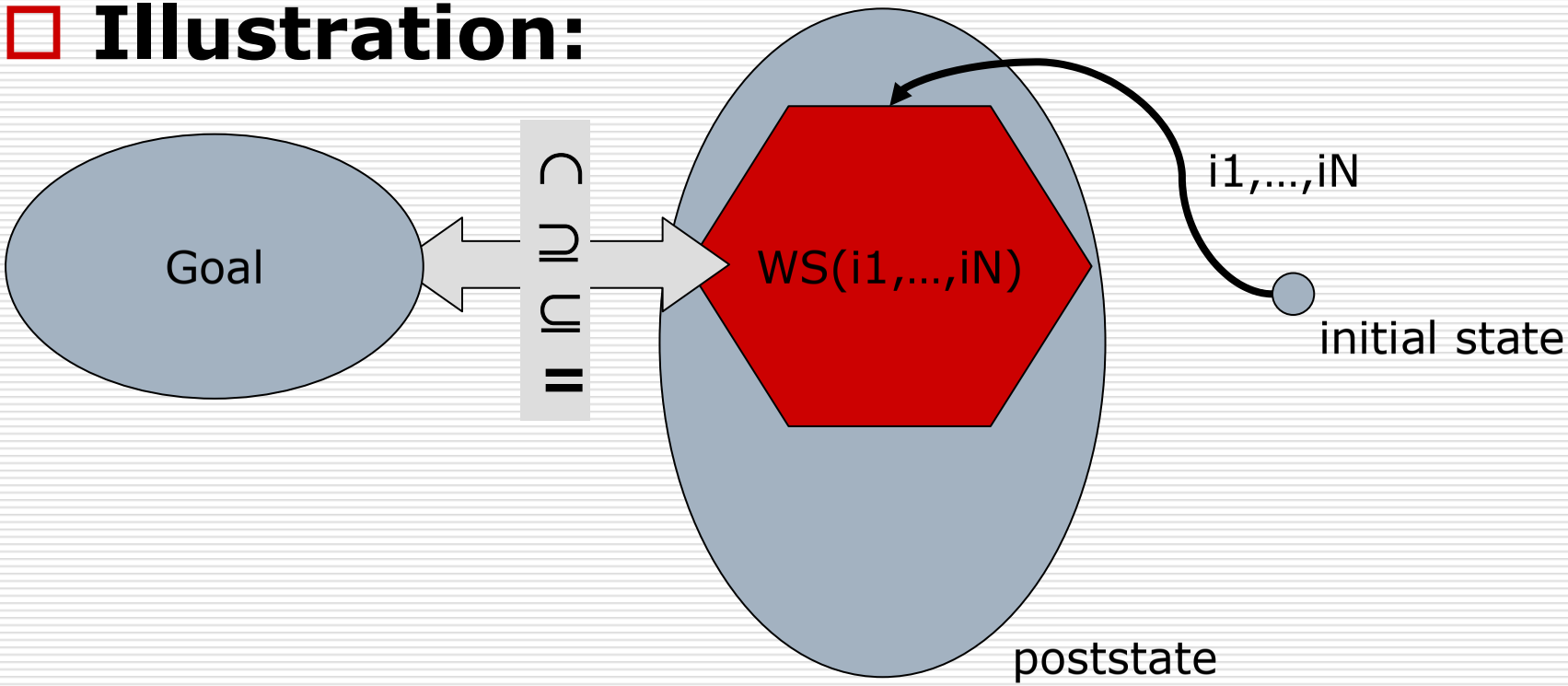
$$\mathbf{goal(x)} \leftrightarrow \varphi\mathbf{(x)}$$

$$\mathbf{ws(x,\ i1,\ldots,\ iN)} \leftrightarrow \psi^{\mathbf{pre}}\mathbf{(i1,\ldots,\ iN)} \wedge \psi^{\mathbf{post}}\mathbf{(i1,\ldots,\ iN,x)}$$

where φ, ψ^{pre}, ψ^{post} are *arbitrary WSML-FOL* formulas

WS Discovery: Modelling IIIb

- **Illustration:**



WS Discovery: Modelling III

- Now …

  Matching criteria can distinguish between single executions of a WS and an arbitrary number of such executions

  - *Plugin-Match*
    - **goal(x) ↔ φ(x)**
    - **ws(x, i1,…, iN) ↔ ψ^{pre}(i1,…, iN) ∧ ψ^{post}(i1,…, iN,x)**
    - **with single execution**
      - exists i1,…,iN (forAll x (goal(x) → ws(x,i1,…,iN))
    - **with an arbitrary number of executions**
      - forAll x (exists i1,…,iN (goal(x) → ws(x,i1,…,iN))

From Web Service Discovery to Service Discovery

- **First**

- **Second ste**
  - A
    ovided (by
    goal

- **Note …**

  By using „*heavyweight*“ *modelling*, one can in principle have an intermediate step between (1) and (2), where a web service is executed solely on a logical level (not in reality)

WOOGLE!

An Example … (I)

- A goal …
  - *„I want to know about all sports events in Frankfurt in the period of Sept 29th to Oct 14th 2004“*

```
goal(x) ↔ exists ?D (
   x[where hasValue frankfurt,
     when hasValue ?D] memberOf SportsEvent
   and date-after-eq(?D, 29.09.2004)
   and date-before-eq(?D, 14.10.2004))
```

An Example … (II)

- A web service description
  - *„The service delivers all events for a given day in the current year in a given region in Germany“*

```
ws(x, ?iDay,?iRegion) ↔
exists ?C(
   x[where hasValue ?C,
     when hasValue ?iDay] memberOf Event and
   located-in(?C, ?iRegion) and
year-of-date(?iDay, currentYear) and
located-in(?iRegion, germany)
```

An Example … (III) NO MATCH!

- **The proofobligation …**
  - User selects Plugin-Match (coverage) with single invocation

```
exists ?iDay, iRegion (forAll x (
    exists ?D ( x[where hasValue frankfurt,
      when hasValue ?D] memberOf SportsEvent
    and date-after-eq(?D, 29.09.2004)
    and date-before-eq(?D, 14.10.2004))
->
    exists ?C(
        x[where hasValue ?C,
          when hasValue ?iDay] memberOf Event and
        located-in(?C, ?iRegion) and
        located-in(?iRegion, germany) and
        year-of-date(?iDay, currentYear)
    ))
```

An Example ... (IV)

MATCH!

- **The proofobligation ...**
  - User selects Plugin-Match (coverage) with multiple invocation

```
forAll x (exists ?iDay, iRegion (
    exists ?D( x[where hasValue frankfurt,
      when hasValue ?D] memberOf SportsEvent
    and date-after-eq(?D, 29.09.2004)
    and date-before-eq(?D, 14.10.2004)
->
    exists ?C(
        x[where hasValue ?C,
          when hasValue ?iDay] memberOf Event and
        located-in(?C, ?iRegion) and
        located-in(?iRegion, germany) and
        year-of-date(?iDay, currentYear)
    ))
```

?iRegion = Hessen
?iDay = ?D
?C = frankfurt

Woogle: A prototypical implementation …

- **Translation-based** Approach
  - Supports full WSML-FOL
  - Translate WSML-FOL to First-order Logic
- **Main inference engine**
  - First-order Logic Theorem Prover (with Equality)
  - Currently: TPTP format for loose coupling with prover
- **Does it scale ?**
  - Sorry, we don‘t know … … **yet** :- (!
  - Interesting approach: Tuplespace-based architecture
  - Prepocessing techniques to extract relevant axioms
  - Exploit Template/Instance Scenario if possible (SWF)

A prototypical implementation …

- **Pros …**
  - Supports full WSML-FOL
  - Generality gives Flexibility
  - Prover can be reused for different tasks
- **Cons …**
  - Full First-order Logic Theorem Prover
  - Scalability is a concern
  - Perhaps WSML-FOL too heavy for many applications ?

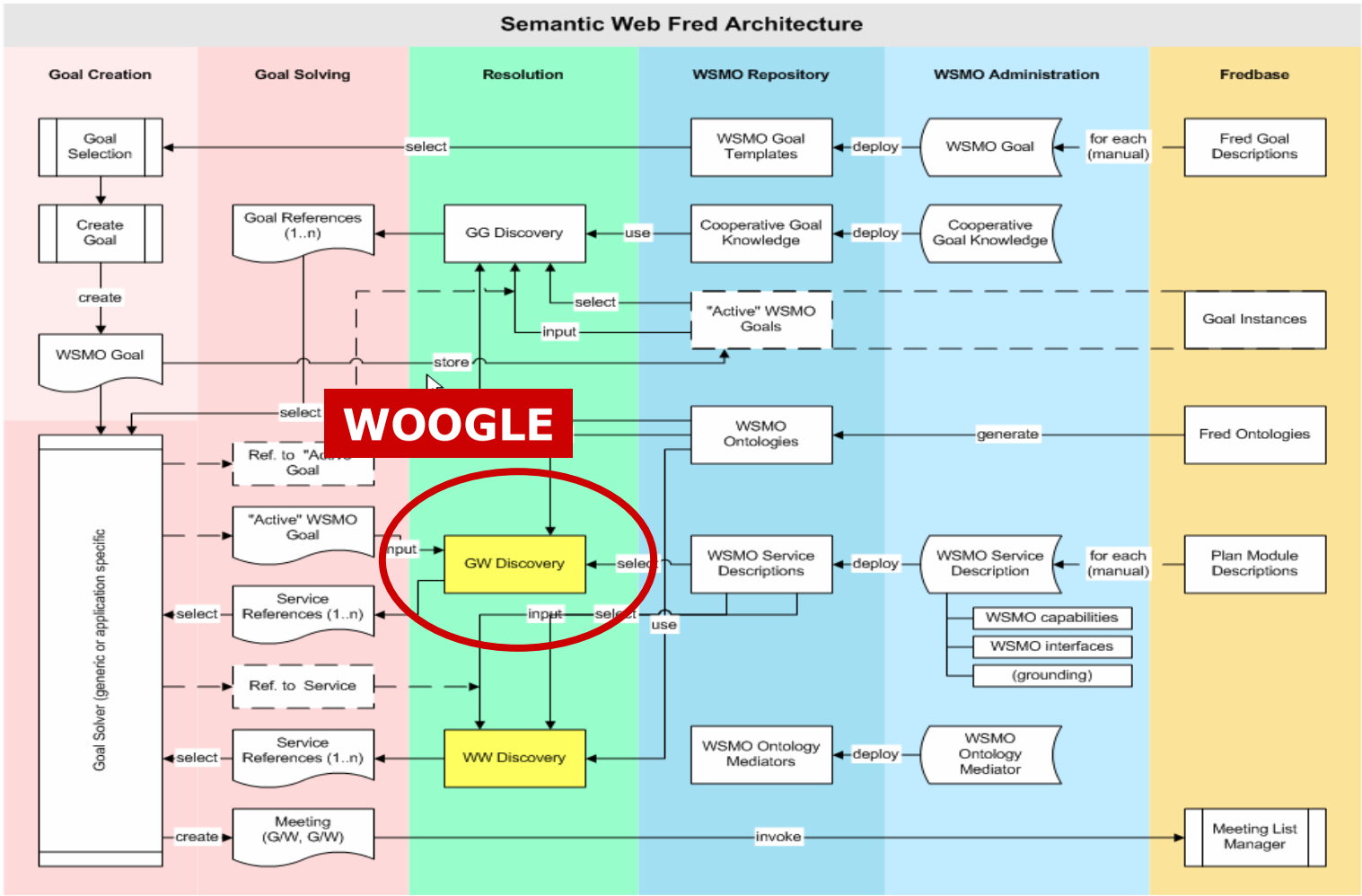
A WSMO-enabled Application: **Semantic Web Fred**

- What is **Semantic Web Fred**?
  - **Not a weird Internet junkie ;-)**
  - **But …**

    *An agent-based system that supports **automated goal-resolution** in a **cooperative** environment*

    - A **WSMO-enabled** software system
  - **It combines techniques from …**
    - *Agent community,*
    - *Semantic Web and*
    - *Web Services*

Semantic Web Fred (III)



Conclusions & Future Work

- **Currently**
  - General framework for Service Discovery has been outlined
  - We are in the process of implementing a prototype implementation
  - Implementation is used in SWF project
- **Future work**
  - Evaluation of modelling approach
  - Evaluation of the Woogle prototype