

第一届全国智能信息处理学术会议

基于主体的语义 Web 服务

史忠植

shizz@ics.ict.ac.cn

中国科学院计算技术研究所



中国科学院计算所
INSTITUTE OF COMPUTING TECHNOLOGY

内容提要

- 引言
- 动态描述逻辑
- **Web**服务的语义标识
- 自动**Web**服务发现
- 自动**Web**服务组合
- 语义**Web**服务平台**SWSBroker**
- 结束语

Web 服务



中科院计算所
INSTITUTE OF COMPUTING
TECHNOLOGY



Web 服务

Dynamic

Web Services

UDDI, WSDL, SOAP



Bringing the computer back as
a device for computation

Static

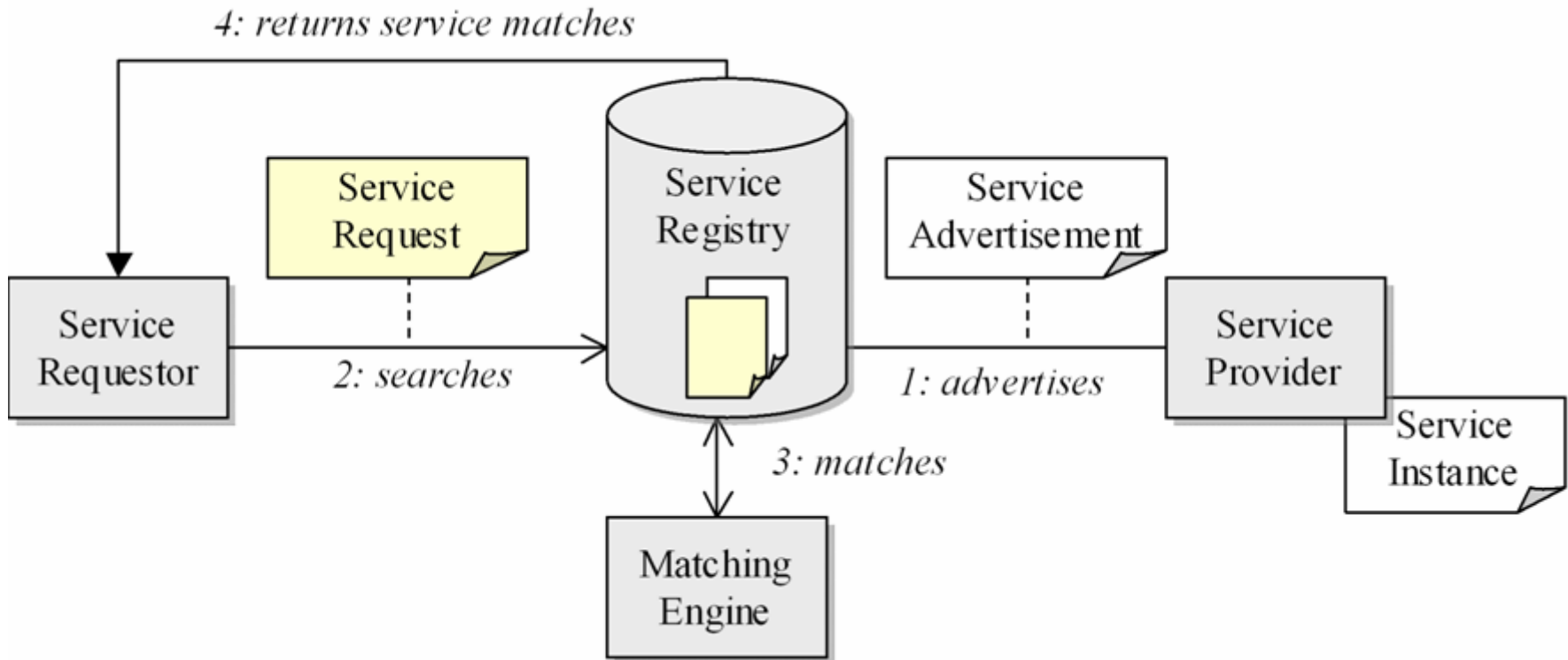
WWW

URI, HTML, HTTP

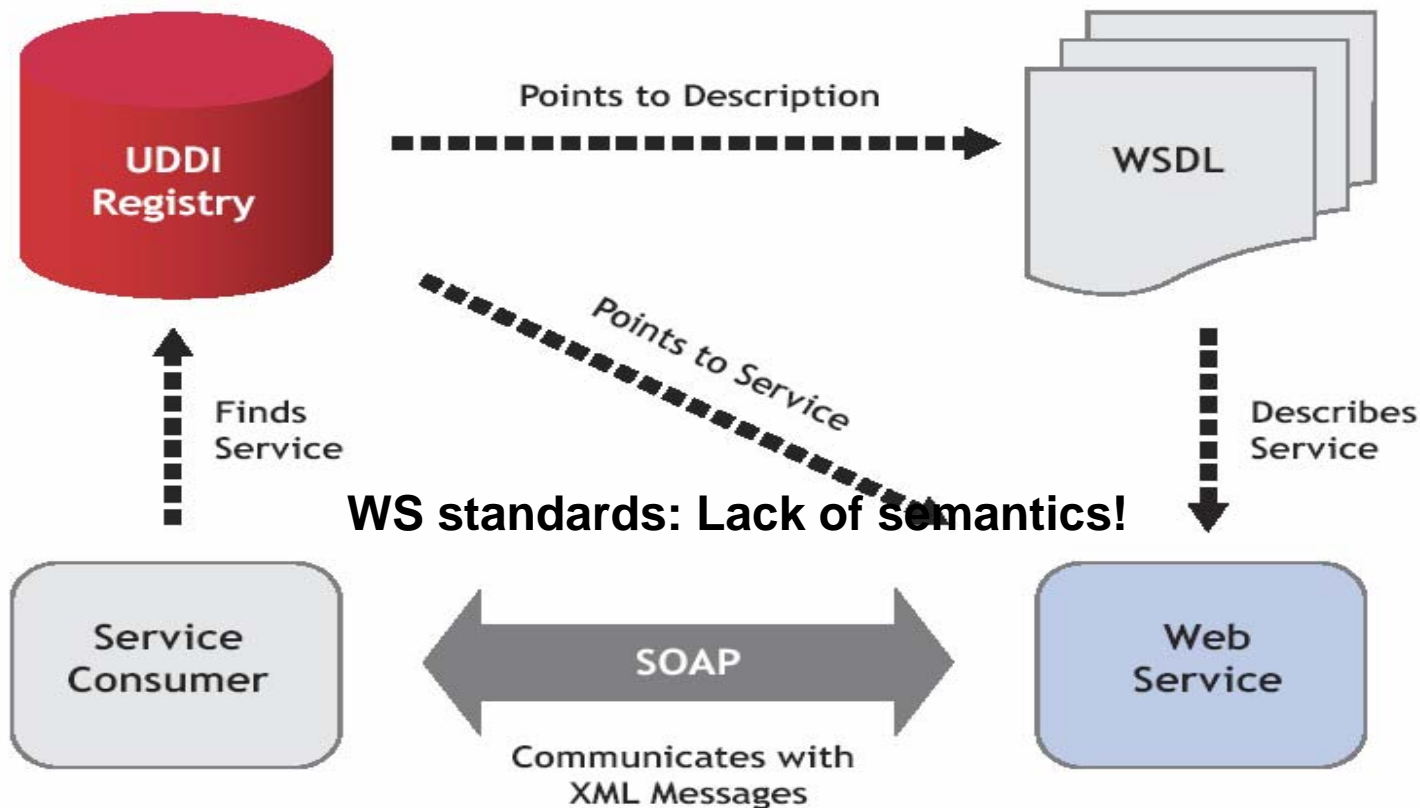
Web 服务协议

???	???	Management	Quality of Service	Security
Routing, Reliability and Transaction	???			
Workflow	WSFL			
Service Discovery, Integration	UDDI			
Service Description	WSDL			
Messaging	SOAP			
Transport	HTTP, FTP, SMTP			
Internet	IPv4, IPv6			

WS 参考架构



Web 服务

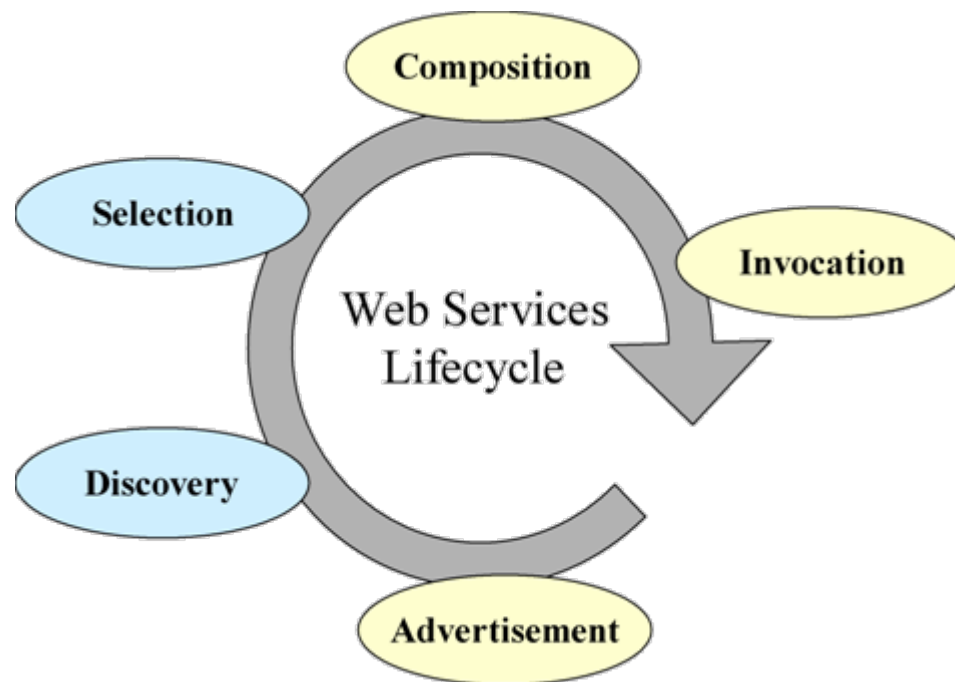


WS standards: Lack of Semantics!

Web Service Architecture

Web 服务

- *programmatic interfaces for applications (i.e., business logic), available over the WWW infrastructure and developed with XML technologies.*



语义 Web的目的

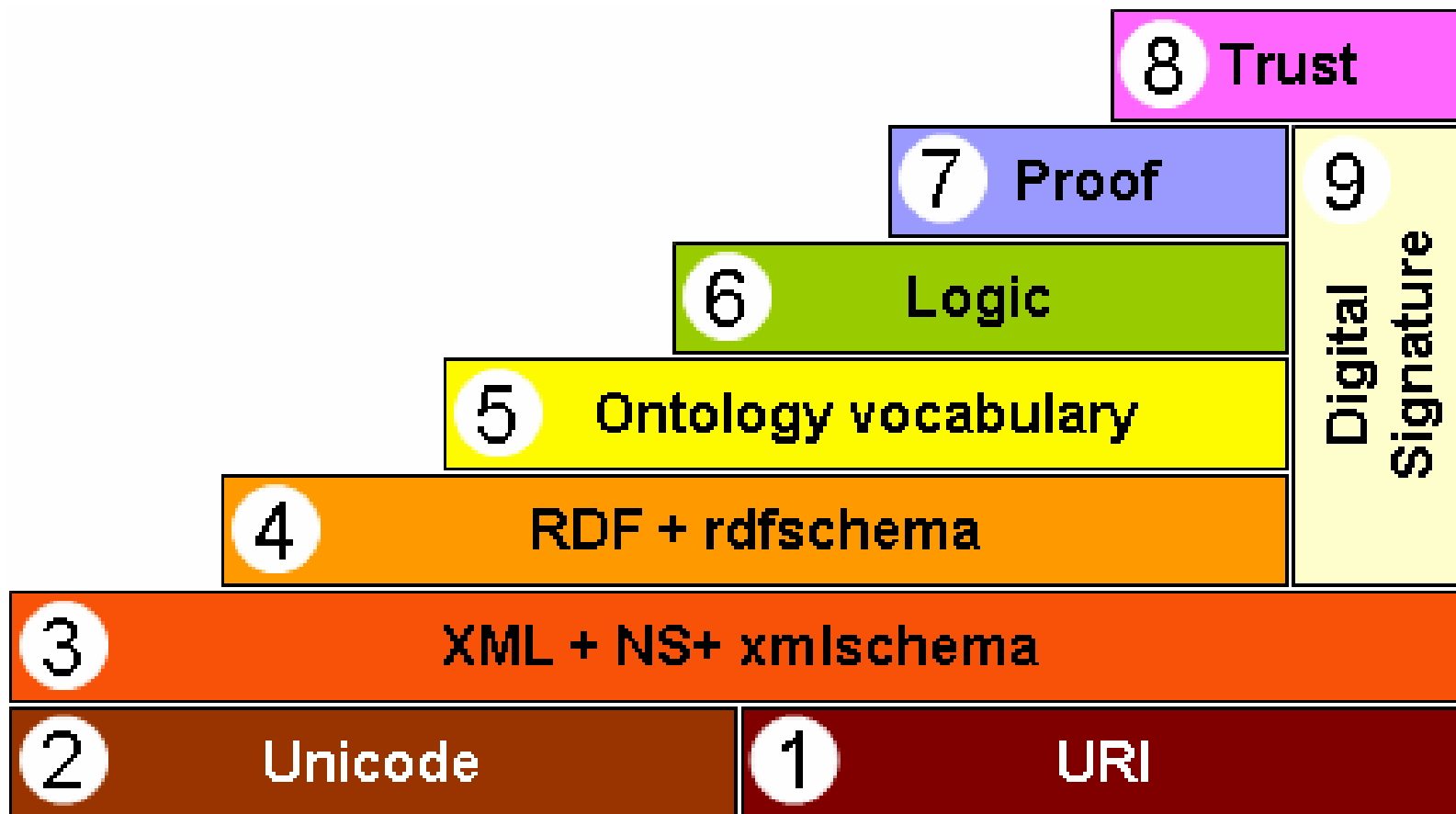
	Static	Dynamic	Syntax	Semantic
Encoding	HTML	+ RDBMS	+ XML	+ RDF/OWL
Creation	Manually	Generated by server-side applications	Generated by applications based on schema	Generated by applications based on models
Users	Humans	Humans	Humans and applications	Humans and applications
Paradigm	Browse	Create/Query/Update	Integrate	Interoperate
Applications	Browsers	Browsers	Process Integration, EAI, BPMS, Workflows	Intelligent agents, Semantic engines

1995

2000

2005

语义 Web 的架构



Web Ontology Language (OWL)

- OWL is built on top of RDF
- OWL is for processing information on the web
- OWL was designed to be interpreted by computers
- OWL was not designed for being read by people
- OWL is written in XML
- OWL is a web standard

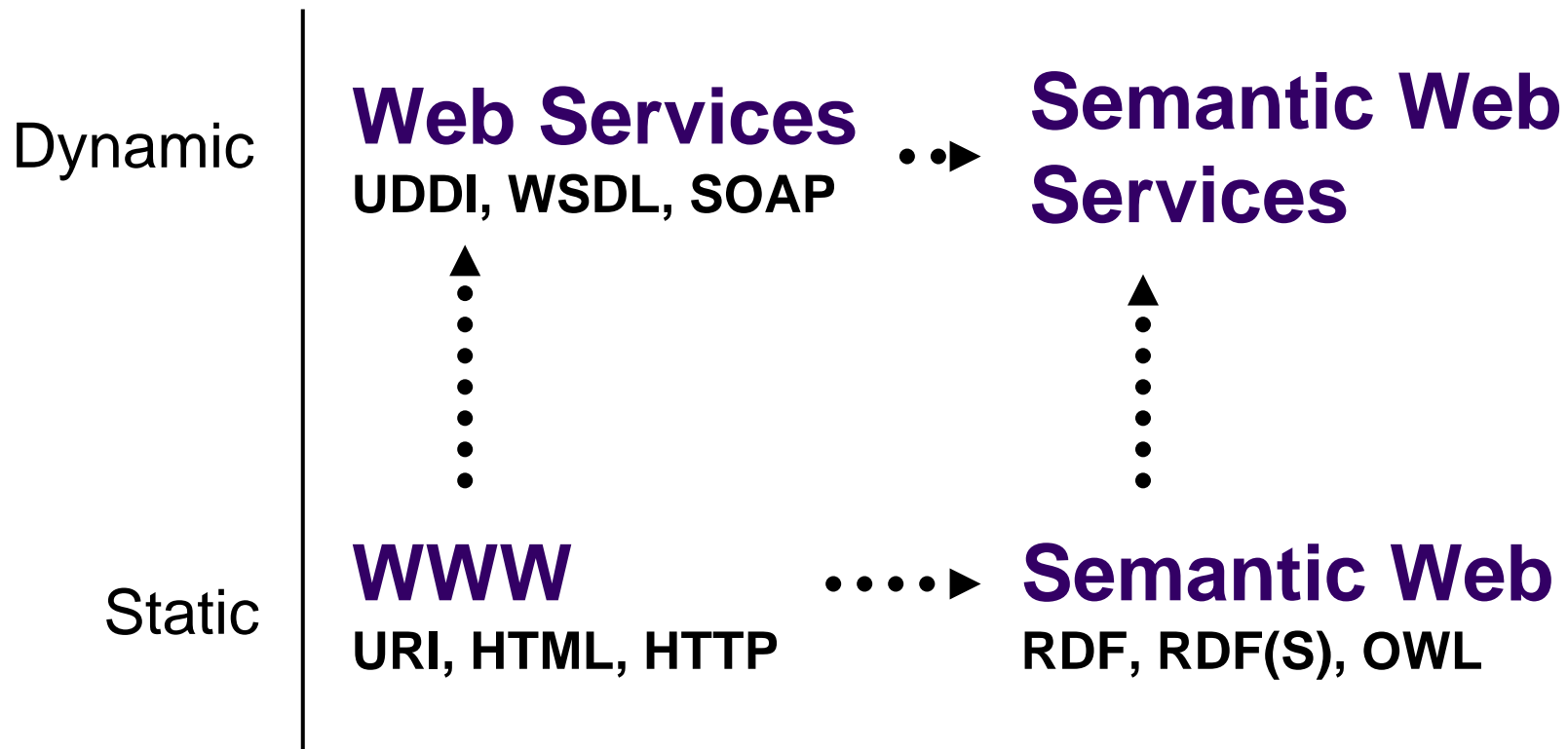
OWL

- Ability to be distributed across many systems
- Scalability to Web needs
- Compatibility with Web standards for accessibility and internationalization
- Openness and extensibility

OWL 推理机

- [KAON2](#) is a reasoner for OWL extended with the DL-safe subset of SWRL; it also provides an OWL API.
- FaCT -- a DL reasoner. see [WonderWeb project](#), [Bechhofer 15 Sep](#).
- Racer -- a DL reasoner. see [Horrocks 12Sep](#)
- Cerebra from [Network Inference](#) - **owl syntax checker**, nearly complete OWL DL [Horrocks 12Sep](#)
- [cwm](#) -- useful but incomplete OWL Full
- Euler -- useful but incomplete OWL Full, see [De Roo 11 Jul](#): 51 / 234 tests
- [surnia](#) -- OWL full reasoner based on otter. [Hawke 26Aug](#)
- [Jena/HP](#) ([Reynolds/HP 7 May](#)) will support OWL reasoning.
- Vampire [Horrocks 17 Jul](#) - uses a first-order theorem prover to do OWL DL
- [Pellet](#) is a reasoner built in Java that was designed specifically for OWL reasoning. [Hendler/Sirin/Parsia 15Sep](#)).
- [SWI-Prolog Semantic Web Library](#) contains *owl.pl* - an OWL reasoning package.
- [F-OWL](#) is an f-logic based Owl tool from UMBC.
- [E-wallet](#) is an e-commerce and mobile computing tool based on a rule-based OWL reasoner.

语义 Web 服务



语义 Web 服务

Semantic Web Services

=

Semantic Web Technology

+

Web Service Technology

语义 Web 服务

- Define exhaustive description frameworks for describing Web Services and related aspects (**Web Service Description Ontologies**)
- Support ontologies as underlying data model to allow machine supported data interpretation (**Semantic Web aspect**)
- Define semantically driven technologies for automation of the Web Service usage process (**Web Service aspect**)

语义 Web 服务

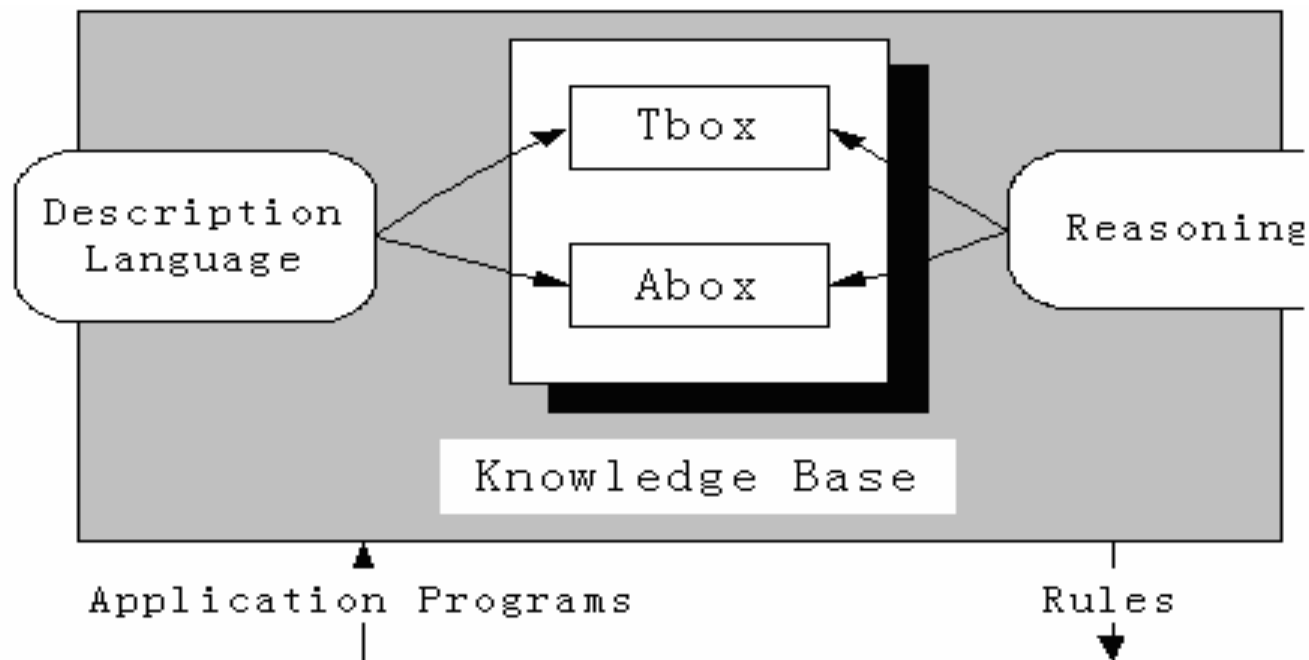
- What should S+WS ontologies provide?
(Mainly) Automation of the Usage Process:
 - Publication
 - Discovery
 - Selection
 - Composition
 - Execution
 - Monitoring

内容提要

- 引言
- 动态描述逻辑
- **Web**服务的语义标识
- 自动**Web**服务发现
- 自动**Web**服务组合
- 语义**Web**服务平台**SWSBroker**
- 结束语

描述逻辑 Description Logics

- A description logic system consists of four parts: constructors, Tbox, Abox and reasoning mechanism of Tbox and Abox.



语法和语义



Operator	syntax	semantics	example
Atomic concept	A	$A^I \subseteq \Delta^I$	
Atomic role	R	$R^I \subseteq \Delta^I \times \Delta^I$	has-child
On concept C,D and role R \Rightarrow			
conjunction	$C \sqcap D$	$C^I \cap D^I$	Human \sqcap Male
disjunction	$C \sqcup D$	$C^I \cup D^I$	Doctor \sqcup Lawyer
Negative	$\neg C$	$\Delta^I \setminus C^I$	\neg Male
Exist	$\exists R.C$	$\{x/ \exists y. \langle x,y \rangle \in R^I \wedge y \in C^I\}$	\exists has-child.Male
8/13/2007 all	$\forall R.C$	$\{x/ \forall y. \langle x,y \rangle \in R^I \Rightarrow y \in C^I\}$	\forall has-child.Doctor

描述逻辑的推理

- 1) Subsumption
- 2) Consistency
- 3) Satisfiability
- 4) Instance checking

K

TBox(Scheme)

Man = Human \sqcap Male

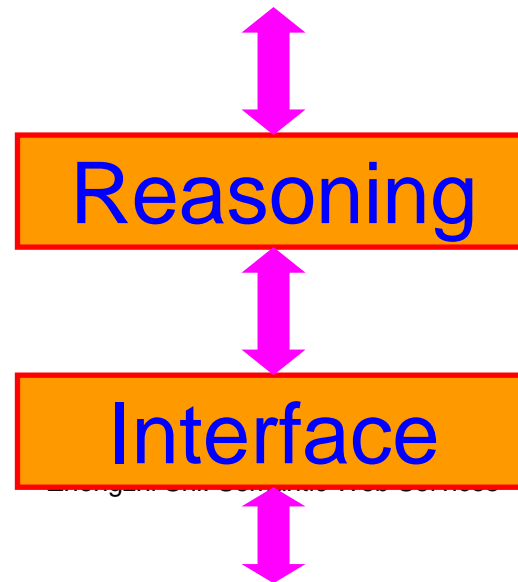
B

Happy-father = Human $\sqcap \exists$ Has-child. Female $\sqcap \dots$

Abox(Data)

John: Happy-father

<John, Mary> : Has-child



Dynamic Description Logic

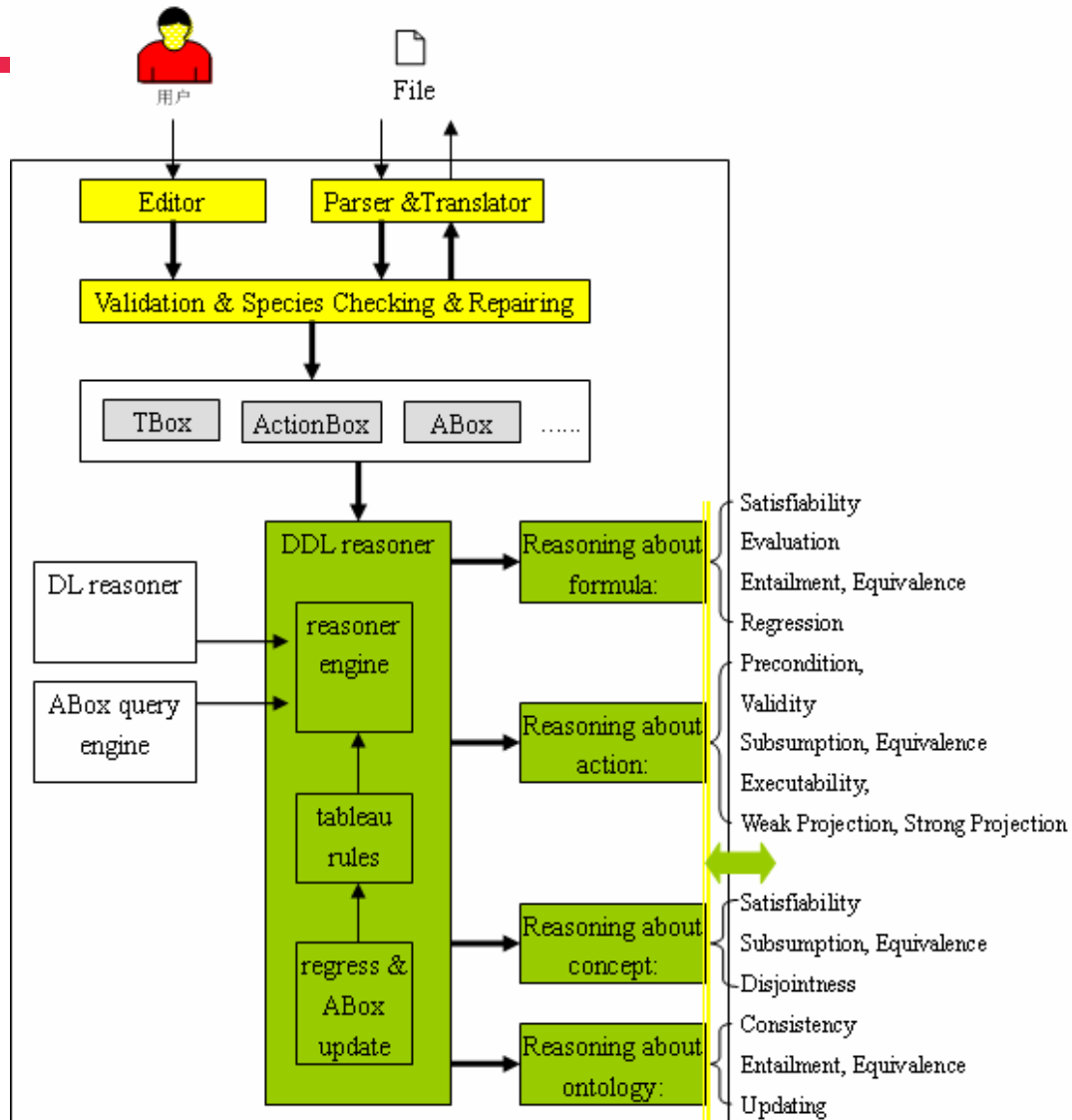
An action description is the form of

$$A(x_1, \dots, x_n) = (P_A, E_A)$$

where

- (1) A is the action name.*
- (2) x_1, \dots, x_n are individual variables, which denote the objects the action operate on.*
- (3) P_A is the set of preconditions, which must be satisfied before the action is executed.*
- (4) E_A is the set of results, which denote the effects of the action.*

DDL 推理机



DDL 推理机



DDL Version 1.0 Intsci ICT

File View Bookmarks Resource Holder Advanced About

Address:

Add Add Add

Add GCI Remove Rename

Add Action Add Con... Add For...

Action Concept Formula Individual Property

graduate

succDegreeDefend

failDegreeDefend

degreeDefend

succMasterDefend

failMasterDefend

masterDefend

succPhdDefend

failPhdDefend

PhdDefend

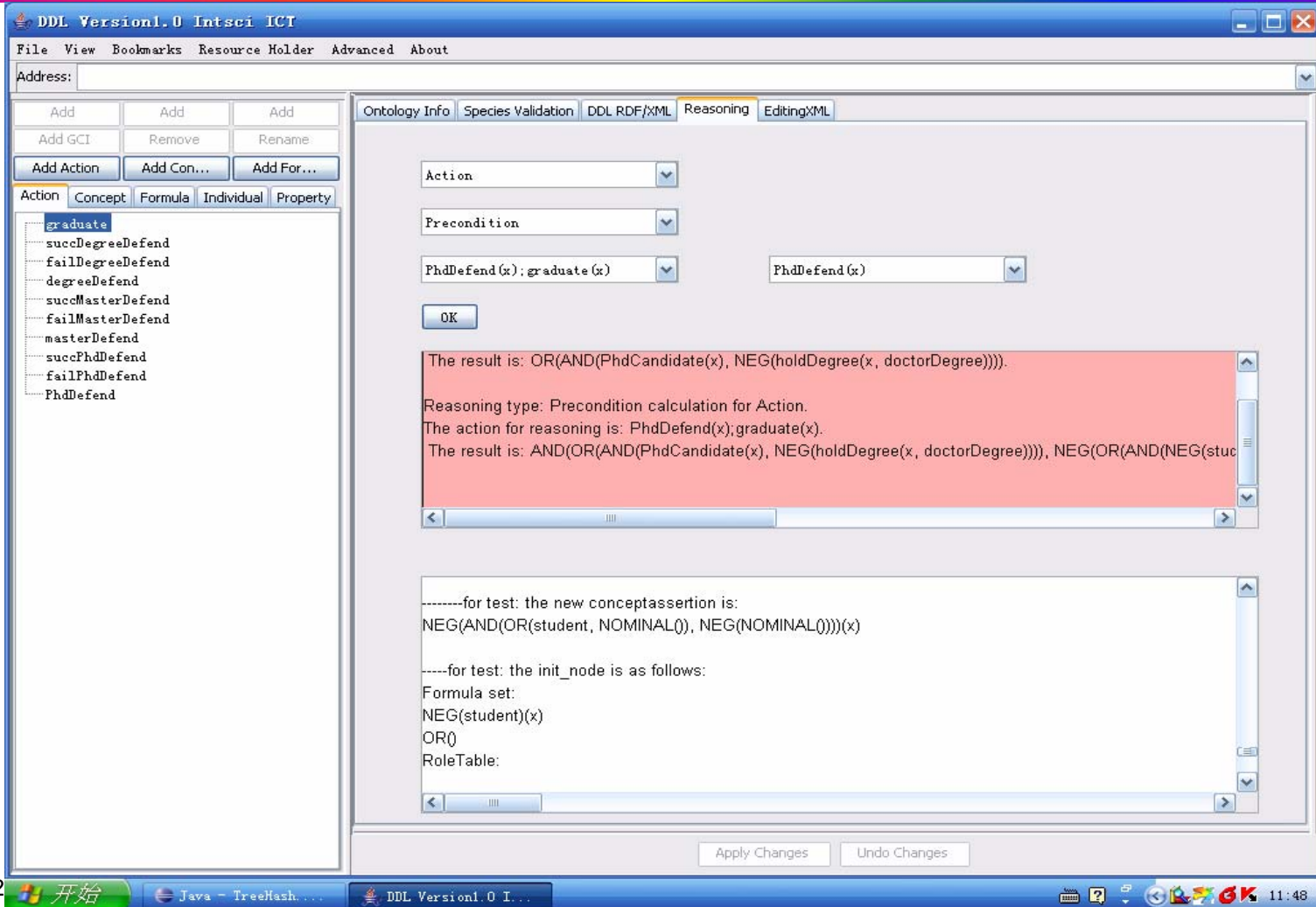
Ontology Info Species Validation DDL RDF/XML Reasoning EditingXML

```
<ddl:actionAxiom xmlns:ddl="http://www.intsci.ac.cn/ddl/reasoner/ddl#" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <ddl:actionParameters>
    <ddl:actionParameter ddl:ID="v"/>
  </ddl:actionParameters>
  <ddl:actionPreconditions>
    <ddl:actionPrecondition>
      <ddl:conceptAssertion>
        <ddl:conceptReferred rdf:resource="#student"/>
        <ddl:individualReferred ddl:resource="v"/>
      </ddl:conceptAssertion>
    </ddl:actionPrecondition>
  </ddl:actionPreconditions>
  <ddl:actionEffects>
    <ddl:actionEffect>
      <ddl:negFormula>
        <ddl:conceptAssertion>
          <ddl:conceptReferred rdf:resource="#student"/>
          <ddl:individualReferred ddl:resource="v"/>
        </ddl:conceptAssertion>
      </ddl:negFormula>
    </ddl:actionEffect>
  </ddl:actionEffects>
</ddl:actionAxiom>
```

Apply Changes Undo Changes

8/13/2007 11:40

DDL 推理机



The screenshot shows the DDL Version 1.0 Intsci ICT software interface. The window title is "DDL Version 1.0 Intsci ICT". The menu bar includes "File", "View", "Bookmarks", "Resource Holder", "Advanced", and "About". The address bar is empty. The main interface is divided into several sections:

- Left Panel:** A tree view showing a hierarchy of concepts. The selected concept is "graduate". Below it are "succDegreeDefend", "failDegreeDefend", "degreeDefend", "succMasterDefend", "failMasterDefend", "masterDefend", "succPhdDefend", "failPhdDefend", and "PhdDefend".
- Top Buttons:** A row of buttons for "Add", "Add GCI", "Remove", "Rename", "Add Action", "Add Con...", and "Add For...".
- Tabbed Interface:** The "Reasoning" tab is active. It contains:
 - Dropdown menus for "Action" and "Precondition".
 - Dropdown menus for "PhdDefend(x) : graduate(x)" and "PhdDefend(x)".
 - An "OK" button.
 - A large text area displaying the result of a reasoning operation:

```
The result is: OR(AND(PhdCandidate(x), NEG(holdDegree(x, doctorDegree))))).  
Reasoning type: Precondition calculation for Action.  
The action for reasoning is: PhdDefend(x), graduate(x).  
The result is: AND(OR(AND(PhdCandidate(x), NEG(holdDegree(x, doctorDegree))))), NEG(OR(AND(NEG(stu
```
 - A scrollable text area at the bottom showing test assertions:

```
-----for test: the new conceptassertion is:  
NEG(AND(OR(student, NOMINAL()), NEG(NOMINAL())))(x)  
  
-----for test: the init_node is as follows:  
Formula set:  
NEG(student)(x)  
OR()  
RoleTable:
```
- Bottom Buttons:** "Apply Changes" and "Undo Changes".

内容提要

- 引言
- 动态描述逻辑
- **Web服务的语义标识**
- 自动**Web**服务发现
- 自动**Web**服务组合
- 语义**Web**服务平台**SWSBroker**
- 结束语

OWL的目标

Design Goals for OWL

- **Shareable**
- **Changing** over time
- **Interoperability** between ontologies
- **Inconsistency** detection (requires a logic)
- Balancing **expressivity and complexity**
- **Ease of use**
- Compatible with **existing standards**
- **Internationalisation**

本体的作用

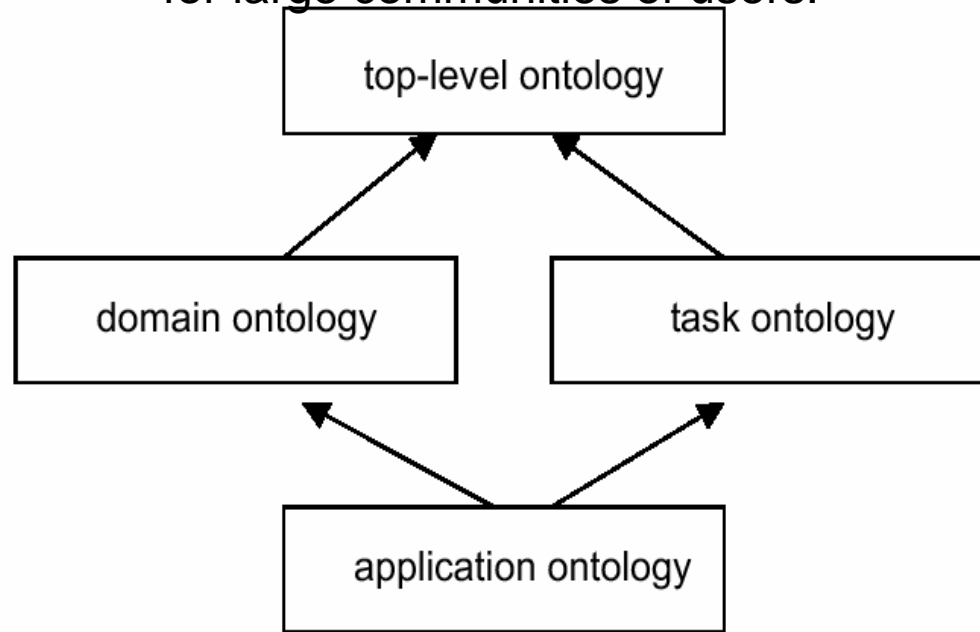
- To make domain assumptions **explicit**
 - Easier to change domain assumptions
 - Easier to understand and update legacy data
- To separate **domain knowledge** from operational knowledge
 - Re-use domain and operational knowledge separately
- A **community reference** for applications
- To **share a consistent understanding** of what information means

本体的类型



Describe **very general concepts** like space, time, event, which are independent of a particular problem or domain. It seems reasonable to have unified top-level ontologies for large communities of users.

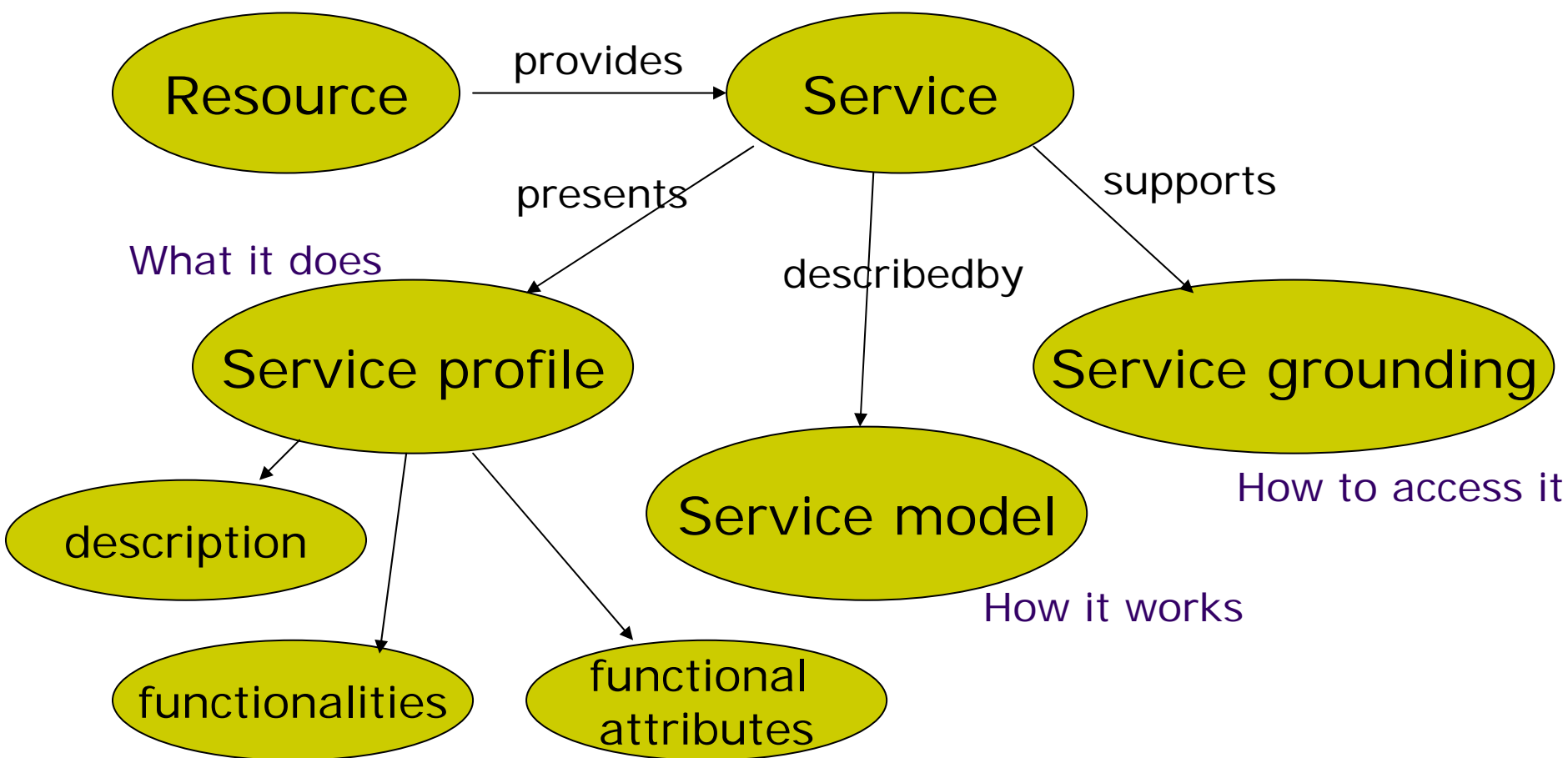
Describe the vocabulary related to a **generic domain** by specializing the concepts introduced in the top-level ontology.



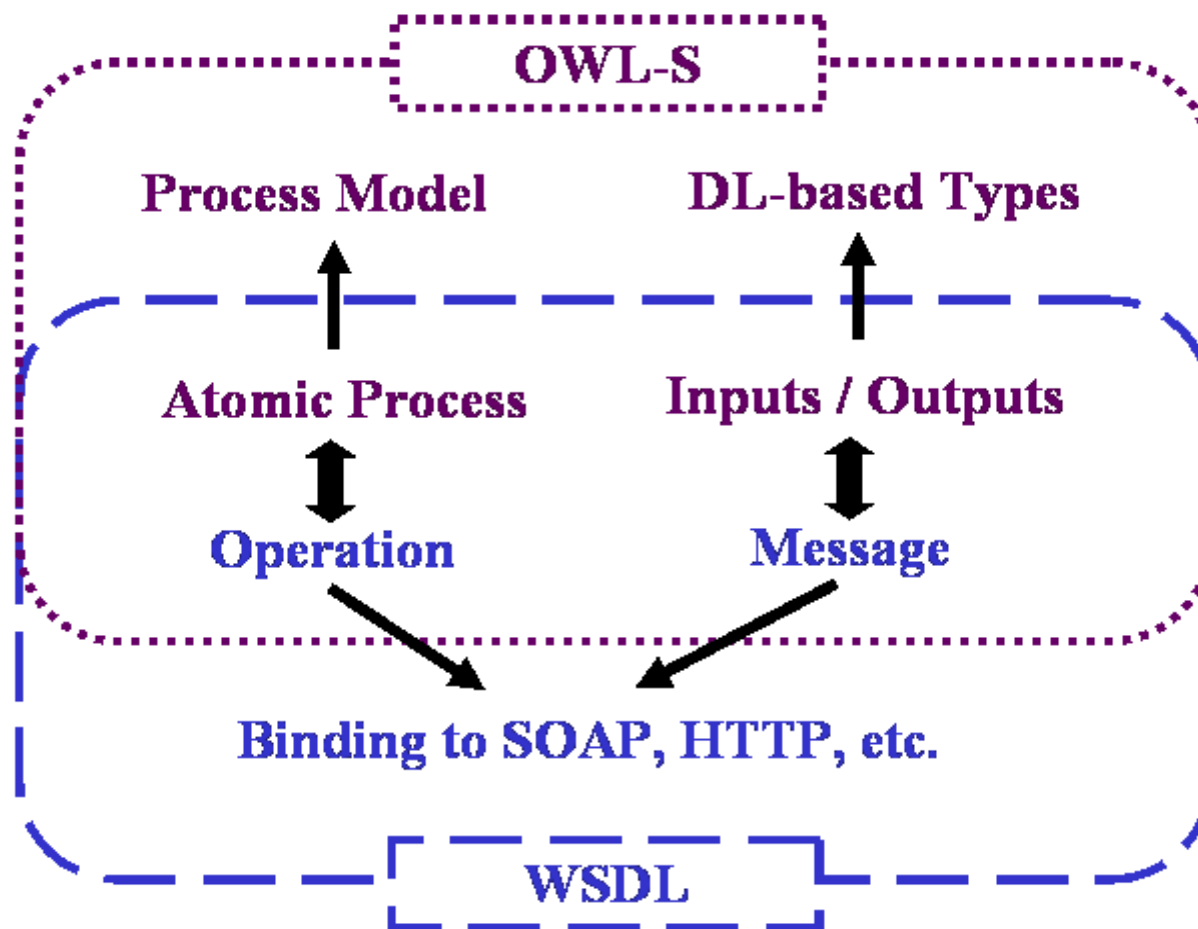
Describe the vocabulary related to a **generic task or activity** by specializing the top-level ontologies.

These are the most specific ontologies. Concepts in application ontologies often correspond to **roles played by domain entities while performing a certain activity**.

Web服务的上位本体OWL-S



OWL-S



Semantic Web Service



- Semantic Web Service Develop environment for:
 - Automated discovery
 - Automated composition
 - Automated invocation
 - Automated annotation
 - Automated mediation

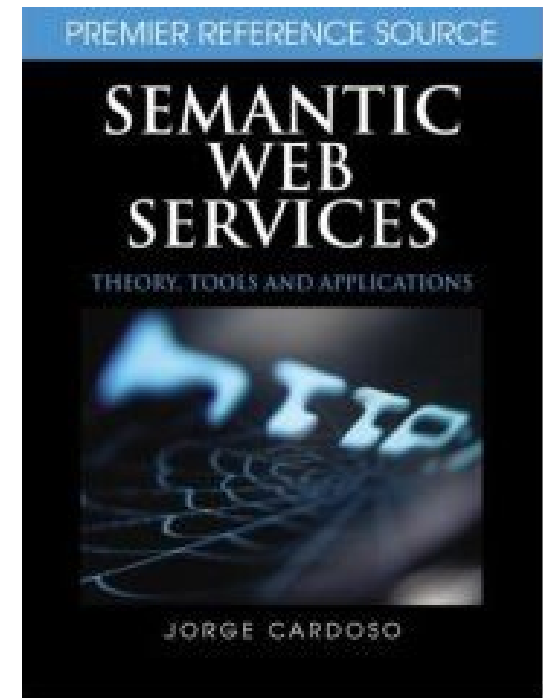
内容提要

- 引言
- 动态描述逻辑
- **Web**服务的语义标识
- 自动**Web**服务发现
- 自动**Web**服务组合
- 语义**Web**服务平台**SWSBroker**
- 结束语

服务匹配

Degree of Match (DoM)

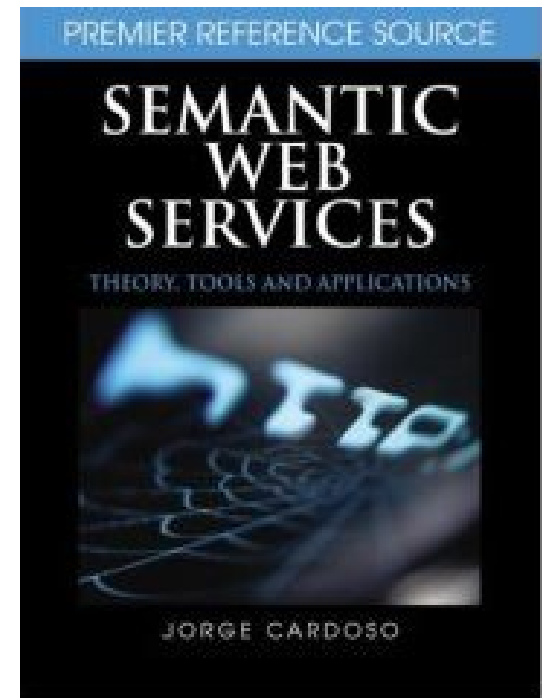
- *A value that expresses how similar two entities are, with respect to some similarity metric(s)*
- Important feature of most SWS matchmaking approaches
- Allows for ranking of discovered services
- Example DoM set: exact, plugin, subsumes, subsumed-by, fail



匹配方法

Variety of Matchmaking Approaches

- Direct
 - Return only single services that match the request
- Indirect
 - Compute service compositions (or “chains” in the simplest case)
- Logic-based
 - Description Logics and First Order Logic reasoning
- Similarity-based (IR techniques)
 - Linguistic similarity, term frequency, ...
- Graph matching



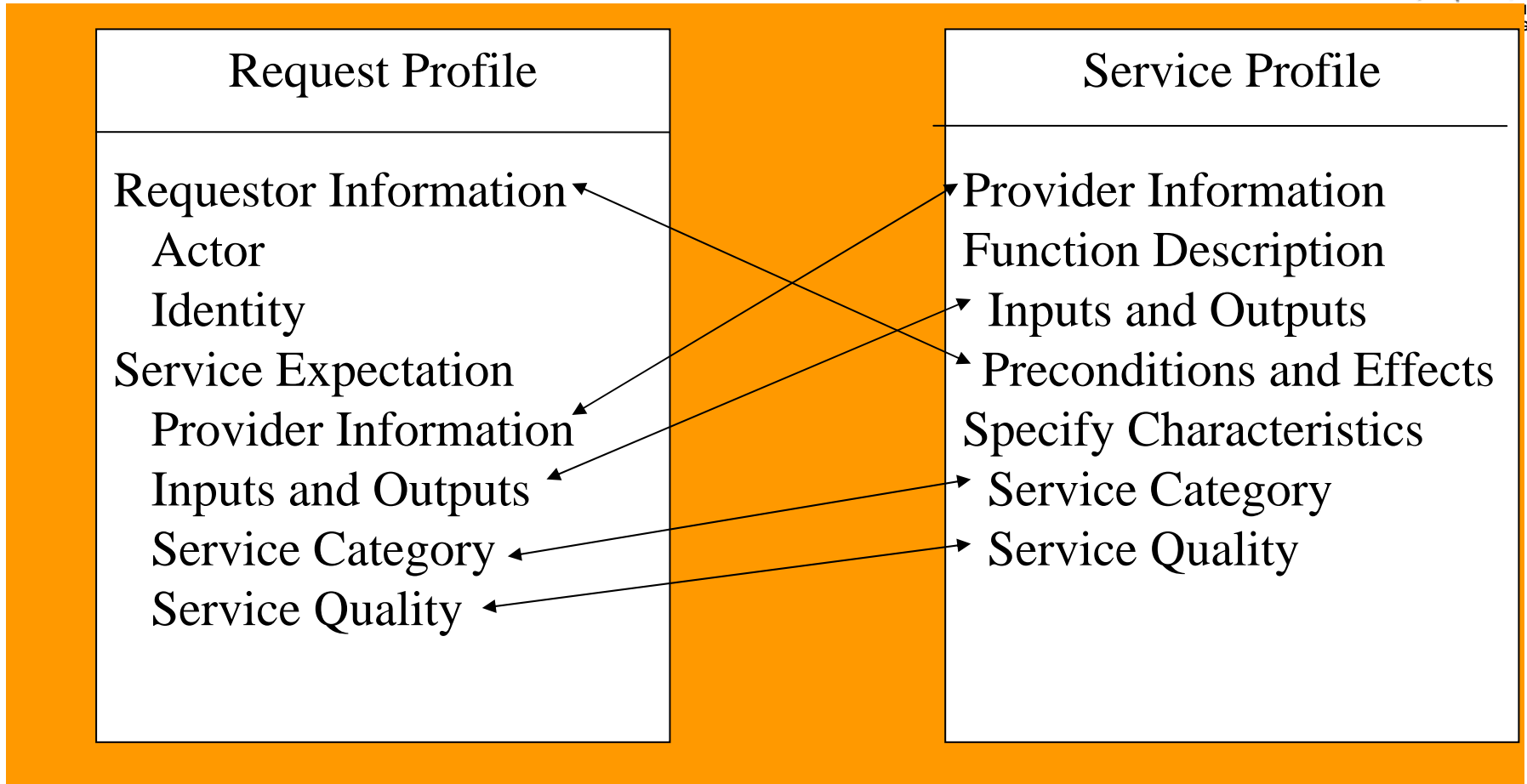
Web 服务发现

- Keyword based
 - UDDI – Universal Description, Discovery, and Integration
 - WSMX (WSMF)
- Semantic discovery
 - Matchmaking – compares advertised and requested service capabilities
 - Subsumption of classes, properties as well as equivalence can be considered
 - The process can be domain ontology dependent

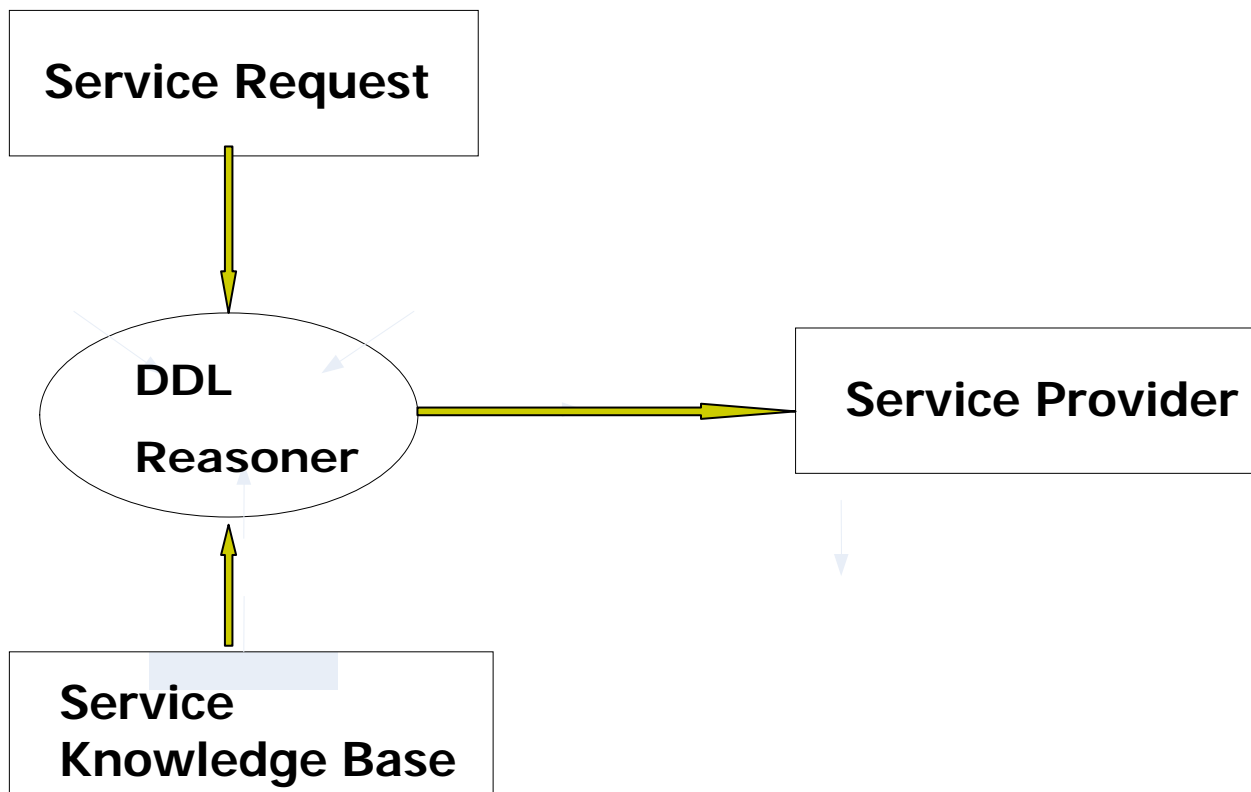
Semantic discovery implementations

- MINDSWAP, Univ. of Maryland
 - Composer – demo only;
 - Matchmaker and planner licensed to FujitsuLabs (no sources)
- Carnegie Mellon, Atlas project
 - Matchmaker, OWL-S -> UDDI
 - No sources
- IBM Emerging Technologies Toolkit (Alpha)
 - Demo available; no sources
- TU Berlin Matchmaker
 - Open source; demo, based on transformation to

匹配 Match Making



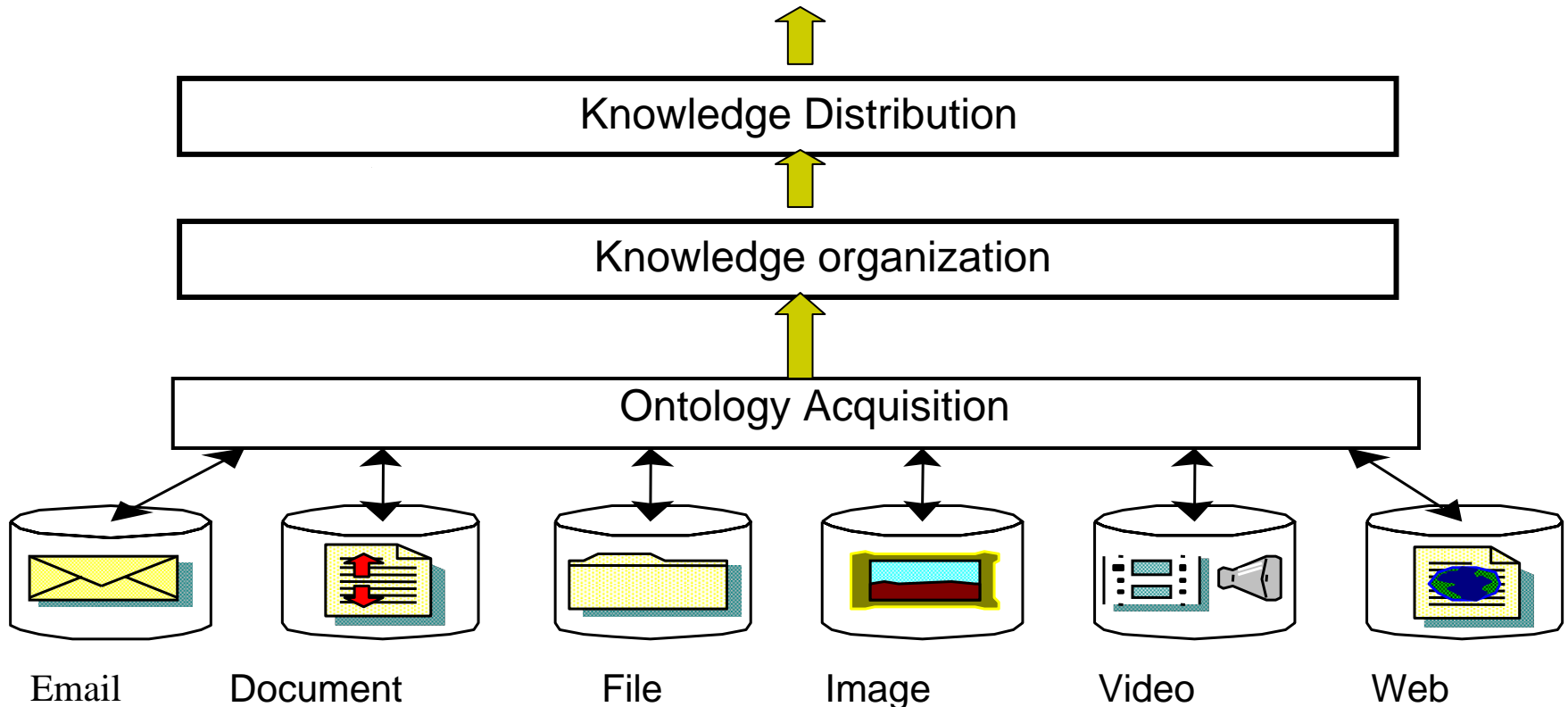
采用DDL推理机匹配



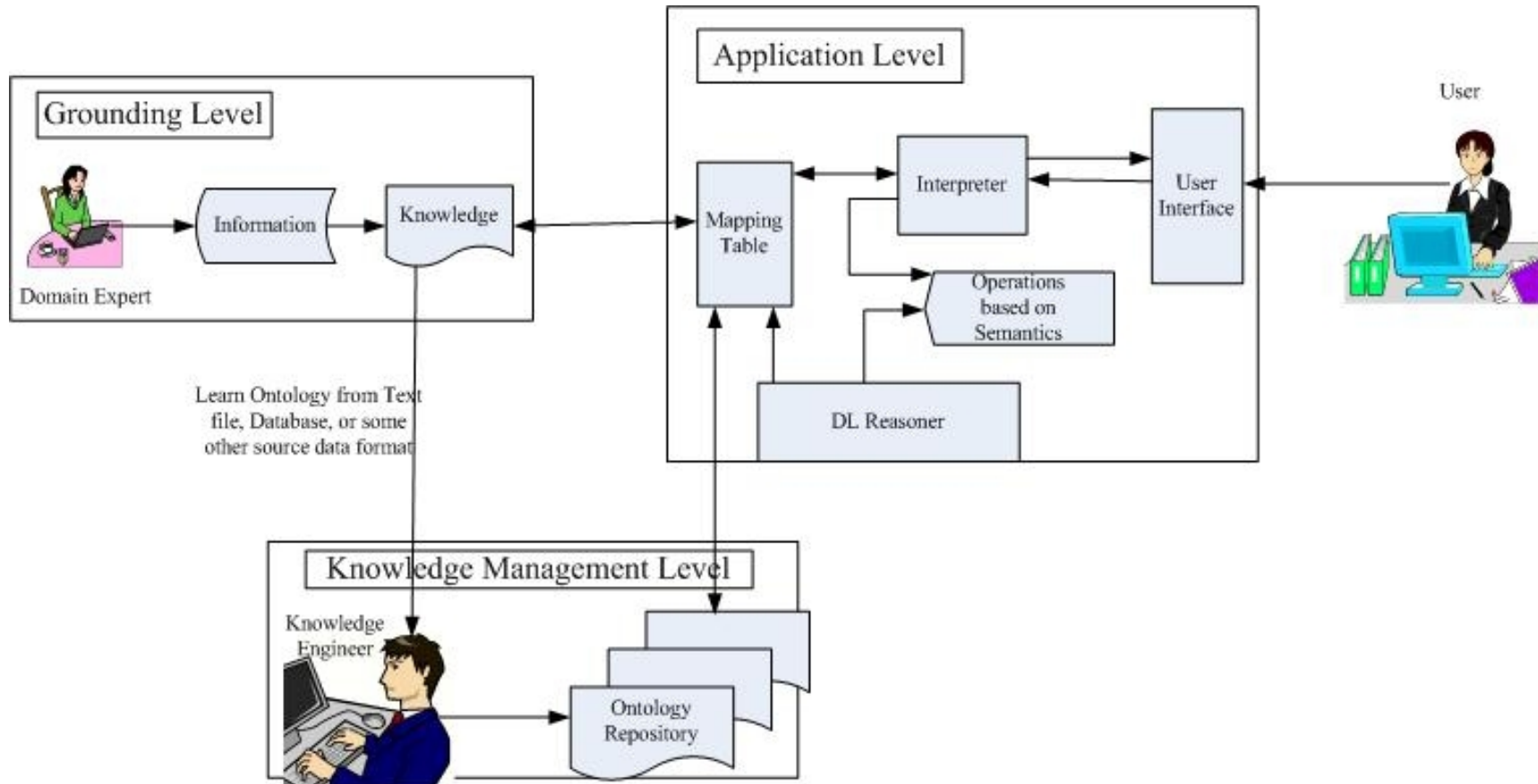
知识管理 KMSphere



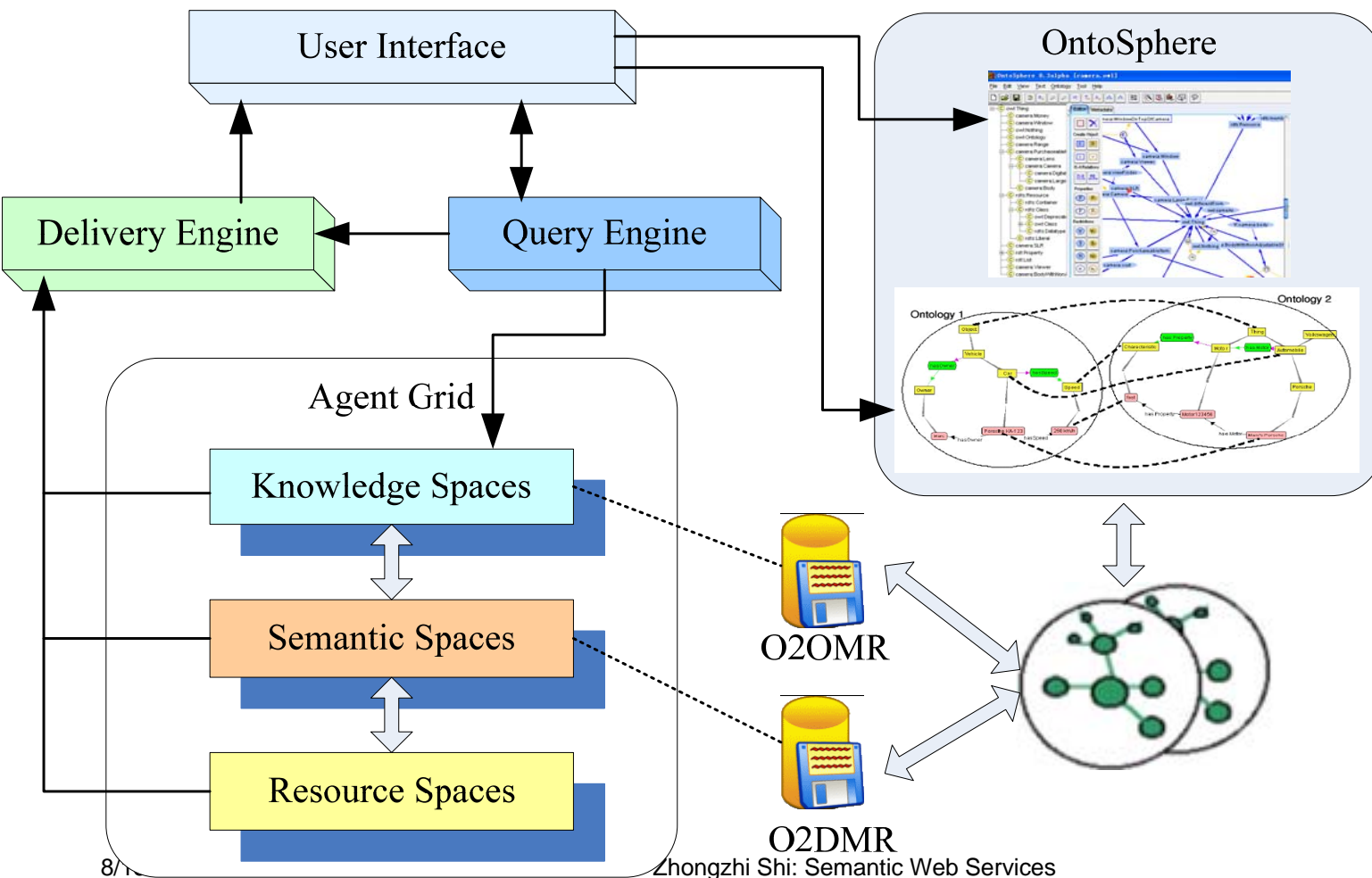
中科院计算所
INSTITUTE OF COMPUTING
TECHNOLOGY



KMSphere 系统结构



KMSphere workflow



KMSphere Demo



中科院计算所
INSTITUTE OF COMPUTING
TECHNOLOGY

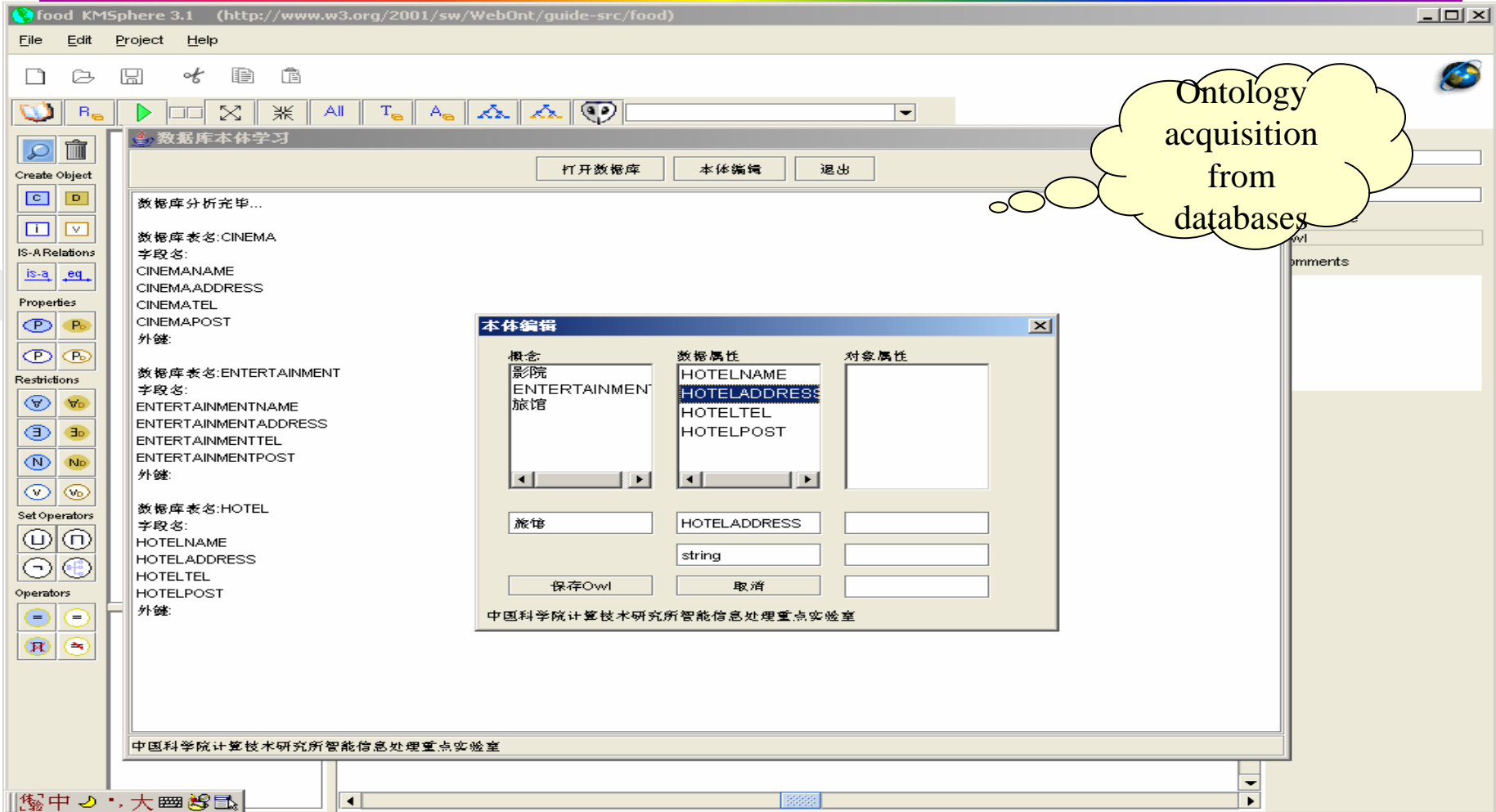
The screenshot shows the KMSphere 3.1 software interface. The main workspace displays an ontology diagram with the following structure:

- Thing** (Root Class)
 - Food** (Subclass)
 - Dessert** (Subclass)
 - Cake** (Instance)
 - Drink** (Subclass)
 - Wine** (Instance)

A yellow thought bubble is positioned near the **Wine** instance, containing the text: "Create ontology by hand".

The interface includes a menu bar (File, Edit, Project, Help), a toolbar with various icons, and a left sidebar with sections for "Create Object", "IS-A Relations", "Properties", "Restrictions", and "Operators". The right sidebar contains a "Class Name" field (set to "Wine"), "Label", "Namespace" (set to "<default>"), and "Comments" fields.

KMSphere Demo



Food KMSphere 3.1 (http://www.w3.org/2001/sw/WebOnt/guide-src/food)

File Edit Project Help

数据库本体学习

打开数据库 本体编辑 退出

数据库分析完毕...

数据库表名: CINEMA
字段名:
CINEMANAME
CINEMAADDRESS
CINEMATEL
CINEMAPOST
外键:

数据库表名: ENTERTAINMENT
字段名:
ENTERTAINMENTNAME
ENTERTAINMENTADDRESS
ENTERTAINMENTTEL
ENTERTAINMENTPOST
外键:

数据库表名: HOTEL
字段名:
HOTELNAME
HOTELADDRESS
HOTELTEL
HOTELPOST
外键:

本体编辑

概念	数据属性	对象属性
影院 ENTERTAINMENT 旅馆	HOTELNAME HOTELADDRESS HOTELTEL HOTELPOST	

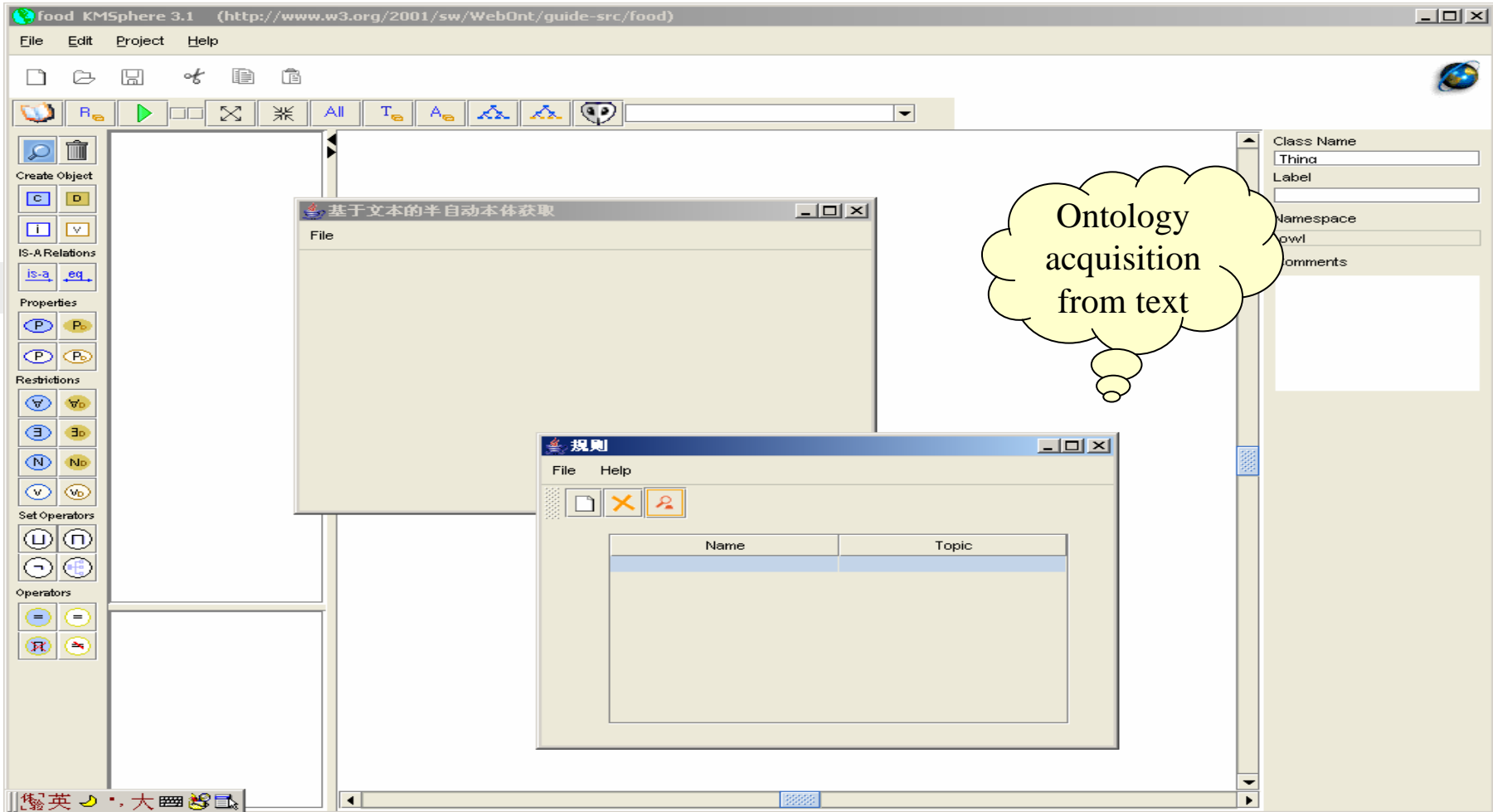
旅馆 HOTELADDRESS string

保存Owl 取消

中国科学院计算技术研究所智能信息处理重点实验室

Ontology acquisition from databases

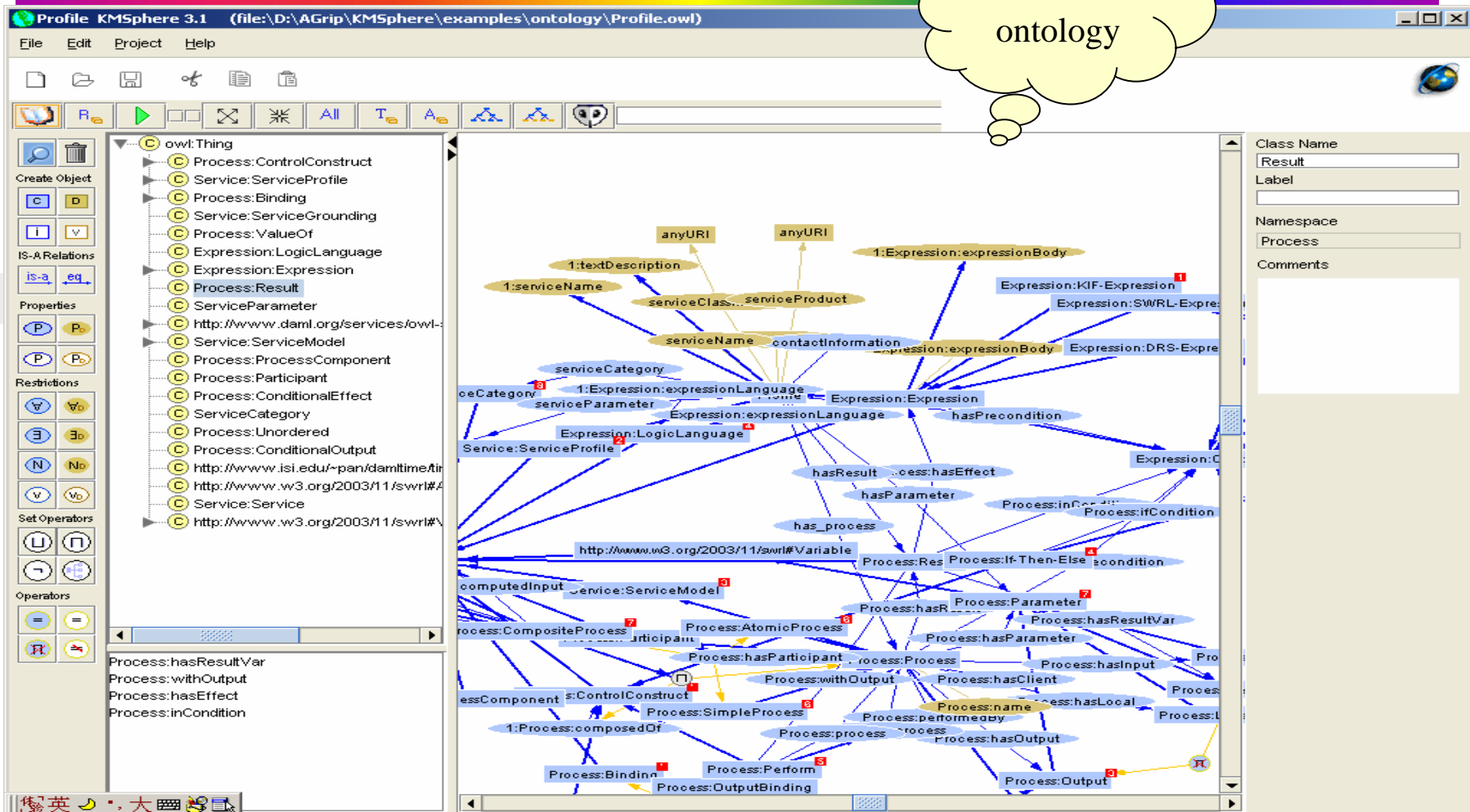
KMSphere Demo



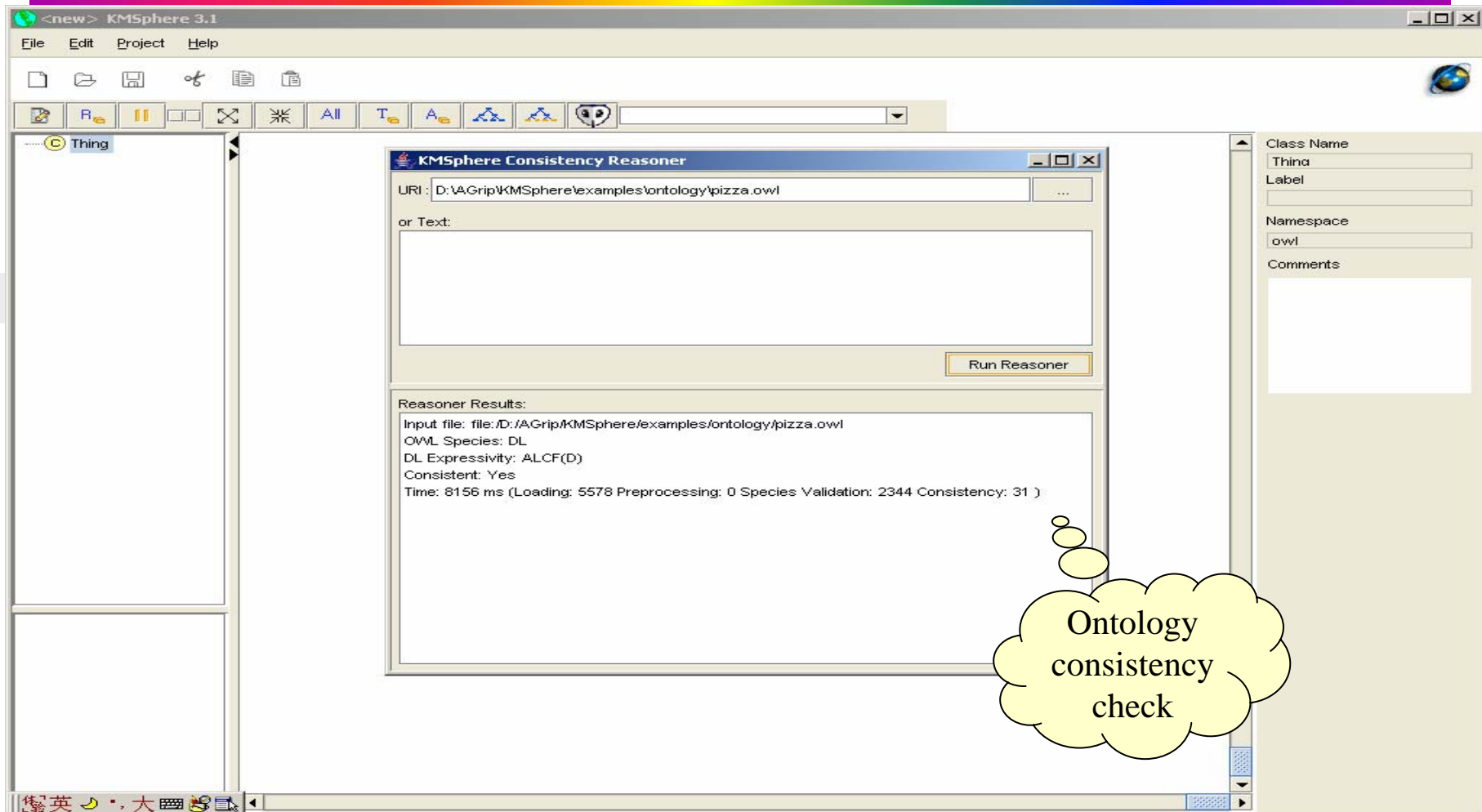
The screenshot shows the KMSphere 3.1 application window. The title bar reads "food KMSphere 3.1 (http://www.w3.org/2001/sw/WebOnt/guide-src/food)". The menu bar includes "File", "Edit", "Project", and "Help". The toolbar contains various icons for file operations and editing. On the left, there is a vertical toolbar with sections for "Create Object", "IS-A Relations", "Properties", "Restrictions", "Set Operators", and "Operators". The main workspace is divided into several panes. A central pane displays a window titled "基于文本的半自动本体获取" (Semi-automatic ontology acquisition from text) with a "File" field. Below it, a "规则" (Rule) window is open, showing a table with columns "Name" and "Topic". A yellow thought bubble with the text "Ontology acquisition from text" is positioned over the central workspace. On the right, a vertical pane shows a "Class Name" field with "Thing" entered, and other fields for "Label", "Namespace", "owl", and "Comments". The Windows taskbar at the bottom shows the system tray with a clock and other icons.

KMSphere Demo

Edit ontology



KMSphere Demo



The screenshot displays the KMSphere 3.1 application window. The main interface includes a menu bar (File, Edit, Project, Help), a toolbar with various icons, and a central workspace. A floating window titled "KMSphere Consistency Reasoner" is open, showing the following details:

- URI: D:\AGrip\KMSphere\examples\ontology\pizza.owl
- or Text: (empty text area)
- Run Reasoner button
- Reasoner Results:
 - Input file: file:D:/AGrip/KMSphere/examples/ontology/pizza.owl
 - OWL Species: DL
 - DL Expressivity: ALCF(D)
 - Consistent: Yes
 - Time: 8156 ms (Loading: 5578 Preprocessing: 0 Species Validation: 2344 Consistency: 31)

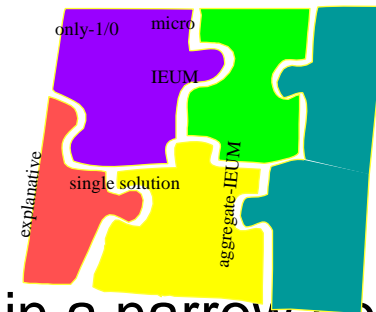
On the right side of the main window, there is a panel for class information with fields for Class Name (Thing), Label, Namespace (owl), and Comments. A yellow thought bubble with the text "Ontology consistency check" is positioned in the lower right area of the screenshot.

内容提要

- 引言
- 动态描述逻辑
- **Web**服务的语义标识
- 自动**Web**服务发现
- 自动**Web**服务组合
- 语义**Web**服务平台**SWSBroker**
- 结束语

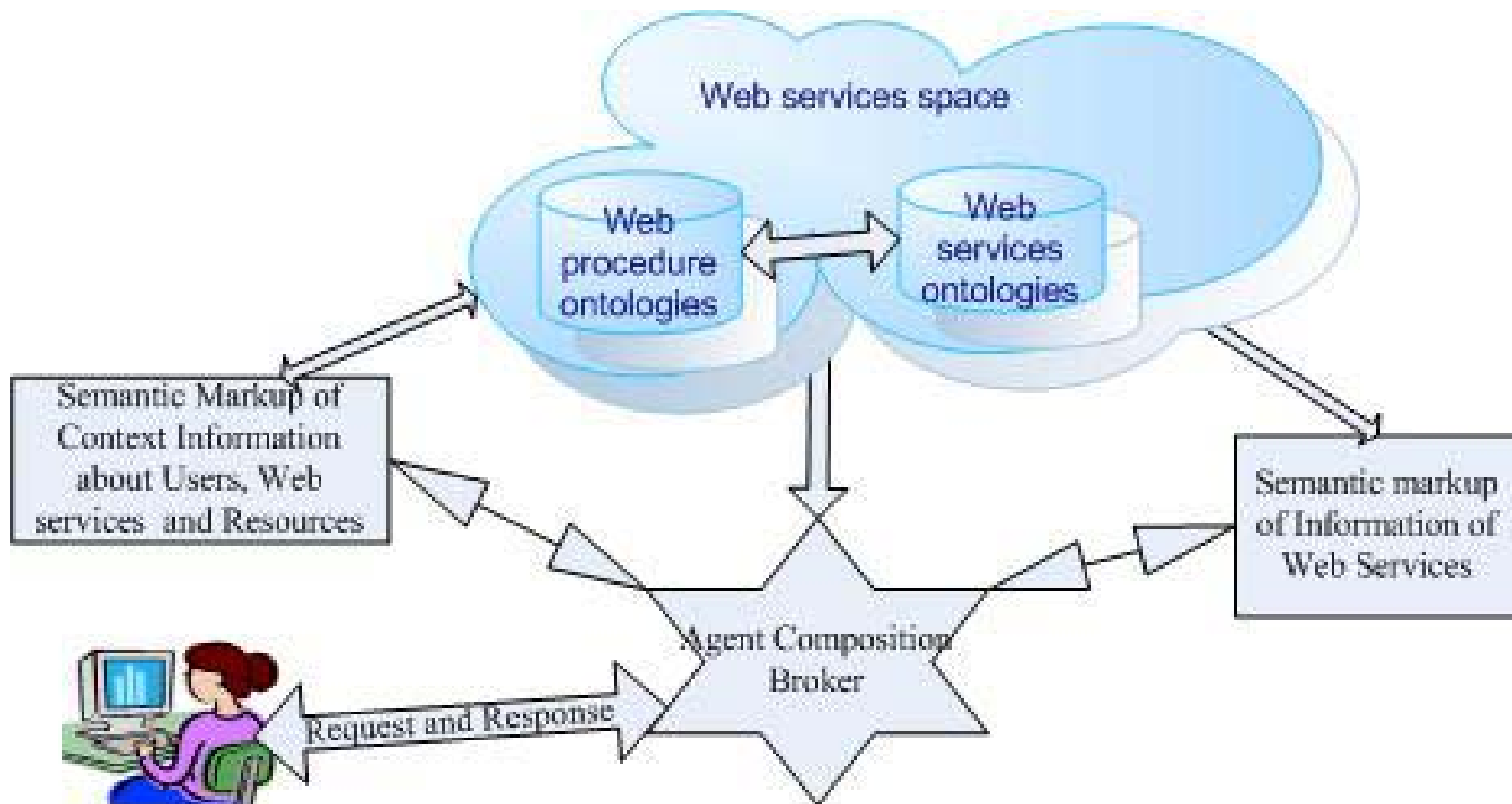
服务组合 Services Composition

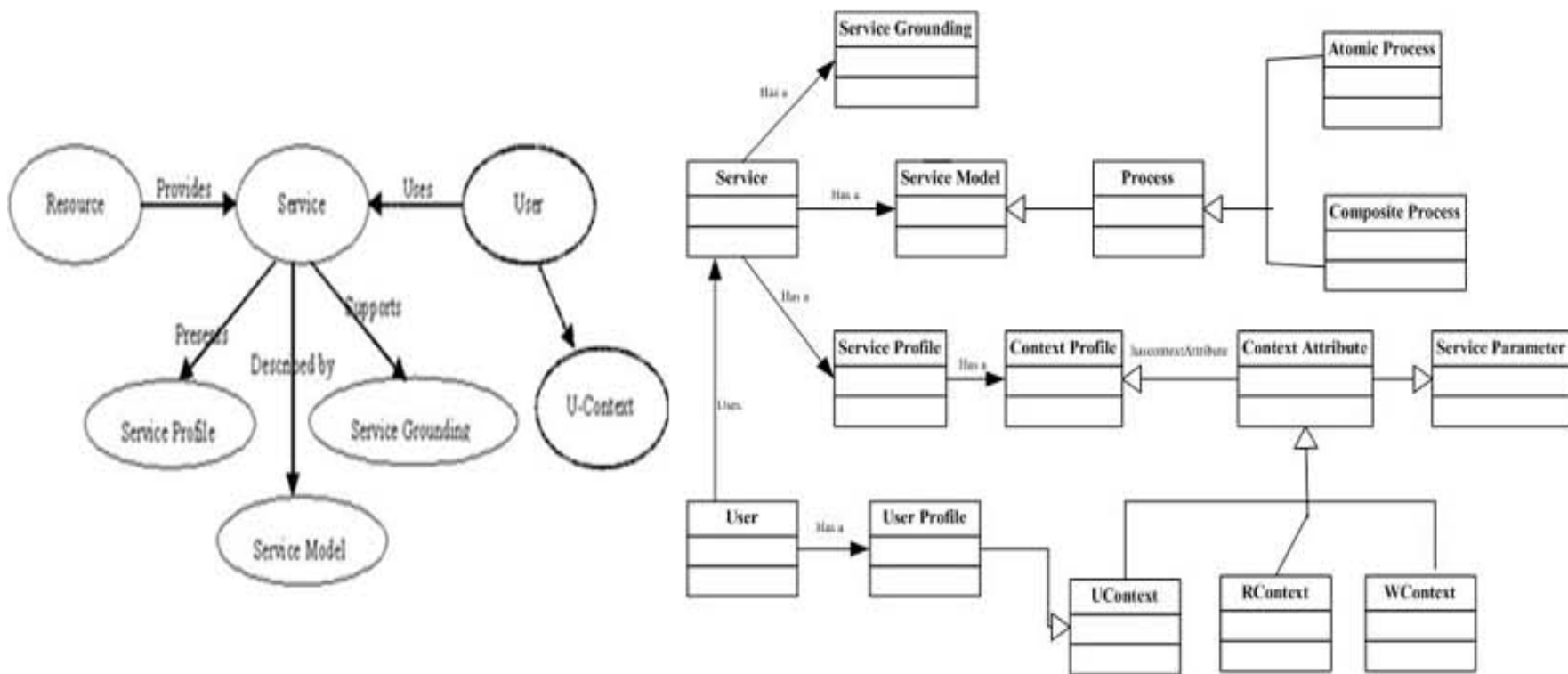
- Planning based approaches:
 - construct a plan from elementary services to obtain a required functionality.
 - reasoning based only on component specifications
 - plan built every time from scratch
- Knowledge based approaches:
 - re-use preconfigured templates
 - reasoning with specialized knowledge in a narrow domain
 - sophisticated domain knowledge is needed



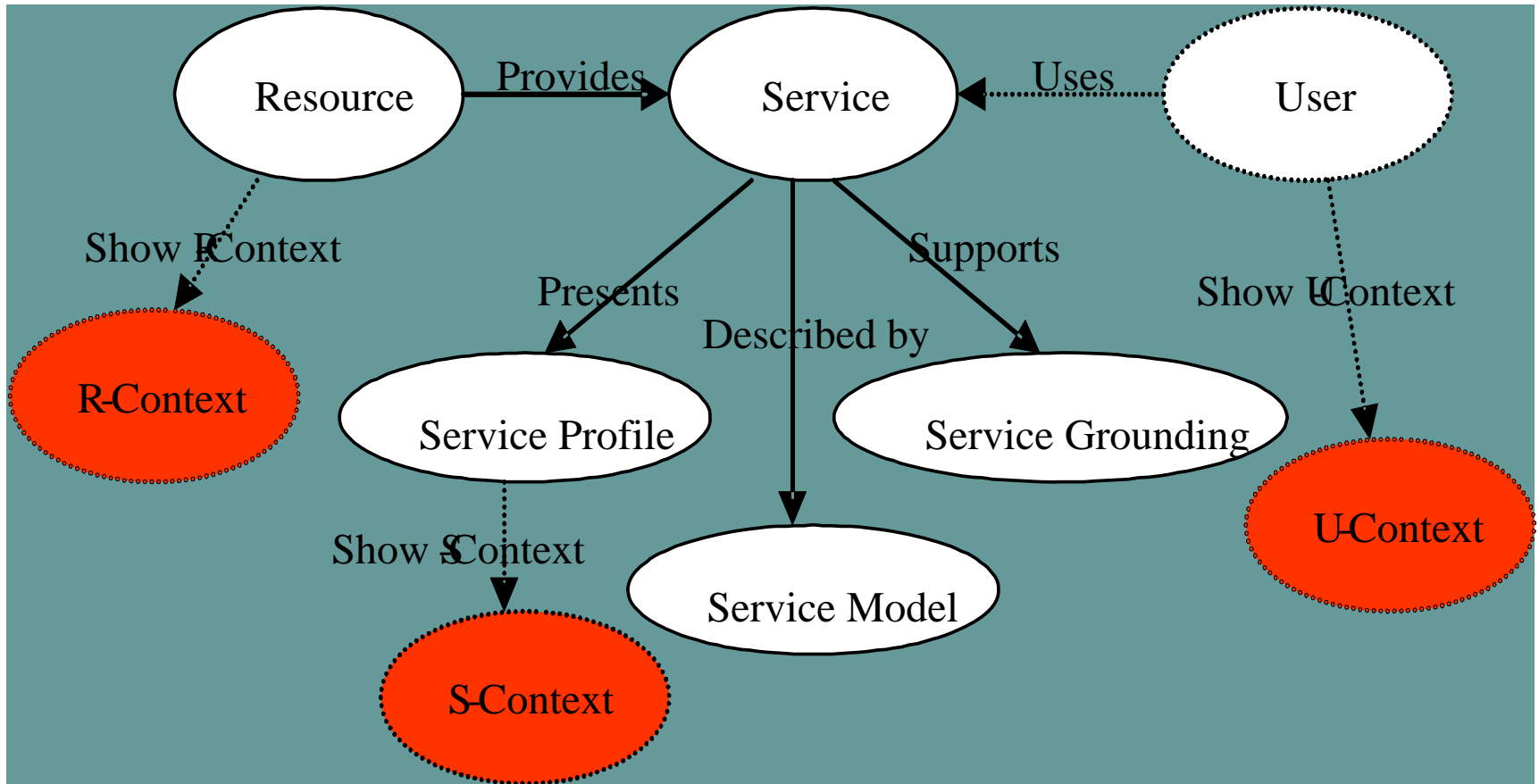
服务组合

Services Composition

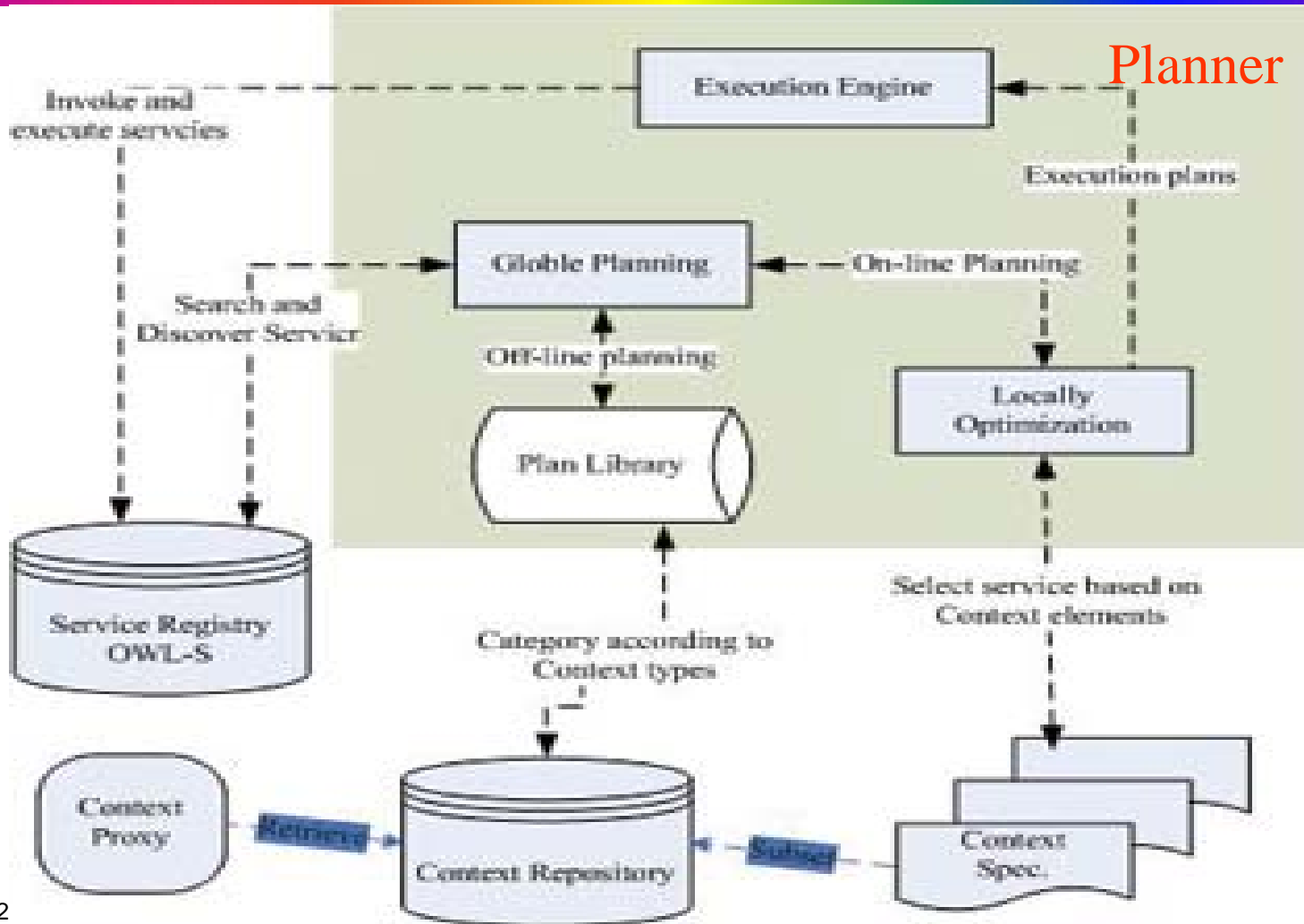




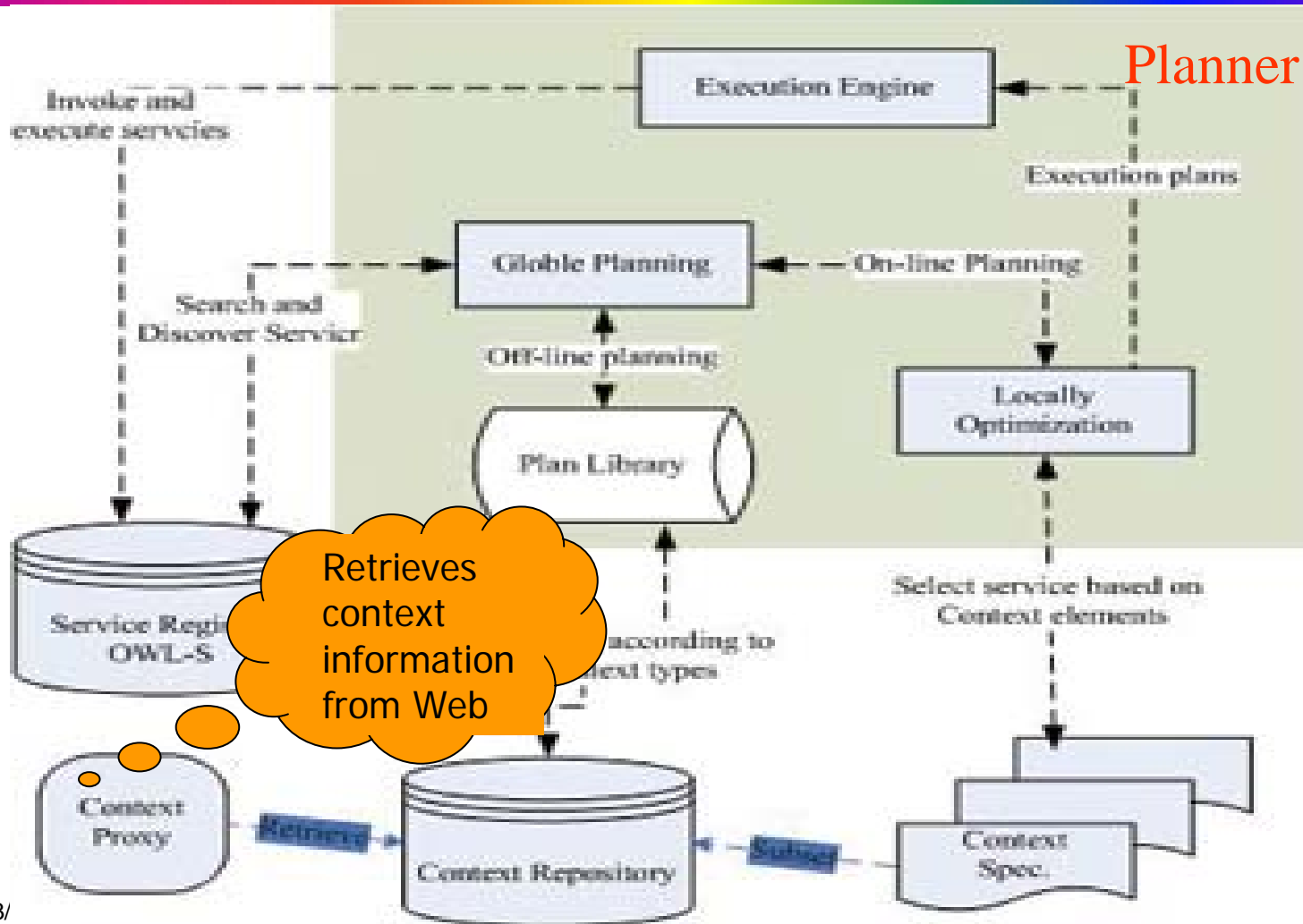
OWL-S 语境



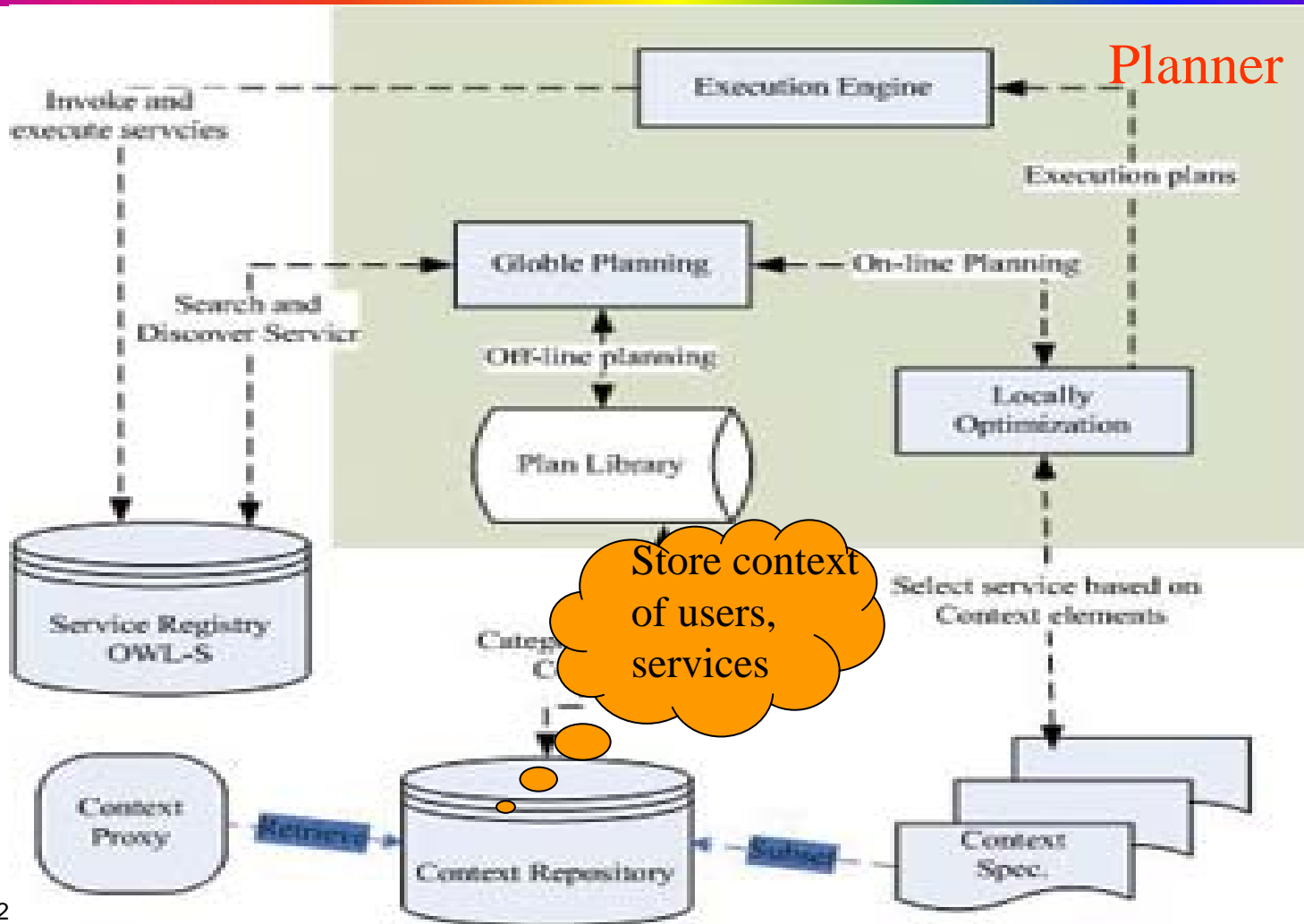
服务组合的语境规划



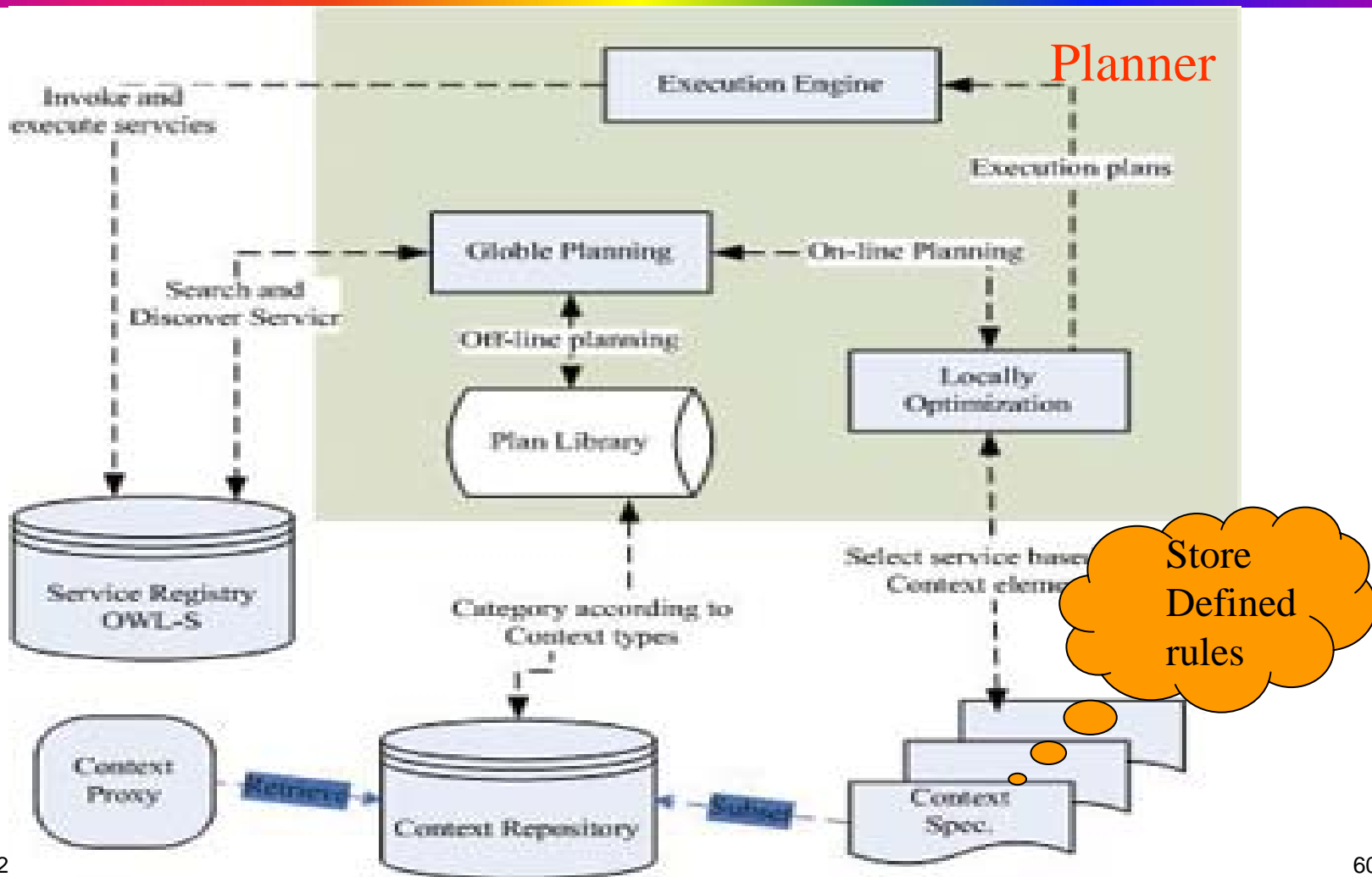
服务组合的语境规划



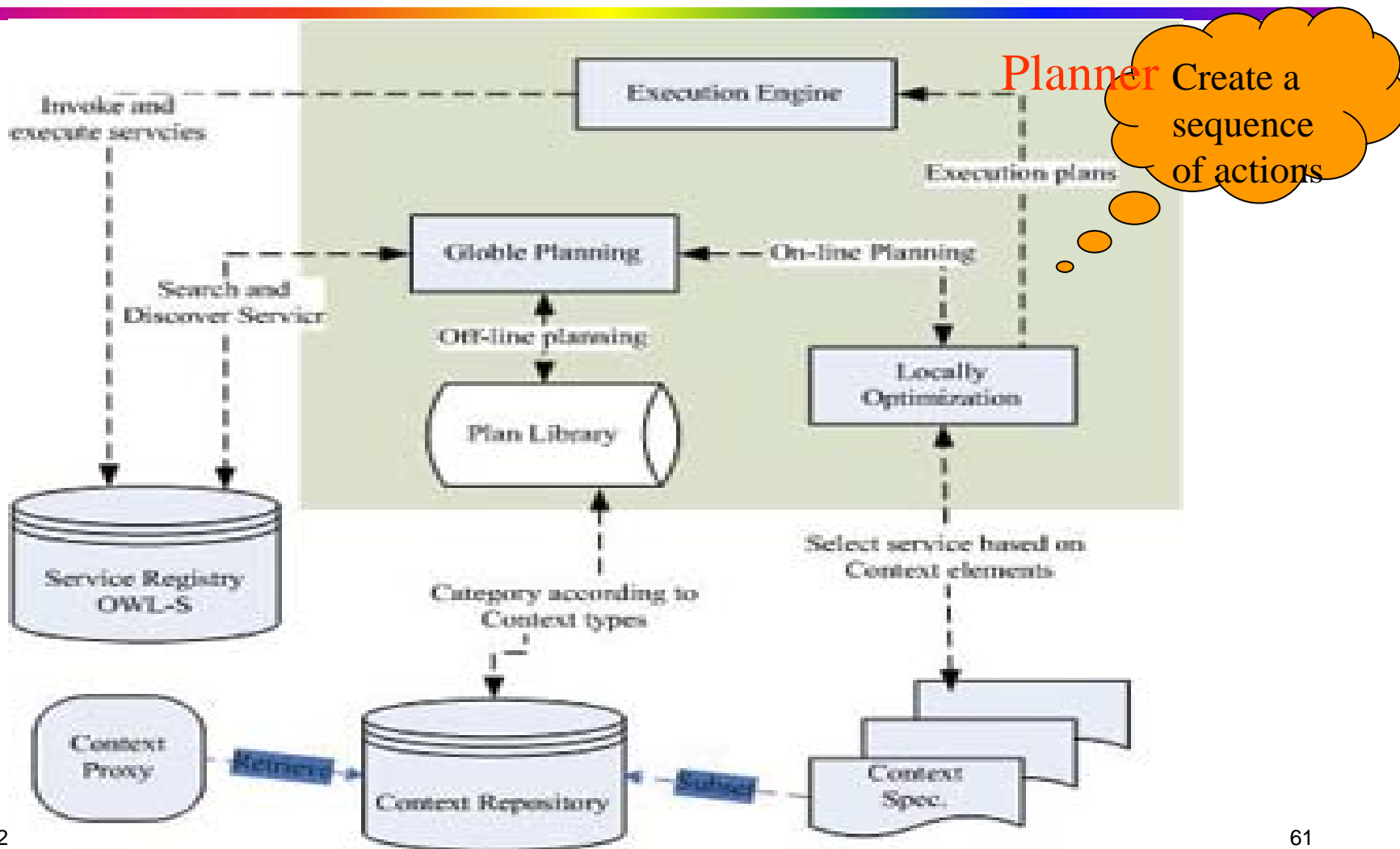
服务组合的语境规划



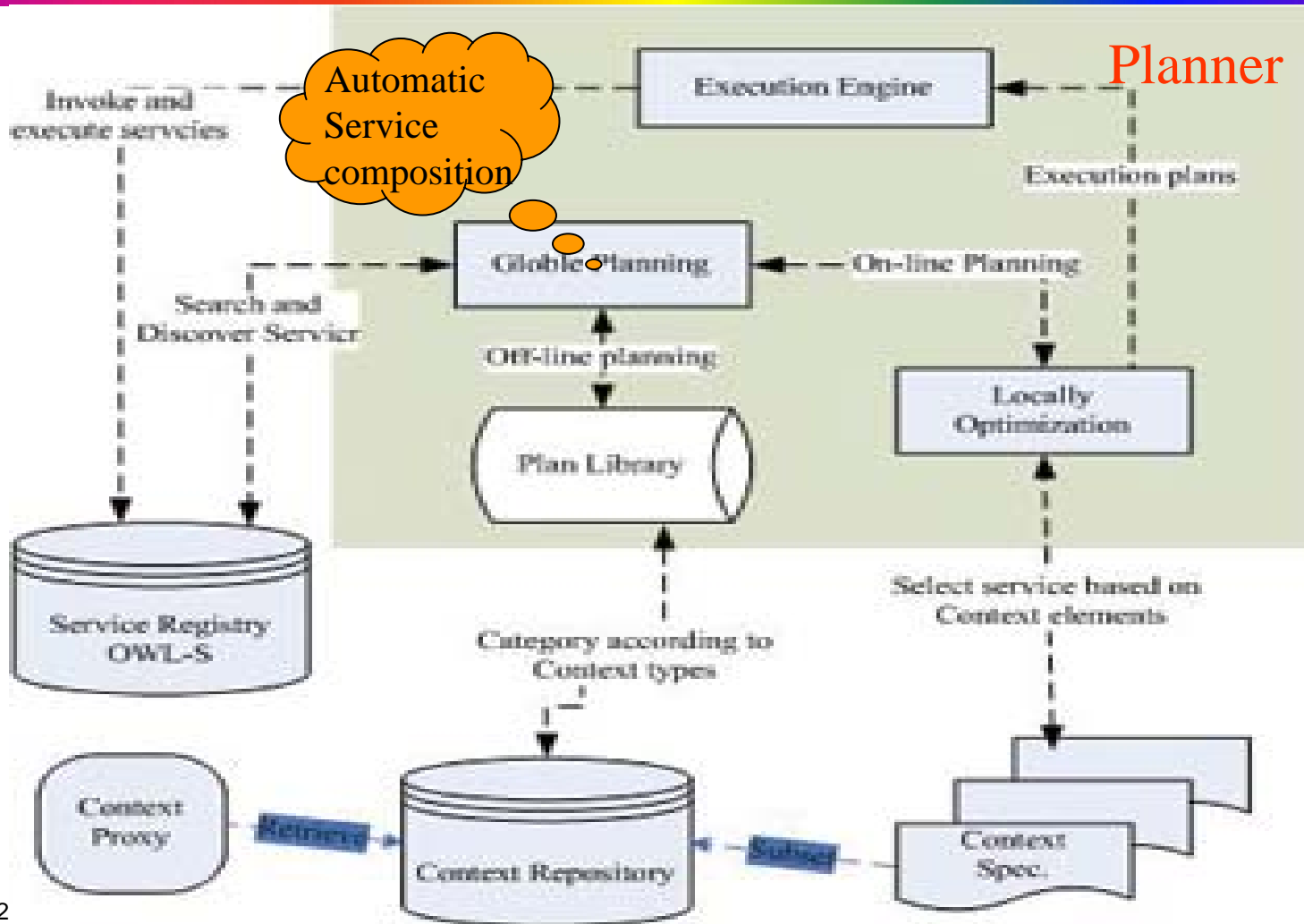
服务组合的语境规划



服务组合的语境规划



服务组合的语境规划



规划域

$$D = (S, A, C, I, \Gamma, \Upsilon)$$

- (1) S is the set of states.
- (2) A is the set of actions (services).
- (3) C is the set of context elements.
- (4) I is the set of initial states and $I \subseteq S$;
- (5) Γ is the transitional function, $S \times A \rightarrow 2^S$, which associates to each current state $s \in S$ and to each action $a \in A$ the set $\Gamma(s, a) \subseteq S$ of next states;
- (6) Υ is the context function, $S \times C \rightarrow A$, which associates to a state $s \in S$ and a context $c \in C$, an action $a \in A$ to be executed, $\Upsilon(s, c) \in A$.

规划任务

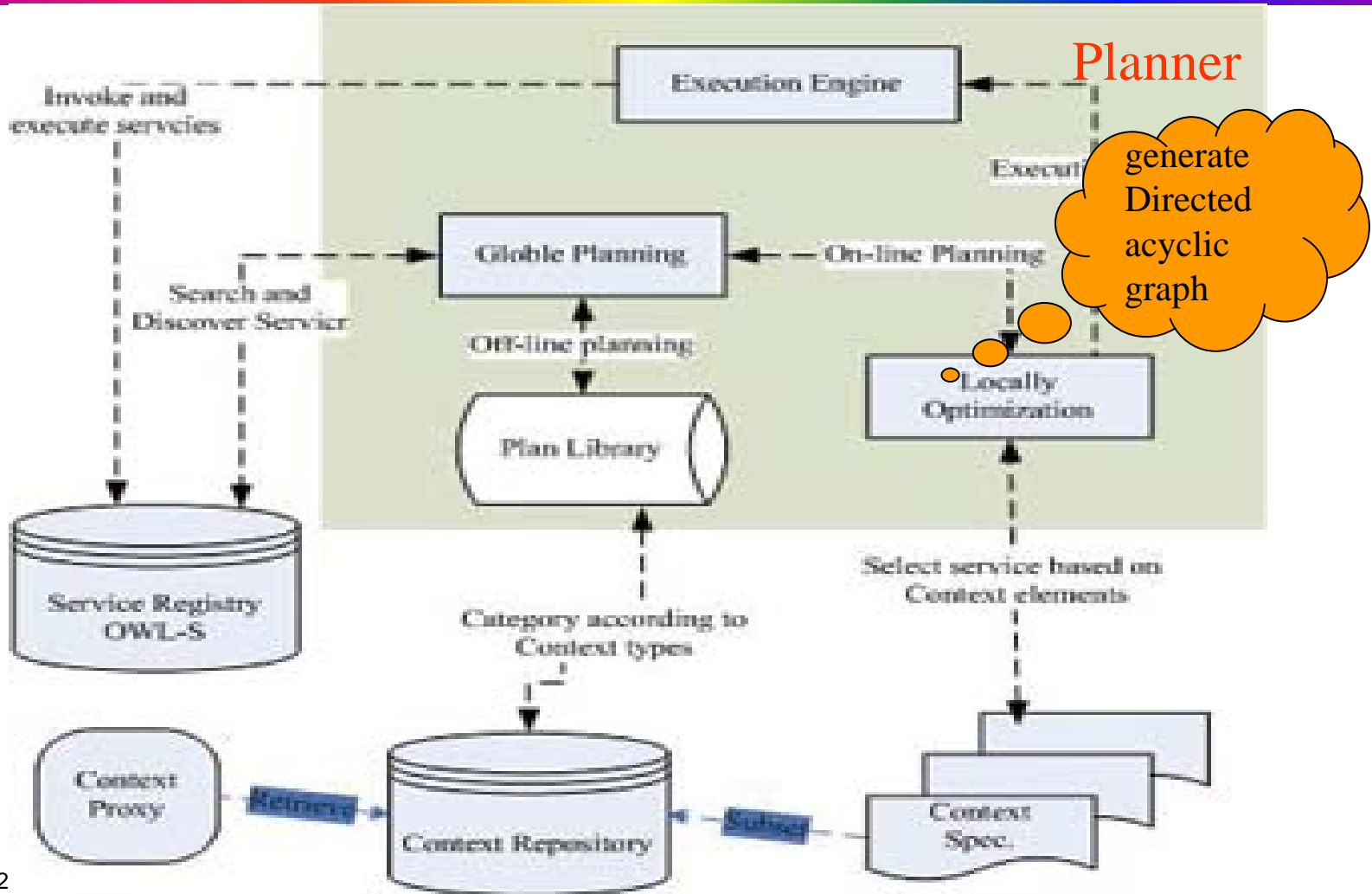
- $T=(S_1, S_n, D)$
- S_1 : initial states
- S_n : goal state
- D : Domain description

全局规划算法

Algorithm 1 Global Planning Algorithm

```
1: Given a planning Problem  $(S_1, S_n, D)$ 
2: Search the Plan Library in domain  $D$ 
3: if find the plan which initial states is  $S_1$  and goal states is  $S_n$  then
4:   return this plan
5: else
6:   if find a plan which goal states is  $S_n$  and initial state is  $S_i$  then
7:     execute another planning task  $(S_1, S_i, D)$ 
8:   else
9:     if find a plan which initial state is  $S_1$  and goal state is  $S_j$  then
10:      check the context consistency of  $S_j$  and  $S_n$ 
11:      execute another planning task  $(S_j, S_n, D)$ 
12:      return break
13:   end if
14: end if
15: end if
```

服务组合的语境规划



DAG 表示

$$\text{DAG } G = G(V, E)$$

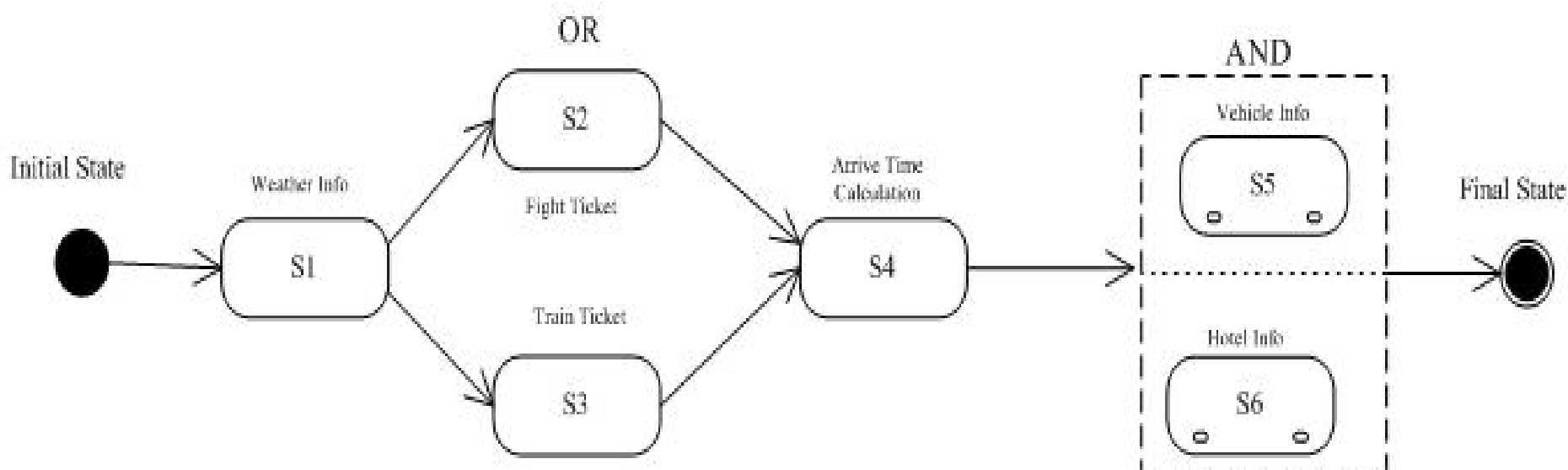
- (1) The DAG has at least two nodes Start and Finish.
- (2) $a_i = (a_{i1}, a_{i2}, \dots, a_{it})$ is the action set for state S_i in ST and $Eff(a_i) \supseteq S_i$
- (3) The DAG has one node for each action (a_1, a_2, \dots, a_n) .
- (4) The DAG contains an edge from action a_i to action a_j iff S_j is a direct successor of S_i .
- (5) The edge from a_i to a_j is weighed by the context elements $c \in C$ and $\Upsilon(S_i, c) \rightarrow a_i$.

局部优化算法

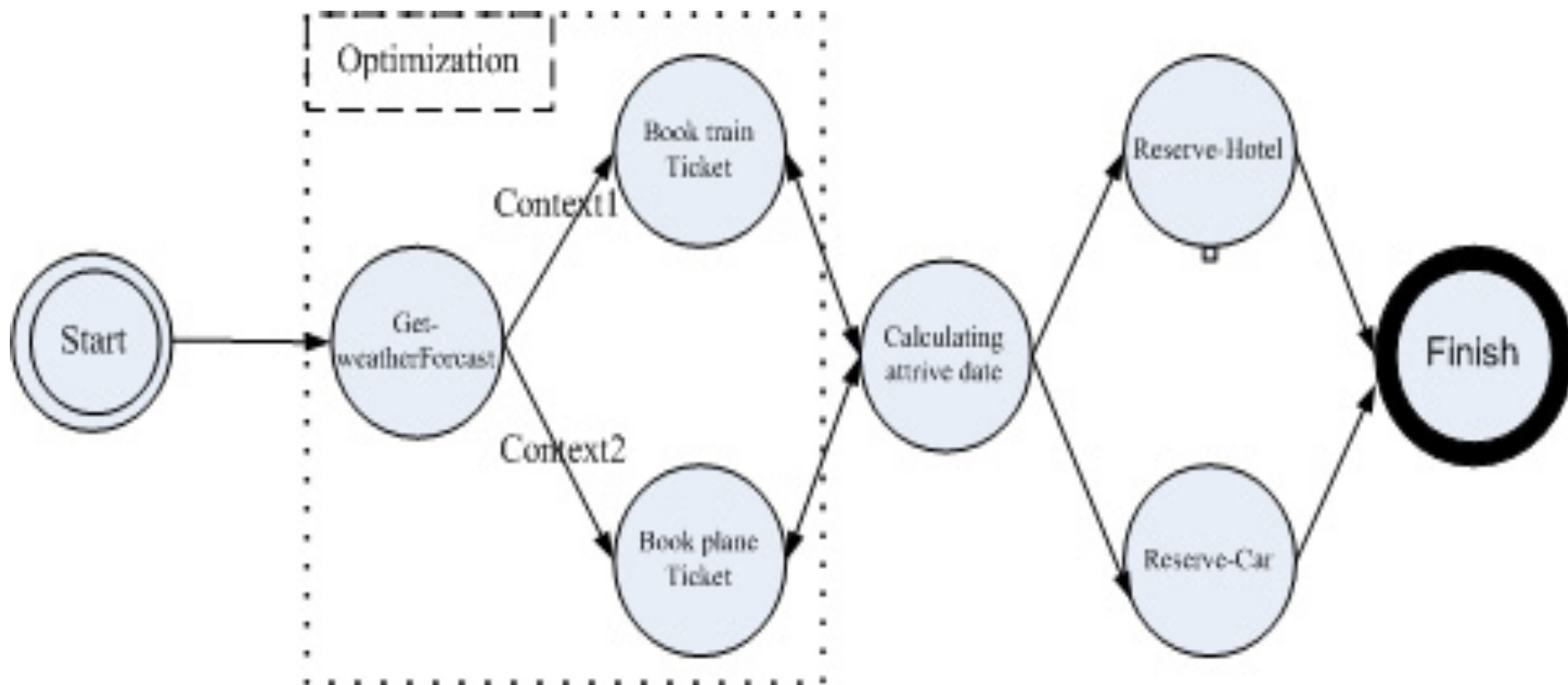
Algorithm 2 Local Optimization Algorithm

```
1: Given a statechart ST [ $S_1, S_2, \dots, S_n$ ],
2:  $RestofST \leftarrow ST$ ,
3: Generate two nodes  $Start$  and  $Finish$ 
4:  $Pre(Start) \in I \cap Eff(Start) = S_1$  and  $Eff(finish) = \emptyset$ 
5: Link  $Start$  to  $Finish$ , stored in  $DAGCurrentPath, DAGDBPath$ 
6: while  $RestofST \neq \emptyset$  do
7:   if the first state in ST isn't OR-States then
8:      $CS \leftarrow getFirstState(RestofST)$ 
9:      $CA \leftarrow GetBestCandidate(CS)$ 
10:     $RestofST = ST - CS$ 
11:    link  $CA$  to  $DAGCurrentPath$ 
12:  else
13:    copy  $DAGDBPath \leftarrow DAGCurrentPath$ 
14:     $CT \leftarrow findRelatedContextTypes(c)$ 
15:     $OT \leftarrow orderContextTypes(CT)$ 
16:    for all  $c \in OT$  do
17:       $T \leftarrow getRelevantSelection(r)$ 
18:    end for add OR1 part to  $DAGCurrentPath1$  add OR2 part to
       $DaGCurrentPath2$ 
19:    add OR2 part to  $DAGCurrentPath2$ 
20:  end if
21: end while
```

任务状态流



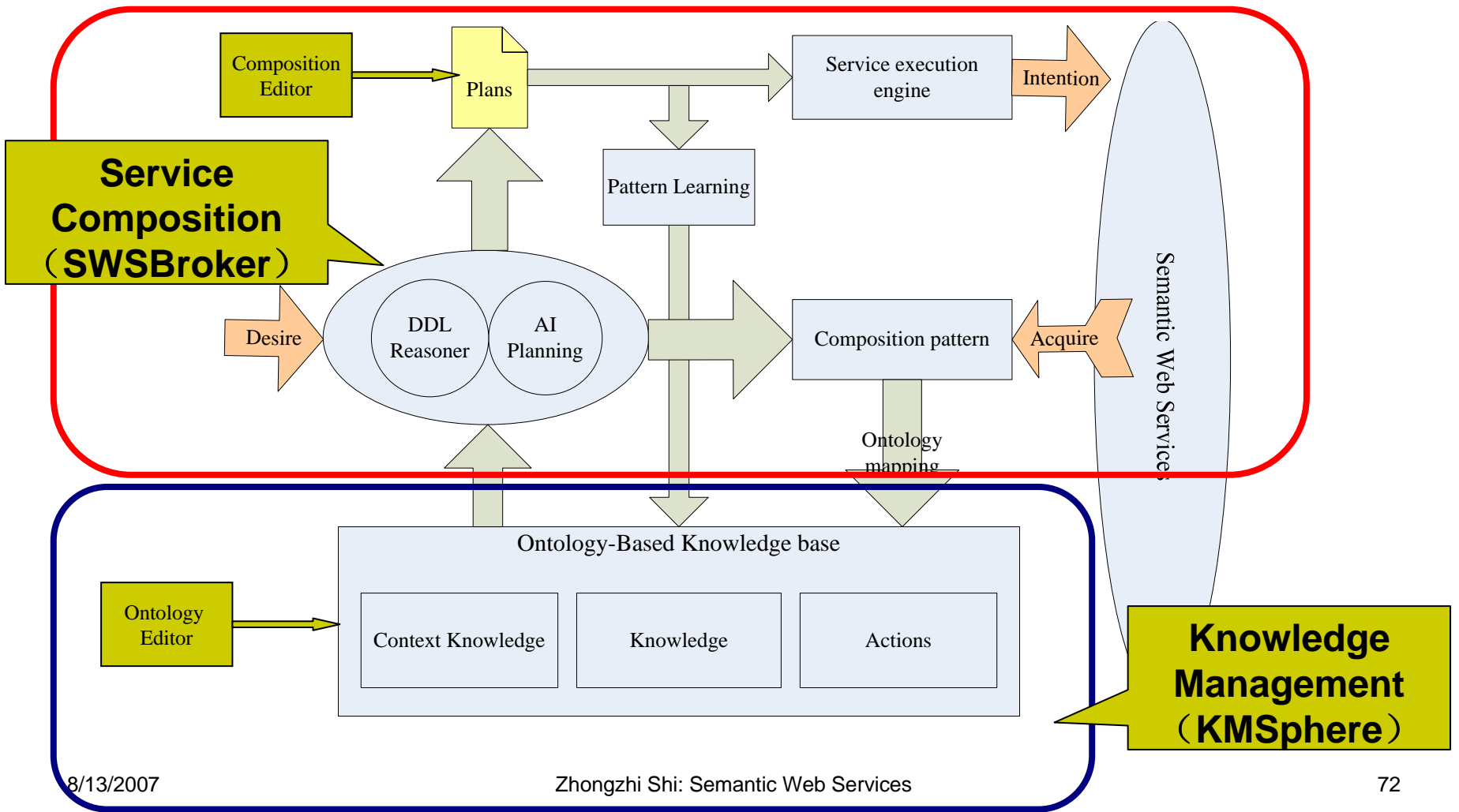
DAG表示的最终执行路径



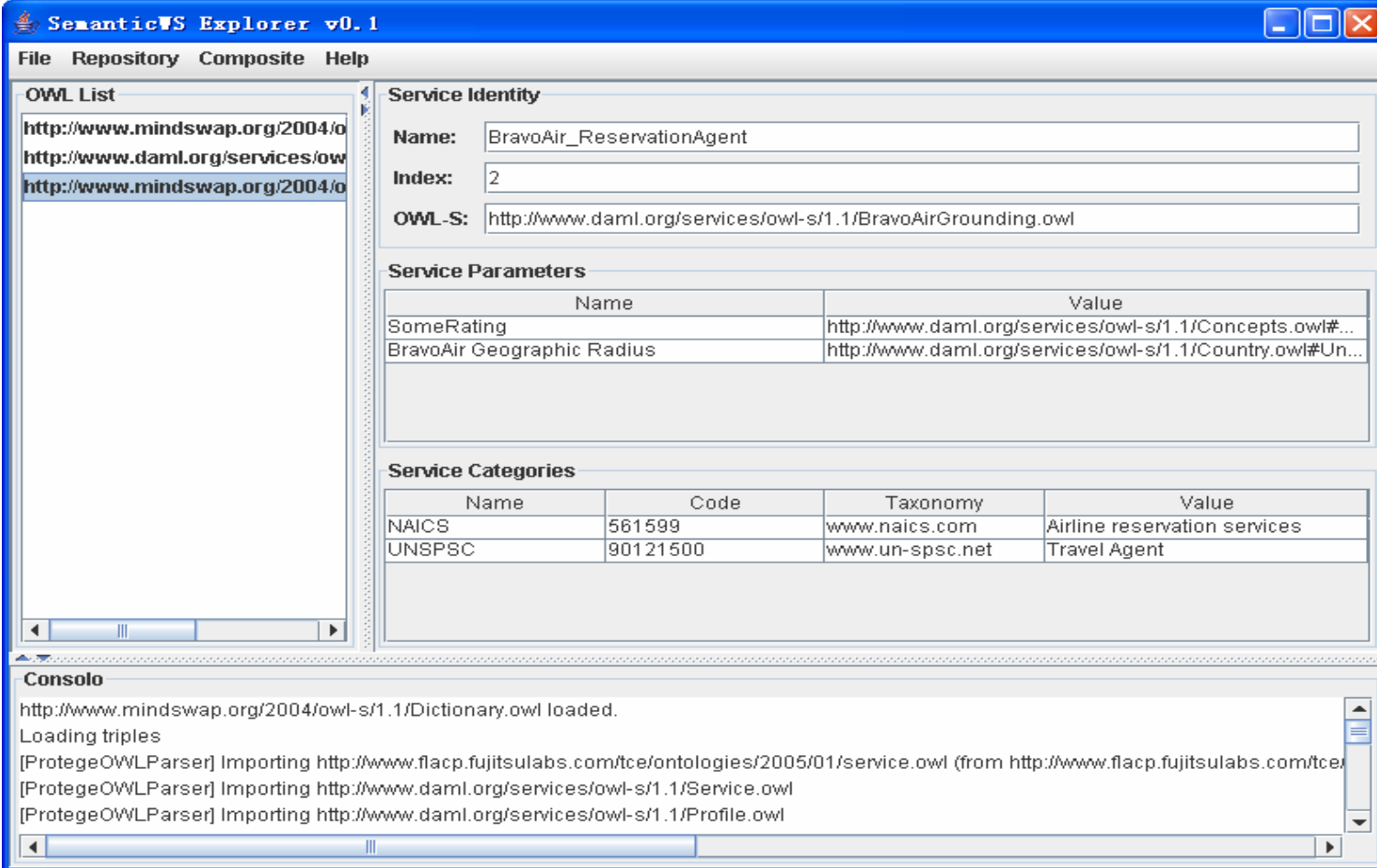
内容提要

- 引言
- 动态描述逻辑
- **Web**服务的语义标识
- 自动**Web**服务发现
- 自动**Web**服务组合
- 语义**Web**服务平台**SWSBroker**
- 结束语

语义Web服务平台 SWSBroker



SWSBroker 界面



SemanticWS Explorer v0.1
File Repository Composite Help

OWL List

- <http://www.mindswap.org/2004/owl-s/>
- <http://www.daml.org/services/owl-s/>
- <http://www.mindswap.org/2004/owl-s/>

Service Identity

Name:

Index:

OWL-S:

Service Parameters

Name	Value
SomeRating	http://www.daml.org/services/owl-s/1.1/Concepts.owl#...
BravoAir Geographic Radius	http://www.daml.org/services/owl-s/1.1/Country.owl#Un...

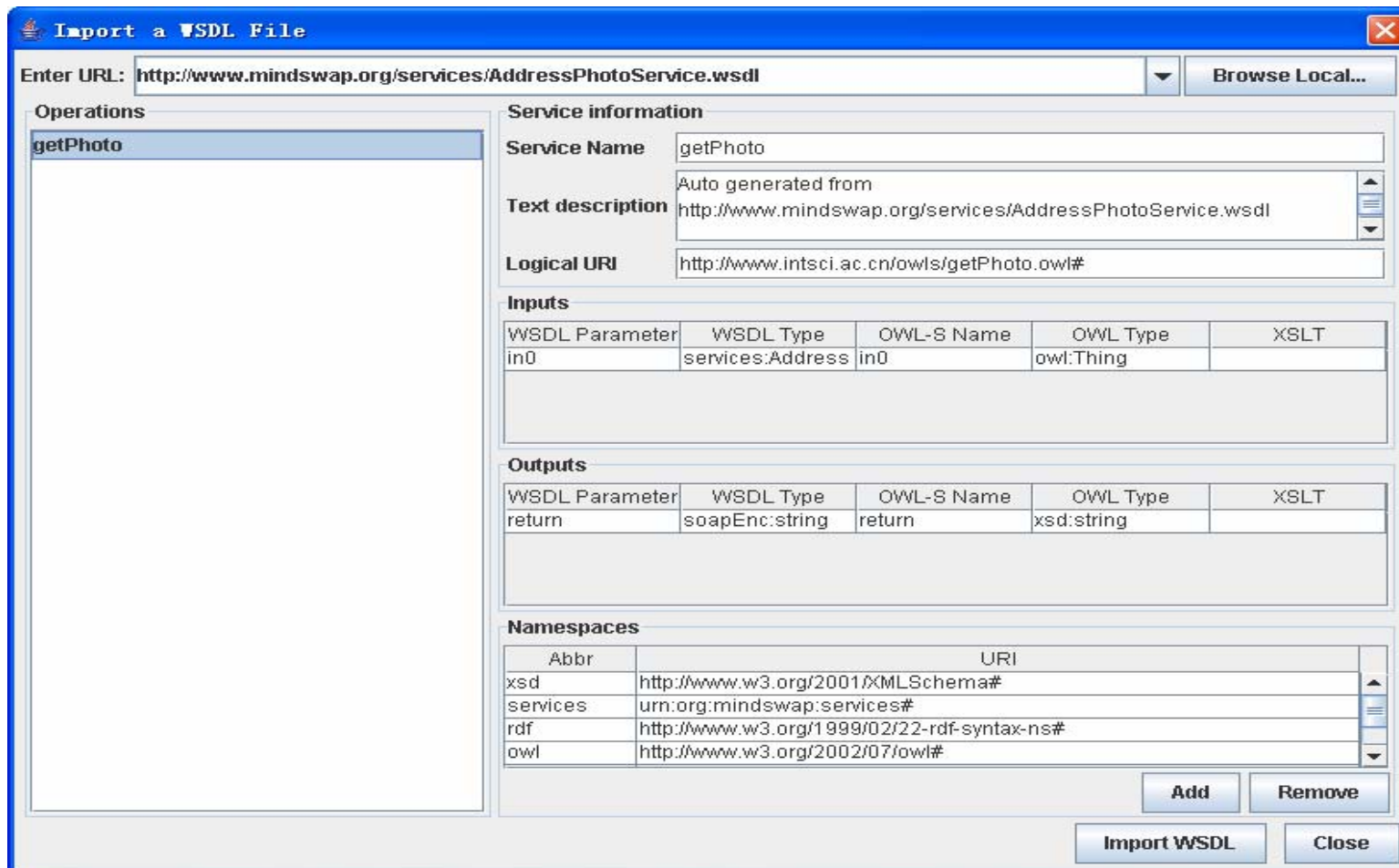
Service Categories

Name	Code	Taxonomy	Value
NAICS	561599	www.naics.com	Airline reservation services
UNSPSC	90121500	www.un-spsc.net	Travel Agent

Console

```
http://www.mindswap.org/2004/owl-s/1.1/Dictionary.owl loaded.  
Loading triples  
[ProtegeOWLParser] Importing http://www.flacp.fujitsulabs.com/tce/ontologies/2005/01/service.owl (from http://www.flacp.fujitsulabs.com/tce/...  
[ProtegeOWLParser] Importing http://www.daml.org/services/owl-s/1.1/Service.owl  
[ProtegeOWLParser] Importing http://www.daml.org/services/owl-s/1.1/Profile.owl
```

SWDL To OWL-S



The screenshot shows a software dialog box titled "Import a WSDL File". It contains the following sections:

- Enter URL:** A text field containing "http://www.mindswap.org/services/AddressPhotoService.wsdl" and a "Browse Local..." button.
- Operations:** A list box containing "getPhoto".
- Service information:**
 - Service Name:** getPhoto
 - Text description:** Auto generated from http://www.mindswap.org/services/AddressPhotoService.wsdl
 - Logical URI:** http://www.intsci.ac.cn/owl#getPhoto.owl#
- Inputs:** A table with columns: WSDL Parameter, WSDL Type, OWL-S Name, OWL Type, XSLT.

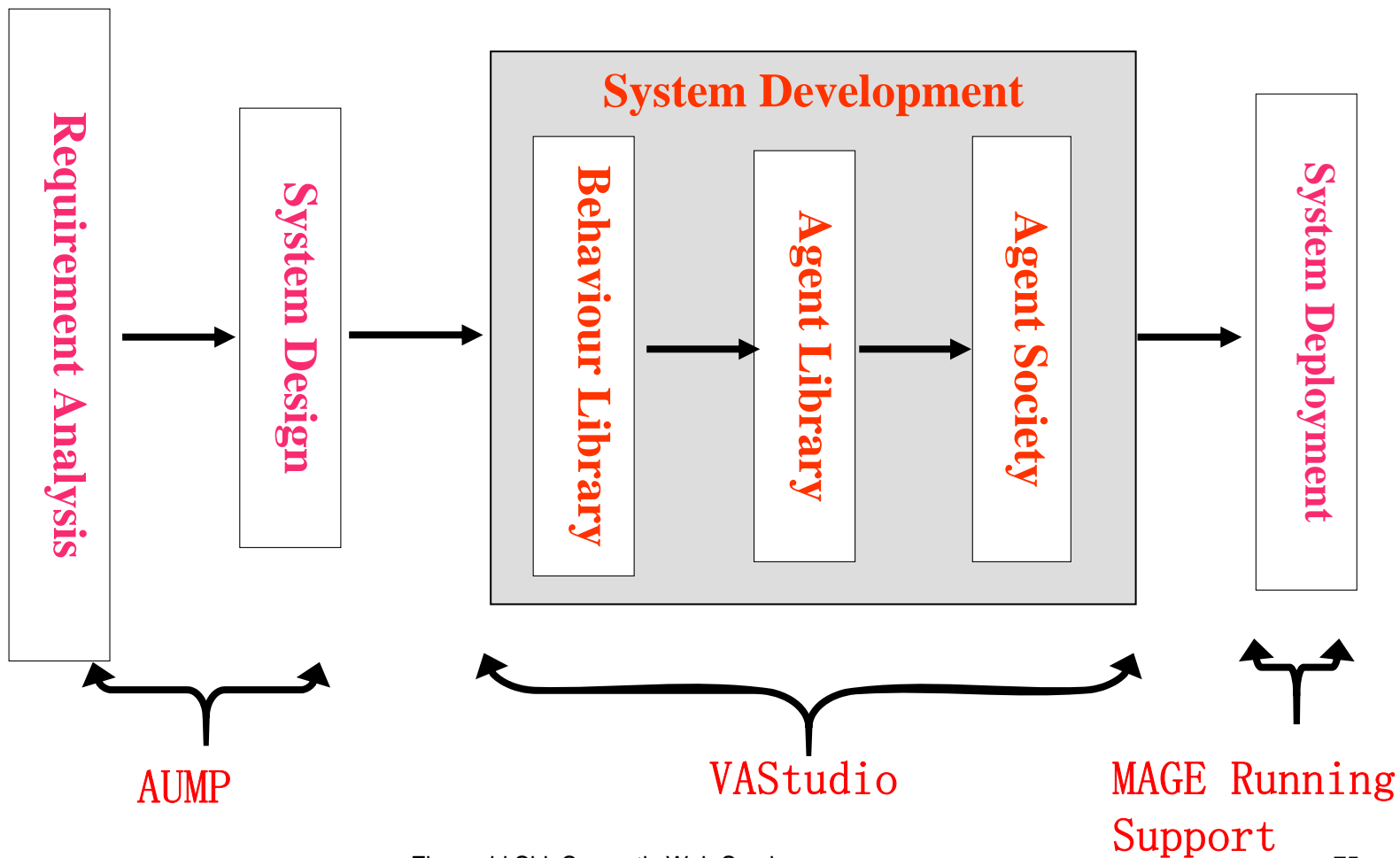
WSDL Parameter	WSDL Type	OWL-S Name	OWL Type	XSLT
in0	services:Address	in0	owl:Thing	
- Outputs:** A table with columns: WSDL Parameter, WSDL Type, OWL-S Name, OWL Type, XSLT.

WSDL Parameter	WSDL Type	OWL-S Name	OWL Type	XSLT
return	soapEnc:string	return	xsd:string	
- Namespaces:** A table with columns: Abbr, URI.

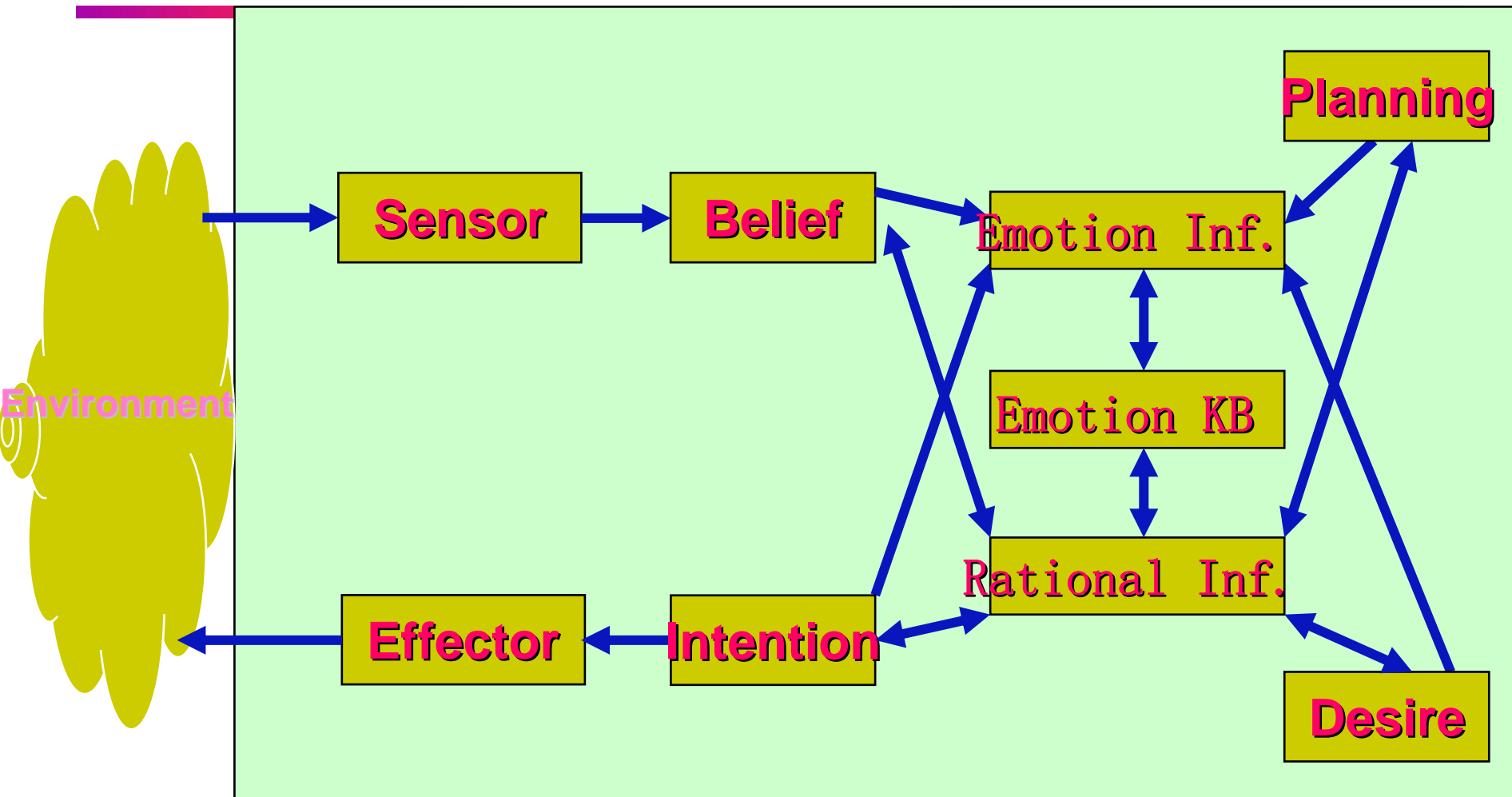
Abbr	URI
xsd	http://www.w3.org/2001/XMLSchema#
services	urn:org:mindswap:services#
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
owl	http://www.w3.org/2002/07/owl#

Buttons at the bottom: "Add", "Remove", "Import WSDL", and "Close".

多主体环境 MAGE

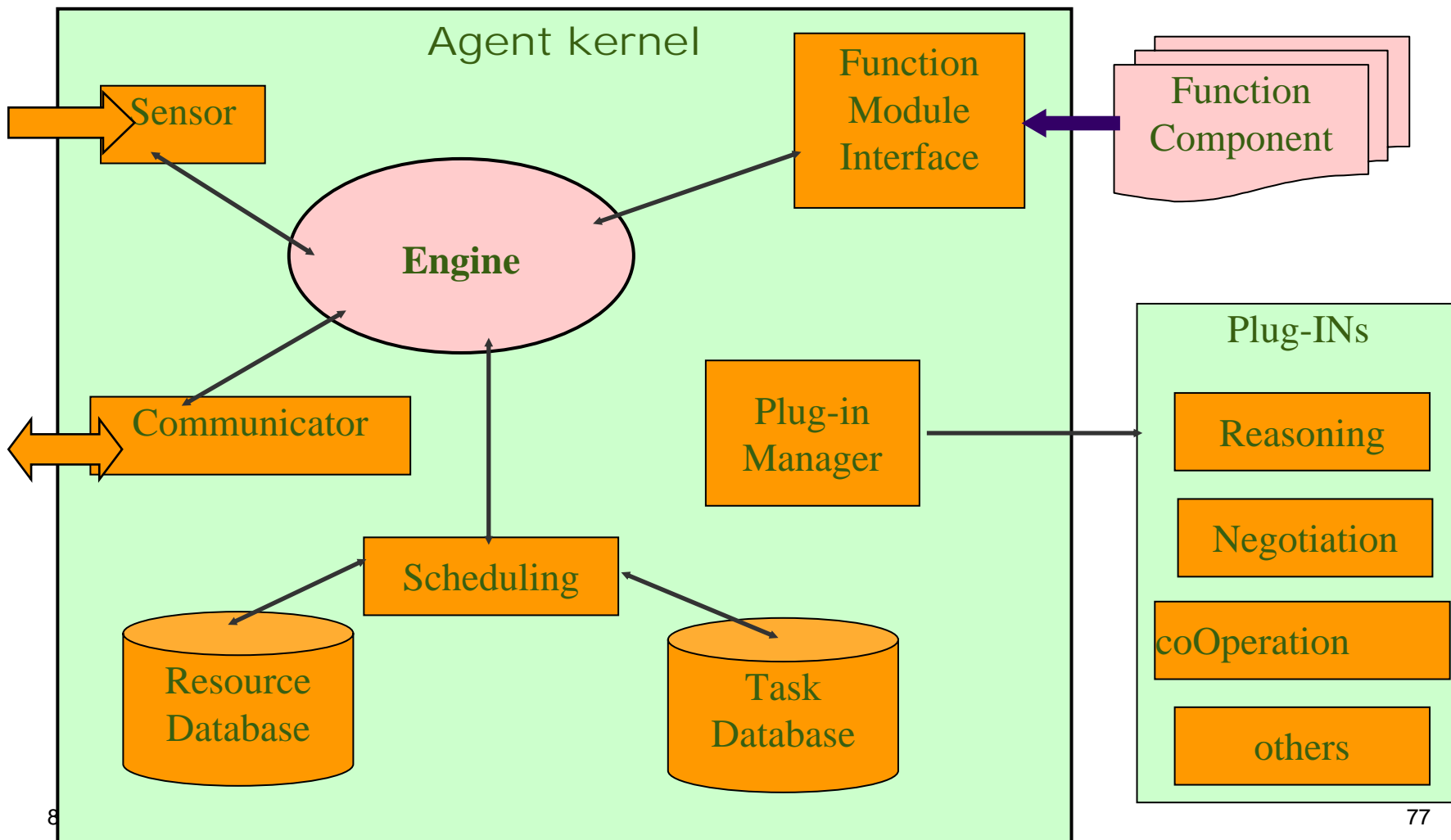


情感主体 Emotion Agent

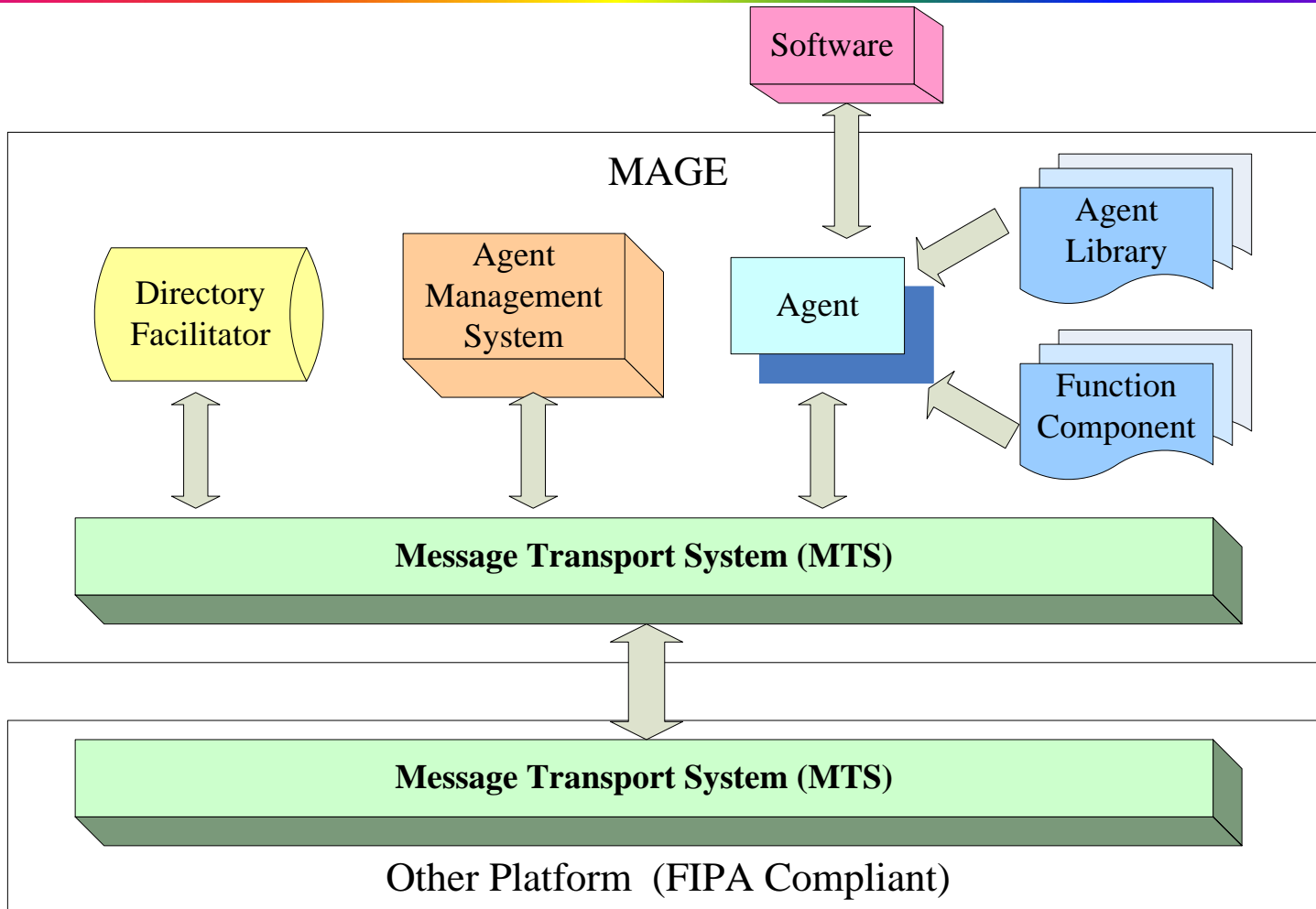


主体结构

Agent Architecture

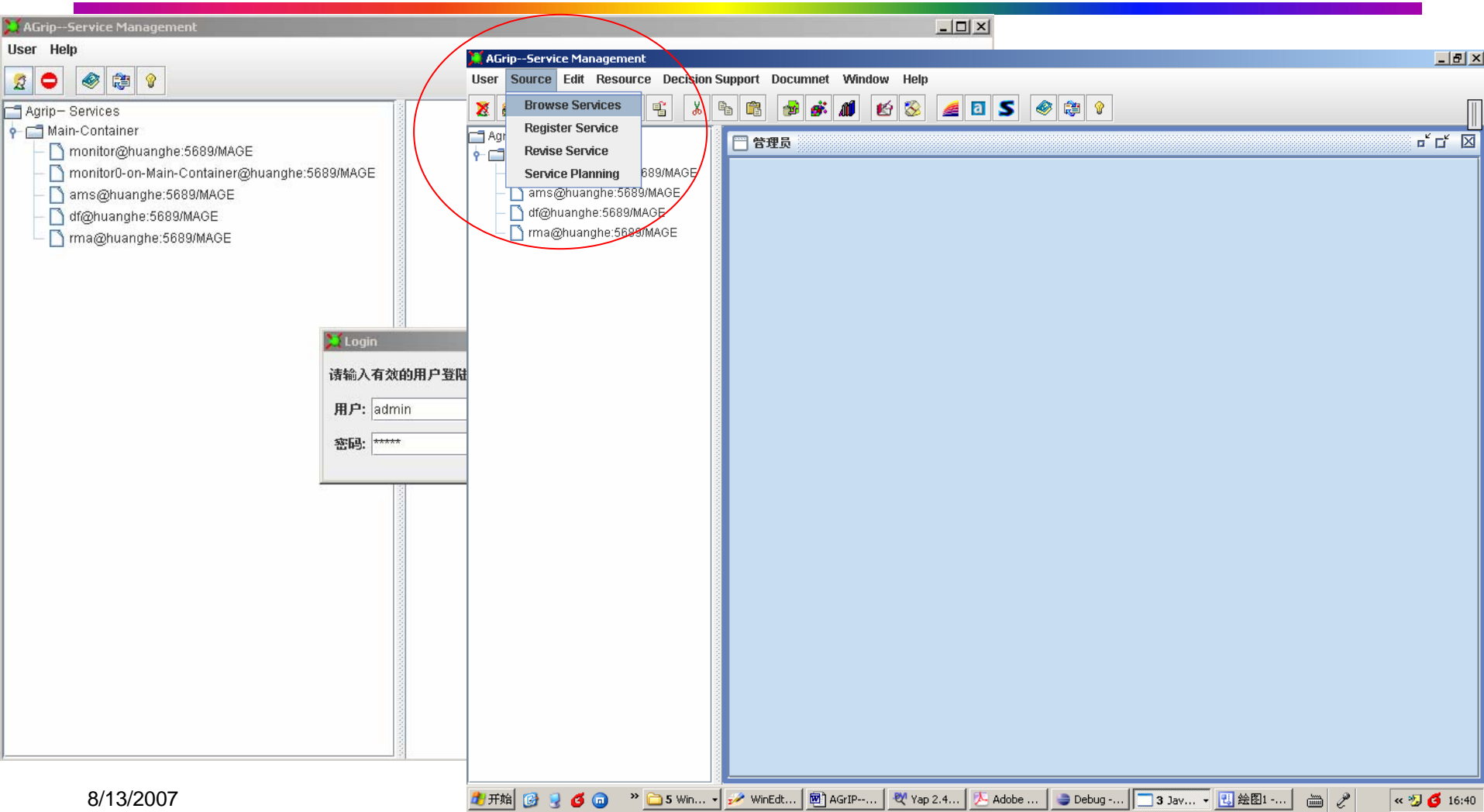


多主体环境 MAGE



基于主体的服务组合

Agent-Based Service Composition



基于主体的服务组合

Agent-Based Service Composition



中科院计算所
INSTITUTE OF COMPUTING
TECHNOLOGY

The screenshot displays the AGrip--Service Management application interface. It features a menu bar (User, Source, Edit, Resource, Decision Support, Documnet, Window, Help) and a toolbar. The main workspace is divided into several panes:

- Left Pane:** A tree view showing a hierarchy of services under 'Agrip-- Services', including 'Main-Container', 'Container-1', and 'ManageAgentServices__200606041701033121'.
- Top Center Pane:** A 'Service Profile' window listing five services with IDs ranging from 20060604140814 to 20060604143420.
- Bottom Left Pane:** An 'Input Parameters' form for a service, with fields for ServiceID, AgentID, and ProviderID. Below it are sections for 'Address' (WebURL), 'Service Process' (Preconditions, Effects, Inputs, Outputs, Source, Atomic).
- Right Pane:** A 'ManageAgentServices' window showing a table of services and a detailed configuration form for a selected service.

ServiceID	ServiceCategory
1 Service__200606041436191713	ServiceCategory.Operational Consulting_Reviews

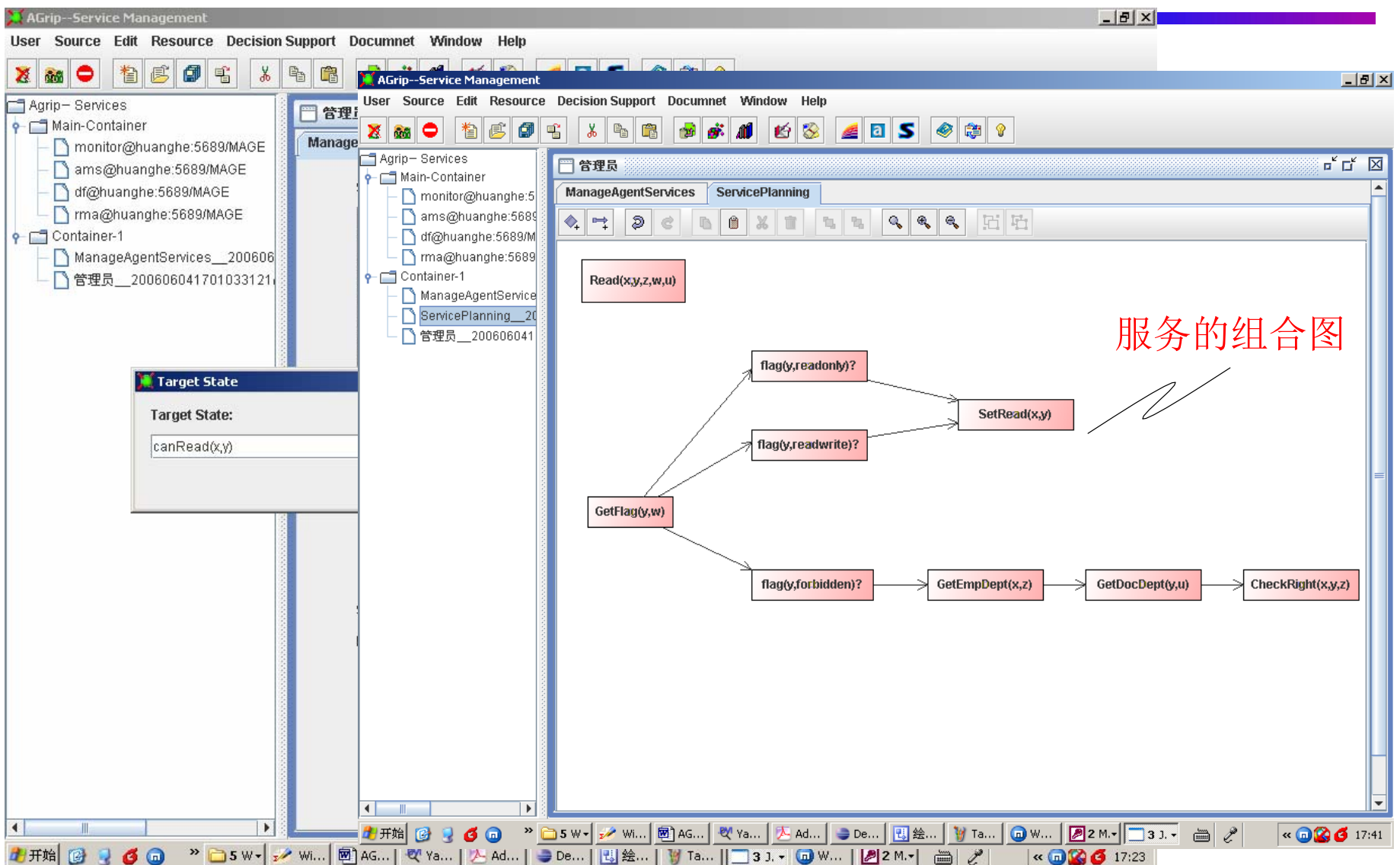
AgentID	Agent__200606041436191717	ServiceID	Service__200606041436191713
ProviderID	Provider__200606041436191873		
ServiceCate...	ServiceCategory.Operational Consulting_Reviews		
Description	CheckRight(x,y,z)		

Name	Zhao	Title		Email	
Phone				Fax	
Address					
WebURL	http://www.intsci.ac.cn/				

Service Process	
Preconditions	Employee(x), Document(y)
Effects	{Department(z), worksIn(x,z), belongsTo(y,z)/canRead(x,y), {Department(z), manages(x,z), belongsTo(y,z)
Inputs	x:String, y:String
Outputs	depart:String, canRead:boolean, canWrite:boolean
Source	C:\Documents and Settings\He Huang\My Documents\CheckRight.jar
Atomic	是
Process	

8/13/2007

基于主体的服务组合 Agent-Based Service Composition



The screenshot displays the AGrip--Service Management interface. On the left, a tree view shows the service hierarchy: Agrip- Services, Main-Container, and Container-1. A 'Target State' dialog box is open, showing 'canRead(x,y)'. The main window displays a 'ServicePlanning' diagram with the following components and flow:

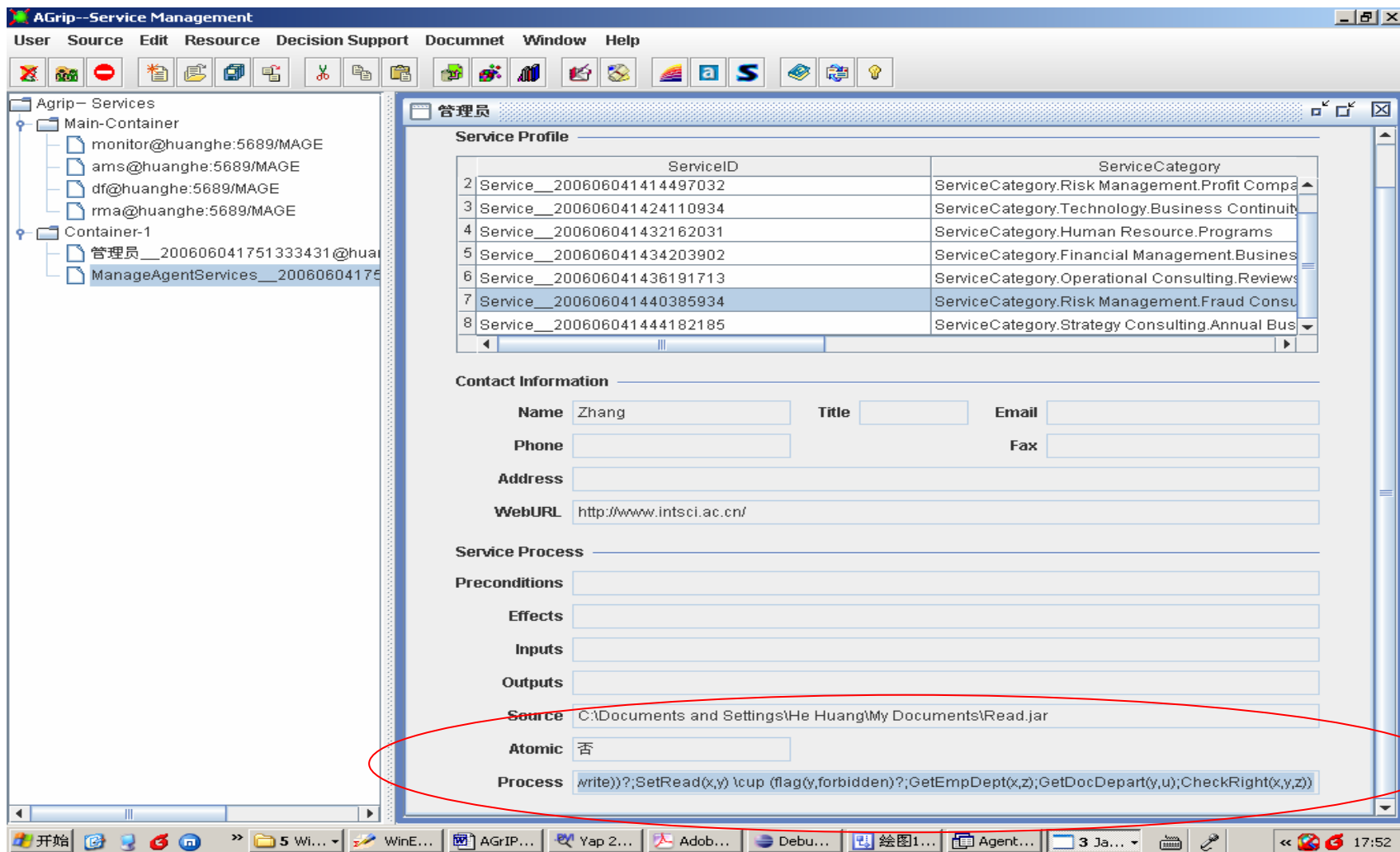
- Read(x,y,z,w,u)
- GetFlag(y,w)
- flag(y,readonly)?
- flag(y,readwrite)?
- SetRead(x,y)
- flag(y,forbidden)?
- GetEmpDept(x,z)
- GetDocDept(y,u)
- CheckRight(x,y,z)

Flow: Read(x,y,z,w,u) and GetFlag(y,w) lead to flag(y,readonly)? and flag(y,readwrite)?, which both lead to SetRead(x,y). GetFlag(y,w) also leads to flag(y,forbidden)?, which leads to GetEmpDept(x,z), then GetDocDept(y,u), and finally CheckRight(x,y,z).

服务的组合图

基于主体的服务组合

Agent-Based Service Composition



AGrip--Service Management

User Source Edit Resource Decision Support Documnet Window Help

Agrip - Services

- Main-Container
 - monitor@huanghe:5689/MAGE
 - ams@huanghe:5689/MAGE
 - df@huanghe:5689/MAGE
 - rma@huanghe:5689/MAGE
- Container-1
 - 管理员__200606041751333431@hua...
 - ManageAgentServices__20060604175...

管理员

Service Profile

ServiceID	ServiceCategory
2 Service__200606041414497032	ServiceCategory.Risk Management.Profit Compa
3 Service__200606041424110934	ServiceCategory.Technology.Business Continu
4 Service__200606041432162031	ServiceCategory.Human Resource.Programs
5 Service__200606041434203902	ServiceCategory.Financial Management.Busines
6 Service__200606041436191713	ServiceCategory.Operational Consulting.Reviews
7 Service__200606041440385934	ServiceCategory.Risk Management.Fraud Consu
8 Service__200606041444182185	ServiceCategory.Strategy Consulting.Annual Bus

Contact Information

Name: Zhang Title: Email: Phone: Fax: Address: WebURL: <http://www.intsci.ac.cn/>

Service Process

Preconditions

Effects

Inputs

Outputs

Source: C:\Documents and Settings\He Huang\My Documents\Read.jar

Atomic: 否

Process: write));SetRead(x,y) \cup (flag(y,forbidden)?;GetEmpDept(x,z);GetDocDepart(y,u);CheckRight(x,y,z))

结束语

- **DDL as a logic foundation of semantic Web**
- **Ontology lets Web services with semantics**
- **Knowledge base support Web services discovery**
- **Context-based planning for service composition**
- **Semantic Web Services Platform SWSBroker**

References

- Zhongzhi Shi, Mingkai Dong, Yuncheng Jiang, Haijun Zhang. **A Logic Foundation for the Semantic Web**. *Science in China, Series F Information Sciences*, 48(2):161-178, 2005
- Cardoso, J. (2007). *Semantic Web Services: Theory, Tools and Applications*. Idea Group.. ISBN 978-1-59904-045-5.
- Michael Stollberg, Omair Shafiq, John Domingue, Liliana Cabral, Emilia Cimpian, Dumitru Roman, Michal Zaremba: **Semantic Web Services – State of Affairs**, *ASWC2006*, Sept. 4-7, Beijing
- Lei Li and Ian Horrocks. **A software framework for matchmaking based on semantic web technology**. In *Proc. of the Twelfth International World Wide Web Conference (WWW 2003)*, 2003
- Daniel J. Mandell and Sheila A. McIlraith. **Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation**. In *Proceedings of the Second International Semantic Web Conference (ISWC2003)*,
- Preist, C.: **A Conceptual Architecture for Semantic Web Services**. In *Proceedings of the 3rd International Semantic Web Conference (ISWC 2004)*, 2004, pp. 395 - 409.
- Stollberg, M.; Keller, U.; Fensel, D.: **Partner and Service Discovery for Collaboration on the Semantic Web**. *Proc. 3rd Intl. Conference on Web Services (ICWS 2005)*, Orlando, Florida, July 2005.

References

- Fen Lin, Lirong Qiu, He Huang, Qing Yu, Zhongzhi Shi: **Description Logic Based Composition of Web Services**. *PRIMA 2006*: 199-210
- Lirong Qiu, Zhongzhi Shi, and Fen Lin.: **Context Optimization of AI Planning for Services Composition**. *2006 IEEE International Conference on e-Business Engineering, Shanghai, 2006*
- Daniela Berardi, Liliana Cabral, Emilia Cimpian, Dr. John Domingue, Massimo Mecella, Michael Stollberg, Professor Katia Sycara: **Semantic Web Services**. *ESWC2005 Tutorial, Heraklion, Greece, May 2005*
- M. Vukovic and P. Robinson. **Adaptive, Planning Based, Web Service Composition for Context Awareness**. In: *Advances in Pervasive Computing*. Volume 176. (2004) 247–252
- Steve Battle, etc. : **Semantic Web Services Ontology (SWSO)**, Version 1.0, W3C proposal, 2005
- Steve Battle, etc. : **Semantic Web Services Language (SWSL)**, Version 1.0, W3C proposal, 2005

Thank You

Question!

Intelligence Science
<http://www.intsci.ac.cn/>

