

# Software Engineering for Large-scale Multi-Agent Systems (SELMAS06)



中科院计算所  
INSTITUTE OF COMPUTING TECHNOLOGY

## Agent-Grid Intelligence Platform for Collaborative Working Environment

**Zhongzhi Shi**  
**shizz@ics.ict.ac.cn**

**Institute of Computing Technology**  
**Chinese Academy of Sciences**



# Acknowledgement

- China Important Basic Research Programme 973
- China High-Tech Programme 863
- National Natural Science Foundation of China
- Ministry of Information Industry, China
- Knowledge Innovative Programme of CAS



# Self-Introduction

- Zhongzhi Shi is Professor at the Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, IEEE Senior member.
- He is Chair of WG 12.2 of IFIP. He also serves as Vice President of Chinese Association for Artificial Intelligence.
- He received the 2nd Grade National Award of Science and Technology Progress in 2002. In 1998 and 2001 he received the 2nd Grade Award of Science and Technology Progress from the Chinese Academy of Sciences.
- His research interests include intelligence science, multiagent systems, Semantic Web, machine learning and data mining.
- He published 10 books, edited 11 books and more than 350 technical papers.



# Outline

- **Introduction**
- **Agent Model**
- **Multiagent Environment— MAGE**
- **Agent Collaboration**
- **Agent Grid Intelligence Platform**
- **Applications**
- **Conclusions**



# Agent Computing

**Agent computing is particularly well suited to the collaborative work. The agent-based computing paradigm has following features:**

- **Autonomy - agent operate without intervention;**
- **Social ability – agents interact each other using an agent communication language;**
- **Goal driven – agent exhibit goal-directed behavior;**
- **Reactivity – agents perceive and respond to their environment.**



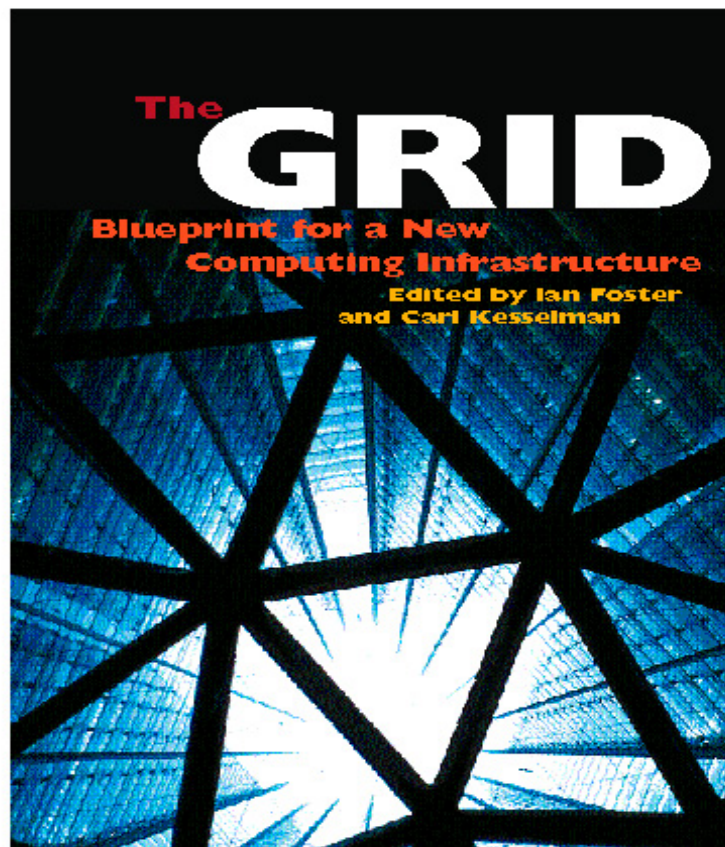
# Semantic Web



Semantic Web will  
provide well-  
defined meaning  
which better  
enabling  
computers and  
people to work in  
collaboration



# Grid Computing



- Flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resource

From "The Anatomy of the Grid: Enabling Scalable Virtual Organizations"

- Grid Forum
  - [www.gridforum.org](http://www.gridforum.org)



# Brain Meet Brawn

Ian Foster, Nicolas R. Jennings, Carl Kasselmann. **Brain Meet Brawn: Why Grid and Agents need each others.**  
AAMAS'04

**The Grid community has historically focused on "brawn": infrastructure, tools, and applications for reliable and secure resource sharing within dynamic and geographically distributed virtual organizations.**

**The agents community has focused on "brain": autonomous problem solvers that can act flexibly in uncertain and dynamic environments.**



# Research on MAS

- 1994: Multiagent Processing Environment **MAPE**
- 1996: Agent-based **CSCW**
- 1998: Common Agent Request Broker Architecture **CARBA**,  
**MAPE2**
- 2000: Multiagent Environment **MAGE**
- 2002: Agent Grid Intelligence Platform **AGrIP**
- 2003: Dynamic Description Logic **DDL**
- 2004: Visual Agent Developing Tool **VASudio**
- 2005: Ontology-based Knowledge Management **KMSphere**

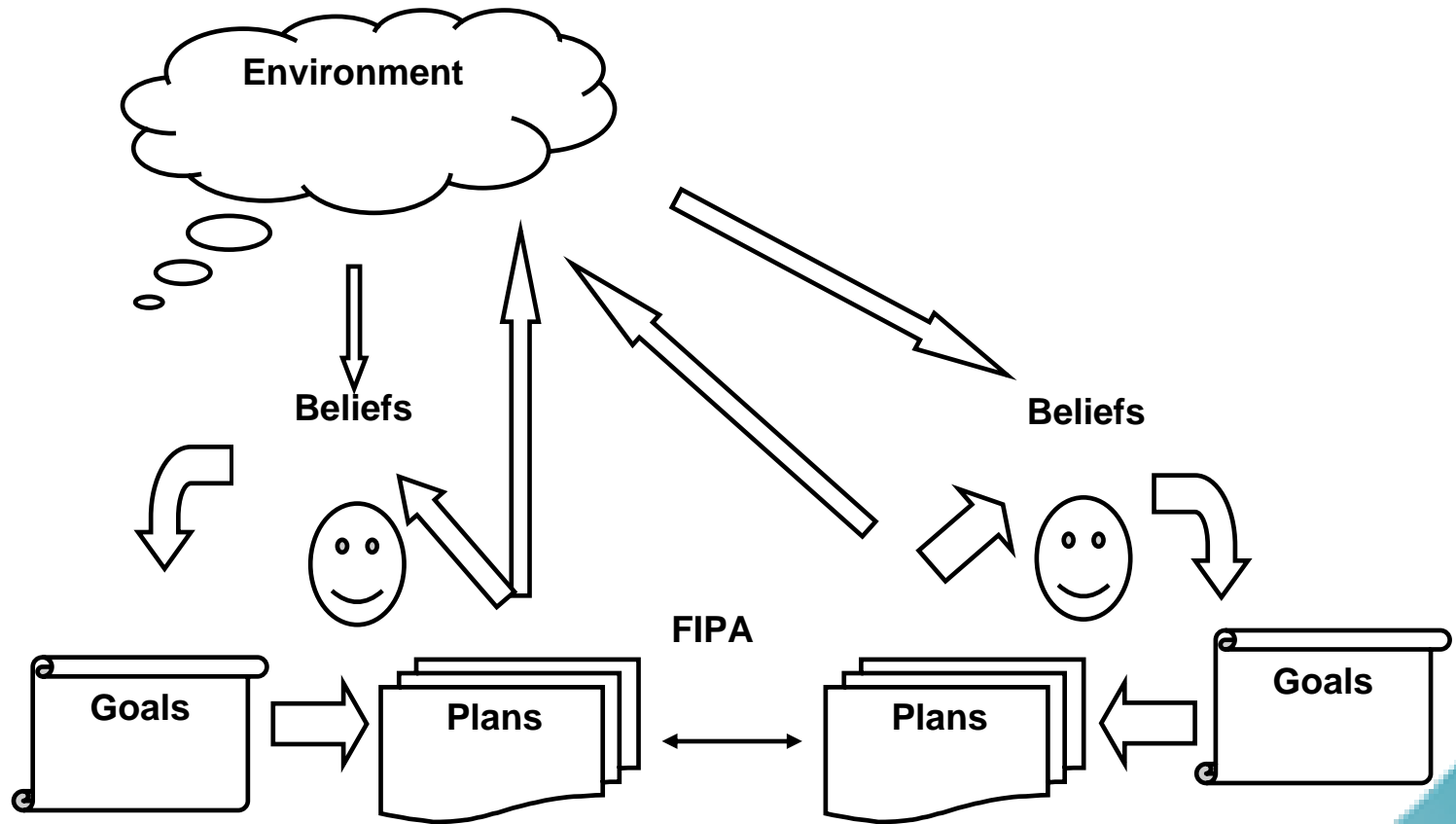


# Outline

- Introduction
- Agent Model
- Multiagent Environment— MAGE
- Agent Collaboration
- Agent Grid Intelligence Platform
- Applications
- Conclusions



# BDI Model





# Description Logic

- Concepts and Role
- Tbox——Assertions
- Abox——Instance
- Reasoning mechanism in terms of Tbox and Abox



# Reasoning in DL

- 1) **S**ubsumption
- 2) consistency
- 3) satisfiability
- 4) instance checking

K  
B

## TBox(Scheme)

Man = Human  $\sqcap$  Male

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

## Abox(Data)

John: Happy-father

<John, Mary> : Has-child

Reasoning

Interface



# Dynamic Description Logic

## *The primitive symbols*

- **Concept name:**  $C_1, C_2, \dots$ ;
- **Role name:**  $R_1, R_2, \dots$ ;
- **Individual constant:**  $a, b, c, \dots$ ;
- **Individual variable:**  $x, y, z, \dots$ ;
- **Concept operation:**  $\neg, \sqcap, \sqcup, \exists, \forall$ ;
- **Axiom operation:**  $\neg, \wedge, \rightarrow, \forall$ ;
- **Action:**  $A_1, A_2, \dots$ ;
- **Action constraction:**  $;$  (composition),  $\cup$  (alternation),  $*$  (repeat),  $?$  (test);
- **Action variable:**  $\alpha, \beta, \dots$ ;
- **Axiom variable:**  $\varphi, \psi, \pi, \dots$ ;
- **State variable:**  $u, v, w, \dots$ ;



# Dynamic Description Logic

*Concepts in DDL are defined as the following:*

- (1) *Primitive concept  $P$ , top  $\top$  and bottom  $\perp$  are concepts.*
- (2)  *$\neg C$ ,  $C \sqcap D$ ,  $C \sqcup D$  are concepts.*
- (3)  *$\exists R.C$ ,  $\forall R.C$  are concepts.*



# 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 Semantics

- *Actions in DDL are defined as the following:*
- *Atom action  $A(a_1, \dots, a_n)$  is action.*
- *If  $\alpha$  and  $\beta$  are actions, then  $\alpha ; \beta$ ,  $\alpha \cup \beta$ ,  $\alpha^*$  are actions;*
- *If  $\varphi$  is an assertion formula, then  $\varphi ?$  is action.*



# Metal State Model

**Mental State:**  $\langle \mathcal{K}, \mathcal{A}, \mathcal{G}, \mathcal{P}, I \rangle$ ,

**Where**

- $\mathcal{K}$  *belief*
- $\mathcal{A}$  **action**
- $\mathcal{G}$  *goal*
- $\mathcal{P}$  **plan**
- $I$  **intention**



# Belief

$$\mathcal{K} = \langle \mathcal{T}, S, \mathcal{B} \rangle$$

*$\mathcal{T}$ : Ontology*

*$S$ : Constrains*

*$\mathcal{B}$ : Current belief*



# Belief Revision

```
AddBelief( $F, \mathcal{B}$ ) {  
     $F' \leftarrow \text{Extend}(F)$ ;  
    Foreach  $\varphi \in F'$  do  
        If  $\neg \varphi \in \mathcal{B}$  Then  $\mathcal{B} \leftarrow \mathcal{B} - \{\neg \varphi\}$ ;  
     $\mathcal{B}' \leftarrow \text{Extend}(\mathcal{B} \cup F')$ ;  
    If Consistent( $\mathcal{B}'$ ) Then Return  $\mathcal{B}'$  ;  
    Else{  
        Let  $\{\psi, \neg \psi\} = \text{ConflictSet}(\mathcal{B}')$  ;  
        If  $\psi \in \mathcal{B}$  Then Return  $\mathcal{B}' - \{\psi\}$ ;  
  
        Else If  $\neg \psi \in \mathcal{B}$  Then Return  $\mathcal{B}' - \{\neg \psi\}$ ;  
        Else Return error;  
    }  
}
```



# Goal

Let  $\mathcal{A}$  be a set of actions,  $\mathcal{L}$  be a set of assertions,  $\mathcal{G}$  can be defined recursively:

- (1)  $\mathcal{A} \subseteq \mathcal{G}$ ,  $\mathcal{A}$  is basic action;
- (2) If  $\varphi \in \mathcal{L}$ , then  $achieve(\varphi) \in \mathcal{G}$ ;
- (3) If  $\varphi \in \mathcal{L}$ , then  $\varphi? \in \mathcal{G}$ ;
- (4) If  $\delta_1, \delta_2 \in \mathcal{G}$ ,  
then  $\delta_1; \delta_2 \in \mathcal{G}$ ,  $\delta_1 \cup \delta_2 \in \mathcal{G}$ ,  $\delta_1^* \in \mathcal{G}$ ;



# Static Plan

$$\delta_1 \Leftarrow \varphi \mid \delta_2$$

$$\delta_1 \in \mathcal{G}, \quad \delta_2 \in \mathcal{G}, \quad \varphi \in \mathcal{L}.$$

$\delta_1$ : Rule header

$\delta_2$ : Rule body

$\varphi$ : Rule guard condition

***$go\_floor(x) \Leftarrow @At(elevator, y) \mid [y < x?; up(x) \cup y > x?; down(x)]$***



# Dynamic Plan

*Plan*( $\delta, B$ ) //Goal  $\delta$ ,  $B$  Current belief set

{

If perform action  $\alpha$   $B \vdash T^\delta$  Then

{ Return  $B$ ; *Enqueue*( $\alpha, P$ ); }

Else {

look for subgoals to reach goal ;

computing  $E_\alpha$  and  $\delta$  for each subgoal;

computing the order of subgoals

select optimal subgoal plan;

sorting subgoals and form sequence  $\delta_1, \dots, \delta_n$

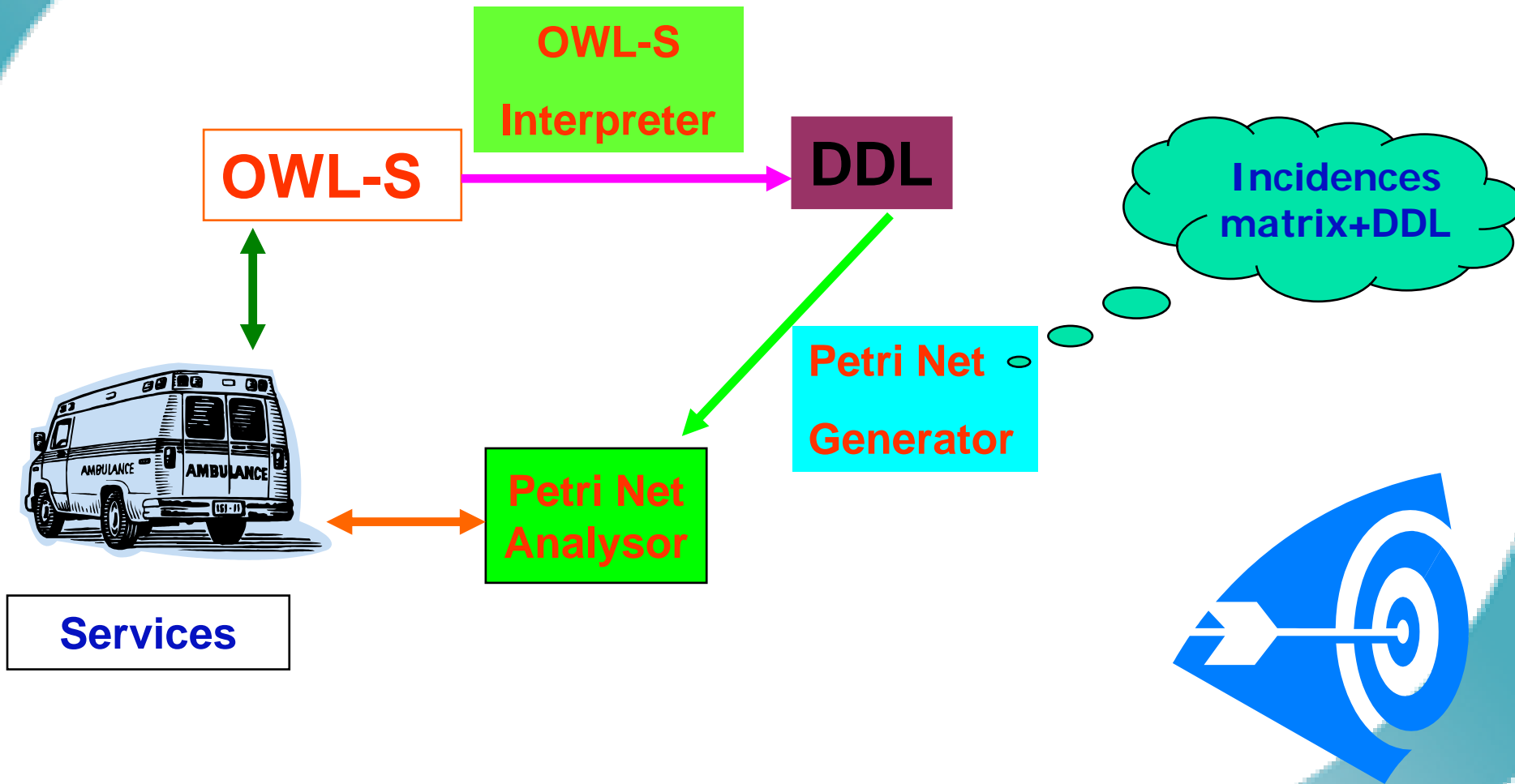
For  $\pi = \delta_1$  to  $\delta_n$  do

$B = \text{Plan}(\pi, B)$

}  
2006-6-3  
{

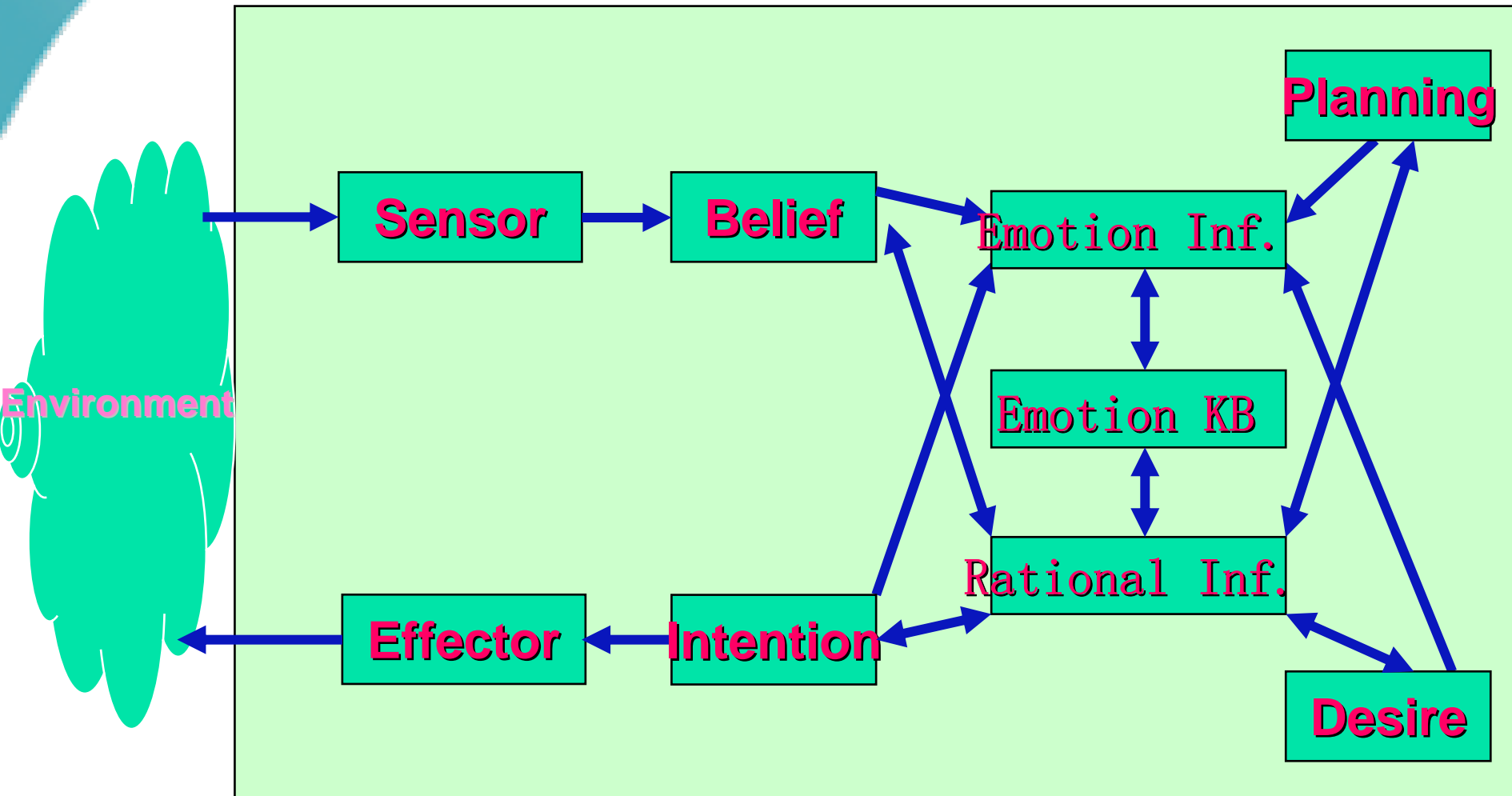


# Petri Net



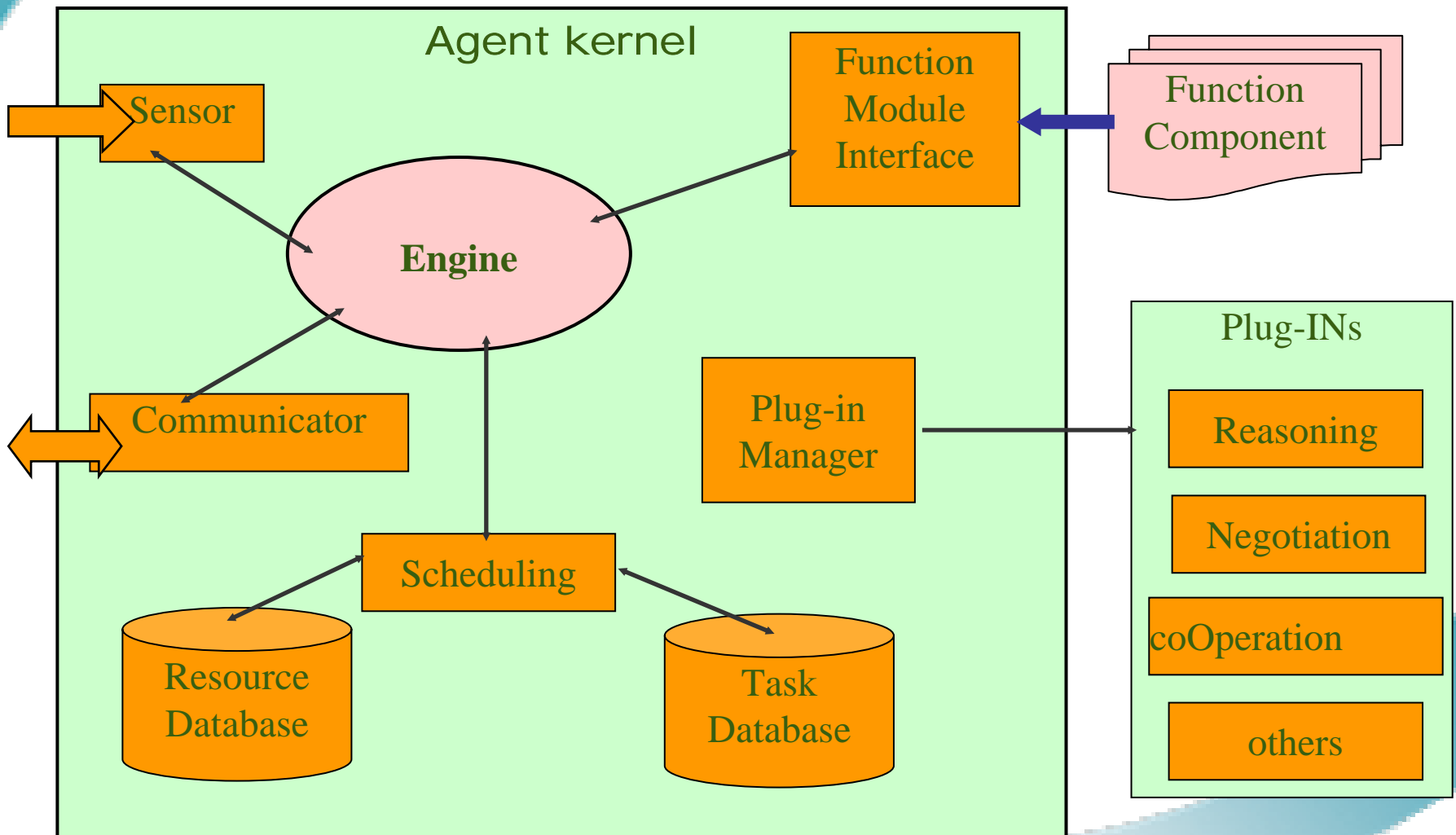


# Emotion Agent



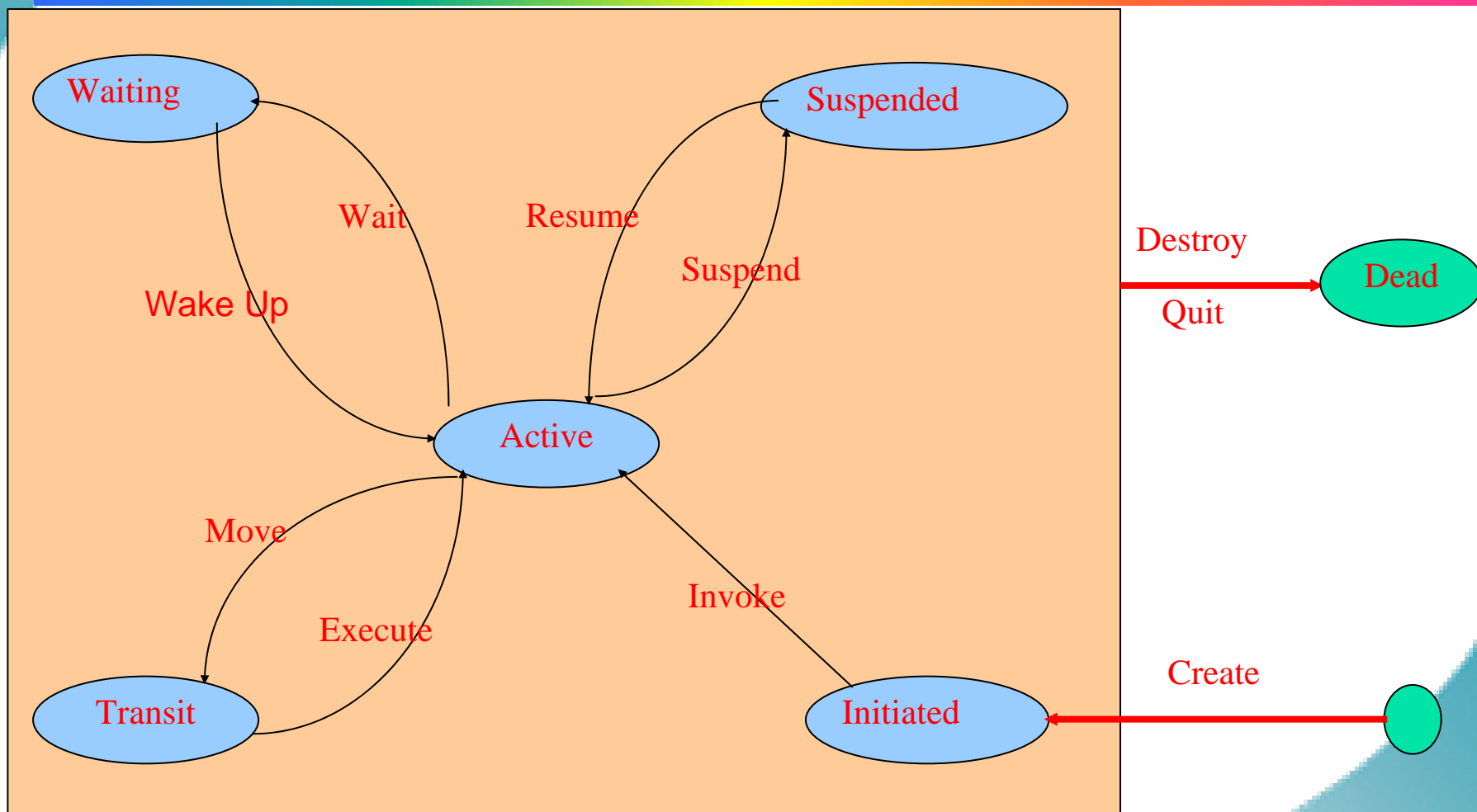


# Agent Architecture





# Agent Life Cycle



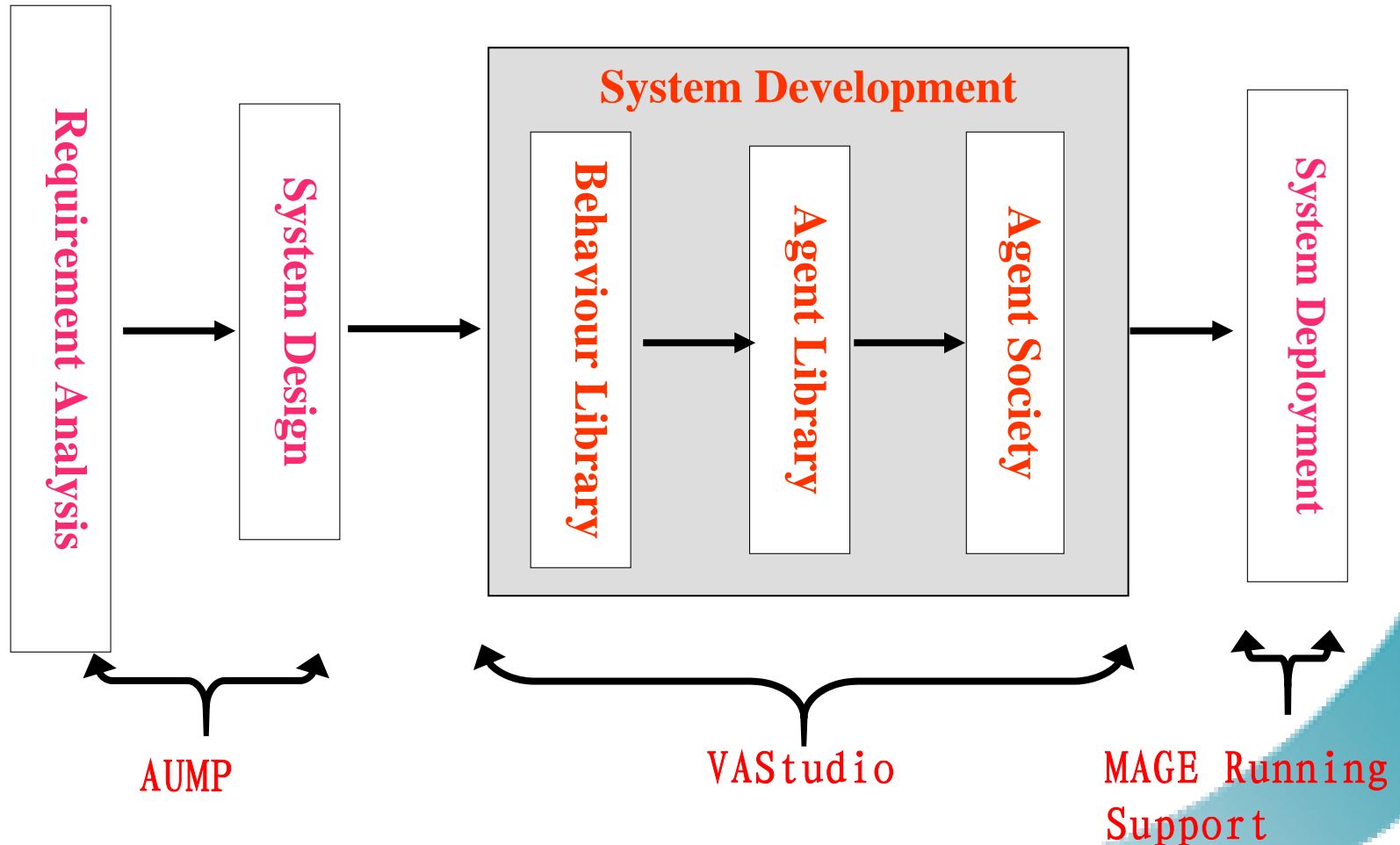


# Outline

- Introduction
- Agent Model
- **Multiagent Environment— MAGE**
- Agent Collaboration
- Agent Grid Intelligence Platform
- Applications
- Conclusions

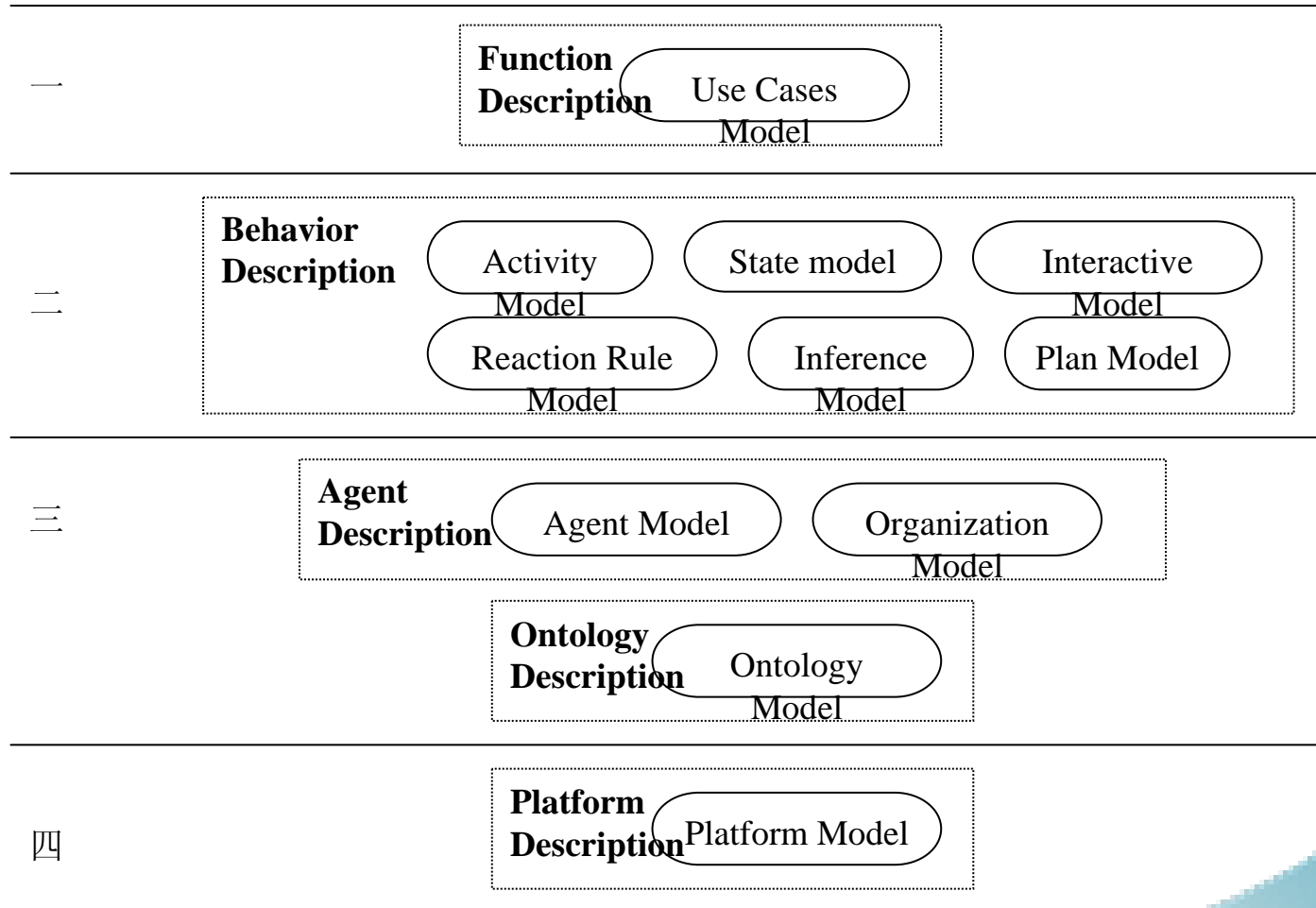


# Multiagent Environment MAGE



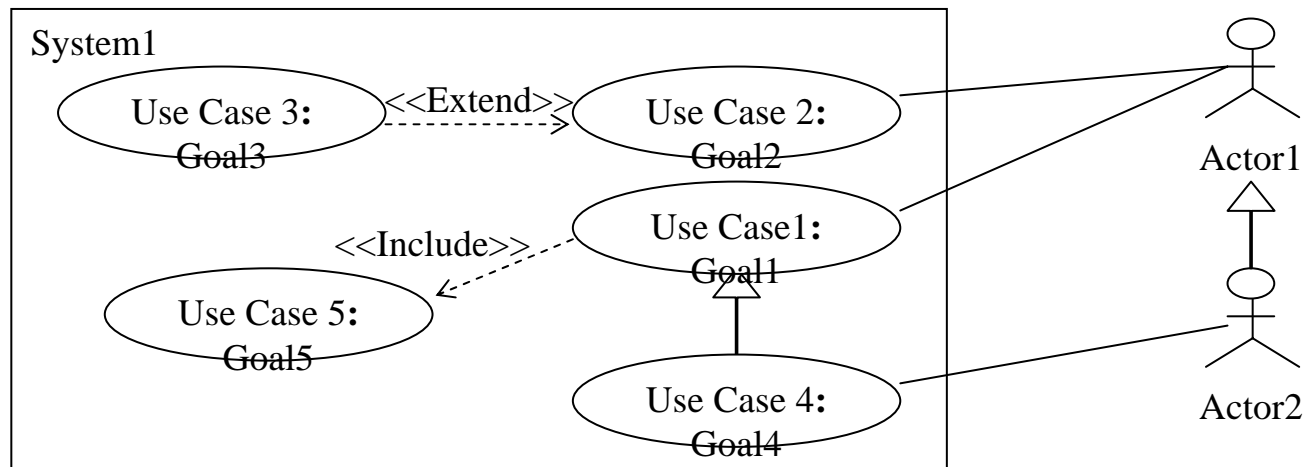
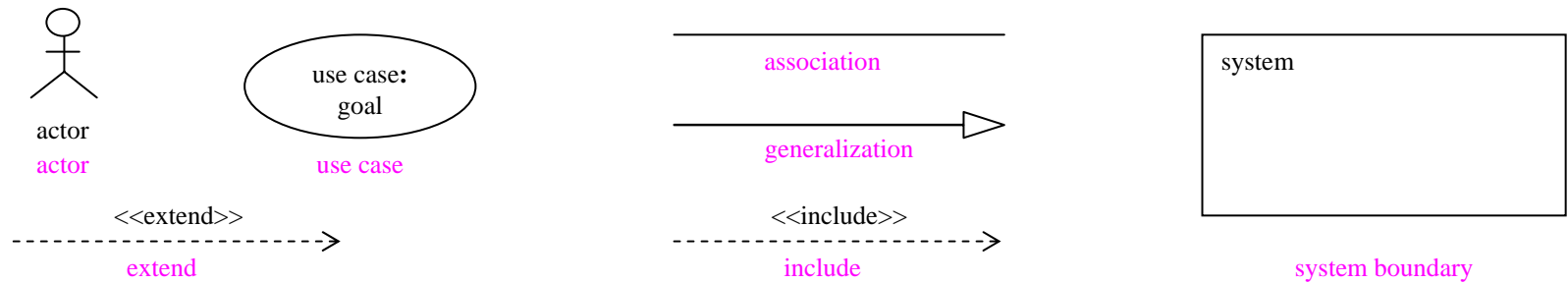


# Multiagent System Design Procedure



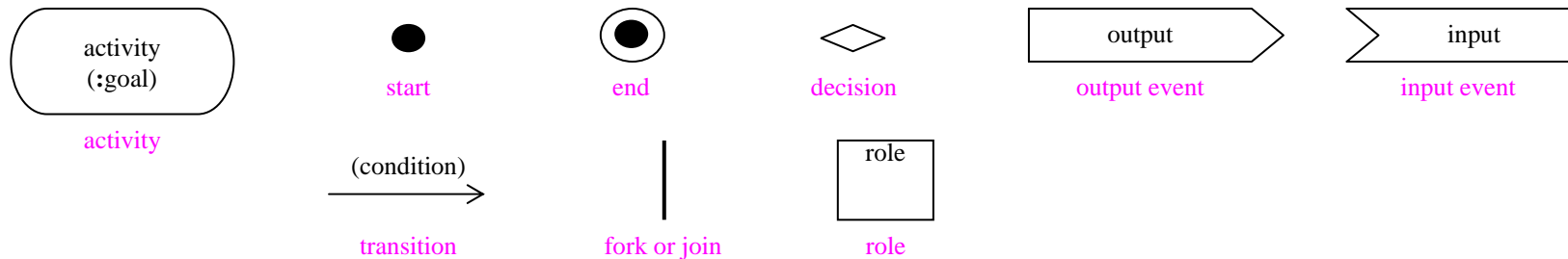


# AUML—Use Cases

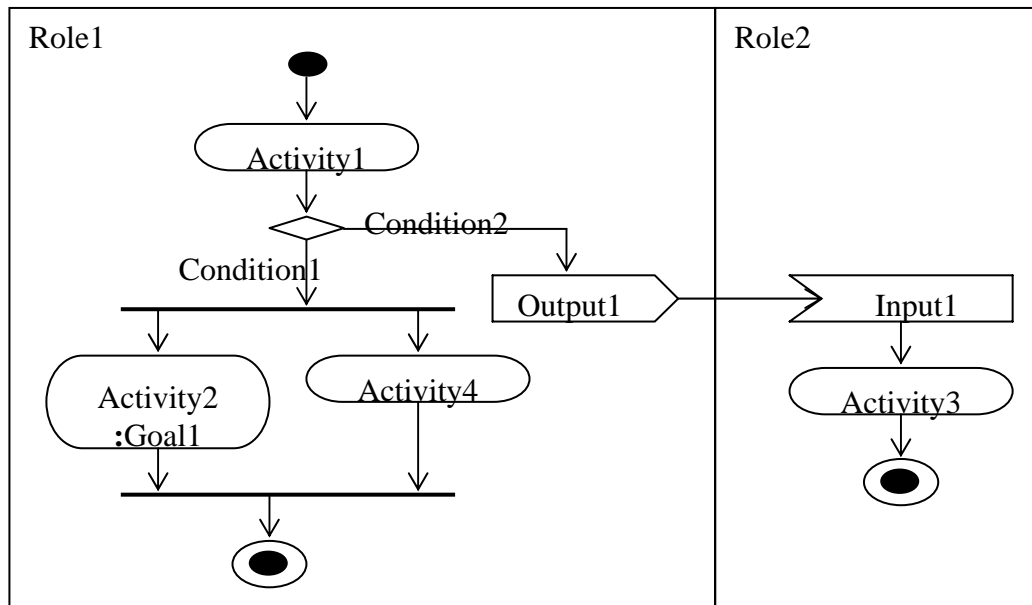




# AUML—Activity Diagram

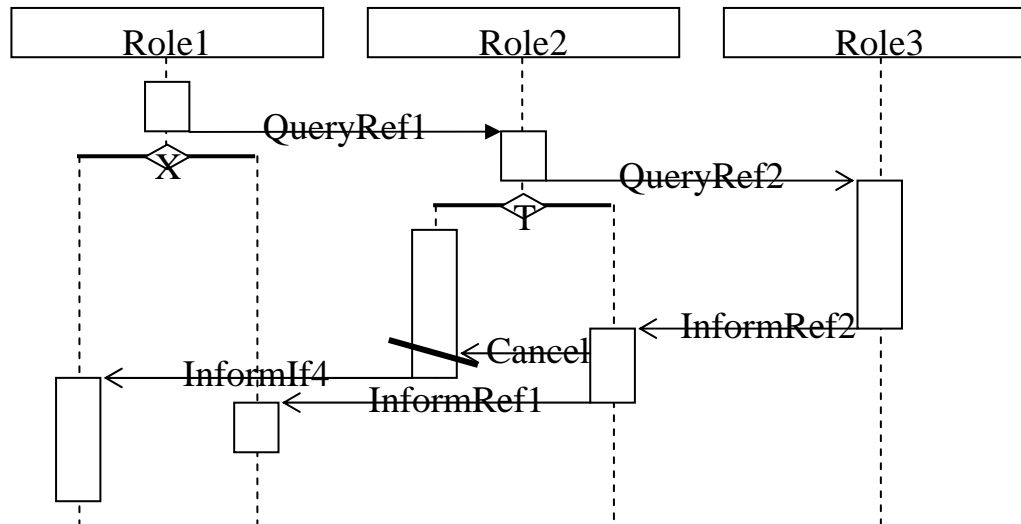
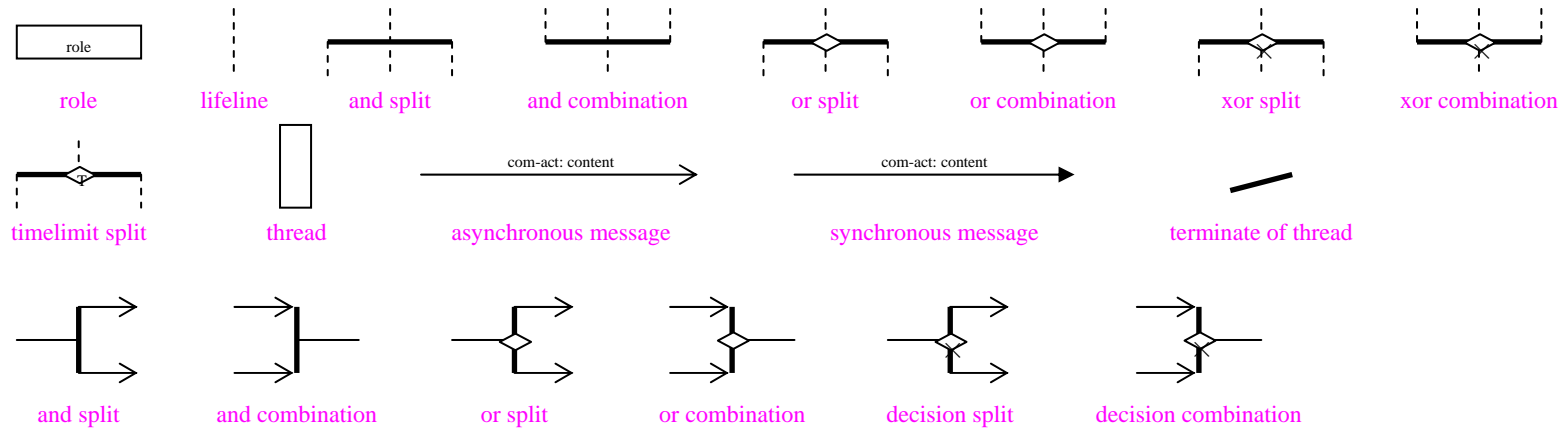


## Symbols in Activity Diagram



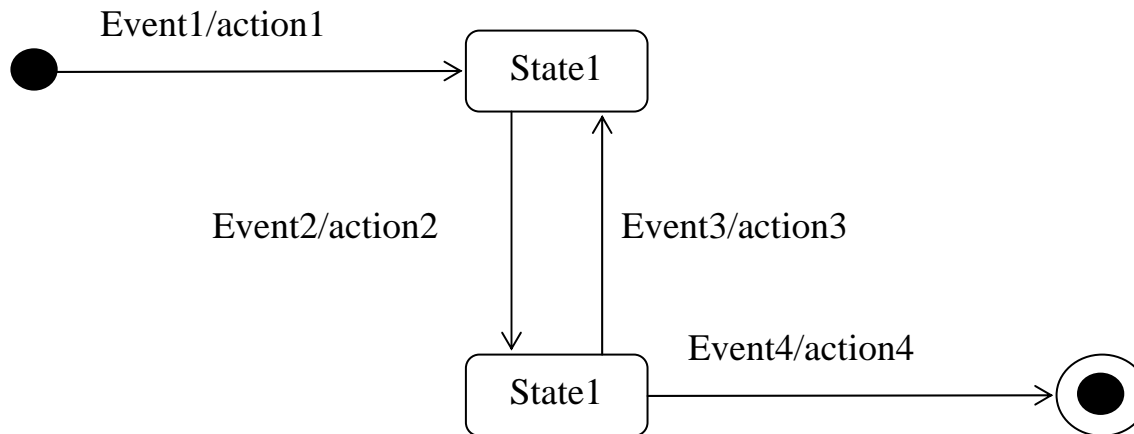


# AUML—Interactive Diagram





# AUML—State Diagram





# AUML—Reaction Diagram

## Event List (自定义事件表)

Event Name	In Params	Out Params	Description
Event1		int param1	Description of Event1 and param1.

## Action List (自定义动作表)

Action Name	In Params	Out Params	Description
Action1	int param1		Description of Action1 and param1.

## Reaction Rule List (反应规则表)

Rule Name	Event	Action	Description
Rule1	Event1	Action1	Description of Rule1.

## Reaction Rule Lib List (反应规则库表)

RuleLib Name	Rules
ReactionRuleLib1	Rule1, Rule2,



# AUML—Inference Model

## Fact Predication List (事实谓词表)

<b>Predication</b>	<b>Meanings</b>
Predication1	Meanings of Predication1

## Rational Rule List (推理规则表)

<b>Rule Name</b>	<b>Conditions</b>	<b>Conclusion</b>	<b>Description</b>
Rule1	Condition1	Conclusion1	Description of Rule1.

## Rational Rule Lib List (推理规则库表)

<b>RuleLib Name</b>	<b>Rules</b>
RationalRuleLib1	Rule1, Rule2,



# AUML—Plan Model

## Environment Predication List (环境描述谓词表)

Predication	Meanings
Predication1	Meanings of Predication1

## Action Template List (动作模板表)

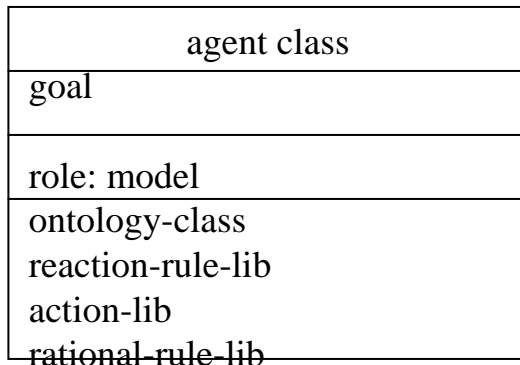
Action Template Name	Preconditions	Post Effects	Roles	Description
Action1	Condition1	Effect1	Role1,Role2	Description Action1.

## Action Lib (动作库)

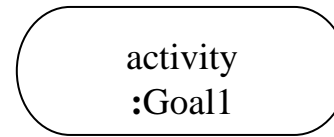
Action Template	Role	Agents
Action1	Role1	Agent1,Agent2
Action1	Role2	Agent1



# AUML—Agent Class Diagram



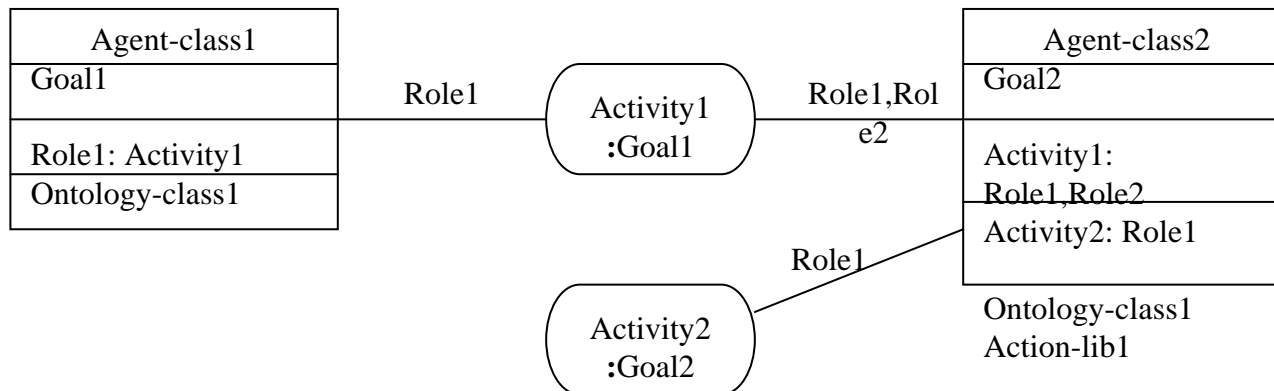
agent class



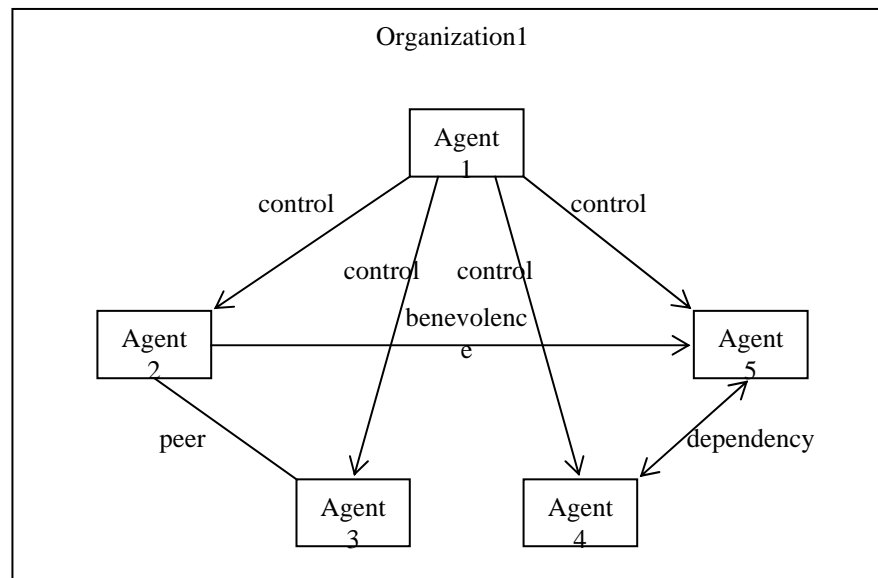
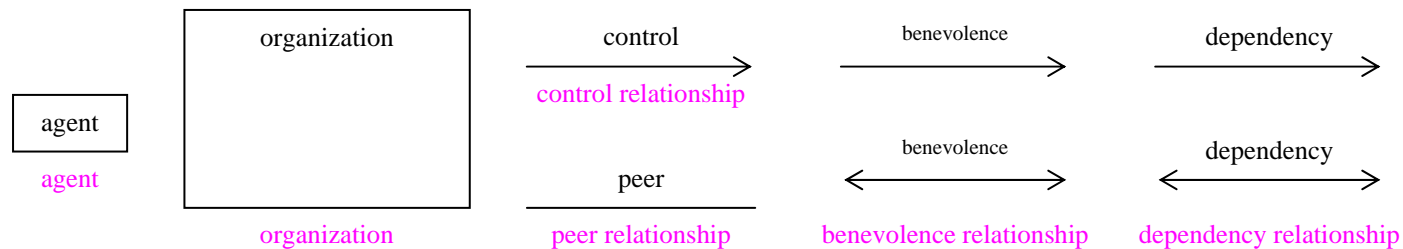
activity



association

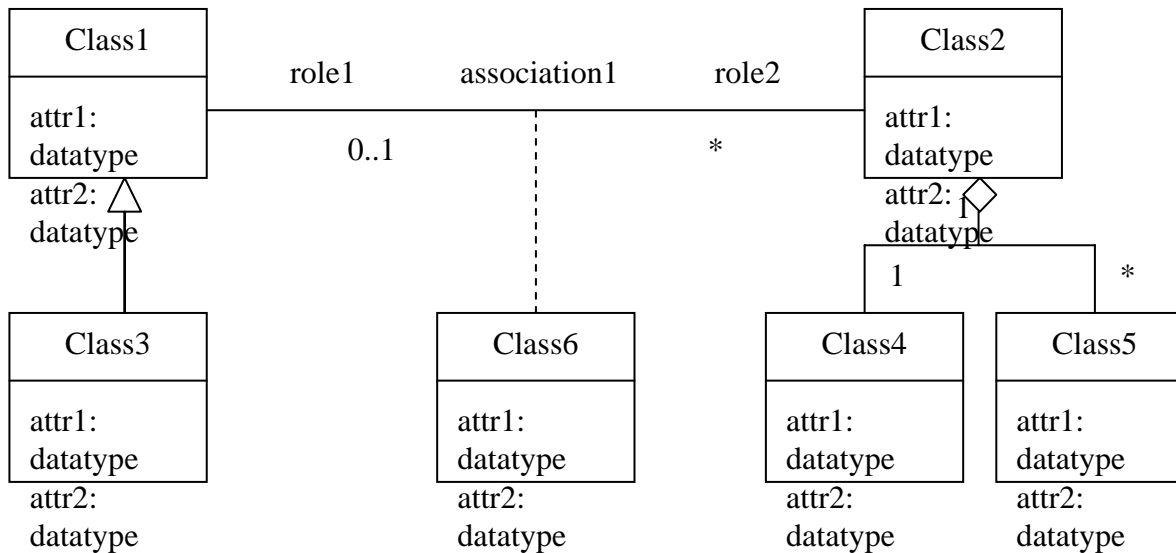
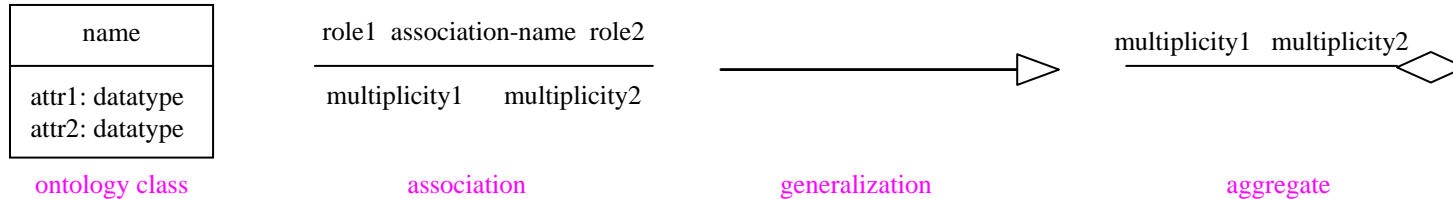


# AUML—Organization Diagram



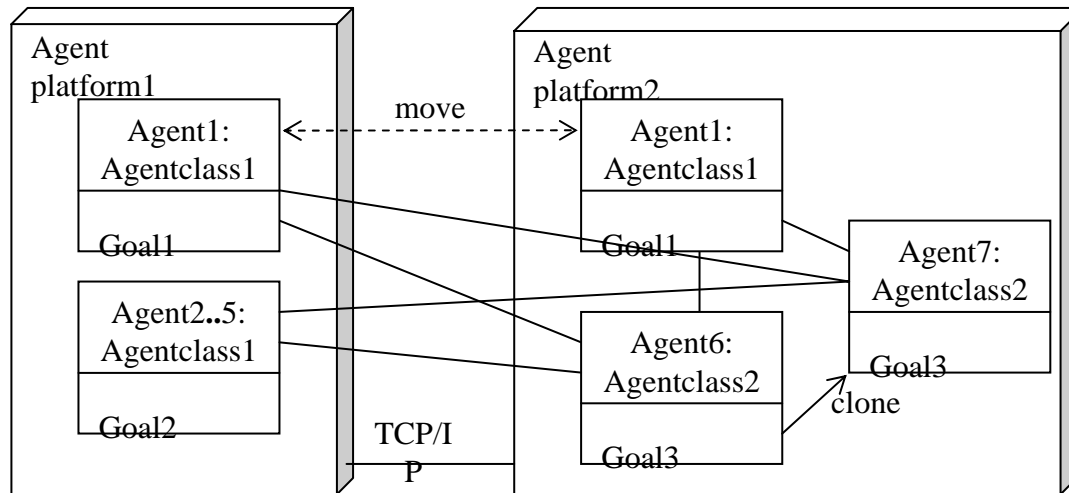


# AUML—Ontology Diagram



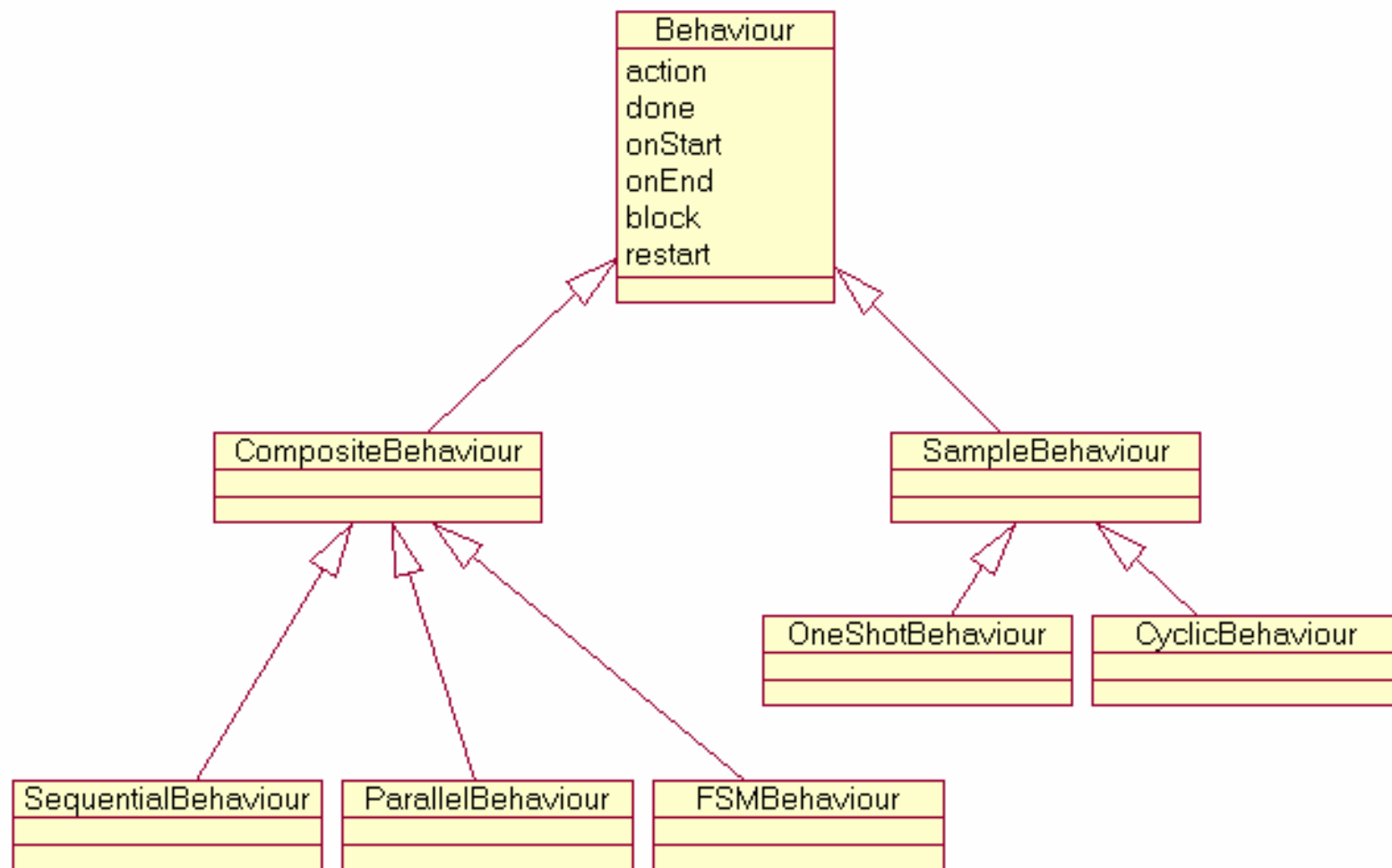


# AUML—Platform Diagram



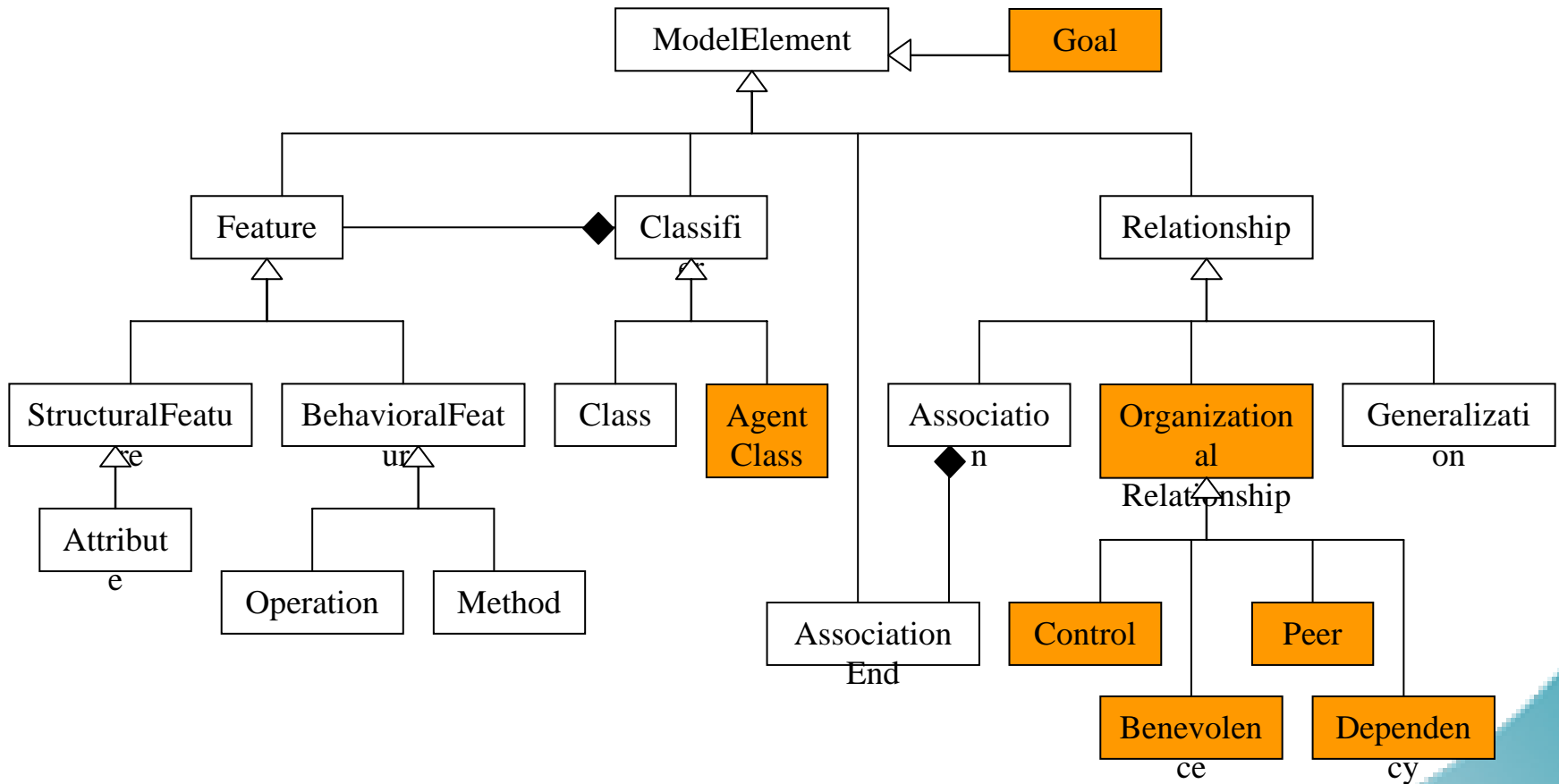


# AUMP — Source Code



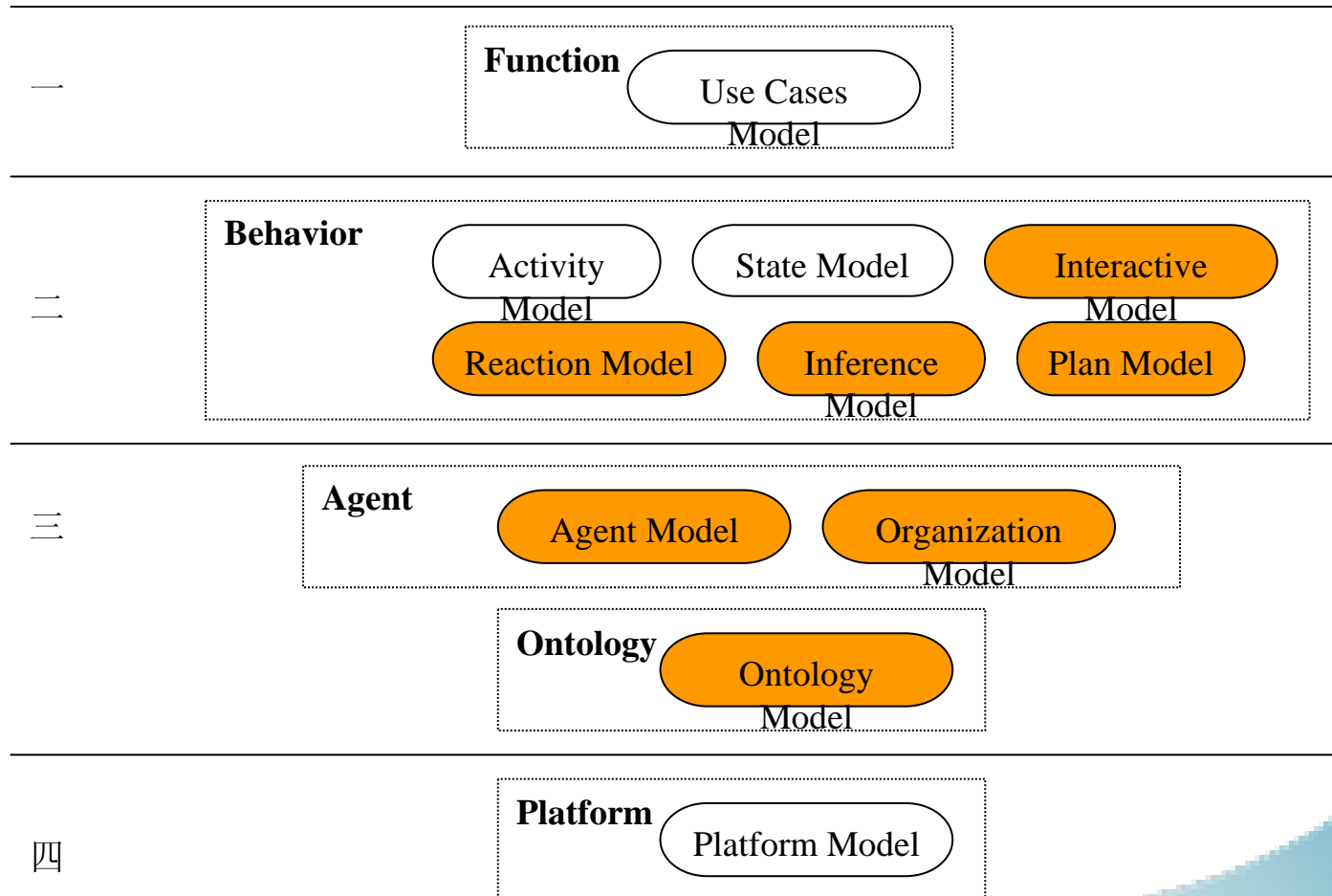


# Compare AUML with UML



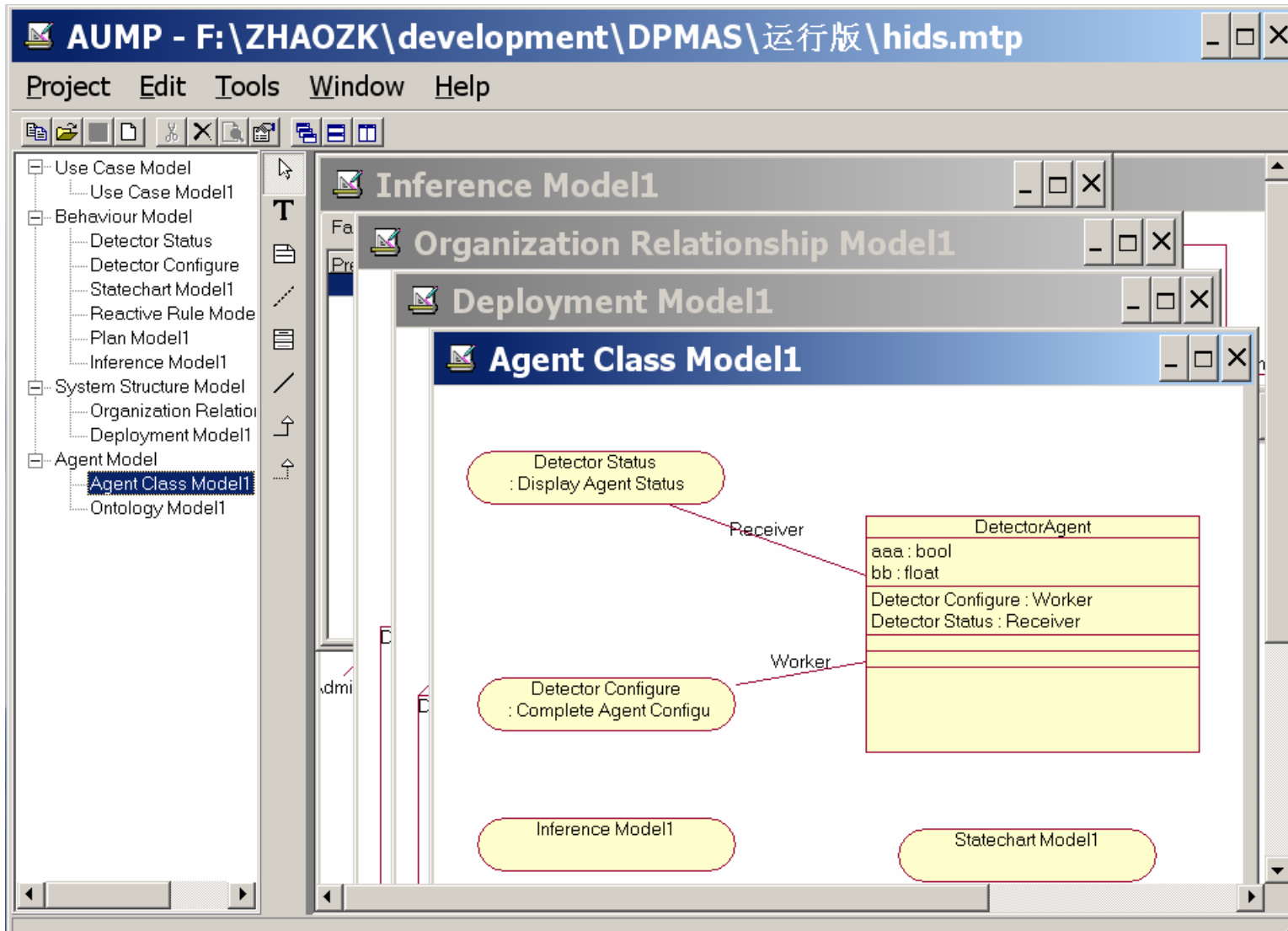


# Compare AUML with UML



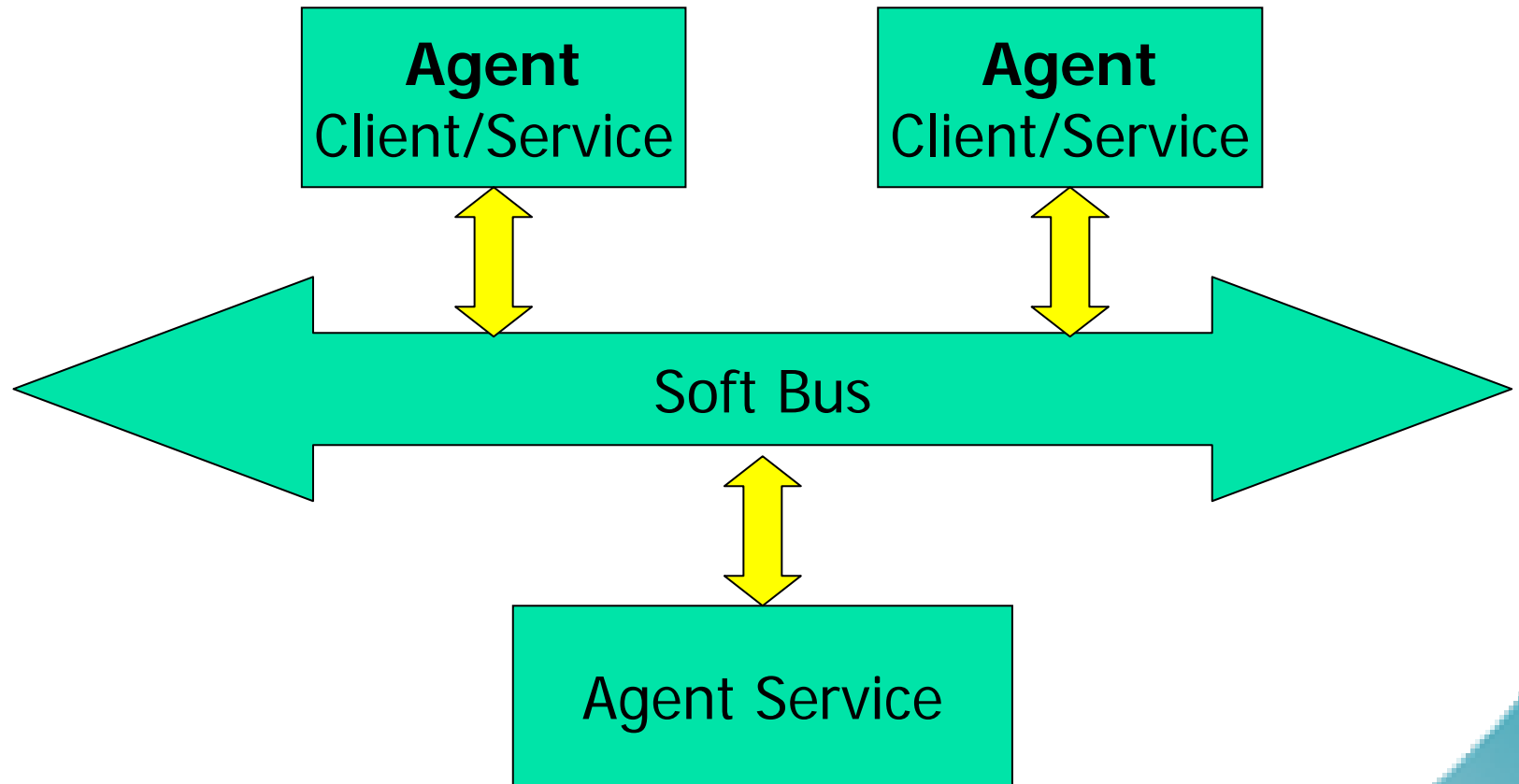


# Agent Unified Modeling Platform —AUMP



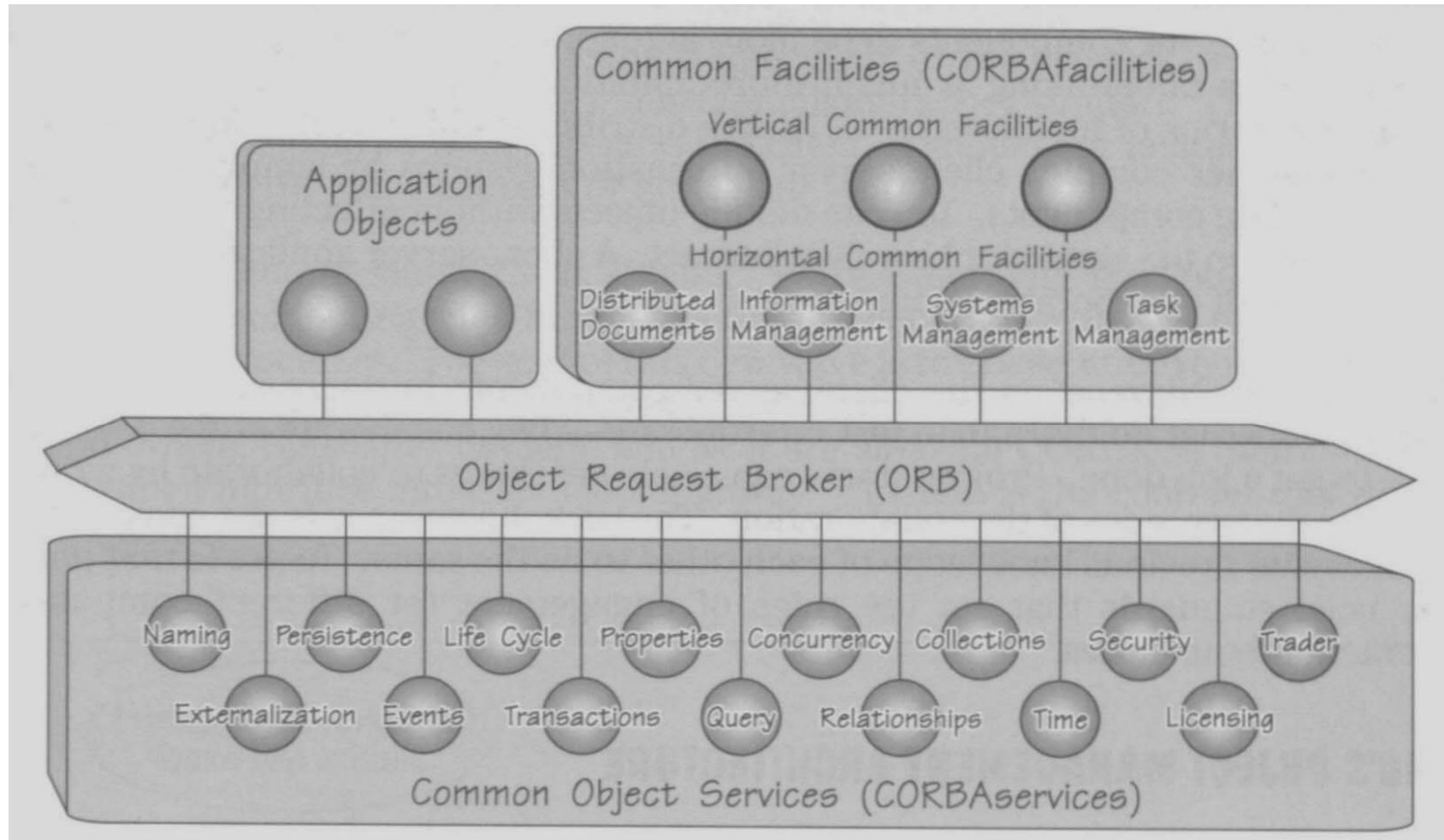


# Computation Model for Agent



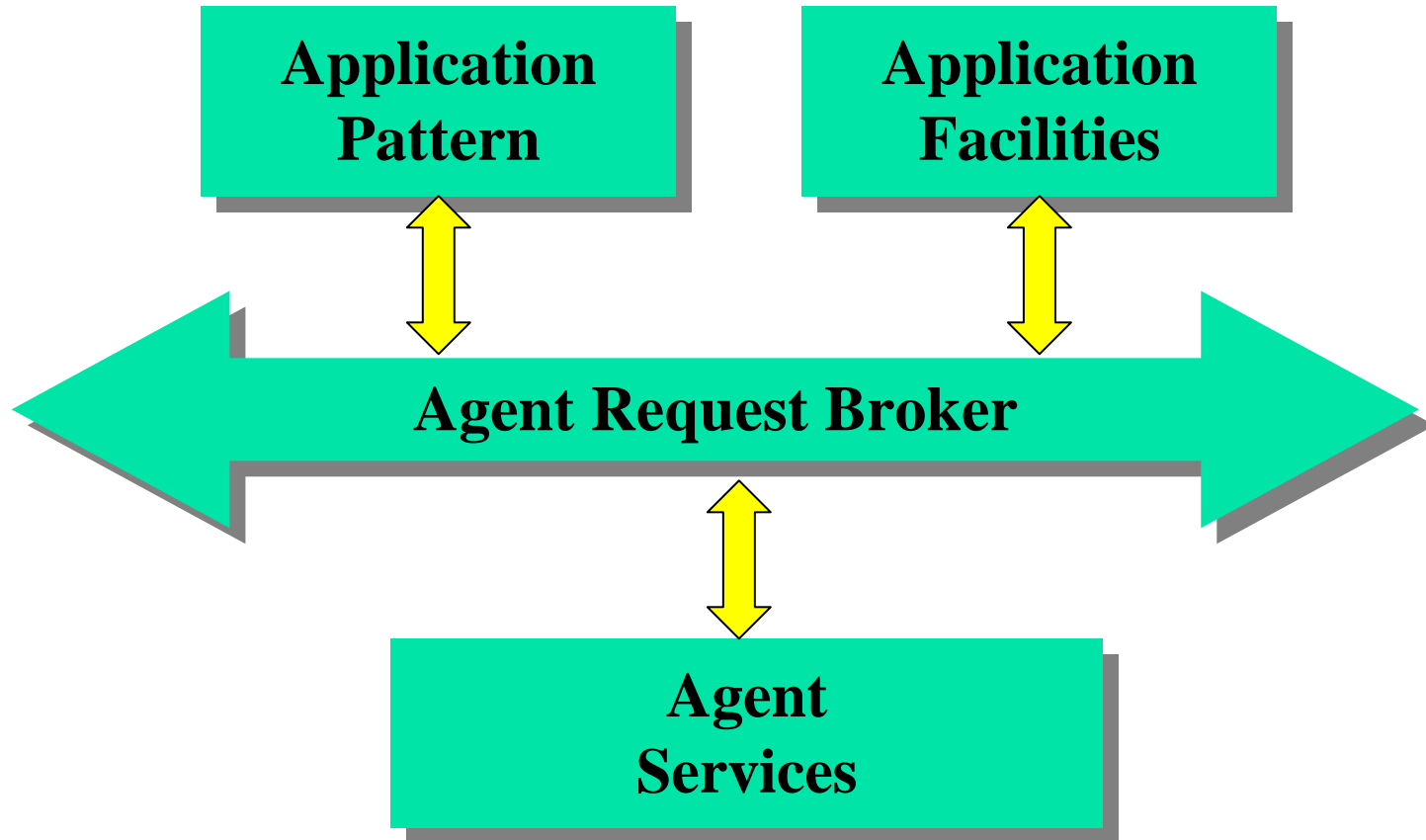


# Common Object Request Broker Architecture



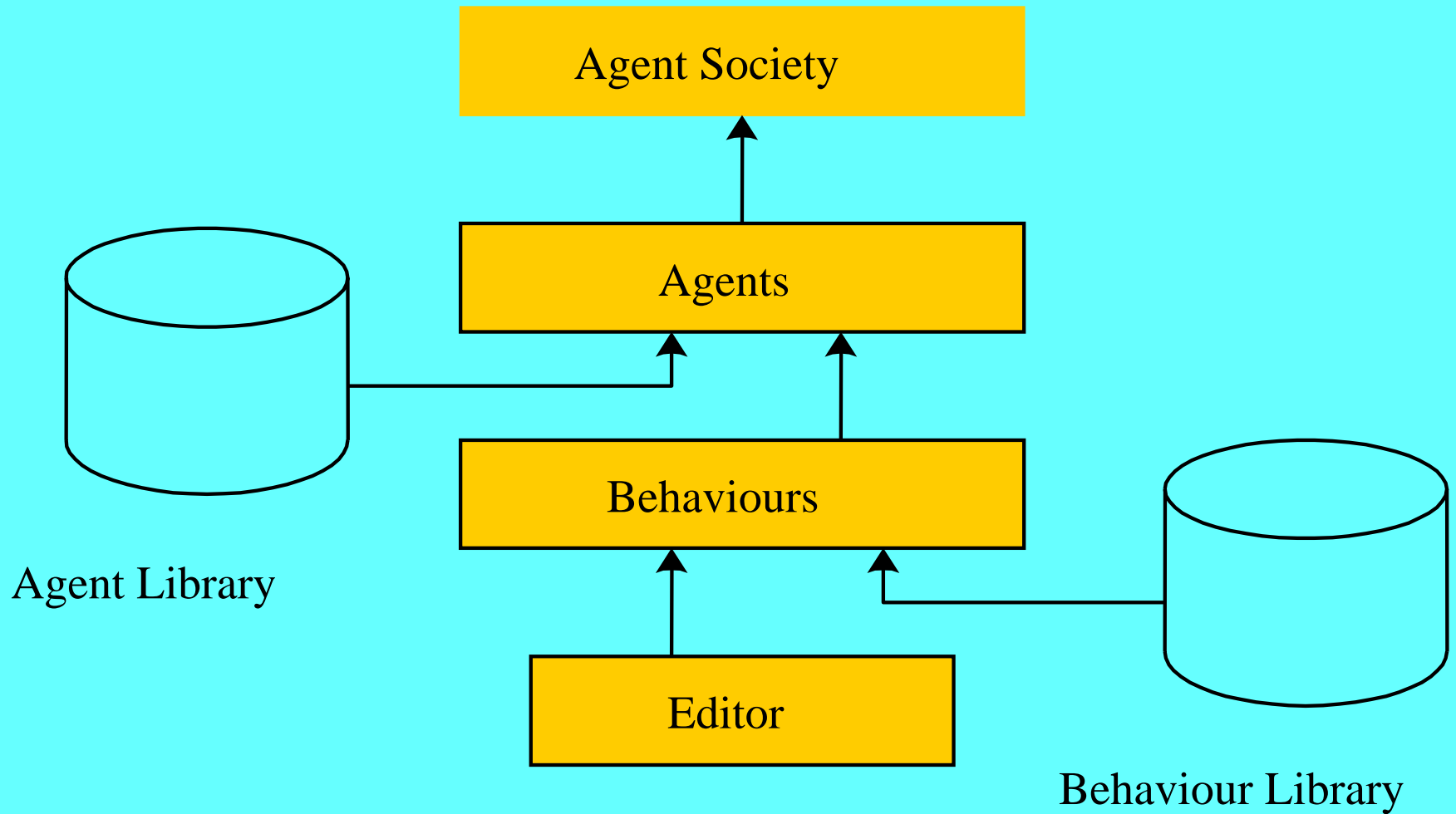


# Common Agent Request Broker Architecture



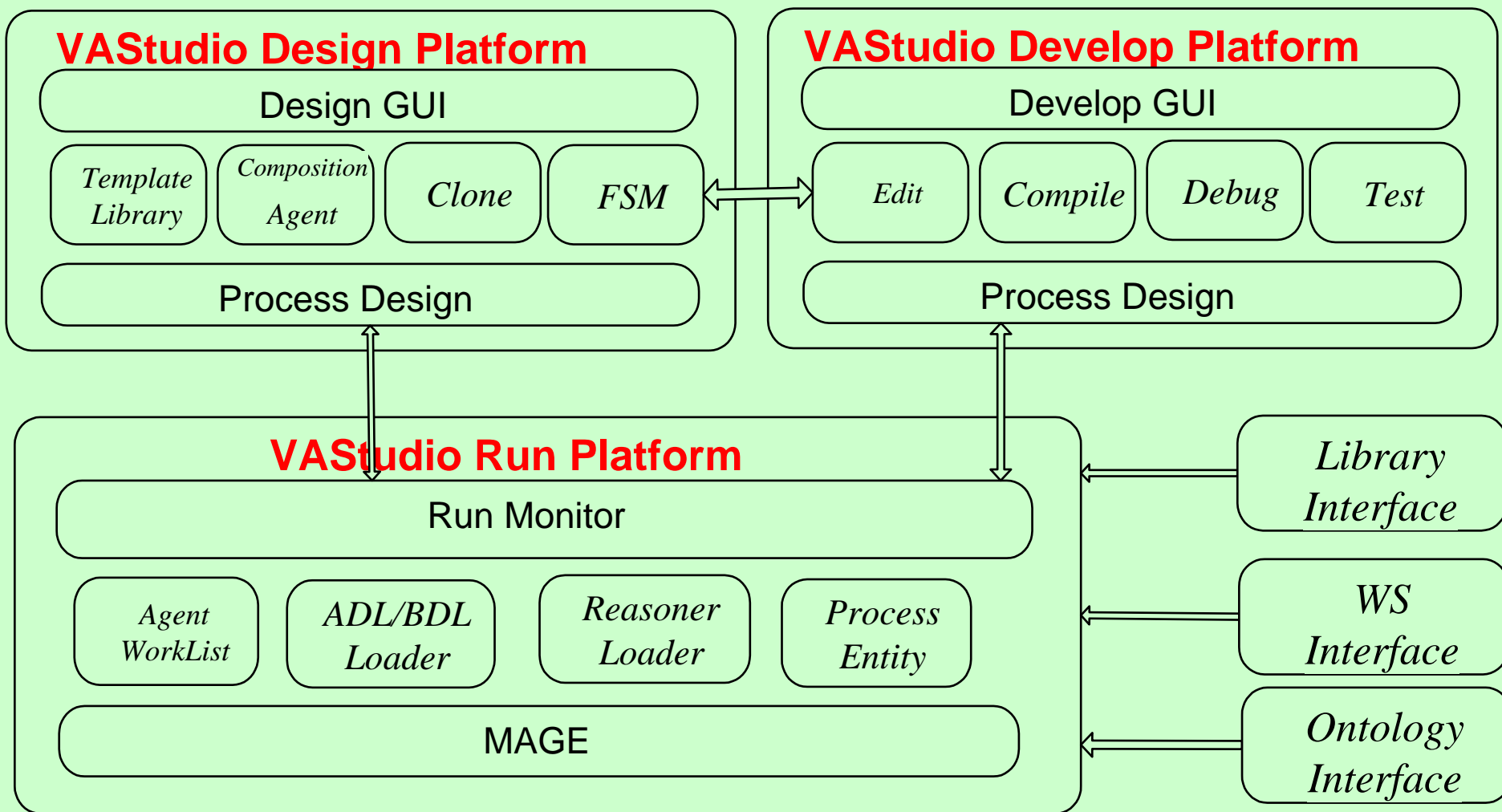


# VASstudio Architecture





# VASudio Architecture





# Behavior Library

- Data Package

```
package intsci.ace.data
```

- Neural Network Package

```
package intsci.ace.neural
```

- **Learning Package**

```
package intsci.ace.learning
```

- Data Mining Package

```
package intsci.ace.mining
```

- Language Processing Package

```
package intsci.ace.language
```



# Behavior Library

- **Graphic Package**  
`package intsci.ace.graphics`
- **Image Package**  
`package intsci. ace .image`
- **Search Package**  
`package intsci.ace.search`
- **Expert System Package**  
`package intsci.ace.expert`
- **Model Package**  
`package intsci.ace.model`
- **Decision Making Package**  
`package intsci.ace.decision`



# Screenshot of VASstudio

The screenshot displays the Visual Agent Studio (VASstudio) Ver3.0.00 interface. The main window is titled "Visual Agent Studio - VASstudio Ver3.0.00" and features a menu bar with options: File, Edit, Search, Insert, View, Behaviour, Agent, Society, Option, Tools, Window, and Help. Below the menu bar is a toolbar with various icons for file operations and development tools.

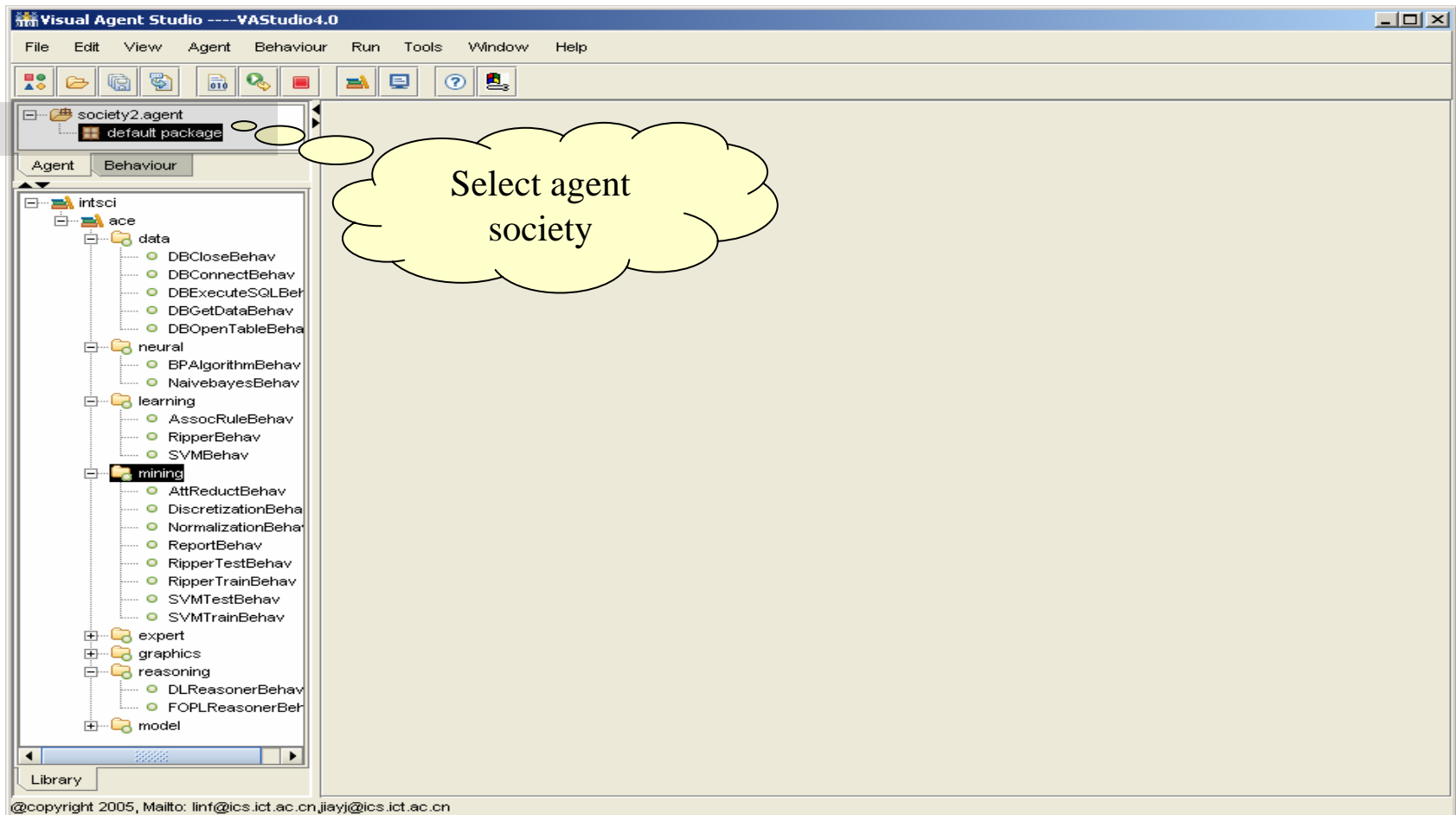
The interface is divided into several panes:

- Behaviours Pane:** Located on the left, it shows a tree view of behaviours: DBQueryBehav, DrawCircleBehav, DrawRectangleBehav (highlighted), and MyBehaviour.
- Code Editor:** The central area displays the code for "DrawRectangleBehav.java". The code includes a copyright notice for Key Lab of IIP, Kexueyuan South Road, Hai Dian, and a list of imports for VASstudio and standard Java classes (Applet, Swing, AWT, Util, SQL). A context menu is open over the code, listing actions such as Undo (Ctrl-Z), Redo (Alt-Z), Cut (Ctrl-X), Copy (Ctrl-C), Paste (Ctrl-V), Delete Line (Ctrl-D), Select All (Ctrl-A), Find & Replace (Ctrl-F), Save (Ctrl-S), Compile, and Close (Alt-F4).
- Console Pane:** At the bottom, it shows the current directory path: "E:\space\com\stu\vastudio\build\class>".
- Bottom Bar:** Contains tabs for Console, Compile, Methods, and Search.

At the bottom of the window, the status bar indicates "Current Window : DrawRectangleBehav" and "Line: 4/177 Column: 24:24 - 2%".

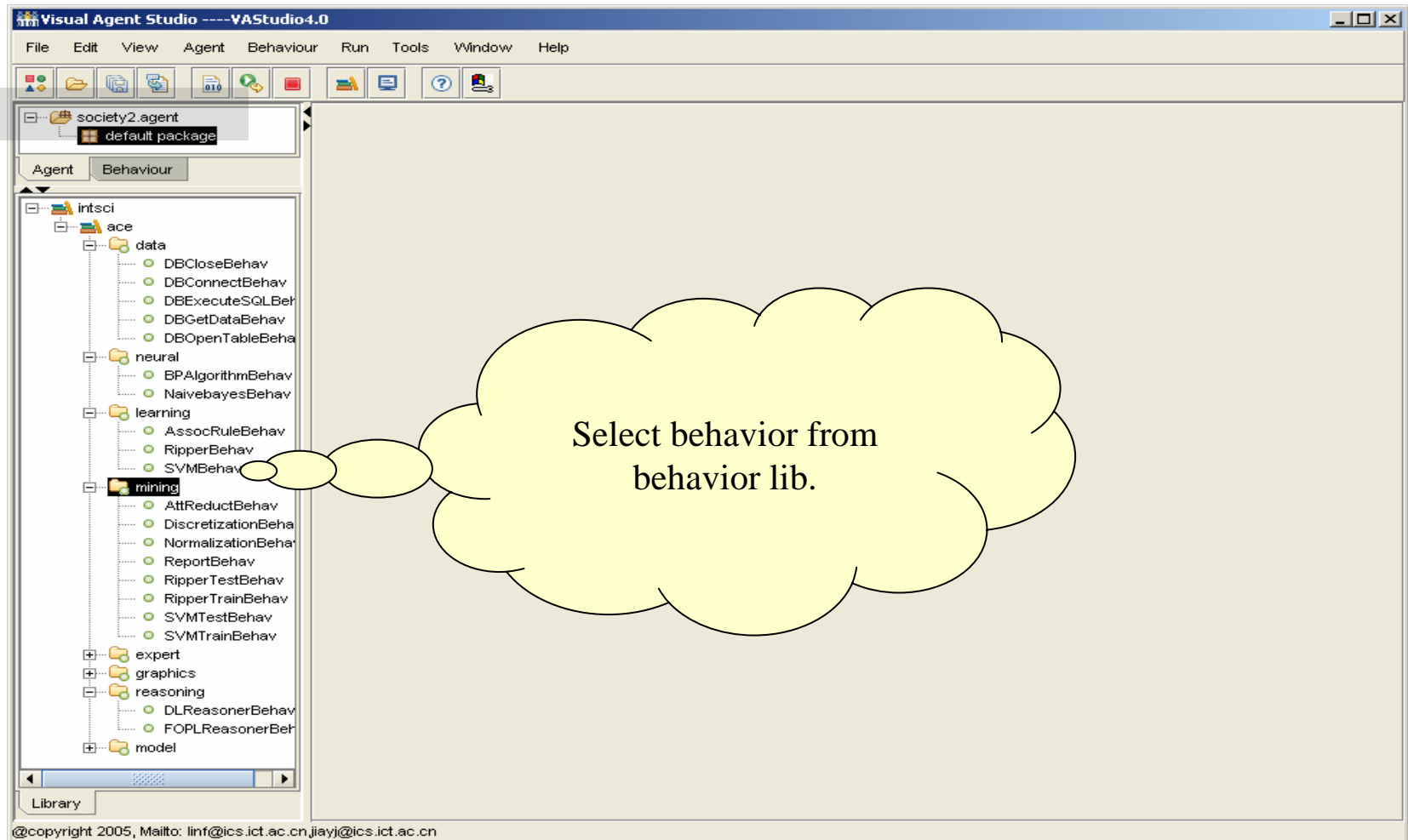


# VASstudio: Visual Agent Studio





# VASstudio: Visual Agent Studio





# Create Agent by VAStudio

Visual Agent Studio ---VAStudio4.0

File Edit View Agent Behaviour Run Tools Window Help

Agent1

VAStudio Template ...

Behaviour Information

This Behaviour aimed at drawing circle.  
There are three parameters:

1) parameter-1: X coordinate of the center of the circle;  
2) parameter-2: Y coordinate of the center of the circle;  
3) parameter-3: Radius of the circle.

Parameters

(---Please Select Parameter---)

Parameters Type

parameters Description

ACLMessage:  
Action: INFROM  
Content: "draw (Case Sensitive)"

Add template behaviour to the agent

all the behaviours that have been added to this agent

Design Source

@copyright 2005, Mailto: linf@ics.ict.ac.cn,jiayj@ics.ict.ac.cn



# Create Agent by VAStudio

Visual Agent Studio ----VAStudio4.0

File Edit View Agent Behaviour Run

Set Behaviour Parameters

Behaviour Information

3) parameter-3 Radius of the circle.

ACLMessage:

Action: INFROM

Content: "draw a circle"  
(Case Sensitive)

1 The first parameter is X co... 60

2 The second is Y coordin... 80

3 The third is radius (int) 50

OK Cancel

Fill out behavior parameters

Content: "draw a circle"  
(Case Sensitive)

Parameters

(---Please Select Parameter---)

Parameters Type

parameters Description

all the behaviours that have been added to this agent  
intsci.ace.graphics.DrawCircleBehav

Design Source

@copyright 2005, Mailto: linf@ics.ict.ac.cn,jiayj@ics.ict.ac.cn



# Create Agent by VASstudio

```
55  * execution model is still sequential and no behaviour scheduling.
56  * is active yet..
57  *.
58  * This method can be used for ordinary startup tasks such as.
59  * <code>DF</code> registration, but is essential to add at least a
60  * <code>Behaviour</code> object to the agent, in order for it to be
61  * able to do anything..
62  * @see mage.core.Agent#addBehaviour (Behaviour b).
63  * @see mage.core.behaviours.Behaviour.
64  */.
65  protected void setup() { //VASTUDIOAGENTBOOKMARK: setup.
66
67      try{
68          Class c = Class.forName("intsci.ace.graphics.DrawCircleBehav");.
69          Class tmpAgent = Class.forName("mage.core.Agent");.
70          Class[] para = {tmpAgent};.
71          java.lang.reflect.Constructor con = c.getConstructor(para);.
72          mage.core.Agent[] as = {this};.
73          vastudio.spec.InterfaceSpec is = (vastudio.spec.InterfaceSpec)con.newInstance(as);.
74          is.setParameter(0, ""+80");.
75          is.setParameter(1, ""+80");.
76          is.setParameter(2, ""+50");.
77          this.addBehaviour((Behaviour)is);.
78      }.
79      catch (Exception e){.
80          e.printStackTrace();.
81      }.

```

Build agent (code generation)

@copyright 2005, Mailto: linf@ics.ict.ac.cn,jiayj@ics.ict.ac.cn



# Build Multiagent System by VASstudio

Visual Agent Studio ---- VASstudio4.0

File Edit View Agent Behaviour Run Tools Window Help

InitAgent

netprotocol.HandleProposeBehav

netprotocol.HandleOtherMsgBe...

(PROPOSE) (NOT-UNDERSTOOD)

Auction protocol using FSM

@copyright 2005, Mailto: linf@ics.ict.ac.cn jiajy@ics.ict.ac.cn



# Build Multiagent System by VASstudio

Seller and buyer using clone

@copyright 2005, Mailto: linf@ics.ict.ac.cn,jiayj@ics.ict.ac.cn



# Build Multiagent System by VASstudio

Visual Agent Studio ----VASstudio4.0

File Edit View Agent Behaviour Run Tools Window Help

Init.Agent Res.Agent1 Res.Agent2

Init.Agent\*

netprotocol.InformStartOfAuctionBehav

(PROPOSE) (NOT-UNDERSTOOD)

netprotocol.HandleProposeBehav netprotocol.HandleOtherMsgB

Compile ok, no errors or warnings!

Compile Prompt Console Search

@copyright 2005, Mailto: linf@ics.ict.ac.cn,jiayj@ics.ict.ac.cn



# Build Multiagent System by VASstudio

Visual Agent Studio --- VASstudio4.0

File Edit View Agent Behaviour Run Tools Window Help

InitAgent ResAgent1 ResAgent2

InitAgent\*

Deploy auction to run

Please input the container number used in your Agent society, the default value is Main Container: 1

Main Container: ams change

Agent Adjustment between Containers

All Agents that can be added to this contai... All Agents in this container

ams  
rma  
df  
auction.agent.InitAgent  
auction.agent.ResAgent1  
auction.agent.ResAgent2

formStartOfAuctionBehav

(BE) (NOT-UNDERSTOOD)

behav netprotocol.HandleOtherMsgB

Compile ok, no errors or warnings!

Compile Prompt Console Search

@copyright 2005, Mailto: linf@ics.ict.ac.cn,jiayj@ic

Run the auction system in MAGE



# Build Multiagent System by VASstudio

The screenshot displays the MAGE Remote Agent Management GUI. The window title is "RMA@linfen:5689/MAGE - MAGE Remote Agent Management GUI". The menu bar includes "File", "Actions", "Tools", "Platforms", and "Help". The toolbar contains various icons for agent management. The left sidebar shows a tree view of "AgentPlatforms" with a sub-entry for "linfen:5689/MAGE" containing a "Main-Container" with several agent instances, including "auction.agent.R...". The main area shows the configuration for "Name: auction.agent.ResAgent2" and "Class: auction.agent.ResAgent2". Below this, there are sections for "Current State" and "Sent/Received messages". The "Sent/Received messages" section is active, showing a table of messages.

*	Time	Performative	Sender/Receiver	Content
	2006-03-10 1...	REQUEST	ams@linfen:...	((action (age...
	2006-03-10 1...	AGREE	ams@linfen:...	((action (age...
	2006-03-10 1...	INFORM	ams@linfen:...	((done (actio...
	2006-03-10 1...	INFORM	auction.agent...	inform-start...
	2006-03-10 1...	CFP	auction.agent...	2000
	2006-03-10 1...	PROPOSE	auction.agent...	2789
	2006-03-10 1...	ACCEPT-PR...	auction.agent...	accept-propo...

Display the agent running procedure

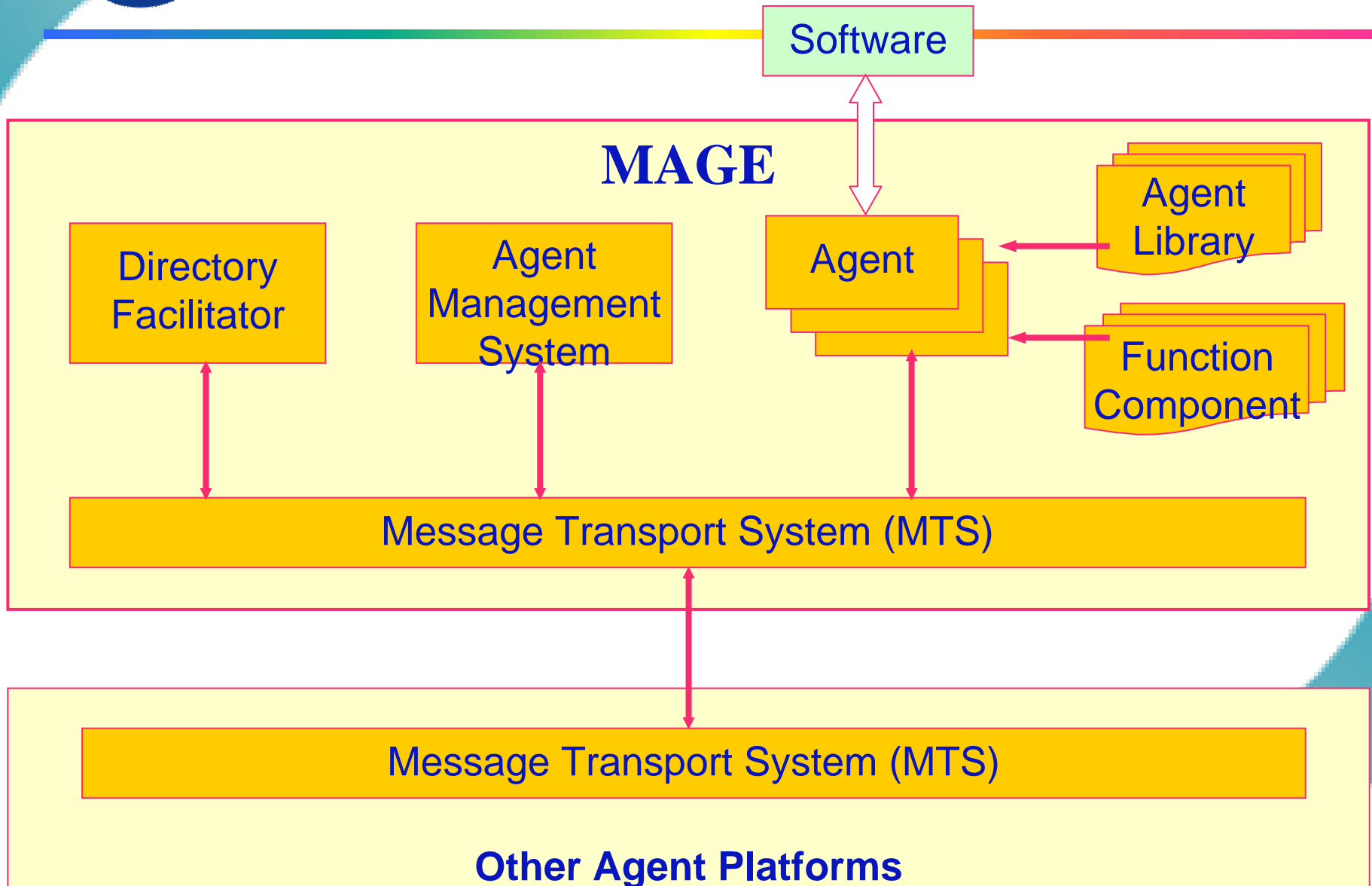


# Build Multiagent System by VASstudio

The screenshot displays the MAGE Remote Agent Management GUI. The window title is "RMA@linfen:5689/MAGE - MAGE Remote Agent Management GUI". The interface includes a menu bar (File, Actions, Tools, Platforms, Help), a toolbar with various icons, and a "MAGE" logo in the top right. On the left, a tree view shows the "AgentPlatforms" structure, including "linfen:5689/MAG" and "Main-Container" with several "auction.ag" agents. The main area shows the configuration for an agent named "auction.agent.InitAgent" with Class "auction.agent.InitAgent". Below this, the "Current State" section shows the agent is "Active". A "Sent/Received messages" table is visible, showing a sequence of messages including REQUEST, AGREE, INFORM, CFP, PROPOSE, ACCEPT-PRO, and REJECT-PROP. A yellow callout bubble with the text "Seller agent running" is overlaid on the interface.

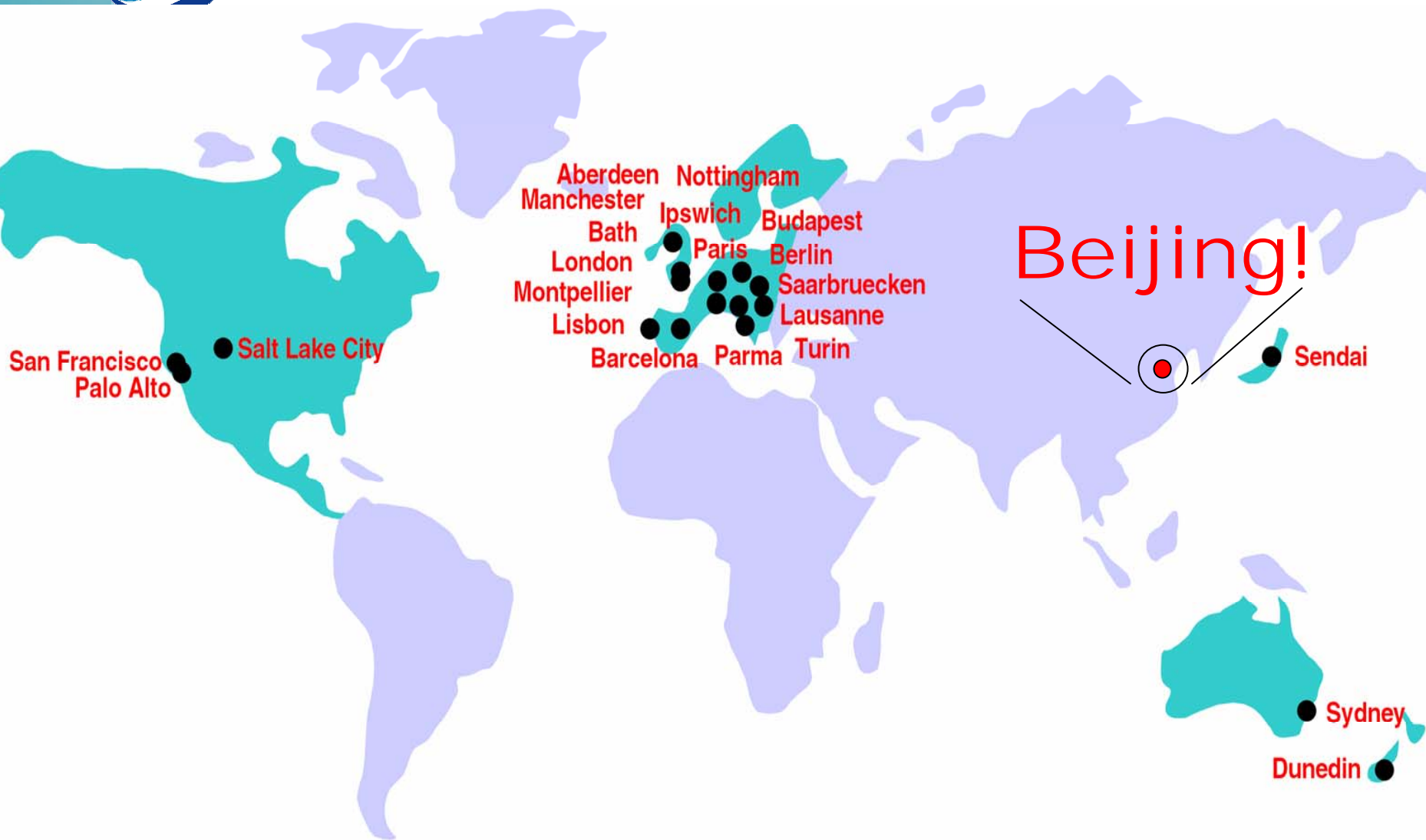
*	Time	Performative	Sender/Receiver	Content
	200...	REQUEST	ams@linfen:5689/MAGE;	((action...
	200...	AGREE	ams@linfen:5689/MAGE	((action...
	200...	INFORM	ams@linfen:5689/MAGE	((done (...
	200...	INFORM	auction.agent.ResAgent1...	inform...
	200...	CFP	auction.agent.ResAgent1...	2000
	200...	PROPOSE	auction.agent.ResAgent2...	2789
	200...	PROPOSE	auction.agent.ResAgent1...	2533
	200...	ACCEPT-PRO...	auction.agent.ResAgent2...	accept...
	200...	REJECT-PROP...	auction.agent.ResAgent1...	reject-p...

# Run-Time Platform





# Agent Cities



# MAGE Comparison



Agent Platform Phase		AgentBuilder	Jack	Zeus	MAGE
Analysis	Completeness:	★★★	★★★	★★★★	★★★★★
	Applicability:	★★★	★★★	★★★	★★★★
	Complexity:	★★★★★	★★★	★★★★★	★★★★
	Reusability	★★★	★★★	★★★	★★★★★
Design	Completeness:	★★★	★	★★★★	★★★★★
	Applicability:	★★★	★	★★★	★★★★
	Complexity:	★★★★★	★	★★★	★★
	Reusability	★★	★★	★★★	★★★★
Development	Completeness:	★★★★★	★★★★	★★★★★	★★★★★
	Applicability:	★★★	★★★★★	★★★	★★★★
	Complexity:	★★★★★	★★	★★★★	★★★
	Reusability	★★	★★★★★	★★	★★★
Deployment	Completeness:	★★★	★★★	★★★★	★★★★★
	Applicability:	★★★	★★★★	★★★	★★★
	Complexity:	★★★★★	★★★	★★★★	★★★★★
	Reusability				★



# Outline

- Introduction
- Agent Model
- Multiagent Environment— MAGE
- **Agent Collaboration**
- **Agent Grid Intelligence Platform**
- Applications
- Conclusions



# Agent Collaboration

- **ACL**
- **Working Flow**
- **Ontology-based Knowledge Management**
- **Policy Driven**
- **Planning**

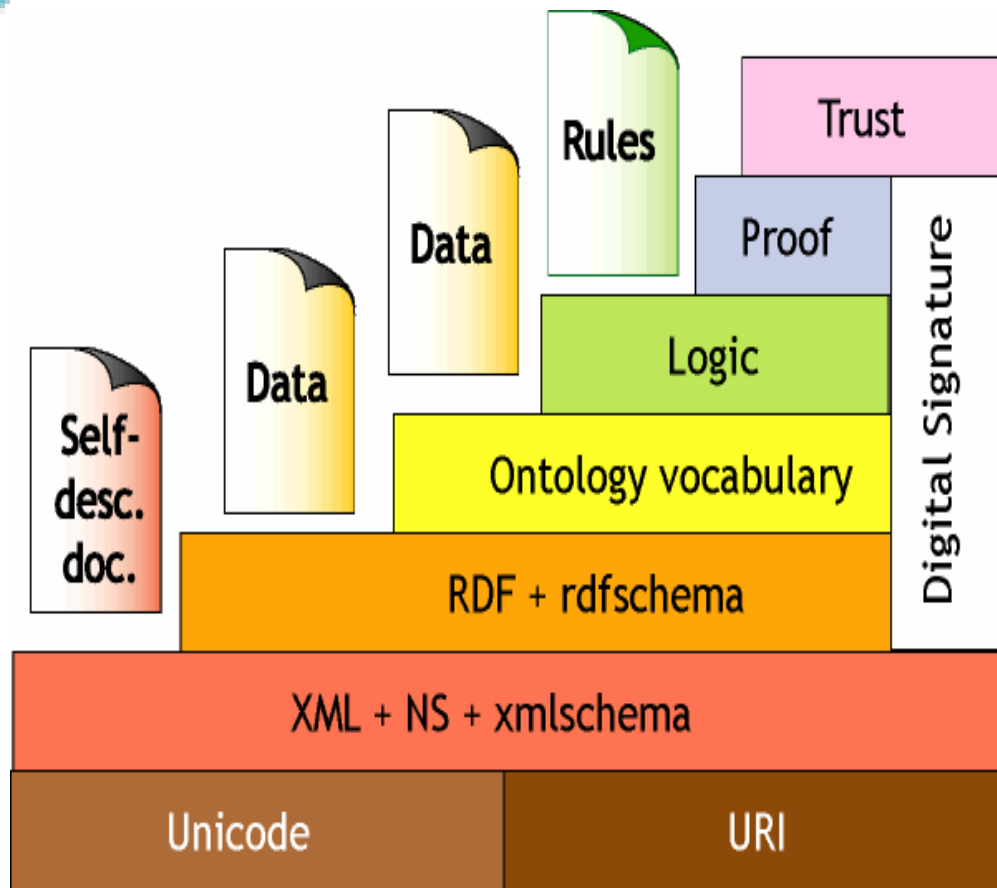


# Ontology

- In philosophy, an ontology is a theory about the nature of existence.
- An ontology is a document or file that formally defines the relations among terms.
- An ontology is a *formal, explicit* specification of a *shared conceptualization*.
- The most typical kind of ontology for the Web has a taxonomy and a set of inference rules.



# Semantic Web Layer Cake

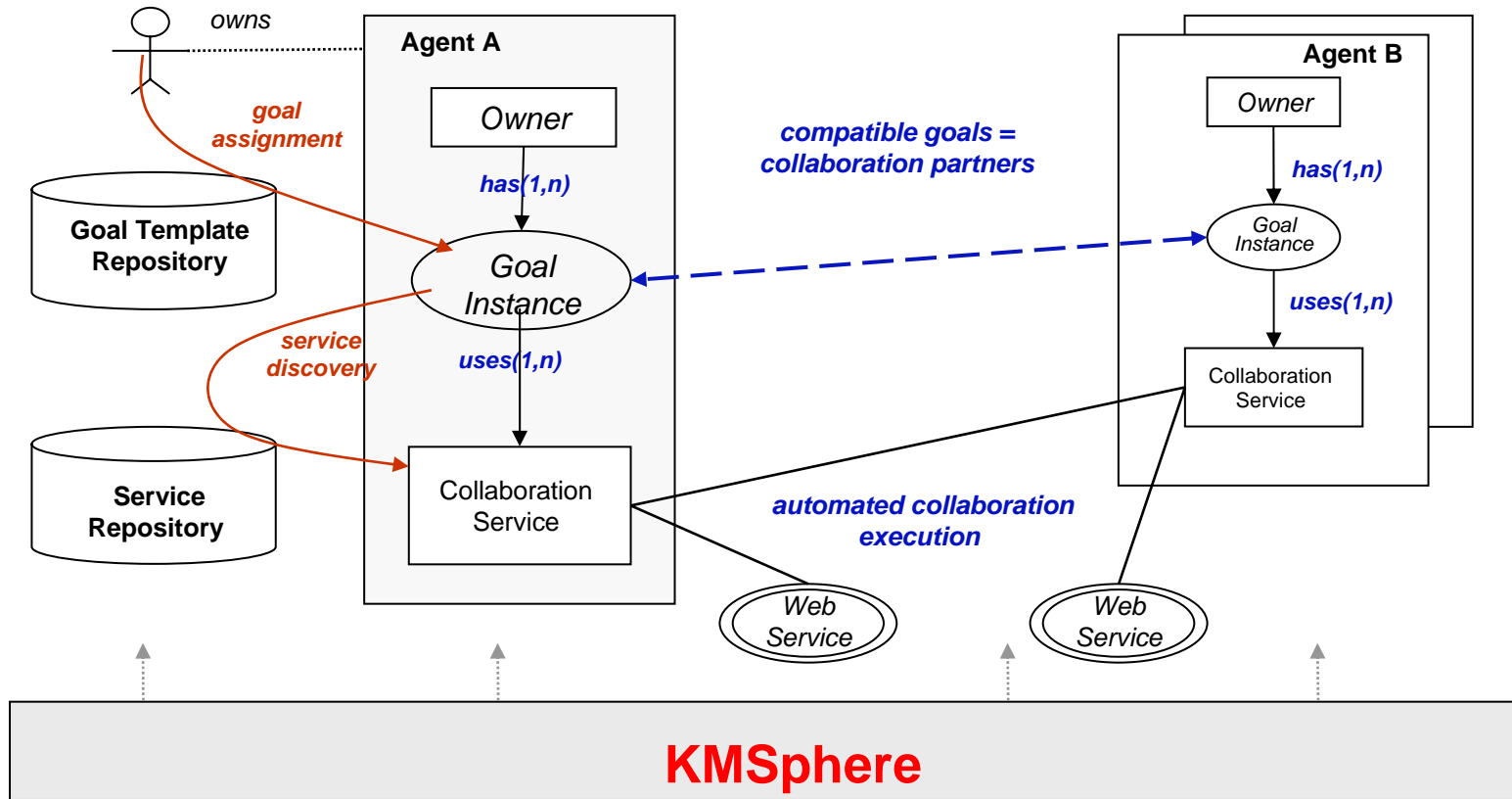


- 10 Feb 2004
- W3C
- Recommendation
- OWL

by Tim Berners-Lee

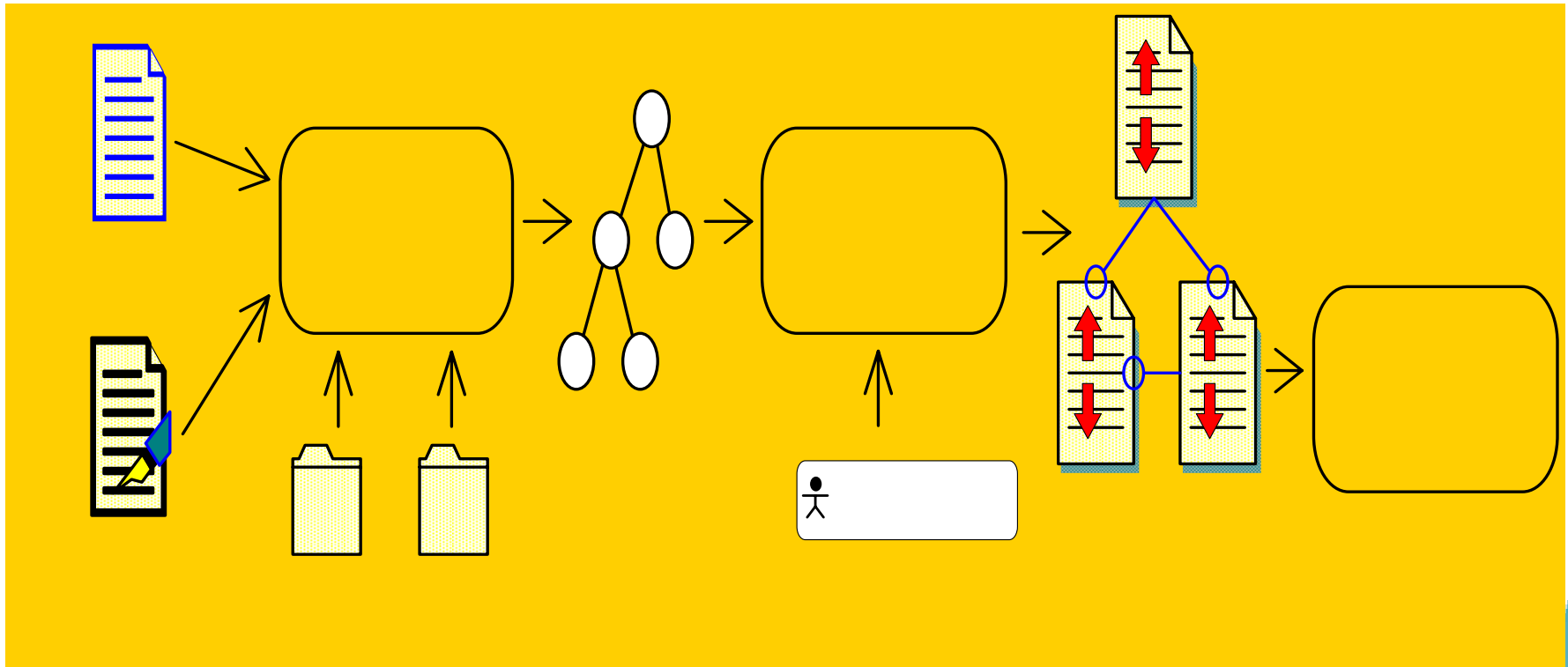


# Collaborative Architecture





# KMSphere

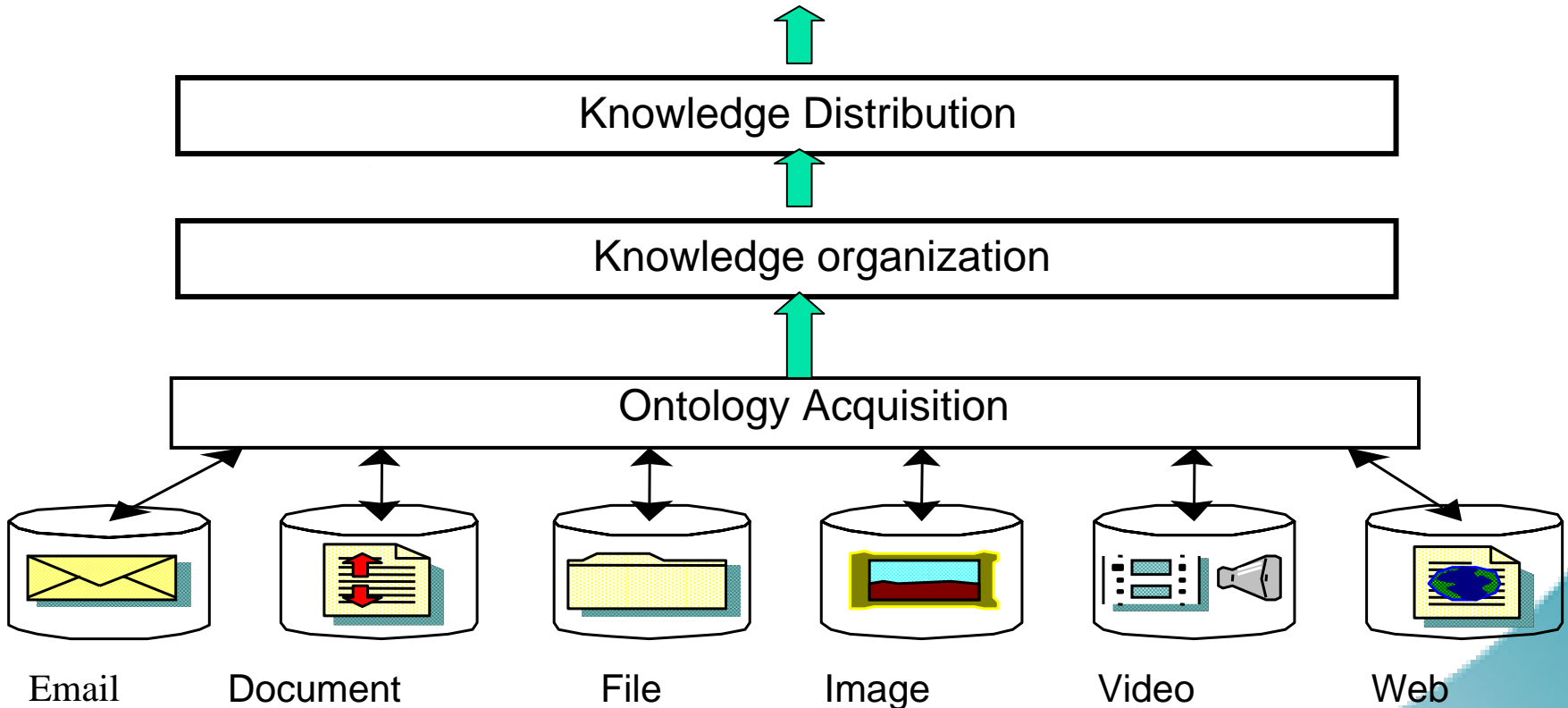




# KMSphere

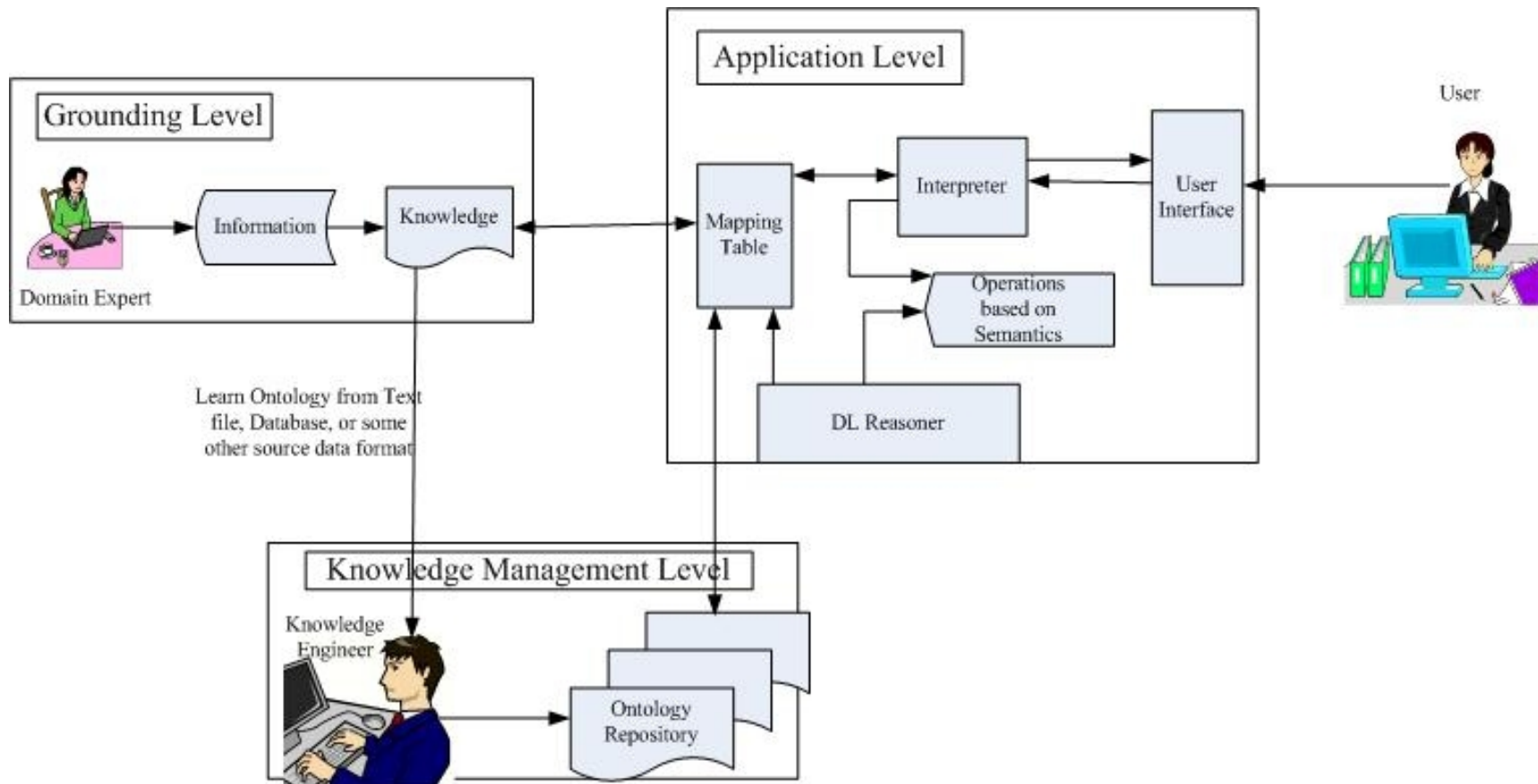


Knowledge  
Application



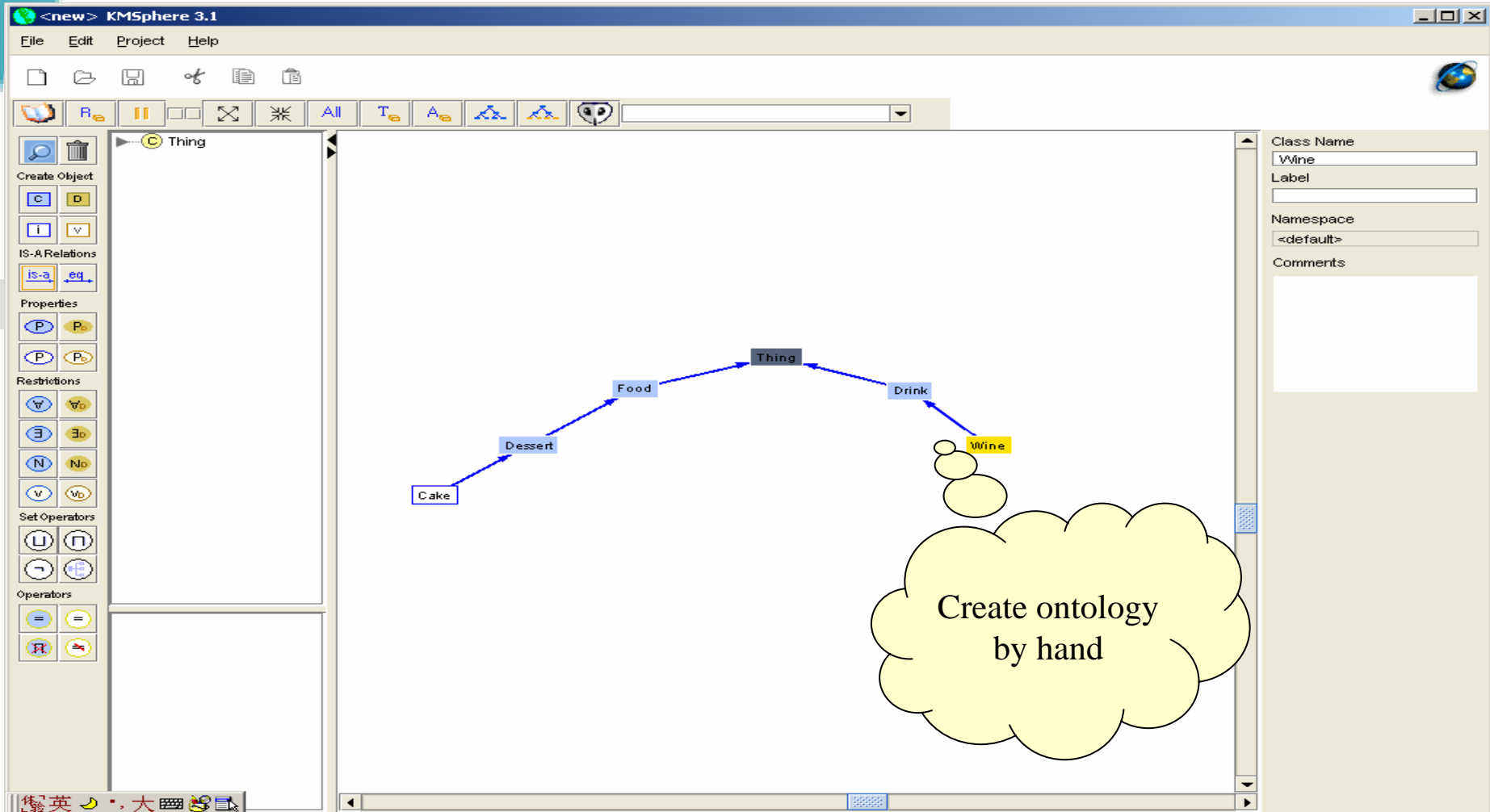


# KMSphere





# KMSphere Demo





# 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	HOTELNAME	
ENTERTAINMENT	HOTELADDRESS	
旅馆	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

中国科学院计算技术研究所智能信息处理重点实验室



# KMSphere Demo

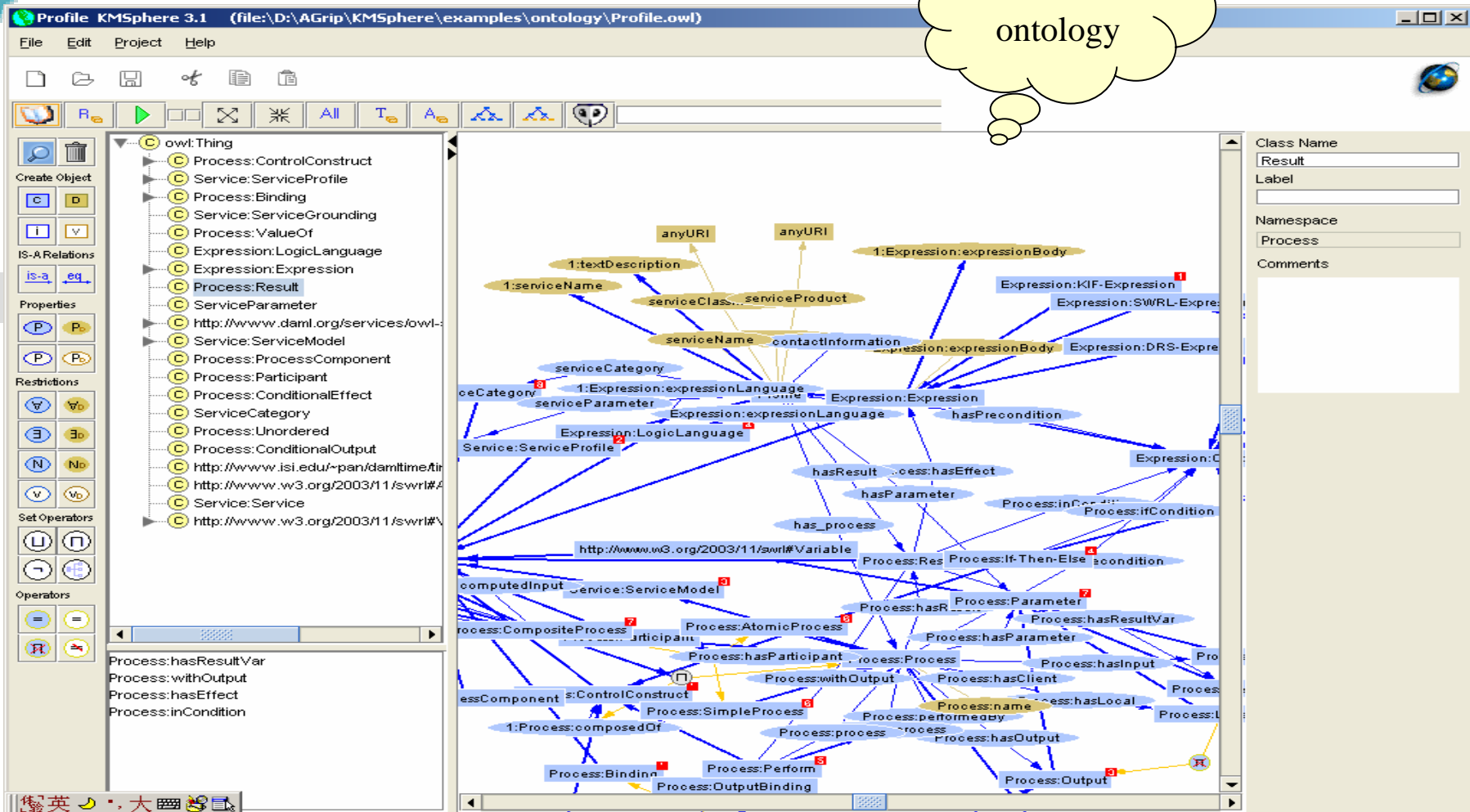
The screenshot displays the KMSphere 3.1 web-based ontology editor. The main window title is "food KMSphere 3.1 (http://www.w3.org/2001/sw/WebOnt/guide-src/food)". The interface includes a menu bar (File, Edit, Project, Help), a toolbar with various icons, and a left sidebar with tool categories: Create Object, IS-A Relations, Properties, Restrictions, Set Operators, and Operators. A central workspace contains two windows: "基于文本的半自动本体获取" (Semi-automatic ontology acquisition from text) and "规则" (Rules). A yellow thought bubble with the text "Ontology acquisition from text" is positioned over the main workspace. On the right, a panel shows ontology metadata fields: Class Name (Thing), Label, Namespace, owl, and comments. The "规则" window contains a table with columns "Name" and "Topic".

Name	Topic



# 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 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.



# 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 interface includes a menu bar (File, Edit, Project, Help), a toolbar, and a left-hand sidebar with sections for "Create Object", "IS-A Relations", "Properties", "Restrictions", "Set Operators", and "Operators".

A yellow callout bubble is positioned over the main query area, containing the text: "RDQL (RDF Data Query Language)".

The main query area contains the following text:

```
http://www.w3.org/2001/sw/WebOnt/guide-src/food  
food:Meat)  
http://www.w3.org/TR/2003/CR-owl-guide-20030818/food#>,  
http://www.w3.org/2002/07/owl#>,  
http://www.w3.org/1999/02/22-rdf-syntax-ns#>,  
rdfs:FOR http://www.w3.org/2000/01/rdf-schema#>,  
wine FOR <http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine#>
```

Below the query is a "Run Query" button. The "Query Results:" section displays the following output:

```
Meat  
-----  
food:RoastBeef  
food:Pork  
food:Veal  
food:GarlickyRoast  
food:BeefCurry  
food:Steak
```

On the right side of the interface, there is a panel with the following fields:

- Class Name: Thing
- Label: [empty]
- Namespace: owl
- Comments: [empty]

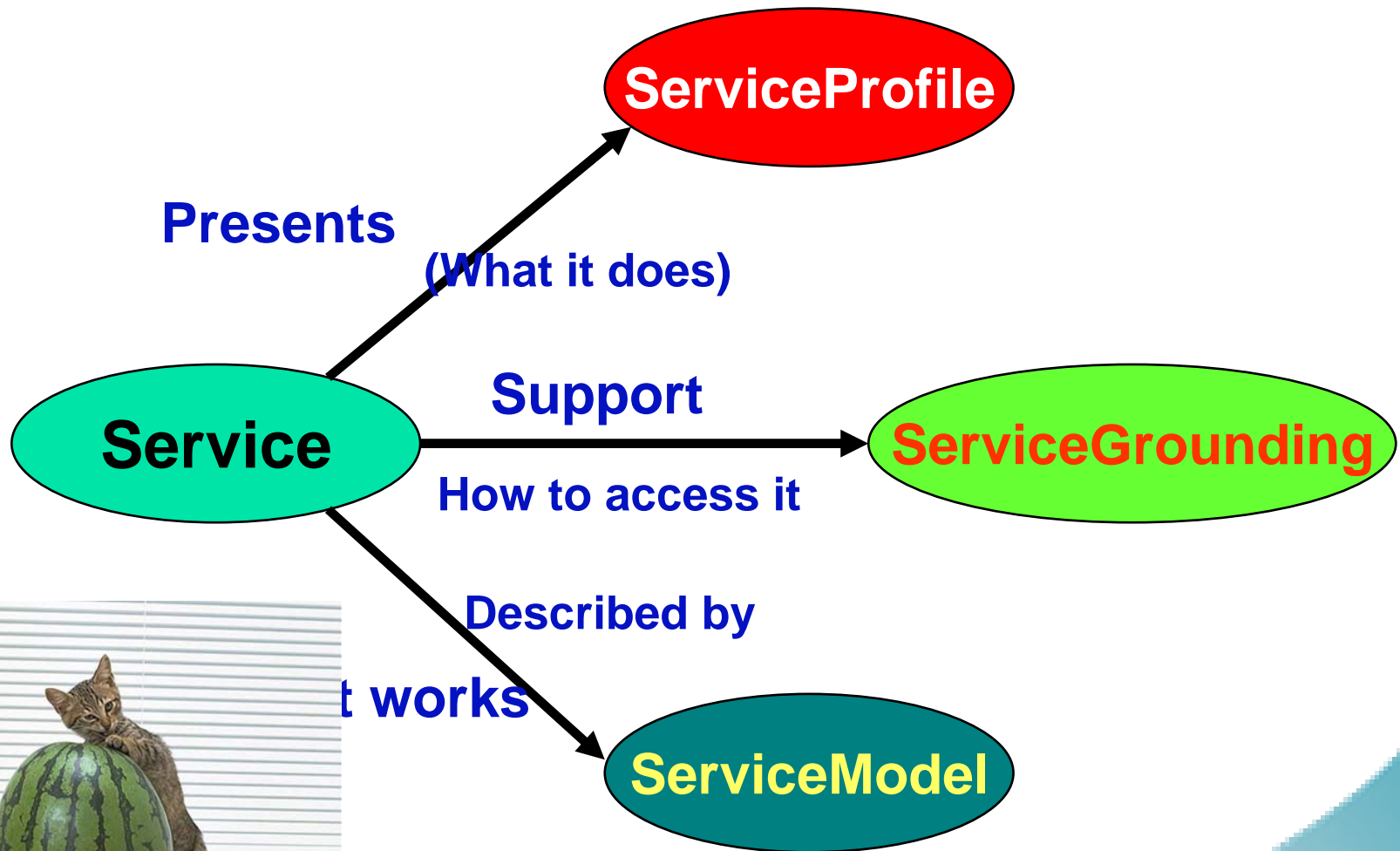


# Semantic Web Services

- 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)**

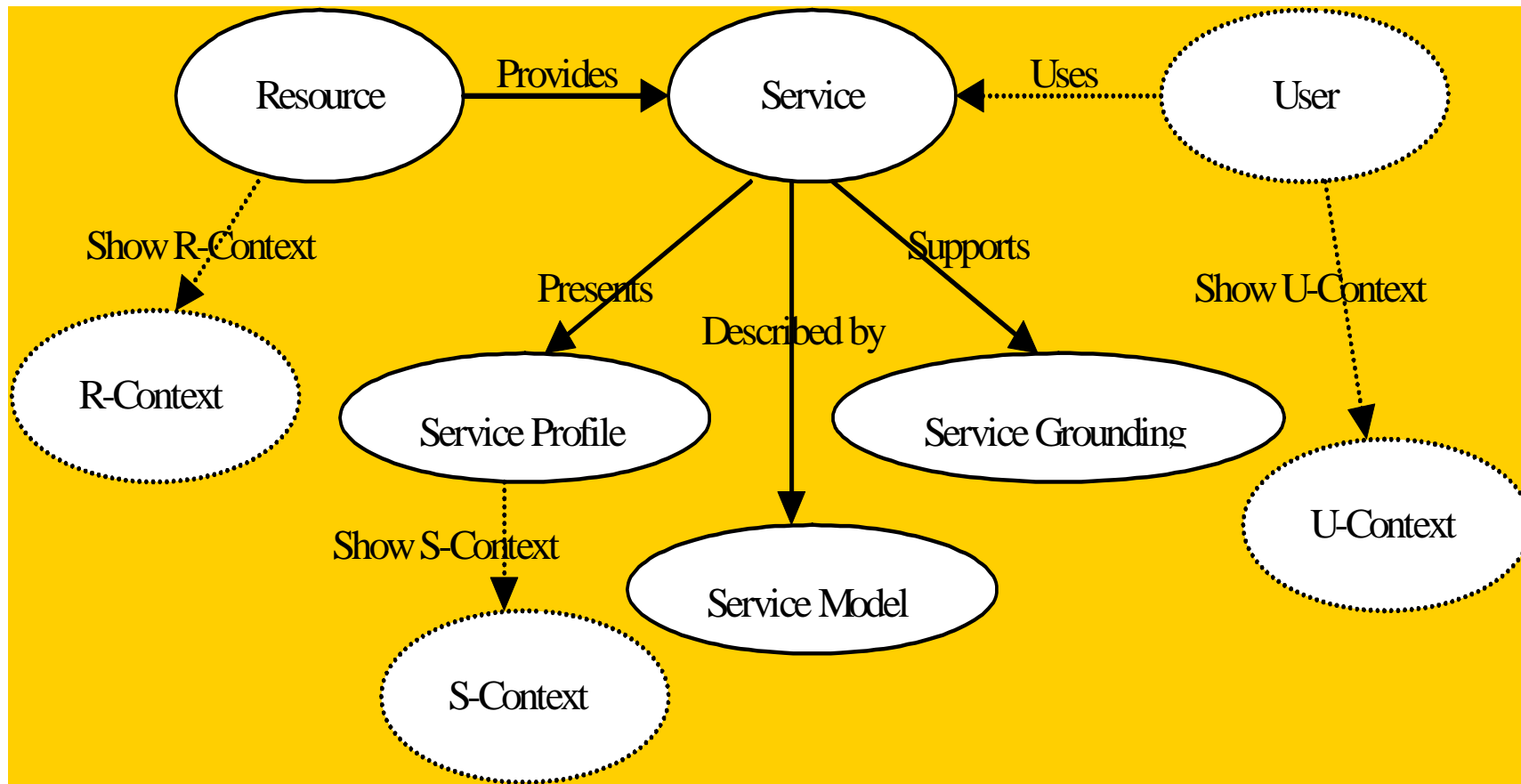


# OWL-S





# OWL-S Context





# Service Description Language SDLSIN

```
<asdl-descr> ::= (ctype
  :service-name name
  :context context-name+
  :types (type-name = <modifier> type)+
  :isa name+
  :inputs (variable: <modifier> put-type-name)+
  :outputs (variable: <modifier> put-type-name)+
  :input-constraints (constraint)+
  :output-constrains (constraint)+
  :io-constrains (constraint)+
  :concept-description (ontology-name = ontology-body)+
  :state-language name
  :concept-language name
  :attributes (attributes-name : attributes-value)+
  :text-description name
)
```



# Policy Driven

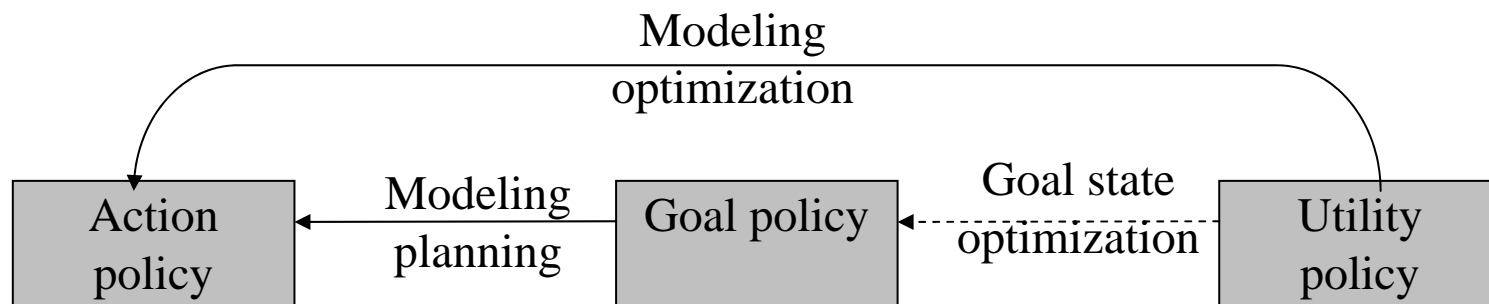
$$p = \langle S_{\text{trigger}}, A, S_{\text{goal}}, U \rangle$$

$S_{\text{trigger}}$ : set of states for policy execution

$A$ : set of actions

$S_{\text{goal}}$ : set of goals

$U$ : set of utility functions





# Action Policy

<ActionPolicy> ::= =

<Name><Parent><Performative><Type><Subject>

<Object><Action><Precondition><ConstraintLanguage>

<Duration><Priority>

- Access Control Policy
- Obligation Policy



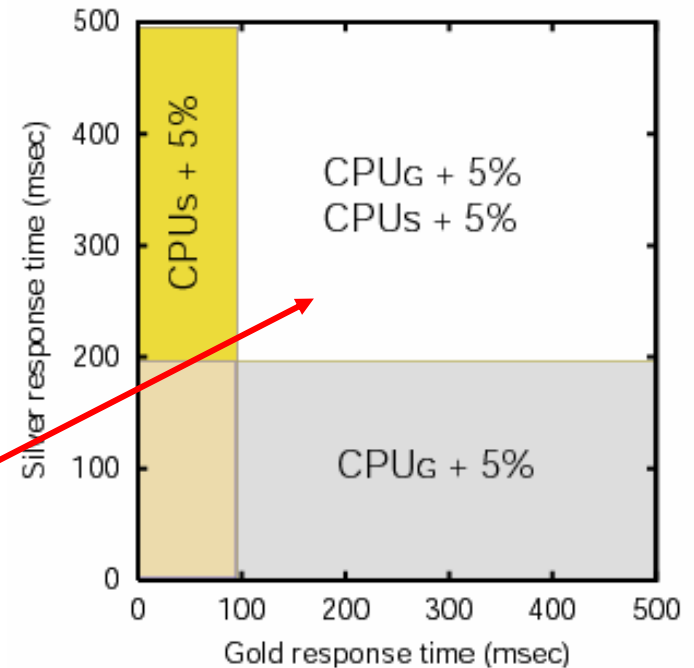
# Action Policy

**G:** IF ( $RT_G > 100$  msec)  
THEN (Increase  $CPU_G$  by 5%)

**S:** IF ( $RT_S > 200$  msec)  
THEN (Increase  $CPU_S$  by 5%)

**Overlapping Action Policies**

**Conflict if CPU (almost) fully utilized!**





# Goal Policy

**<GoalPolicy> ::=**

**<Name><Performative><Subject><Object><Precondition><Postcondition><ConstraintLanguage><Duration><Priority>**

```
(  
  :name goalpolicy1  
  :performative "Achieve"  
  :subject ftpagent  
  :precondition between(clienthost, 192.168.0.0,  
192.168.0.255)  
  :postcondition greater(bandwidth(clienthost), 1M)  
)
```



# Optimization

<UtilityPolicy> ::=

<Name><Performative><Subject><Object><Precondition><UtilityFunction><ConstraintLanguage><Duration><Priority>

```
(  
  :name utilitypolicy1  
  :performative "Optimize"  
  :subject httpagent  
  :utilityfunction  $f = w_1 \sum_i bandwidth(client_i) + w_2 \sum_i responsetime(client_i)$   
)
```



# Plan Representation

A plan  $\Pi$  is a triple  $\langle SO, OO, CS \rangle$

*SO*: a set of action-steps,

*OO*: a set of ordering constraints on the actions in *SO*.

*CS*: a set of variable-binding constraints between the variables of the action-steps in *SO* and other variables or constants.



# Distributed Multiagent Plan Algorithm

*Distributed – MultiAgent – Plan*( $OPS, CaS, I, G$ )

Input: a set of action-templates  $OPS$ ,  
a set of capabilities of agents  $CaS$ ,  
initial state  $I$ , and  
goal  $G$

Output: Plan  $\Pi$

# Distributed Multiagent Plan Algorithm

Procedure:

1. Inform all agents that I will start planning, and wait for them finishing current actions and entering plan state.
2. Initialize: Build an empty plan  $\Psi = \langle SO, OO, CS, CLS \rangle$  where  $SO = \{Init, Goal\}$ ,  $OO = \phi$ ,  $CS = \phi$ ,  $CLS = \phi$ . Decompose goal  $G$  into a set of sub-goals  $GS$  through adding  $\langle g, Goal \rangle$  to  $GS$  according to every literal  $g \in G$ .
3. Invoke recursive function  $\varphi = Generate - Next - Step(\Psi, GS)$
4. Taking effect: For each action executed by other agent, request it to add the action. For each constraint related to the actions executed by other agents, request it them to add the constraints.
5. Inform all agents the planning finishes, thus they can continue their plans.
6. If the invoking succeeds, then return success, otherwise return fault.



# Outline

- Introduction
- Agent Model
- Multiagent Environment— MAGE
- Agent Collaboration
- **Agent Grid Intelligence Platform**
- Applications
- Conclusions



# Agent Grid

**Applications Service**

IMS

Power Supply

Environment

Biology

E-Business

**Developing Toolkits**

Distributed Computing Toolkit

Data-Intensive Applications Toolkit

Collaborative Applications Toolkit

Remote Visualization Applications Toolkit

Problem Solving Applications Toolkit

Remote Instrumentation Applications Toolkit

**Agent Environment**

Resource-independent and application-independent services

authentication, authorization, resource location, resource allocation, events, accounting, remote data access, information, policy, fault detection

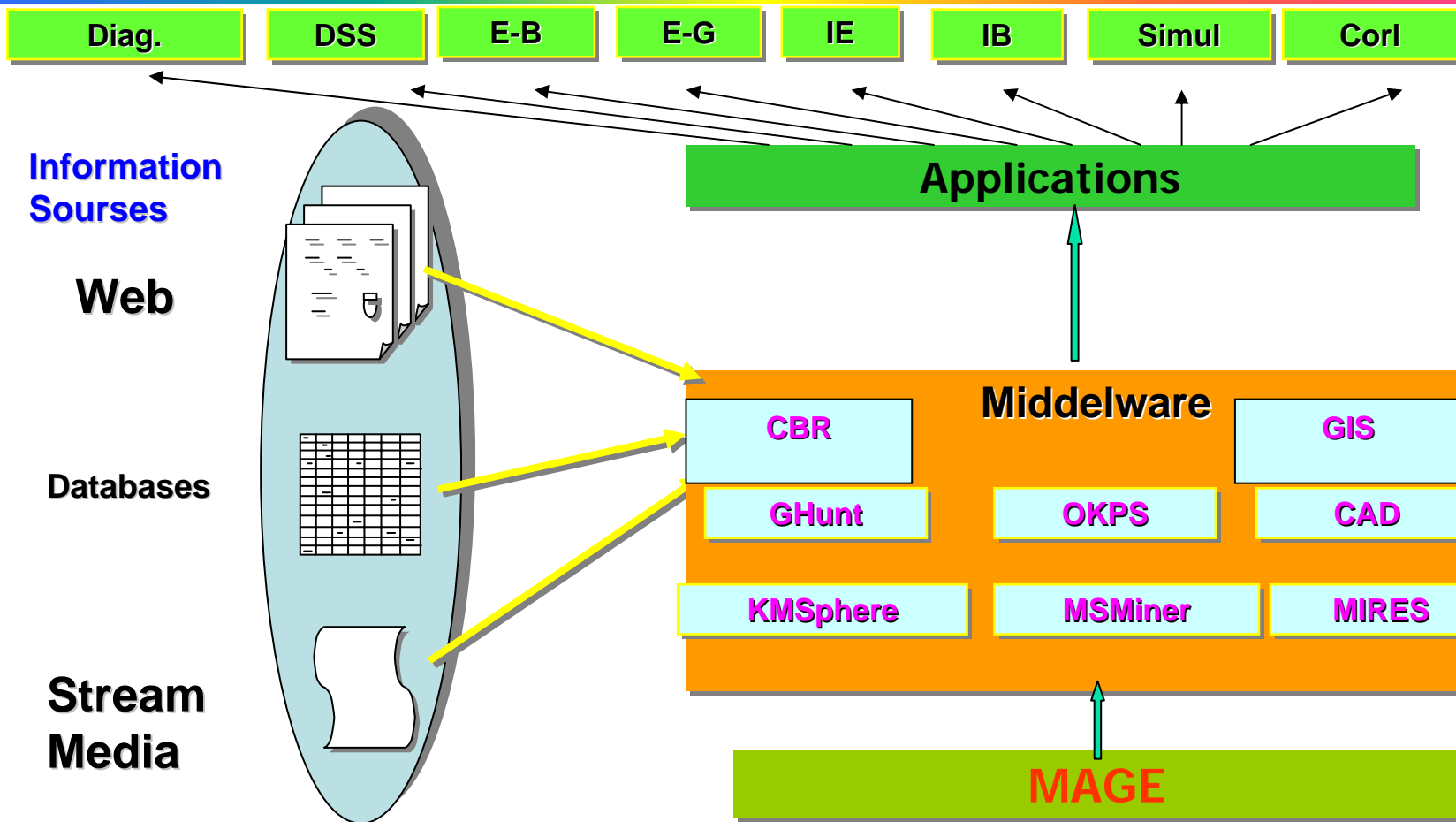
**Common Resources**

Resource-specific implementations of basic services

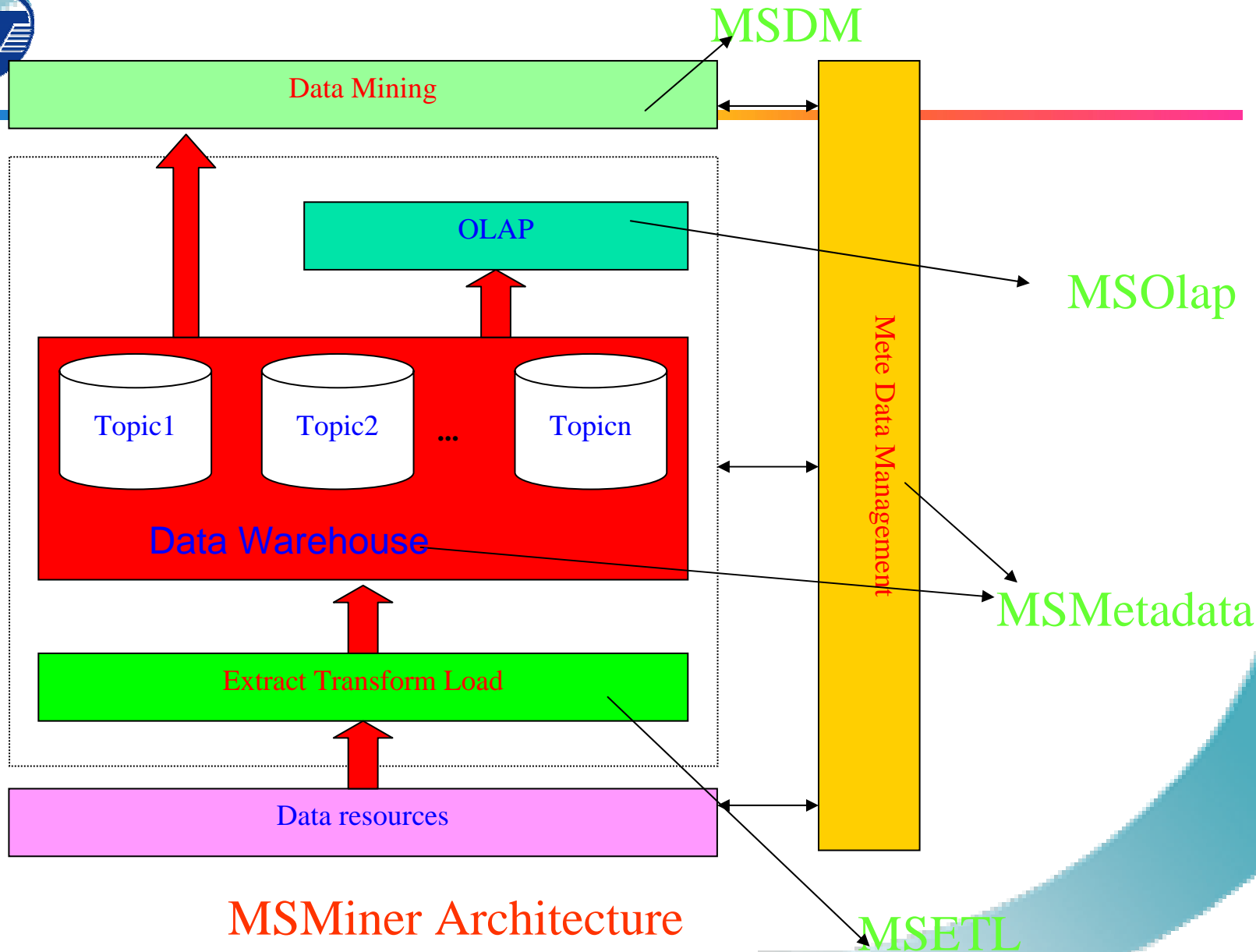
E.g., Transport protocols, name servers, differentiated services, CPU schedulers, public key infrastructure, site accounting, directory service, OS bypass



# Agent Grid Intelligence Platform AGrIP

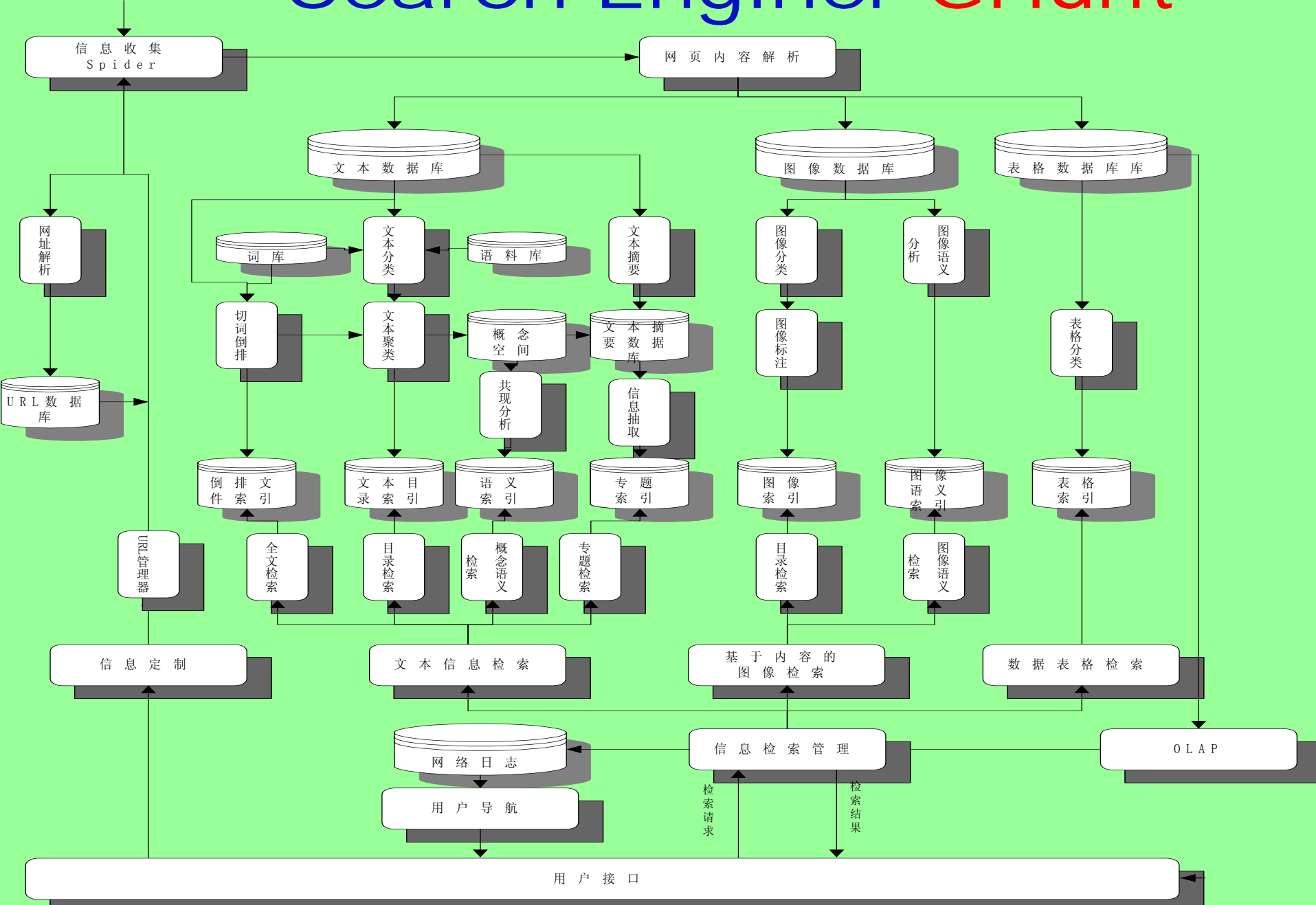


# Data Mining Platform MSMiner



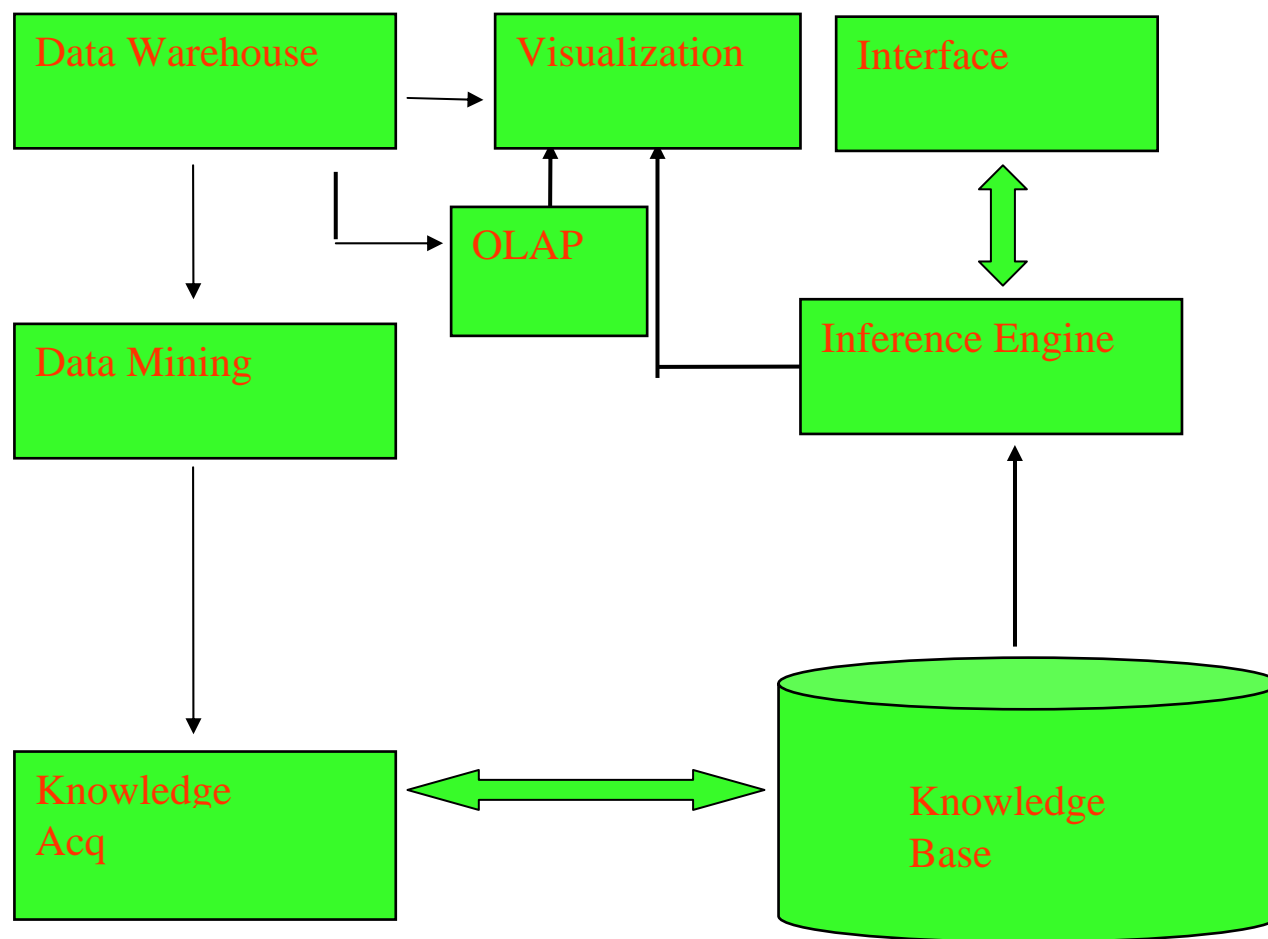
MSMiner Architecture

# Search Engine GHunt





# Expert System OKPS

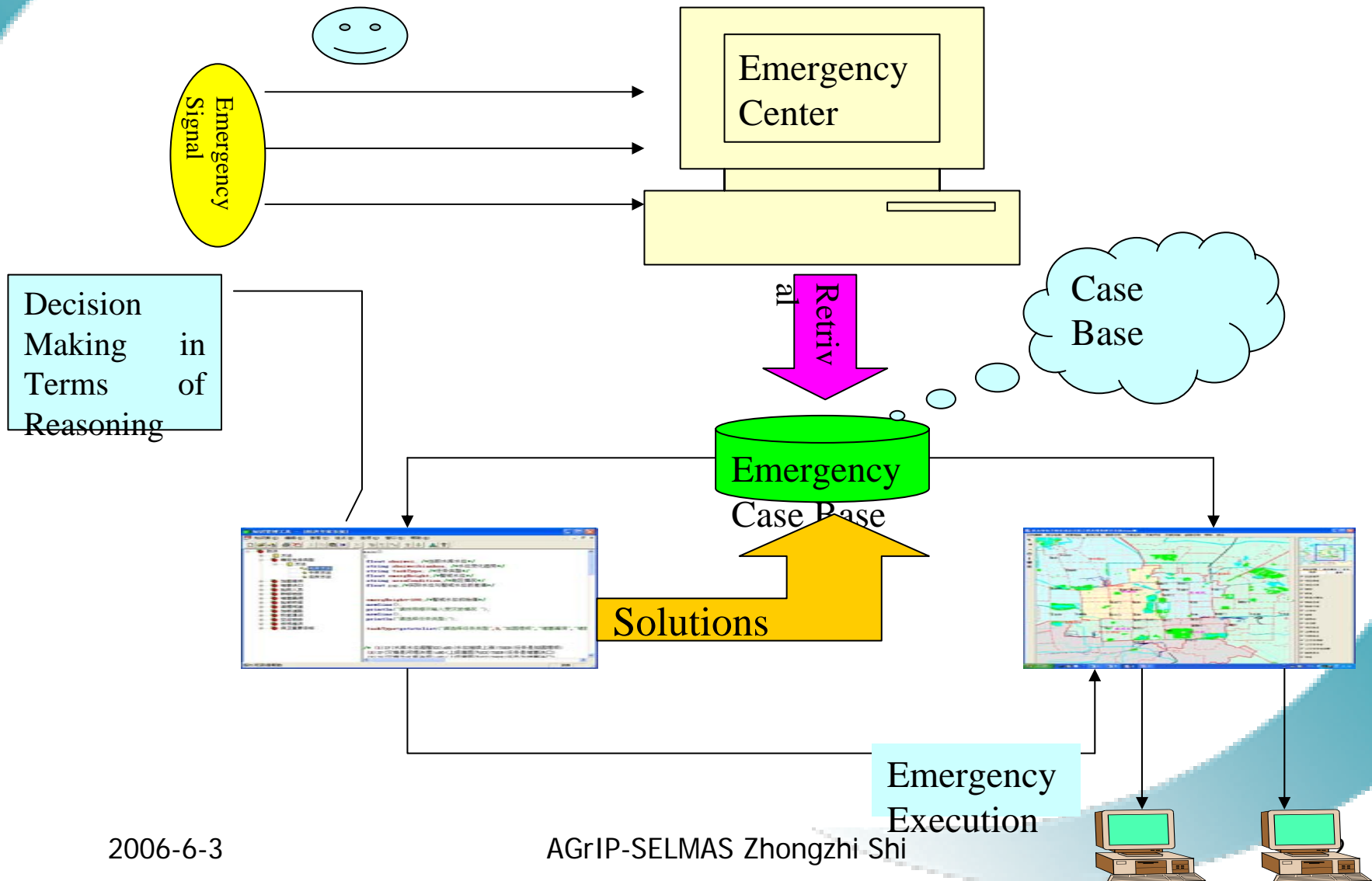




# Outline

- Introduction
- Agent Model
- Multiagent Environment— MAGE
- Agent Collaboration
- Agent Grid Intelligence Platform
- Applications
- Conclusions

# Emergency Interactive System GEIS





# GEIS Interface

index.jsp - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

后退 前进 刷新 地址 搜索 收藏夹 媒体

地址(D) http://192.168.0.212:8080/yjld/logon.do 转到 链接 >>

## 城市应急联动与社会综合服务系统

中国科学院计算技术研究所 北京曙光天演信息技术有限公司  
北京军泰科技开发有限责任公司 北京市科瑞讯科技发展有限公司

接警台 处警台 地图台 社会综合服务系统

请使用1024\*768分辨率 2005版权所有

Intelligent Science Research Group, at Key Lab of ICT, CAS, China

Internet



# Receive Crime Interface

接警台 - Microsoft Internet Explorer

## 接警台

>>基本信息<<

警情序号:  接警员:

警情地点:

报警人名:  性别:  男  女

报警人电话:  接警员:

编号	接警员	值班领导	事故类型	发案地点	接警时间	警情状态
282	施智平	尹超	1	三里屯	2006-01-11	正在处理.....
283	施智平	尹超	6	西直门	2006-01-13	正在处理.....
287	施智平	尹超	6	西直门	2006-02-18	正在处理.....
286	施智平	尹超	1	西直门	2006-02-18	正在处理.....
				西直门	2006-02-23	正在处理.....
				三里屯	2006-03-07	正在处理.....

>>警情细节<<

事故等级:  天气:

指挥长:  处警员:

事故对象:  事故分:

火情:  有  无 死伤人:

危险品:  有  无 丧失行数:

损失折款:  元

警情概要:

来警!!! - Microsoft Internet Explorer

中国科学院计算技术研究所 智能信息处理重点实验室 2005版权所有



# Process Crime Interface

外警页面 - Microsoft Internet Explorer

## 处警台

### >>基本信息<<

警情序号:  接警员:

警情地点:

报警人名:

报警人电话:  接警时间:

### >>外警协同<<

外警操作:

到达现场时间:

反馈时间:

处理完毕时间:

### >>警情细节<<

事故等级:  天气:

指挥长:  处警员:

事故对象:  事故分类:

火情:  有  无 死伤人数: 死:  伤:

危险品:  有  无 损毁车辆数:  辆

警情概要:

损失折款:  元

### >>处警方案<<

交警单位:	<input type="text" value="西城交警支队"/>	<input type="button" value="路线"/>	<input type="button" value="优化"/>	人数:	<input type="text" value="3"/>
医护单位:	<input type="text" value="人民医院"/>	<input type="button" value="路线"/>	<input type="button" value="优化"/>	车数:	<input type="text" value="2"/>
消防单位:	<input type="text" value="北京西城区西直门消防中队"/>	<input type="button" value="路线"/>	<input type="button" value="优化"/>	车数:	<input type="text" value="2"/>
路政单位:	<input type="text" value="东城区路政"/>	<input type="button" value="路线"/>	<input type="button" value="优化"/>	人数:	<input type="text" value="3"/>

处理方法:

- 1、消防人员、器材、车辆赴现场灭火和抢险救援。
- 2、警察维持秩序，保证灭火工作进行顺利，必要时对事故现场可采取临时交通管制措施。
- 3、医务人员和救护车赴现场抢救伤员。
- 4、交警勘察现场、拍照，认定事故原因。
- 5、交警尽快处理现场，路政人员迁移肇事车辆，恢复交通。

编号	接警员	值班领导	事故类型	发案地点	接警时间	警情状态
282	施智平	尹超	1	三里屯	2006-01-11	正在处理.....
283	施智平	尹超	6	西直门	2006-01-13	正在处理.....
287	施智平	尹超	6	西直门	2006-02-18	正在处理.....
286	施智平	尹超	1	西直门	2006-02-18	正在处理.....
290	施智平	尹超	6	西直门	2006-02-23	正在处理.....
299	施智平	尹超	1	三里屯	2006-02-27	正在处理.....



# Geographical Information System

信息产业部应急联动系统-地理信息子系统 - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

地址(D) <http://192.168.0.212:8080/yjld/default.html>

地图操作  
图标编辑  
标图作业  
空间分析  
统计图形

东城区  
朝阳区  
北京站  
天坛公园  
北京游乐园

Internet

欢迎使用本系统...



# CBR-Based Reserve Plan Against Emergency

预案查询页面 - Microsoft Internet Explorer

地址(D) http://192.168.0.212:8080/yjld/jsp/accidentRef.jsp?AccidentNum=1

预案(CBR) 城市应急联动与社会综合服务系统>>>交通安全指挥子系统>>>预案生成

>>预案警情<<      >>现场警情<<

事故地点： 西单	预案序号：16	事故地点： 西直门	警情序号：1
事故等级：2	天气：晴朗	事故等级：1	天气：晴朗
事故对象：自行车	事故分类：超速驾驶	事故对象：机动货车	事故分类：超速驾驶
火情： <input checked="" type="radio"/> 有 <input type="radio"/> 无	死伤人数：死 1 伤 2	火情： <input checked="" type="radio"/> 有 <input type="radio"/> 无	死伤人数：死 1 伤 2
危险品： <input checked="" type="radio"/> 有 <input type="radio"/> 无	损毁车辆：1 辆	危险品： <input checked="" type="radio"/> 有 <input type="radio"/> 无	损毁车辆：0 辆
损失折款：10000 元		损失折款：10000元	
警情概述： 小客车撞到自行车		警情概述： 有行人受伤	

>>参考预案<<      >>决心方案<<

交警单位	西城交警支队	人数：	3	选用 清除
医护单位	北京市崇文区急救站	车数：	2	选用 清除
消防单位		车数：		选用 清除
路政单位		车数：		选用 清除

建议处警方法：  
1、市公安局、市公安交通管理局迅速组织警力赴现场调查处理事故，维护现场秩序，参加救援工作。  
2、市卫生局迅速调集医务人员和车辆赴现场抢救伤员。  
3、交警勘察现场、拍照，认定事故原因。  
4、交警尽快处理现场，恢复交通。

处警方法：

预案(RBR) 综合预案

Internet



# CBR-Based Reserve Plan Against Emergency

预案查询页面 - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

地址(D) http://192.168.0.212:8080/yjld/jsp/accidentRef.jsp?AccidentNum=1

城市应急联动与社会综合服务系统>>>交通安全指挥子系统>>>预案生成

预案(CBR) 预案(RBR)

RBR结果

**>>现场警情<<**

事故地点:	西直门	警情序号:	1
事故等级:	1	天气:	晴朗
事故对象:	机动货车	事故分类:	超速驾驶
火情:	<input type="radio"/> 有 <input checked="" type="radio"/> 无	死伤人数:	死: 1 伤: 2
危险品:	<input type="radio"/> 有 <input checked="" type="radio"/> 无	损毁车辆:	0 辆
损失折款:	10000元		
道路设施受损描述:	有行人受伤		

**>>决心方案<<**

事故等级:	重大事故
交警人数:	10
路政人数:	5
消防车数:	0
救护车数:	3

处警方法:

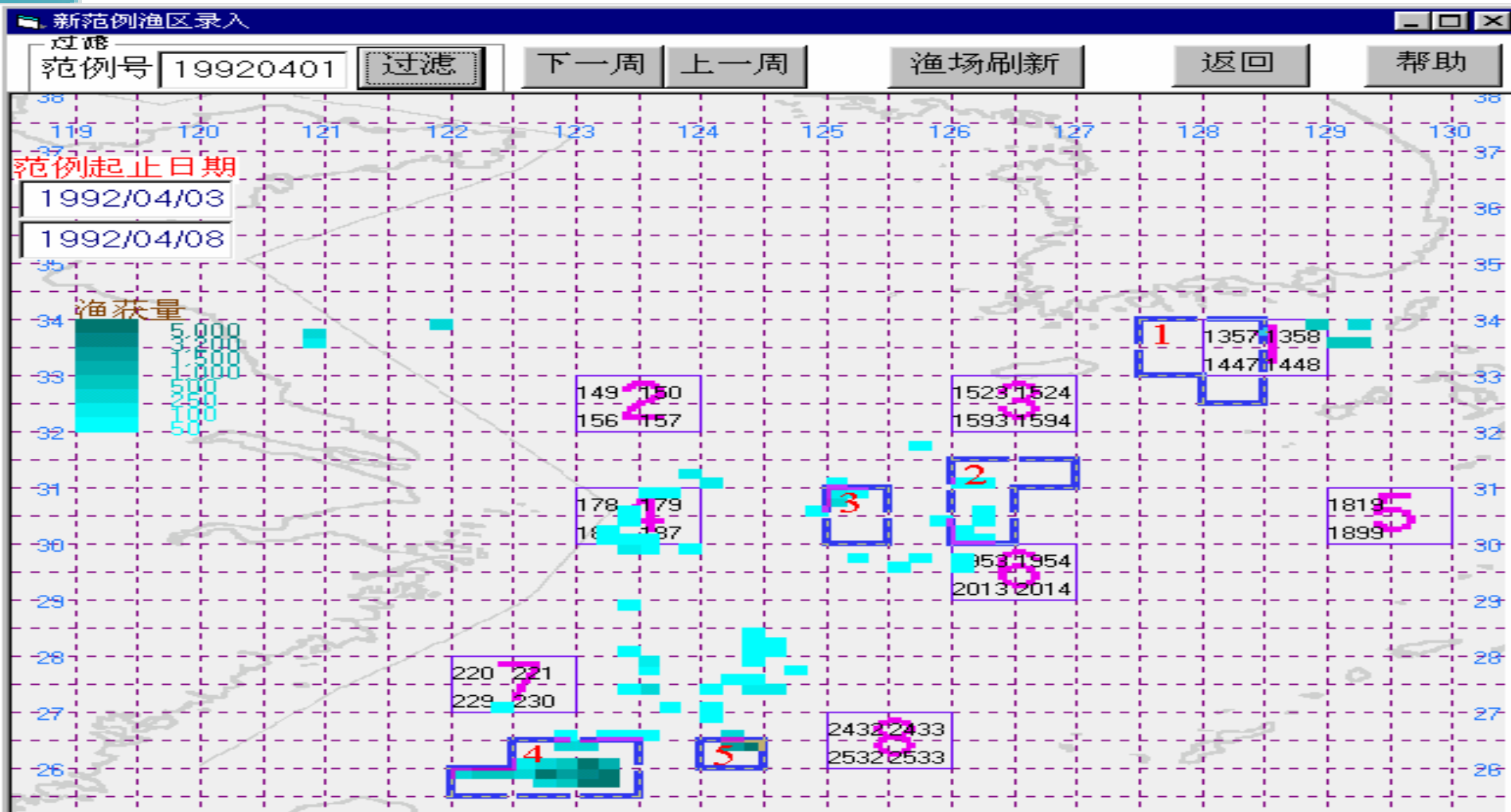
(1) 市公安局、市公安交通管理局迅速组织警力赴现场调查处理事故,维护现场秩序,参加救援工作。(2) 市卫生局迅速调集医务人员和车辆赴现场抢救伤员。(3) 事故发生所在地区县人民政府、事故单位上级主管部门的负责同志立即赴现场处理事故。

综合预案

完毕 Internet

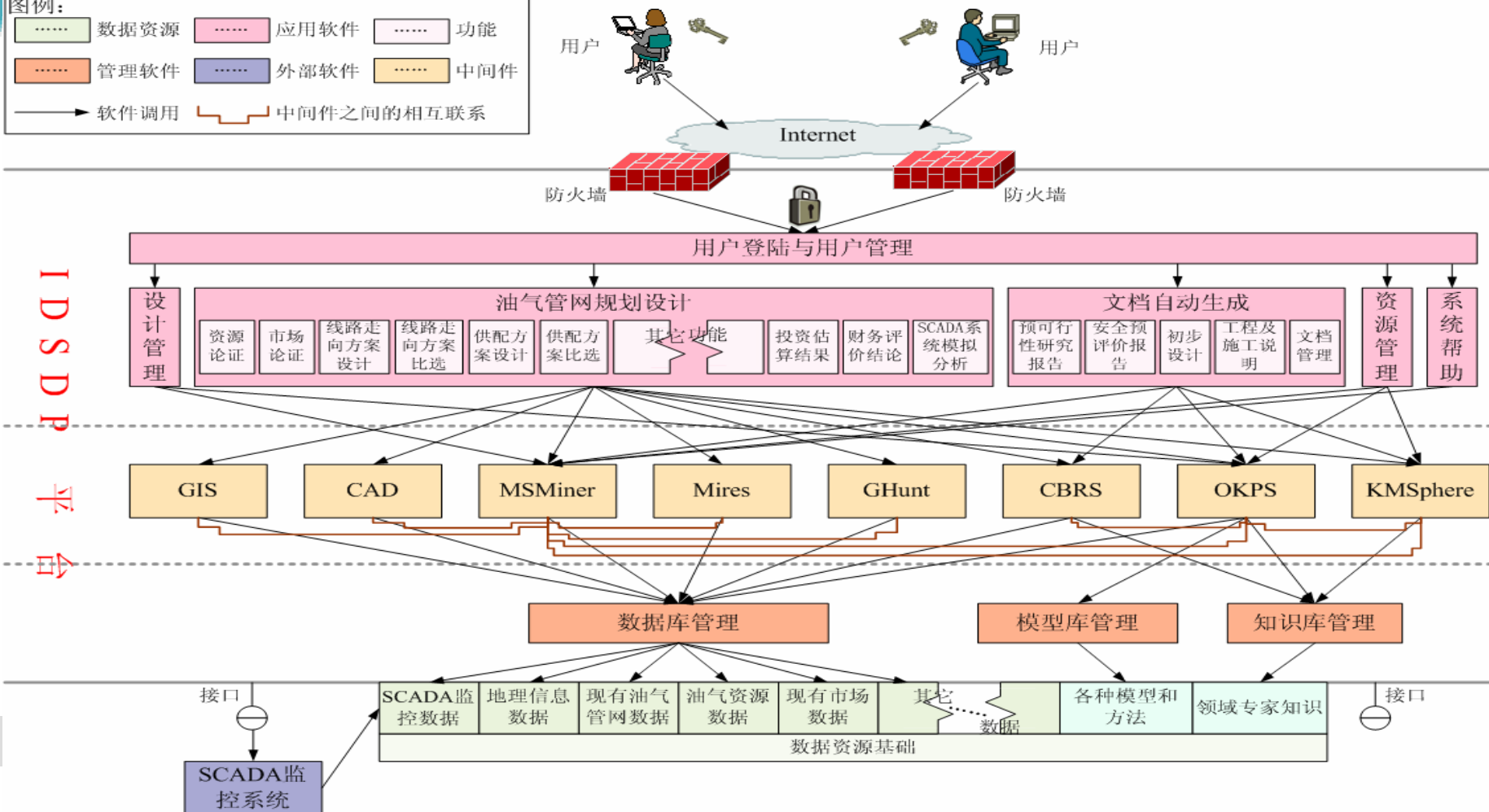
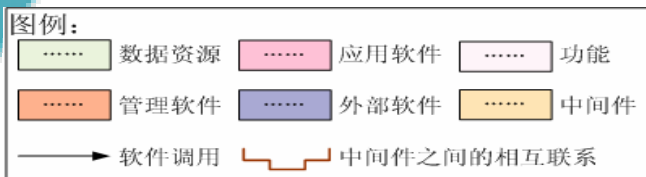


# Fish Field Forecast in Eastern Sea of China





# DSS for Oil Pipeline Design





# Conclusions

- **Proposed Agent Models**
- **MAGE Satisfies the Software Engineering for Large Multiagent Systems**
- **Collaborative Working Agents on Semantic Grid**
- **AGrIP is a Powerful Platform for Constructing Large Complex Systems**



# Published Paper List

- Zhongzhi Shi, He Huang, Jiewen Luo, Fen Lin, Haijun Zhang. [Agent-based Grid Computing](#), JAMM, 2006
- Jiewen Luo, Zhongzhi Shi, Maoguang Wang, Jun Hu: [AGrIP: An Agent Grid Intelligent Platform for Distributed System Integration](#). APWeb Workshops 2006: 590-594
- Zhongzhi Shi. [Autonomic Semantic Grid](#). Keynote speaker. IFIP AIAI2005, Sept., 7-9, 2005
- 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
- Zhongzhi Shi. [Agent-based Semantic Grid](#). China Agent System, IFMAS, 2004, 12, 13-17
- Zhongzhi Shi, Haijun Zhang, Mingkai Dong, [MAGE: Multi-Agent Environment](#), ICCNMC-03, IEEE CS Press, pp.181-188, 2003
- Zhongzhi Shi, Mingkai Dong, Haijun Zhang, Qiujian Sheng. [Agent-based Grid Computing](#). Keynote Speech, International Symposium on Distributed Computing and Applications to Business, Engineering and Science, Wuxi, Dec. 16-20, 2002
- Zhongzhi Shi, Wenpin Jiao, Qiujian Sheng. [Agent-Oriented Software Methodology](#). CEEMAS2001, Cracow, Poland, 2001
- Zhongzhi Shi. [Agent-based E-commerce](#). Invited Speaker, DS-9, Hong Kong, 2001



# Published Paper List

- Zhongzhi Shi, Ju Wang, and Hu Cao, **The Security Mechanism In Multi-Agent System AOSDE**, in: LNAI, Vol. 1733, pp.62-71 PRIMA'99, Dec. 1999.
- Zhongzhi Shi, Qijia Tian, Yunfeng Li. **RAO Logic for Multiagent Framework**. Journal of Computer Science and Technology, July, 1999
- Zhongzhi Shi, Hu Cao, Yunfeng Li, Wenjie Wang, Tao Jiang, **A Building Tool for Multiagent Systems: AOSDE**. IT & Knows, IFIP WCC '98, 1998
- Zhongzhi Shi, Qijia Tian, Yunfeng Li, **RAO Logic for Multiagent Framework**, DAIMAS'97, St. Petersburg, Russia, 1997.
- Zhongzhi Shi, Tao Wang, **An Agent based CSCW System Architecture**, Chinese Journal of Electronics, 6(1), 1996.
- Wenjie Wang Zhongzhi Shi Qijia Tian Tao Wang. **An Agent Belief System for Multi-agent System**. IEEE SMC'96, 2065-2069, 1996.
- Zhongzhi Shi, etc., **MAPE: Multi-Agent Processing Environment**, PRICAI-94, 1994.



# THANK YOU!

Question!



## Intelligence Science

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