

# 多主体系统中的模糊匹配

蒋运承<sup>1,2</sup>, 王 驹<sup>1</sup>

(1. 广西师范大学 计算机科学与信息工程学院, 广西 桂林 541004; 2. 中山大学 计算机科学系, 广东 广州 510275)

**摘 要:**分析了目前主体服务匹配中存在的问题和不足,针对多主体系统中模糊匹配的需求,提出了一种主体模糊服务描述方法.在模糊服务描述的基础上提出了4种模糊匹配算法:基于语法的近似模糊匹配算法、基于语义的近似模糊匹配算法、等价模糊匹配算法和插入模糊匹配算法,并给出了它们的实现方法.这些算法克服了目前主体服务匹配算法中存在的不足.

**关键词:**多主体系统;服务描述;模糊匹配;SDL SIN

**中图分类号:**TP18 **文献标识码:**A **文章编号:**1673-4785(2006)01-0079-05

## Fuzzy service matchmaking in multi-agent systems

JIANG Yun-cheng<sup>1,2</sup>, WANG Ju<sup>1</sup>

(1. School of Computer Sciences and Information Engineering, Guangxi Normal University, Guilin 541004, China; 2. Department of Computer Sciences, Sun Yat-sen University, Guangzhou 510275, China)

**Abstract:** The current work and the existing problem of agent service matchmaking were analyzed in this paper. Aiming at the requirement of fuzzy matchmaking in multi-agent systems, a method of fuzzy service description in multi-agent systems was presented. Based on fuzzy agent service description, four kinds of fuzzy service matchmaking algorithms under fuzzy condition, i. e. syntax based fuzzy matchmaking algorithm, semantics based fuzzy matchmaking algorithm, equivalent fuzzy matchmaking algorithm, and plug-in fuzzy matchmaking algorithm under fuzzy condition, were presented, and the implementation methods of the algorithms were given. The algorithms overcome the shortcoming of the current agent service matchmaking.

**Key words:** multi-agent systems; service description; fuzzy matchmaking; SDL SIN

主体和多主体系统是近年来分布式人工智能领域中的重要研究课题,特别是如何利用主体和多主体系统为用户提供各种有效和高效的服务是目前主体和多主体系统的研究热点.同时,随着 Web 服务、语义 Web 服务和 Grid 服务研究的深入,对主体服务的研究显得尤为重要<sup>[1]</sup>.在多主体系统中,主体服务包含多个环节,主要有:服务描述、服务匹配、服务调用、服务组合、服务验证、服务监控等,文中将研究第2个环节,即服务匹配.

目前国内外对主体服务描述和服务匹配进行了广泛研究. Wickler G. J. 提出了主体能力描述语言 CDL<sup>[2]</sup>, Sycara K. 提出了主体能力描述语言

LARKS<sup>[3]</sup>, Arisha K. 提出了一种主体服务描述语言 SDL<sup>[4]</sup>, 史忠植利用描述逻辑来描述主体服务<sup>[5]</sup>, 蒋运承提出了一种带语义、继承以及支持协商机制的主体服务描述语言 SDL SIN<sup>[1]</sup>. 上述这些主体服务描述语言都有相应的服务匹配算法,其中 CDL 上的服务匹配是基于语法的匹配,其他描述语言上的服务匹配是基于语义的匹配.在上述基于语义的主体服务匹配算法中,它们都存在一个共同的不足:主体服务匹配是在精确服务描述条件下进行的,即服务描述中不能含有模糊或者不确定描述;而在实际应用中不可避免地会出现模糊或者不确定描述.因此,有必要对传统主体服务匹配进一步推广,使之能处理模糊或者不确定的服务描述,即研究在模糊或者不确定服务描述条件下的主体服务匹配.

基于上述原因,文中对文献[1]的主体服务匹配

收稿日期:2006-03-02.

基金项目:国家自然科学基金资助项目(60573010).

算法进行了推广,首先研究了多主体系统中的模糊服务描述问题,给出了判断模糊服务描述近似相等的方法;在模糊服务描述的基础上重点研究了主体服务的模糊匹配及其设计实现问题.

### 1 服务描述

采用文献[1]提出的主体服务描述语言 SDL-SIN 进行服务描述,与文献[1]不同的是,文中允许 SDL-SIN 的属性取模糊值.经过模糊化推广后的 SDL-SIN 形式规范如下:

```

:service-name name
:context f-context-name+
:types (type-name = < modifier > f-type)+
:isa name+
:inputs (variable: < modifier > f-put-type-name)+
:outputs (variable: < modifier > f-put-type-name)+
:input-constraints (f-constraint)+
:output-constraints (f-constraint)+
:io-constraints (f-constraint)+
:concept-description (ontology-name = f-ontology-body)+
:state-language name
:concept-language name
:attributes (attributes-name: f-attributes-value)+
:text-description name)
其中 context-name ::= f-name < * ontology-name >
type-name ::= name
modifier ::= listof | setof
f-type ::= (name: f-name < * ontology-name >)+
f-put-type-name ::= (f-type-name | f-name < * ontology-name >)+
variable ::= name < * ontology-name >
f-ontology-body ::= (expression in fuzzy concept language)
f-attributes-name ::= f-name < * ontology-name >
f-attributes-value ::= f-name
ontology-name ::= name
f-constraint ::= (expression in fuzzy state language)
f-name ::= string
name ::= string

```

上述 service-name、context、types、isa、inputs、

outputs、input-constraints、output-constraints、io-constraints、concept-description、state-language、concept-language、attributes 和 text-description 的含义与文献[1]相同,不同的是这些属性的取值可以用模糊值或模糊变量表示.在推广后的 SDL-SIN 中,模糊概念语言一般采用模糊描述逻辑,而模糊状态语言一般采用模糊逻辑.

### 2 模糊匹配

模糊匹配有 4 种:基于语法的近似模糊匹配、基于语义的近似模糊匹配、等价模糊匹配和插入模糊匹配.其中基于语法和基于语义的近似模糊匹配直接利用 context 属性值之间的匹配可以完成,而等价模糊匹配和插入模糊匹配利用 inputs、outputs、input-constraints、output-constraints 和 io-constraints 属性值来实现匹配判断.

#### 2.1 基于语法的近似模糊匹配

基于语法的近似模糊匹配是利用 context 属性值之间的匹配来实现的,即如果 2 个服务描述(请求服务描述 SPS 和提供服务描述 SRS)的 context 属性值语法描述部分之间近似相等的元素个数与 SPS 和 SRS 所有元素个数之比超过给定的阈值,则认为 SPS 和 SRS 满足近似模糊匹配.首先给出 2 个模糊操作定义.

定义 1 (模糊求交  $\cap_f$ ) 给定 2 个模糊集合  $FS_1 = \{f_{s1}, \dots, f_{sm}\}$  和  $FT_1 = \{f_{t1}, \dots, f_{tn}\}$ , 则  $\{f_{s1}, \dots, f_{sm}\} \cap_f \{f_{t1}, \dots, f_{tn}\} = \{f_n | f_n = f_{si} (\exists f_{tj}, f_{si} \sim f_{tj}), 1 \leq i \leq m, 1 \leq j \leq n\}$ , 其中  $\sim$  表示模糊数据之间的近似相等,可以用贴近度或距离来刻画,即当  $f_s$  和  $f_t$  的贴近度或距离小于等于  $\alpha$ ,  $f_s$  和  $f_t$  近似相等(或  $\alpha$ -相等),记作  $f_s \sim_{\alpha} f_t$ .

贴近度有海明贴近度、欧氏贴近度等;距离有海明距离、欧氏距离等.用户可以根据实际情况采用不同的形式.有关模糊集的贴近度、距离的概念,可以参考文献[6].

定义 2 (模糊合并  $\cup_f$ ) 给定 2 个模糊集合  $FS_1 = \{f_{s1}, \dots, f_{sm}\}$  和  $FT_1 = \{f_{t1}, \dots, f_{tn}\}$ , 则  $\{f_{s1}, \dots, f_{sm}\} \cup_f \{f_{t1}, \dots, f_{tn}\} = (\{f_{s1}, \dots, f_{sm}\} \cup \{f_{t1}, \dots, f_{tn}\}) - (\{f_{s1}, \dots, f_{sm}\} \cap_f \{f_{t1}, \dots, f_{tn}\})$ .

#### 算法 1 基于语法的近似模糊匹配算法 SYBSFM(SPS, SRS)

输入:服务提供主体 SP 的服务描述 SPS 和服务请求主体 SR 的服务描述 SRS.

输出:布尔值(真或者假).

算法:1)如果 SPS 中含有 isa 属性值,则调用实



例化函数 instantiate (SPS) 对 SPS 实例化,假设所得到的服务描述是 ISPS,它的 context 属性值为:  
 $con^{SP} = \{ con_1^{SP}, con_2^{SP}, \dots, con_i^{SP} \} = \{ C_1^{SP} * O_1^{SP}, C_2^{SP} * O_2^{SP}, \dots, C_i^{SP} * O_i^{SP} \}$ .

2) 如果 SRS 中含有 isa 属性值,则调用实例化函数 instantiate (SRS) 对 SRS 实例化,假设所得到的服务描述是 ISRS,它的 context 属性值为:  
 $con^{SR} = \{ con_1^{SR}, con_2^{SR}, \dots, con_i^{SR} \} = \{ C_1^{SR} * O_1^{SR}, C_2^{SR} * O_2^{SR}, \dots, C_j^{SR} * O_j^{SR} \}$ .

3) 如果  $(|\{ C_1^{SP}, C_2^{SP}, \dots, C_i^{SP} \} \cap \{ C_1^{SR}, C_2^{SR}, \dots, C_j^{SR} \}|) / (|\{ C_1^{SP}, C_2^{SP}, \dots, C_i^{SP} \} \cup \{ C_1^{SR}, C_2^{SR}, \dots, C_j^{SR} \}|) > \theta$ , 则返回真,否则返回假. 其中  $\theta \in [0, 1]$  是用户给定阈值.

4) 算法结束.

说明:实例化函数 instantiate () 与文献 [1] 相同,用来转化 isa 属性.

### 2.2 基于语义的近似模糊匹配

与基于语法的近似模糊匹配相同,基于语义的近似模糊匹配也是利用 context 属性值之间的匹配来实现的.

算法 2 基于语义的近似服务匹配算法 SEBF-SM (SPS, SRS)

输入:服务提供主体 SP 的服务描述 SPS 和服务请求主体 SR 的服务描述 SRS.

输出:布尔值(真或者假).

算法:1) 如果 SPS 中含有 isa 属性值,则调用实例化函数 instantiate (SPS) 对 SPS 实例化,假设所得到的服务描述是 ISPS,它的 context 属性值为:  
 $con^{SP} = \{ con_1^{SP}, con_2^{SP}, \dots, con_i^{SP} \} = \{ C_1^{SP} * O_1^{SP}, C_2^{SP} * O_2^{SP}, \dots, C_i^{SP} * O_i^{SP} \}$ .

2) 如果 SRS 中含有 isa 属性值,则调用实例化函数 instantiate (SRS) 对 SRS 实例化,假设所得到的服务描述是 ISRS,它的 context 属性值为:  
 $con^{SR} = \{ con_1^{SR}, con_2^{SR}, \dots, con_i^{SR} \} = \{ C_1^{SR} * O_1^{SR}, C_2^{SR} * O_2^{SR}, \dots, C_j^{SR} * O_j^{SR} \}$ .

3) 令  $\{ O_1^{SP}, \dots, O_i^{SP} \} \cap \{ O_1^{SR}, \dots, O_j^{SR} \} = \{ O_l \} (O = O_k^{SP} (\exists O_i^{SR}, O_k^{SP} \subseteq_f O_i^{SR} \quad O_k^{SP} \supseteq_f O_i^{SR})) (O = O_l^{SR} (\exists O_k^{SP}, O_l^{SR} \subseteq_f O_k^{SP} \quad O_l^{SR} \supseteq_f O_k^{SP}))$ ,  $1 \leq k \leq i, 1 \leq l \leq j$ , 如果  $|O_l| / (|\{ O_1^{SP}, O_2^{SP}, \dots, O_i^{SP} \} \cap \{ O_1^{SR}, O_2^{SR}, \dots, O_j^{SR} \}|) > \theta$ , 则返回真,否则返回假. 式中  $\theta \in [0, 1]$  是用户给定的阈值,  $\approx$  表示近似相等,即  $O_l^{SR} \approx O_k^{SP}$  表示  $O_l^{SR}$  和  $O_k^{SP}$  近似等价(即  $O_l^{SR} \subseteq_f O_k^{SP}$ );  $\subseteq_f$  和  $\supseteq_f$  按模糊描述逻辑<sup>[7]</sup>中的语义进行解释, $O_l^{SR} \subseteq_f O_k^{SP}$  表示  $O_l^{SR}$

模糊包含  $O_l^{SR}$  (即  $O_l^{SR} \subseteq_f O_k^{SP}$ );  $O_l^{SR} \supseteq_f O_k^{SP}$  表示  $O_l^{SR}$  模糊包含  $O_k^{SP}$  (即  $O_l^{SR} \supseteq_f O_k^{SP}$ ).

4) 算法结束.

说明:算法 1 和算法 2 中的近似相等  $\approx$  也可以用模糊相等  $\approx_f$  来代替,有关  $\approx_f, \subseteq_f$  和  $\supseteq_f$  的定义见文献[8].

### 2.3 等价模糊匹配

等价模糊匹配要求两个服务描述在语义上是模糊等价的,利用 inputs、outputs、input-constraints、output-constraints 和 io-constrains 属性值来实现匹配判断.

算法 3 等价模糊匹配算法 EFM (SPS, SRS)

输入:服务提供主体 SP 的服务描述 SPS 和服务请求主体 SR 的服务描述 SRS.

输出:布尔值(真或者假).

算法:1) 如果 SPS 中含有 isa 属性值,则调用实例化函数 instantiate (SPS) 对 SPS 实例化,假设所得到的服务描述是 ISPS,它的 inputs 属性值为:  
 $I^{SP} = \{ I_1^{SP}, I_2^{SP}, \dots, I_{i_{SP}}^{SP} \} = \{ iv_1^{SP} \quad it_1^{SP}, iv_2^{SP} \quad it_2^{SP}, \dots, iv_{i_{SP}}^{SP} \quad it_{i_{SP}}^{SP} \}$ , outputs 属性值为:  
 $O^{SP} = \{ O_1^{SP}, O_2^{SP}, \dots, O_{j_{SP}}^{SP} \} = \{ ov_1^{SP} \quad ot_1^{SP}, ov_2^{SP} \quad ot_2^{SP}, \dots, ov_{j_{SP}}^{SP} \quad ot_{j_{SP}}^{SP} \}$ , input-constraints 属性值为:  
 $S_I^{SP} = \{ S_{I1}^{SP}, S_{I2}^{SP}, \dots, S_{I_{k_{SP}}}^{SP} \}$ , output-constraints 属性值为:  
 $S_O^{SP} = \{ S_{O1}^{SP}, S_{O2}^{SP}, \dots, S_{O_{m_{SP}}}^{SP} \}$ , io-constrains 属性值为:  
 $S_{IO}^{SP} = \{ LS_{IO1}^{SP} \quad RS_{IO1}^{SP}, \dots, LS_{IO_{m_{SP}}}^{SP} \quad RS_{IO_{m_{SP}}}^{SP} \}$ .

2) 如果 SRS 中含有 isa 属性值,则调用实例化函数 instantiate (SRS) 对 SRS 实例化,假设所得到的服务描述是 ISRS,它的 inputs 属性值为:  
 $I^{SR} = \{ I_1^{SR}, I_2^{SR}, \dots, I_{i_{SR}}^{SR} \} = \{ iv_1^{SR} \quad it_1^{SR}, iv_2^{SR} \quad it_2^{SR}, \dots, iv_{i_{SR}}^{SR} \quad it_{i_{SR}}^{SR} \}$ , outputs 属性值为:  
 $O^{SR} = \{ O_1^{SR}, O_2^{SR}, \dots, O_{j_{SR}}^{SR} \} = \{ ov_1^{SR} \quad ot_1^{SR}, ov_2^{SR} \quad ot_2^{SR}, \dots, ov_{j_{SR}}^{SR} \quad ot_{j_{SR}}^{SR} \}$ , input-constraints 属性值为:  
 $S_I^{SR} = \{ S_{I1}^{SR}, S_{I2}^{SR}, \dots, S_{I_{k_{SR}}}^{SR} \}$ , output-constraints 属性值为:  
 $S_O^{SR} = \{ S_{O1}^{SR}, S_{O2}^{SR}, \dots, S_{O_{m_{SR}}}^{SR} \}$ , io-constrains 属性值为:  
 $S_{IO}^{SR} = \{ LS_{IO1}^{SR} \quad RS_{IO1}^{SR}, \dots, LS_{IO_{m_{SR}}}^{SR} \quad RS_{IO_{m_{SR}}}^{SR} \}$ .

3) 如果  $i_{SP} = i_{SR}$ , 并且  $it_g^{SP} \quad it_g^{SR}, 1 \leq g \leq i_{SP}, 1 \leq g \leq i_{SR}$ , 则 goto 4), 否则返回假.

4) 如果  $j_{SP} = j_{SR}$ , 并且  $ot_g^{SP} \quad ot_g^{SR}, 1 \leq g \leq j_{SP}, 1 \leq g \leq j_{SR}$ , 则 goto 5), 否则返回假.

5) 如果存在一个置换  $\sigma$ , 使得  $S_I^{SP} \Leftrightarrow (S_I^{SR})$ , 则 goto 6), 否则返回假.

6) 如果存在一个置换  $\sigma$ , 使得  $S_O^{SP} \Leftrightarrow (S_O^{SR})$ , 则



goto 7), 否则返回假.

7) 如果存在一个置换, 使得  $S_{i0}^{SP} \Leftrightarrow (S_{i0}^{SR})$ , 则返回真, 否则返回假.

8) 算法结束.

说明:  $\Leftrightarrow$  表示模糊等价, 即  $f_{l1} \Leftrightarrow f_{l2}$  表示模糊真值在 范围内, 模糊逻辑公式  $f_{l1}$  和  $f_{l2}$  等价.

### 2.4 插入模糊匹配

插入服务匹配要求服务提供主体所提供的服务  $M$  在语义上“包含”服务请求主体所请求的服务  $N$ , 即相对于  $N$  来说,  $M$  能提供更多的服务,  $N$  能够“插入”到  $M$  中.

算法 4 插入模糊匹配算法 PIFM(SPS, SRS)

输入: 服务提供主体 SP 的服务描述 SPS 和服务请求主体 SR 的服务描述 SRS.

输出: 布尔值(真或者假).

算法: 1) 如果 SPS 中含有 isa 属性值, 则调用实例化函数 instantiate(SPS) 对 SPS 实例化, 假设所得到的服务描述是 ISPS, 它的 inputs 属性值为:  $I^{SP} = \{I_1^{SP}, I_2^{SP}, \dots, I_{i_{SP}}^{SP}\} = \{iv_1^{SP} \quad i_{i_1}^{SP}, iv_2^{SP} \quad i_{i_2}^{SP}, \dots, iv_{i_{SP}}^{SP} \quad i_{i_{i_{SP}}}^{SP}\}$ , outputs 属性值为:  $O^{SP} = \{O_1^{SP}, O_2^{SP}, \dots, O_{j_{SP}}^{SP}\} = \{ov_1^{SP} \quad o_{i_1}^{SP}, ov_2^{SP} \quad o_{i_2}^{SP}, \dots, ov_{j_{SP}}^{SP} \quad o_{i_{j_{SP}}}^{SP}\}$ , input-constraints 属性值为:  $S_I^{SP} = \{S_{i_1}^{SP}, S_{i_2}^{SP}, \dots, S_{i_{i_{SP}}}^{SP}\}$ , output-constraints 属性值为:  $S_O^{SP} = \{S_{o_1}^{SP}, S_{o_2}^{SP}, \dots, S_{o_{j_{SP}}}^{SP}\}$ , io-constraints 属性值为:  $S_{i0}^{SP} = \{S_{i0_1}^{SP}, S_{i0_2}^{SP}, \dots, S_{i0_{m_{SP}}}^{SP}\} = \{LS_{i0_1}^{SP} \quad RS_{i0_1}^{SP}, \dots, LS_{i0_{m_{SP}}}^{SP} \quad RS_{i0_{m_{SP}}}^{SP}\}$ .

2) 如果 SRS 中含有 isa 属性值, 则调用实例化函数 instantiate(SRS) 对 SRS 实例化, 假设所得到的服务描述是 ISRS, 它的 inputs 属性值为:  $I^{SR} = \{I_1^{SR}, I_2^{SR}, \dots, I_{i_{SR}}^{SR}\} = \{iv_1^{SR} \quad i_{i_1}^{SR}, iv_2^{SR} \quad i_{i_2}^{SR}, \dots, iv_{i_{SR}}^{SR} \quad i_{i_{i_{SR}}}^{SR}\}$ , outputs 属性值为:  $O^{SR} = \{O_1^{SR}, O_2^{SR}, \dots, O_{j_{SR}}^{SR}\} = \{ov_1^{SR} \quad o_{i_1}^{SR}, ov_2^{SR} \quad o_{i_2}^{SR}, \dots, ov_{j_{SR}}^{SR} \quad o_{i_{j_{SR}}}^{SR}\}$ , input-constraints 属性值为:  $S_I^{SR} = \{S_{i_1}^{SR}, S_{i_2}^{SR}, \dots, S_{i_{i_{SR}}}^{SR}\}$ , output-constraints 属性值为:  $S_O^{SR} = \{S_{o_1}^{SR}, S_{o_2}^{SR}, \dots, S_{o_{j_{SR}}}^{SR}\}$ , io-constraints 属性值为:  $S_{i0}^{SR} = \{S_{i0_1}^{SR}, S_{i0_2}^{SR}, \dots, S_{i0_{m_{SR}}}^{SR}\} = \{LS_{i0_1}^{SR} \quad RS_{i0_1}^{SR}, \dots, LS_{i0_{m_{SR}}}^{SR} \quad RS_{i0_{m_{SR}}}^{SR}\}$ .

3) 如果  $i_{SP} = i_{SR}$ , 并且  $it_g^{SP} \supseteq_f it_g^{SR}, 1 \leq g \leq i_{SP}$ , 则 goto 4), 否则返回假.

4) 如果  $j_{SP} = j_{SR}$ , 并且  $ot_g^{SP} \supseteq_f ot_g^{SR}, 1 \leq g \leq j_{SP}$ , 则 goto 5), 否则返回假.

5) 如果存在一个置换, 使得  $S_I^{SP} \Leftarrow (S_I^{SR})$ , 则 goto 6), 否则返回假.

6) 如果存在一个置换, 使得  $S_O^{SP} \Rightarrow (S_O^{SR})$ , 则 goto 7), 否则返回假.

7) 如果存在一个置换, 满足  $\forall S_{i0_i}^{SP} = LS_{i0_i}^{SP} \quad RS_{i0_i}^{SR}, 1 \leq i \leq m_{SP}, \exists S_{i0_j}^{SR} = LS_{i0_j}^{SR} \quad RS_{i0_j}^{SR}, 1 \leq j \leq m_{SR}$ ,

使得  $(LS_{i0_i}^{SP} \Leftarrow (LS_{i0_j}^{SR})) \quad (RS_{i0_i}^{SP} \Rightarrow (RS_{i0_j}^{SR}))$  成立, 则返回真, 否则返回假.

8) 算法结束.

说明:  $\Leftarrow$  表示模糊蕴含, 即  $f_{l1} \Leftarrow f_{l2}$  表示模糊真值在 范围内, 模糊逻辑公式  $f_{l2}$  模糊蕴含模糊逻辑公式  $f_{l1}$ .

### 2.5 举例

例 1 给定服务描述  $S_1$  和  $S_2$ , 其中  $S_1 = \{\text{service-name } s_1; \text{context 常用家电 * 家电, 品牌电脑 * 电脑}\}$ ,  $S_2 = \{\text{service-name } s_2; \text{context 主要家电 * 家用电器, 品牌机 * 计算机}\}$ , 家电 电视机—电冰箱—洗衣机—音箱, 电脑 联想电脑—方正电脑, 家用电器 电视机—电冰箱—洗衣机—音箱, 计算机 联想计算机—方正计算机, 假设阈值  $\alpha = 0.7, \beta = 0.2$ , 常用家电与主要家电的语义距离是 0.3, 品牌电脑与品牌机的语义距离是 0.1, 则  $S_1$  和  $S_2$  满足基于语义的近似模糊匹配要求, 但不满足基于语法的近似模糊匹配要求.

例 2 给定服务描述  $S_1$  和  $S_2$ , 其中  $S_1 = \{\text{service-name } s_1; \text{inputs } xs: \text{Listof Integer}; \text{outputs } ys: \text{Listof Integer}; \text{input-constraints } (le(\text{length}(xs), 100), 0.7); \text{output-constraints } (before(x, y, ys) \geq(x, y), 0.65), (in(x, ys) \quad in(x, xs), 0.7)\}$ ,  $S_2 = \{\text{service-name } s_2; \text{inputs } xs: \text{Listof Real}; \text{outputs } ys: \text{Listof Real}; \text{output-constraints } (before(x, y, ys) \geq(x, y), 0.8), (before(x, y, ys) \quad preceds(x, y), 0.9), (in(x, ys) \quad in(x, xs), 0.85)\}$ , 假设阈值  $\alpha = 0.2$ , 则  $S_1$  和  $S_2$  满足模糊插入匹配要求, 但不满足等价模糊匹配要求; 假设阈值  $\beta = 0.1$ , 则  $S_1$  和  $S_2$  不满足模糊插入匹配要求.

## 3 实现方法

在多主体环境 MAGE<sup>[9]</sup> 中已经实现了文中提出的模糊匹配算法, 它们都是传统主体服务匹配算法(即精确服务描述条件下的匹配算法)的推广. 在 MAGE 中有一个 DF 主体(目录服务主体)和一个 SBroker 主体(服务代理主体), 其中 DF 主体存放主体服务描述及完成服务匹配工作, SBroker 主体完成服务的调用、协商、组合以及监控等工作, 它们一起构成 MAGE 的中间主体<sup>[1,5]</sup>. 服务请求主体把服务请求交给 SBroker 主体, 由 SBroker 主体代理它完成服务推理的所有工作, 最后 SBroker 主体将结果返回给服务请求主体, 该方法吸收了 Grid 服务的思想. 由于在服务匹配时要用到简单的模糊描述逻辑 FALC 和模糊逻辑, 为此实现了一个模糊描述逻



辑推理机 FDLRM 主体和一个模糊逻辑推理机 FL-Reasoner 主体.

多主体系统是一个开放、动态的系统,即主体可以随时加入或者离开多主体系统,在这样一个多主体系统中,如何找到所需要的主体、中间主体就是一种很好的解决方案. MAGE 中的 DF 主体和 SBroker 主体就是为此目的而建立的. 在 DF 主体中,主体服务描述语言采用 SDL SIN,服务匹配算法有多种,包括同构条件下的匹配算法和异构条件下的匹配算法,以及文中提出的主体服务模糊匹配算法,主体可以根据实际情况采用不同的实现算法. 目前 SBroker 主体具有服务调用和服务组合的功能. MAGE 服务环境的体系结构简图见图 1.

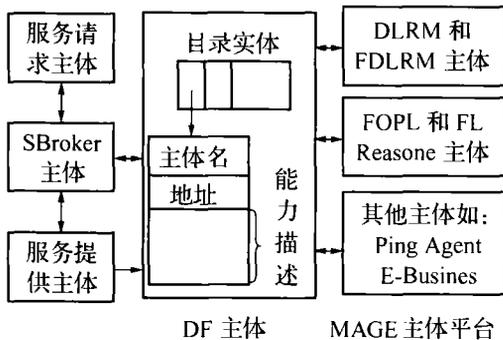


图 1 MAGE 服务环境的体系结构

Fig. 1 Architecture of MAGE service environment

### 4 结束语

目前主体服务匹配是在精确服务描述条件下进行的,即服务描述中不能含有模糊或者不确定描述. 针对该不足,研究了模糊服务描述条件下的主体服务匹配问题,即首先对主体服务描述进行模糊化推广,提出了一种模糊服务描述方法. 在模糊服务描述的基础上根据模糊集理论重点研究了主体服务的模糊匹配及其设计实现问题,提出了 4 种模糊匹配算法. 这些算法克服了目前传统主体服务匹配算法中存在的不足. 进一步工作主要包括:研究异构条件下的模糊匹配,以及适合普适计算的模糊匹配等问题.

### 参考文献:

[1] 蒋运承, 张海俊, 董明楷, 等. 多主体系统中的动态服务匹配[J]. 电子学报, 2004, 32(3): 457 - 461.  
JIANG Yuncheng, ZHANG Haijun, DONG Mingkai, et al. Dynamic service matchmaking in multi-agent systems [J]. Acta Electronica Sinica, 2004, 32(3): 457 - 461.

[2] WICKLER GJ. Using expressive and flexible action representations to reason about capabilities for intelligent a-

gent cooperation[D]. Edinburgh, U K: University of Edinburgh, 1999.

[3] SYCARA K, WIDOFF S, KLUSCH M, et al. LARKS: Dynamic matchmaking among heterogenous software agents in cyberSpace[J]. Autonomous Agents and Multi-Agent Systems, 2002, 5(2): 173 - 203.

[4] ARISHA K, OZCAN F, ROSS R, et al. Impact: a platform for collaborating agents[J]. IEEE Intelligent Systems, 1999, 14(2): 64 - 72.

[5] 史忠植, 蒋运承, 张海俊, 等. 基于描述逻辑的主体服务匹配[J]. 计算机学报, 2004, 27(5): 625 - 635.  
SHI Zhongzhi, JIANG Yuncheng, ZHANG Haijun, et al. Agent service matchmaking based on description logic[J]. Chinese Journal of Computers, 2004, 27(5): 625 - 635.

[6] 何新贵. 模糊知识处理的理论与技术: 第 2 版[M]. 北京: 国防工业出版社, 1998.  
HE Xingui. Fuzzy theories and fuzzy techniques in knowledge processing: 2nd ed. [M]. Beijing: National Defence Industry Press, 1999.

[7] STRACCIA U. Reasoning within fuzzy description logics[J]. Journal of Artificial Intelligence Research, 2001, 14(1): 137 - 166.

[8] 刘普寅, 吴孟达. 模糊理论及其应用[M]. 长沙: 国防科技大学出版社, 1998.  
LIU Puyin, WU Mengda. Fuzzy theory and its application [M]. Changsha: National University of Defense Technology Press, 1998.

[9] SHI Zhongzhi, DONG Mingkai, ZHANG Haijun, et al. Agent-based grid computing[A]. International Symposium on Distributed Computing and Applications to Business, Engineering and Science[C]. Wuhan: Wuhan University of Technology Press, 2002.

### 作者简介:



蒋运承,男,1974年生,博士,副教授,2004年毕业于中国科学院计算技术研究所,现为中山大学计算机科学系博士后,主要研究方向为描述逻辑、语义 Web 和多 Agent 系统,在中国科学、软件学报、计算机学报等刊物上发表论文 20 余篇.

E mail: ycjiang@mailbox.gxnu.edu.cn

王 驹,男,1950年生,博士,研究员,毕业于美国加州大学,主要研究方向为描述逻辑、数理逻辑和人工智能,主持或参加 10 多项国家 863、973、国家自然科学基金课题、子课题,在中国科学、科学通报、软件学报、计算机学报等刊物上发表论文 30 余篇.

