

文章编号: 1001-0920(2013)09-1414-05

一种基于语义的决策服务协作自组织方法

张波¹, 向阳², 黄震华²

(1. 上海师范大学 信息与机电工程学院, 上海 200234; 2. 同济大学 电信学院, 上海 201804)

摘要: 以形式化语义为基础提出一种决策服务协作自组织方法. 首先, 定义相关概念并引入 Horn 子句为语义推理基; 然后, 提出基于决策环境和能力准则的决策服务评估算法; 最后, 提出了自组织算法, 通过自组织矩阵找到最符合需求的决策服务协作参与者. 实验分析表明所提出的方法是有效可行的.

关键词: 决策服务; 协作; 自组织; 语义; 决策环境; 能力评估

中图分类号: TP391

文献标志码: A

Semantic based self-organization method for decision-making service collaboration

ZHANG Bo¹, XIANG Yang², HUANG Zhen-hua²

(1. College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai 200234, China; 2. College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China. Correspondent: ZHANG Bo, E-mail: sh.zhangbo@gmail.com)

Abstract: A self-organization method for decision-making service collaboration is proposed based on formal semantic. Firstly, definitions of decision-making service organization are presented, and the theory of Horn clause is introduced as the rule of semantic reasoning. Meanwhile, evaluation algorithms of decision-making service are proposed based on two criterions of decision-making environment and capacity. Finally, the self-organization algorithm is proposed based on the organizing matrix in order to find out the most competent collaboration participants who can satisfy decision-making requirement. Experimental analysis shows the effectiveness and feasibility of the proposed method.

Key words: decision-making service; collaboration; self-organization; semantic; decision-making environment; capacity evaluation

0 引言

随着人们对决策的需求日益复杂化, 协作决策已成为计算机辅助决策发展的重要方向^[1-3]. 近年来, 网络决策和决策 Web 服务化技术的出现, 促使协作决策可以松散耦合的方式, 通过良好的组织机制挑选最佳协作参与者实现服务组合, 解决复杂需求, 使决策过程更具灵活性.

协作是一种社会行为^[2], Web 服务时代实施协作决策的重要问题之一在于: 必须有效挑选出最合适的协作参与者. 目前, 针对服务协作自组织的众多研究主要集中在: 服务中间件、协作控制、协作安全性、Qos、实时数据管理等方面^[4-7]. 但挑选参与者的

核心标准是参与者的能力和符合协作参与的计算环境^[8]. 因此, 本文以能力和环境为标准, 对候选决策服务 DS (Decision-making Service) 的能力及决策环境进行评估, 提出一种能力和环境可感知的决策服务自组织机制. 该机制通过形式化语义方法描述决策环境特性、决策服务和决策需求等, 进而引入 Horn 子句对服务组织的推理规则进行描述, 最终实现决策服务协作自组织算法.

1 基本概念

1.1 Horn 子句

本文引入 Horn 子句^[9]来实现服务自组织过程的逻辑条件推理及规则描述等.

收稿日期: 2012-05-15; 修回日期: 2012-11-07.

基金项目: 国家自然科学基金项目(61103069, 71171148); 上海市教委科研创新基金项目(13YZ052); 教育部新世纪优秀人才支持计划项目(NCET-12-0413). 上海师范大学校级创新基金项目(DXL125, DCL201302).

作者简介: 张波(1978-), 男, 讲师, 从事语义网、服务计算的研究; 向阳(1962-), 男, 教授, 博士生导师, 从事智能决策等研究.

定义1 Horn子句表示为: $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q$, P_i 和 Q 是命题文字, \wedge 和 \rightarrow 是逻辑连接符, 其语义为命题 Q 可以被实现当且仅当所有前提 P_i 都被满足.

Horn子句的推理可表达为以下等价形式:

1) 同时拥有两类肯定命题文字的Horn子句被称为一个规则, 其形式为 $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q$;

2) 不存在肯定命题的Horn子句被称为一个目标子句, 其形式为 $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow$;

3) 不存在命题文字的Horn子句被称为一个事实, 其形式为 $\rightarrow Q$.

1.2 基本概念

1) 决策环境语义.

决策环境是一种对DS周围客观事物的描述, 可以定义如下:

定义2 决策环境是一个四元组 $\mathfrak{S} = (L, P, S, C)$. 其中 $L = (loc_1, loc_2, \dots)$ 是一组描述DS位置信息的向量集合; $P = (p_1, p_2, \dots)$ 是一组描述DS计划时间的信息; $S = (s_1, s_2, \dots)$ 是一组描述某个DS状态集合; $C = \{c_i(x_j) \mid x_j \in L \vee T \vee S\}$ 是标识该DS所具备的约束条件的集合.

2) DS相关语义.

DS语义包括物理资源和计算资源两个方面.

定义3 DS物理语义可以定义为 $D = (id, I|O, \mathfrak{S}_{real}, \mathfrak{S}_{mobile})$. 其中 id 用来标识DS物理端编号; $I|O$ 表示DS的输入端口和输出端口参数; \mathfrak{S}_{real} 表示物理端的实时环境; \mathfrak{S}_{mobile} 表示物理端潜在的临时动态环境.

定义4 DS计算能力语义是 $\mathfrak{N} = (class, In|Out, price)$. 其中 $class$ 表示该DS计算的类型; $In|Out$ 表示DS计算必需的输入和输出参数; $price$ 表示使用DS所需要付出的价格.

3) 决策任务语义.

决策任务描述用户应用需求, 是DS组织必须满足的前提条件.

定义5 决策任务语义是一组向量 $T = T_1, T_2, \dots$, 其中 $T_i = (TClass_i, Status_i, Goal_i, Envir_i)$ 是组成该任务的若干子任务, $TClass_i$ 是该子任务的类型; $Status_i$ 为初始条件; $Goal_i$ 指明预期任务; $Cost_i$ 和 $Envir_i$ 指明求解的成本和环境要求.

2 环境的评估机制

2.1 环境约束

本文给定环境语义必须遵循的约束条件.

约束1 任何一个环境的语义表达都应服从

Horn子句的事实表达方式, 即形如 $\rightarrow \mathfrak{S}$.

约束2 决策任务执行决策任务时对环境的要求通过Horn子句表达为形如 $\mathfrak{S} \rightarrow T_1.Envir_i$ 的规则.

2.2 决策环境评估算法

首先, 本文给出进行评估的匹配函数定义.

定义6 假设 X 是一组事实, y 是一个需求, 匹配函数 $X \mapsto y$ 是指这组事实 X 能够满足需求 y . 匹配函数可以通过形如 $X \mapsto y \equiv x_1 \wedge x_2 \wedge \dots \wedge x_n \rightarrow y$ 的Horn子句表达, 其中 $x_1 \in X, \dots, x_n \in X$. 依据上述定义, 可给出决策环境匹配公式如下:

$$\begin{cases} L \mapsto loc_1 \equiv \exists loc_2 \in L \wedge loc_2 = loc_1, \\ P \mapsto p_1 \equiv \exists p_2 \in P \wedge p_1.st \leq p_2.st \wedge p_2.et \leq p_1.et, \\ S \mapsto s_1 \equiv \exists s_2 \in S \wedge s_2 = s_1, \\ C \mapsto c_1 \equiv \neg \exists c_2 \in C \wedge c_1 = \neg c_2. \end{cases} \quad (1)$$

在上述公式中, $p_1.st$ 表示决策环境计划时间参数 p_1 的起始时间, $p_1.et$ 表示结束时间. 环境感知评估算法的推理规则可表示为

$$\begin{cases} \mathfrak{S}.L.loc_1 \wedge \mathfrak{S}.L.loc_2 \wedge \dots \rightarrow T.Envir.L, \\ \mathfrak{S}.P.p_1 \wedge \mathfrak{S}.P.p_2 \wedge \dots \rightarrow T.Envir.P, \\ \mathfrak{S}.S.s_1 \wedge \mathfrak{S}.S.s_2 \wedge \dots \rightarrow T.Envir.S, \\ \mathfrak{S}.C.c_1 \wedge \mathfrak{S}.C.c_2 \wedge \dots \rightarrow T.Envir.C. \end{cases} \quad (2)$$

由此, 若决策任务环境要求为 $T.Envir$, 则任一环境 \mathfrak{S} 的感知评估算法如下.

算法1 环境感知评估算法.

Step 1: $N_1 \leftarrow |T.Envir.L| + |T.Envir.P| + |T.Envir.S| + |T.Envir.C|$;

Step 2: 对每一个需求具有关系 $x_i \in T.Envir.L \cup T.Envir.P \cup T.Envir.S \cup T.Envir.C$ 的需求 x_i , \mathfrak{S} 利用评估函数寻找是否存在一组位置事实信息满足 $\mathfrak{S} \mapsto x_i$;

Step 3: 如果存在匹配函数满足 $\mathfrak{S} \mapsto x_i$, 则 $N_1 \leftarrow N_1 - 1$;

Step 4: 重复Step 2和Step 3直到所有需求 x_i 均获得匹配函数的评估;

Step 5: 如果 $N_1 = 0$, 则环境需求 $T.Envir$ 被环境 \mathfrak{S} 所满足, 且存在规则 $\mathfrak{S} \rightarrow T.Envir$, 否则, 说明存在环境需求 x_i 无法获得满足.

3 决策服务能力组织评估

3.1 决策服务的能力约束

首先给出服务能力的约束规则表达式.

约束3 对于任务语义而言, 存在形如 $T_i.Goal \neq \emptyset \wedge T_i.Status \neq \emptyset$ 的约束.

约束 4 任务语义 $T = (T_1, T_2, \dots)$ 的目标约束形如 $\forall T_i \in T \wedge \forall T_j \in T \rightarrow T_i.\text{Goal} \cap T_j.\text{Goal}_j = \emptyset$.

约束 5 决策服务的输入输出应服从如下约束:

$$\begin{cases} \text{DS}.D.I \wedge \text{DS}.\text{N}.In \wedge (\text{DS}.D.I \mapsto \text{DS}.\text{N}.In) \rightarrow, \\ \text{DS}.D.O \wedge \text{DS}.\text{N}.Out \wedge (\text{DS}.D.O \mapsto \text{DS}.\text{N}.Out) \rightarrow. \end{cases} \quad (3)$$

3.2 DS的能力可组织评估算法

对于子任务 T_i , DS_i 能够执行某个决策任务 T_i 当且仅当下列表达的规则能够得到满足:

$$\begin{cases} T_i.\text{Status}_i \rightarrow \text{DS}_j.D.I, \\ T_i.\text{Status}_i \rightarrow \text{DS}_j.\text{N}.In, \\ \text{DS}_j.D.O \wedge \text{DS}_j.\text{N}.Out \rightarrow T_i.\text{Goal}_i, \\ \text{DS}_j.\text{N}.price \rightarrow T_i.\text{Cost}_i. \end{cases} \quad (4)$$

其中 DS 评估的匹配函数语义如下:

$$\begin{cases} T_i.\text{Status} \mapsto \text{DS}_j.D.I \equiv \text{DS}_j.D.i_j \in T_i.\text{Status}, \\ \text{DS}_j.D.O \mapsto T_i.\text{goal} \equiv T_i.\text{goal} \in \text{DS}_j.D.O, \\ T_i.\text{Status}_i \mapsto \text{DS}_j.\text{N}.In \equiv \text{DS}_j.\text{N}.In \in T_i.\text{Status}_i, \\ \text{DS}_j.\text{N}.Out \mapsto T_i.\text{goal} \equiv T_i.\text{goal} \in \text{DS}_j.\text{N}.Out, \\ \text{DS}_j.\text{N}.price \mapsto T_i.\text{cost} \equiv \text{DS}_j.\text{N}.price \in T_i.\text{cost} \end{cases} \quad (5)$$

假设存在子任务 T_i , 可组织评估如下.

算法 2 DS 可组织能力评估算法.

Step 1: $N_2 \leftarrow |T_i.\text{Goal}|$.

Step 2: 对于每一个目标 $\text{goal} \in T_i.\text{Goal}$, DSP 利用匹配函数 $\text{DS}_j.D.O \wedge \text{DS}_j.\text{N}.Out \mapsto T_i.\text{goal}$ 进行目标匹配判定. 如果匹配函数满足, 则 $N_2 \leftarrow N_2 - 1$.

Step 3: 重复 Step 2 直至所有目标均获得匹配函数的判定.

Step 4: 若 $N_2 = 0$, 则算法进入 Step 5, 否则, 算法结束, 对于该 DS 端返回评价结果 $\text{capacity}(\text{node}_j) = 0$.

Step 5: 对于每一个 DS 端的服务 $\text{DS}_j.D.I$ 和 $\text{DS}_j.\text{N}.In$, 评估是否输入参数满足任务语义中的初始条件事实 $T_i.\text{Status}$. 如果无法满足, 则 DS 端退出算法, 并返回评估结果 $\text{capacity}(\text{DS}_j) = 0$, 否则, 继续下一步评估.

Step 6: 对 DS 端进行价格评价. 如果对于决策任务的成本需求, 满足匹配函数 $\text{DS}_j.\text{N}.price \mapsto T_i.\text{cost}$, 则说明 Horn 子句 $\text{DS}_j \rightarrow T_i$ 是有效的, 返回评价价值 $\text{capacity}(\text{DS}_j) = 1$, 否则返回评价价值 $\text{capacity}(\text{DS}_j) = 0.5$.

4 决策服务协作的自组织方法

4.1 决策任务的语义分解方法

本文给出决策任务语义分解计算实现复杂决策任务的语义分解. 假设 T_i 存在 $T_i.\text{Goal}' \subset T_i.\text{Goal}$ 无法单一 DS 求解, 则可通过下列计算定义进行分解.

定义 7 决策任务语义分解计算可以定义为

$$T_i^{\text{dec}} = [T_i.\text{TClass}_i, T_i.\text{Status}_i, (T_i.\text{Goal}_i - T_i.\text{Goal}'), T_i.\text{Cost}_i, T_i.\text{Envir}_i].$$

下面给出如下的决策任务语义分解方法.

算法 3 决策任务语义分解算法.

Step 1: 假设存在决策任务语义 T 和一组 DS. 对于任一 $\text{DS}_j \in \text{DS}$ 而言, 系统建立一个任务表 Q_j 来记录 DS_j 能够解决的任务目标情况;

Step 2: 对于任一 $\text{DS}_j \in \text{DS}$, 系统利用式 (5) 中的匹配函数进行 $T.\text{goal}_i$ 的评估;

Step 3: 若规则 $\text{DS}_j.D.O \wedge \text{DS}_j.\text{N}.Out \rightarrow T.\text{goal}_i$ 成立, 则 $Q_j \leftarrow T.\text{goal}_i$;

Step 4: 重复 Step 2 和 Step 3 直到所有 DS 均获得 Q_j ;

Step 5: 系统计算可能获得的分解组合且满足公式约束 $\bigcup Q_j = T.\text{Goal} \wedge \bigcap Q_j = \emptyset$;

Step 6: 系统通过 Q_j 获得拥有最小数量分解子任务的语义, 即该任务语义被分解为如下子任务: $\text{dec}.T_i = (T.\text{TClass}_i, T.\text{Status}_i, Q_j, T.\text{Cost}, T.\text{Envir})$.

4.2 自组织矩阵

本文提出自组织矩阵的方法, 对经评估可用的候选服务进行优选. 假设存在 $T = [T_1, T_2, \dots, T_n]$, 以及向量 $\text{DS} = [\text{DS}_1, \text{DS}_2, \dots, \text{DS}_m]^T$, 则这两个向量可以组成 $T \times \text{DS}$ 矩阵, 定义如下.

定义 8 DS 自组织的能力矩阵可以定义为如下形式:

$$\Gamma = \begin{bmatrix} \text{capacity}_{11} & \cdots & \text{capacity}_{1m} \\ \vdots & \ddots & \vdots \\ \text{capacity}_{n1} & \cdots & \text{capacity}_{nm} \end{bmatrix}, \quad (6)$$

其中 capacity_{ij} 的取值来源于算法 2.

同样, 定义了环境感知矩阵, 对自组织过程中的 DS 满足环境要求的情况进行优选.

定义 9 DS 自组织的环境矩阵形式如下:

$$\Psi = \begin{bmatrix} \text{environment}_{11} & \cdots & \text{environment}_{1m} \\ \vdots & \ddots & \vdots \\ \text{environment}_{n1} & \cdots & \text{environment}_{nm} \end{bmatrix}, \quad (7)$$

其中 environment_{ij} 值为

$$\text{environment}_{ij} = \begin{cases} 1, & \text{DS}_j.I \rightarrow T_i.\text{Envir}; \\ 0.5, & \text{DS}_j.D.\text{S_mobile} \rightarrow T_i.\text{Envir}; \\ 0, & \text{else.} \end{cases} \quad (8)$$

4.3 决策服务协作自组织算法

决策服务协作自组织算法建立在上述环境感知评估基础上,从能力评价出发,借助自组织矩阵进行服务优选,得到最终入选DS.

算法4 DS协作自组织算法.

Step 1: Task $\leftarrow T_i$; Candidate $\leftarrow \emptyset$; Selected $\leftarrow \emptyset$.

Step 2: 对于决策任务向量 $T = [T_1, T_2, \dots, T_n]$ 和DS向量 $\text{DS} = [\text{DS}_1, \text{DS}_2, \dots, \text{DS}_m]^T$, 计算能力矩阵 Γ 和环境矩阵 Ψ .

Step 3: 对于每一个 $T_i \in \text{Task}$, 若DS满足约束条件 ($\text{capacity}_{ij} + \text{environment}_{ij} \geq \text{capacity}_{ij} \geq 0.5$), 则从矩阵中选择满足条件 $\max[\text{capacity}_{ij} + \text{environment}_{ij}]$ 的 DS_j , 并执行 $\text{Candidate} \leftarrow T_i | \text{DS}_j$, $\text{Task} \leftarrow \text{Task} - T_i$.

Step 4: 对于每一对 $T_i | \text{DS}_j \in \text{Candidate}$, 通过下列公式计算其竞争力参数:

$$\text{competence}(\text{DS}_j) = \frac{\text{DS}_j.N.\text{price}}{T_i.\text{cost}} + \frac{\text{DS}_j.D.\text{S}.P}{T_i.\text{Envir}.P}. \quad (9)$$

Step 5: 针对 $T_i \in \text{Candidate}$, 选择具有 $\min[\text{competence}(\text{DS}_j)]$ 的 DS_j , 并执行 $\text{Select} \leftarrow T_i | \text{DS}_j$, $\text{Candidate} \leftarrow \text{Candidate} - Z$. 这里 $Z = \{x_k | \text{DS}_j | x_k = T_i\}$.

Step 6: 重复 Step 4 和 Step 5 直到 $\text{Candidate} = \emptyset$.

Step 7: 若 $\text{Task} \neq \emptyset$, 则对每一个 $T_i \in \text{Task}$, 选择满足条件 $\text{capacity}_{ij} = 0.5 \wedge \min(\text{DS}_j.N.\text{price}/T_i.\text{cost})$ 的 DS_j , 并执行 $\text{Select} \leftarrow T_i | \text{DS}_j$, $\text{Task} \leftarrow \text{Task} - T_i$.

Step 8: 如果 $\text{Task} \neq \emptyset$, 则通过算法3对 T_i 进行语义分解, 形成新的任务向量 $\text{dec}.T$.

Step 9: $\text{Task} \leftarrow \emptyset$, $\text{Task} \leftarrow \text{dec}.T$.

Step 10: 重复 Step 2 ~ Step 8, 直到 $\text{Task} = \emptyset$.

Step 11: 返回最终选择的DS集 Selected.

5 实验分析

在自主开发的基于服务的决策支持系统原型系统上进行决策服务组织准确性、协作组织效果分析、时间复杂性等分析.

5.1 决策服务组织准确性分析

首先通过人工手段从所有服务中挑选出可以获得的最佳服务, 然后通过作为对比实验, 设定不同选择算法所能得到的DS的情况. 实验分4组: Group 1中挑选对于决策需求类别 $TClass_i$ 服务次数最多的

DS_i ; Group 2中选择满足决策需求目标 Goal_i 最多的决策服务 DS_i ; Group 3选择满足决策需求 Envir_i 最多的服务; Group 4中则采用本文DSCS算法选择. 共进行了200次协作服务组织选择, 记录了准确系数 Ratio1 , 结果如图1所示.

$$\text{Ratio1} = \sum_{i=1}^N \left(\frac{\text{选择出的最佳服务个数}}{\text{预设的最佳决策服务个数}} \right) / N.$$

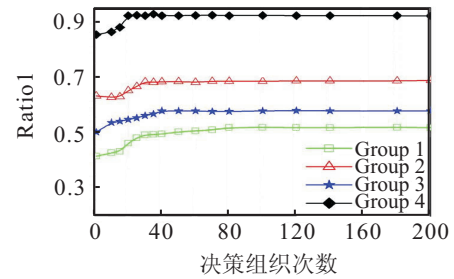


图1 决策服务组织准确性比较

5.2 决策服务组织效果分析

实验目的是检验DS协作组织算法所能达到的最终决策效果. 与上述实验相同, 通过4组对比实验来检验效果. 在记录每次协作组织后, 利用所组织的DS进行协作决策, 并对协作决策的结果打分(该打分为人工打分, 区间为0-100). 记录不同实验组所组织的协作决策能得到的平均得分

$$\text{Ratio2} = \sum_{i=1}^N (\text{协作决策得分}) / N.$$

由图2可见, 采用本文算法进行协作决策组织明显得分高于其他方法.

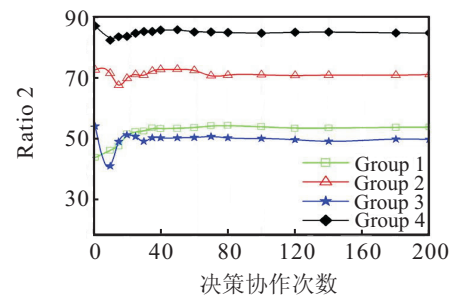


图2 决策服务组织效果分析

5.3 协作组织的时间开销分析

时间性能分析的目的是检测通过本文方法进行协作组织是否会带来巨大的时间开销. 时间性能分析实验与上述2个实验方案相同, 记录了完成协作组织的选择过程和完成整个协作决策过程所需要的两个时间指标, 分别记录在图3和图4中. 从图3中可以看出, 本文算法耗时高于前3组, 分析内容是本文算法中评估和自组织分析带来的开销. 然而, 其平均时间约为8.18s, 属于可接受的范围. 从图4中可以看出, 经本文算法组织后执行整个协作决策的时间已经优

于其中的两组. Group1 的时间虽较快,但出现了服务无法执行完任务并退出的情况. 可见,本文提出的算法在时间开销上是可接受的.

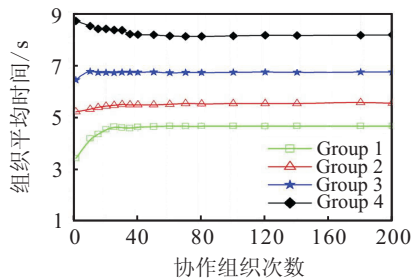


图 3 决策服务组织平均时间开销

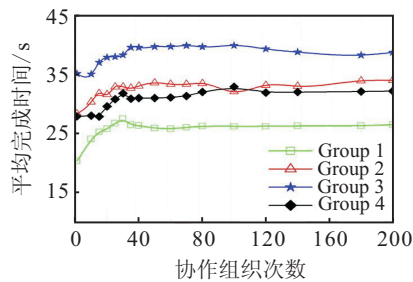


图 4 完成协作总体平均时间开销

6 结 论

本文针对如何选择组织参与协作的决策服务问题,提出了一种基于语义的协作自组织方法.该方法引入 Horn 子句理论作为选择推理的逻辑规则表达基础,通过语义计算基础上的决策环境和决策服务能力评估,挑选出最符合需求的决策服务,最终自组织形成有效的协作服务团队.下一步的工作将集中于利用网络服务间的社会关系,进一步优化自组织的算法.

参考文献(References)

[1] Erm C. Decision support systems: A summary, problems and future trends[J]. Decision Support Systems, 1988, 4(4):

355-363.

- [2] Indiramma M, Anandakumar K R. Collaborative decision making framework for multi-agent system[C]. Proc of the Int Conf on Computer and Communication Engineering 2008. Kuala Lumpur, 2008: 140-1146.
- [3] Stock Christian, Bishop Ian D, O'Connor Alice N, et al. SIEVE: Collaborative decision-making in an immersive online environment[J]. Cartography and Geographic Information Science, 2008, 35(2): 133-144.
- [4] Dabholkar A, Gokhale A. An approach to middleware specialization for cyber physical systems[C]. The 29th IEEE Int Conf on Distributed Computing Systems Workshops. Montreal, 2009: 73-79.
- [5] Swarnkar R, Choudhary A K, Harding J A. A framework for collaboration moderator services to support knowledge based collaboration[J]. J of Intelligent Manufacturing, 2011, 23(5): 1-21.
- [6] Purao Sandeep, Khatri Vijay. Service-oriented architecture and computing[J]. J of Database Management, 2011, 22(2): 1-4.
- [7] Almorsy Mohamed, Grundy John, Ibrahim Amani S. Collaboration-based cloud computing security management framework[C]. Proc of 2011 IEEE Int Conf on Cloud Computing. Beijing, 2011: 364-371.
- [8] 张波,向阳. 基于语义的决策模型能力评估与选择方法[J]. 控制与决策, 2010, 25(9): 1324-1328. (Zhang B, Xiang Y. Semantics based decision model capacity evaluation and selection[J]. Control and Decision, 2010, 25(9): 1324-1328.)
- [9] Alfred Horn. On sentences which are true of direct unions of algebras[J]. J of Symbolic Logic, 1951, 16(1): 14-21.

(上接第1413页)

[9] 杨争斌,郭福成,周一宇. 基于中心差分的机动辐射源单站被动跟踪算法[J]. 电子与信息学报, 2008, 30(12): 2871-2875. (Yang Z B, Guo F C, Zhou Y Y. Central difference based maneuvering emitter tracking with a single passive observer[J]. J of Electronics & Information Technology, 2008, 30(12): 2871-2875.)

[10] 刘学,焦淑红. 自适应迭代平方根 UKF 的单站无源定位算法[J]. 哈尔滨工程大学学报, 2011, 32(3): 372-377. (Liu X, Jiao S H. Adaptive iterated SRUKF for single observer passive location[J]. J of Harbin Engineering University, 2011, 32(3): 372-377.)

[11] Zhang J, Zhan R H. Posterior Cramér-rao bounds analysis for passive target tracking[J]. J of Electronics(China), 2008, 25(1): 84-88.

[12] Guo L, Tang B, Liu G, et al. Posterior Cramér-rao lower bounds for multitarget bearings-only tracking[J]. J of Systems Engineering and Electronics, 2008, 19(6): 1127-1132.

[13] 张贤达. 矩阵分析与应用[M]. 北京: 清华大学出版社, 2004: 64-68. (Zhang X D. Matrix analysis and application[M]. Beijing: Tsinghua University Press, 2004: 64-68.)