

基于服务的决策问题求解新方法*

(南京大学控制与系统工程系 江苏南京 210093)

摘要: 当前的军事决策系统没有真实融入人的经验、智慧、灵感等智能, 求解非结构化问题的效果难以令人满意。针对此问题, 本课题组提出了人件服务的概念, 并实现了人件服务的原型。在此基础上提出了软件服务与人件服务的自动组合算法, 用于建立基于人件服务的决策问题求解流程。案例表明此方法可有效地完成服务组合并形成决策问题求解流程。

关键词: 人件服务、决策问题求解、服务组合

0 引言

随着军事科学技术的发展和进步, 军队的机动能力、火力、突击力、电子对抗能力空前提高, 战役节奏大大加快, 战机稍纵即逝。侦察能力的增强产生了海量信息, 而其中很大一部分是非结构化的。未来的战场态势将会复杂多样, 已经超出了软件可判断的范围。根据 Fitts[1]等人对 人机功能分配的研究, 人擅长感知视觉、听觉信号, 处理非结构化信息, 进行归纳推理, 能对局势有合理的判断, 处理问题具有社会意识。机器擅长快速响应控制信号, 存储、删除大量信息和演绎推理。因此, 面对复杂的决策问题, 仅依靠软件已经不能解决问题, 决策系统需要真实融入人的智能。

历史上, 很多研究者试图将人的智能引入决策问题求解过程。第一个专家系统于 1965 年由费根鲍姆研制成功。专家系统综合了定量分析技术和不确定推理的优势, 能够解决部分定性分析问题[2]。浙江大学的路甬祥、陈鹰等从机械科学的角度提出了一种新型智能系统的设计方法—“人机一体化系统”(Humachine system), 并从其定义、内涵、理论框架、建模、关键技术构成、实现方式及其设计特点出发, 进行了全面的阐述[3]。北京科技大学涂序彦教授等提出了“软件人”(Soft Man)的概念及其结构模型, 并取得了一定的理论与应用研究成果[4]。然而, 以上研究成果还不足以满足未来战争的需求。据 2011 年 9 月美国空军官方杂志《Airman》报道, 自“9.11 事件”后, 美国无人机技术取得了显著的发展。但与此同时带来的问题是每天大约会拍摄 1500 小时的视频和 1500 张照片, 空军需要从中发现有用的信息以确定下一步的行动方案。这些信息分析的工作都是由人来完成。此类任务必须要人参与, 但如果仅依靠本地的人, 那么任务非常繁重, 并且很可能无法及时完成任务。因此, 迫切地需要一种新的决策问题求解方法, 适时快速地优化组合网络上的软件资源和对问题求解有帮助的人, 实现人机智能良好的结合, 打赢信息化条件下的未来战争。

针对以上的挑战, 本课题组提出了“人件服务”的概念, 并实现了人件服务的原型。本文在此基础上提出了软件服务与人件服务的自动组合算法, 将人的智能以服务的形式融入求解过程, 用服务组合的方法完成问题求解。

1 人件服务简介

1.1 面向服务架构

面向服务的体系结构 (Service-Oriented Architecture, SOA) 是一个组件模型, 它将应用程序的不同功能单元 (称为服务) 通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的, 它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种各样的系统中的服务可以使用一种统一和通用的方式进行交互。SOA 具有许多明显的特征和优势, 如服务封装、服务共享、接口标准化且开放、松耦合、跨平台等。

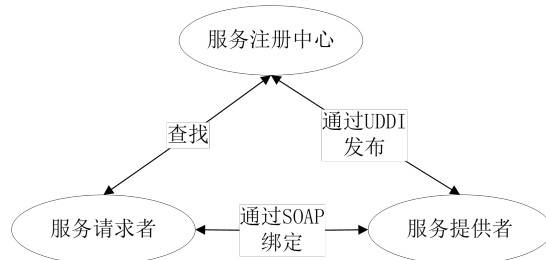


图 1 SOA 体系架构

*项目资助: 1 基于“人即服务”理念的决策支持系统设计理论和方法研究, 国家自然科学基金 (71171107); 2 基于人件的新型决策支持系统设计的关键技术研究, 江苏省普通高校研究生科研创新计划 (CXZZ12_0069)。

SOA 体系结构中共有三种角色，如图 1 所示：

服务提供者：发布自己的服务，并且对服务请求进行响应。

服务注册中心：注册已经发布的服务，对其进行分类，并提供搜索服务。

服务请求者：利用服务注册中心查找所需的服务，然后使用该服务。

1.2 人件服务的概念

英国哲学家波兰尼把知识分成缄默知识和显性知识两类。缄默知识是指以整体经验为基础的只可意会，不可言传的知识。显性知识是指可以用语言文字表达出来的知识，又可分为可编码化的部分和不可编码化的部分。人件服务的提出是为了把人从系统的使用者变成系统的组件，把人的缄默知识和显性知识中不可编码的部分融入系统

定义 1：人件。“人件”是从决策角度专指参与决策活动并对决策问题求解过程提供知识、经验和智慧等支持的人。

定义 2：人件服务。“人件服务”是人件作为军事决策系统组件时，人件的特定功能（如分析、判断、建议、决定等）的服务化实现形态，是人智慧发挥作用的载体，并以和软件服务一致的形式被描述、发布、发现、组装和调用。

2 人件服务与软件服务的组合算法

含有人件服务的决策问题求解流程如图 2 所示。

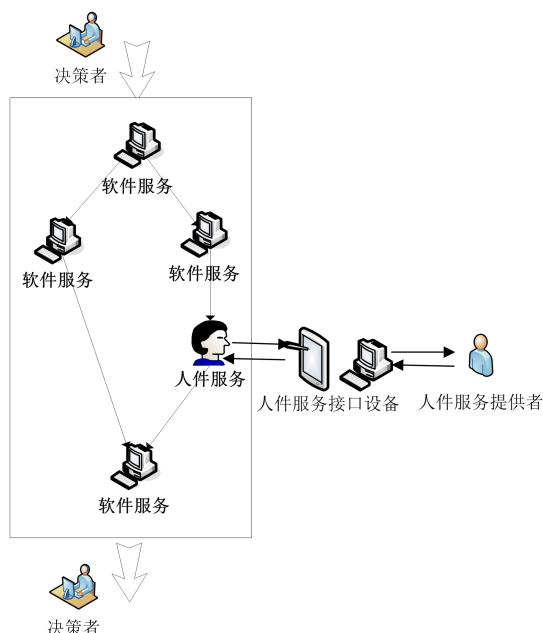


图 2 决策问题求解流程

在图 2 中，有些问题软件服务可以求解，有些问题则不能，求解流程需要调用人件服务。此时流程将暂停，将需要人件服务求解的问题通过人件服务接口设备显示，人件服务提供者看到问题的具体内容后利用其经验、智慧求解问题，并将结果通过接口设备输入流程。流程得到输入的内容后继续执行，直至结束。

要实现上述的问题求解，关键是对软件服务和人件服务进行组合。然而，软件服务与人件服务的接口在实现上是不同的。软件服务的输入要求计算机能理解，人件服务的输入要易于人件服务提供者理解。此处理解指的是能进行处理并给出结果。若采用传统的基于接口匹配，会造成虽然匹配成功，但服务却不能执行的问题。同样是输入一个字符串，人件服务会根据字符串的意义进行求解，而软件服务只能对其中的字符进行操作。所以本文针对此问题提出了根据输入输出接口参数进行分类匹配的算法。

定义 3：软件服务和人件服务形式化描述 ($WS(Input, Output)$)。其中 $Input$ 为服务的输入参数集合， $Output$ 为服务的输出参数集合。

定义 4：输入 ($Input$)。输入可表示为三元组：

$$Input = \{I_s, I_h, I_c\},$$

其中 I_s 表示仅计算机可以理解的输入参数集合，例如复杂的数值计算； I_h 表示仅人可以理解的输入参

数集合, 例如需要人的经验才能理解的一段话; I_c 表示计算机和人都可以理解的输入参数集合, 例如简单的逻辑推理。若一个服务为软件服务, 则 I_h 为 \emptyset , 因为对软件服务输入仅有人可以理解的参数是没有意义的。同样的, 若一个服务为人件服务, I_s 为 \emptyset , 对人件服务输入仅有软件可以理解的参数也是没有意义的。

定义 5: 输出 (*Output*)。输出可表示为三元组:

$$Output = \{O_s, O_h, O_c\},$$

其中 O_s 表示仅计算机可以理解的输出集合, O_h 表示仅人可以理解的输出集合, O_c 表示计算机和人都可以理解的输出集合。

输入和输出由软件服务的开发者或者人件服务提供者定义。输入参数集合描述了服务的类别 (软件服务或者人件服务), 输出参数集合描述了此服务的结果可供哪种服务使用。

定义 6: 概念间的语义关系。对于三个概念 C_i 、 C_j 和 C_k , 在领域本体中, 如果 C_i 被定义为 C_j 的 *equivalentClass*, 则称概念 C_i 和概念 C_j 语义相等, 记为 $C_i \equiv C_j$; 如果 C_j 被定义为 C_i 的 *subClassOf*, 则称概念 C_i 语义包含 C_j , 记为 $C_i \supseteq C_j$; 如果 C_k 被定义为 C_i 和 C_j 的 *unionOf*, 则称概念 C_k 为 C_i 和 C_j 的并集, 记为 $C_k = C_i \cup C_j$ 。

对于两个概念集合 SC_i 和 SC_j , 如果对 SC_i 中的任一概念 C_i , 在 SC_j 中都存在一个概念 C_j , 满足 $C_i \equiv C_j$ 或 $C_i \supseteq C_j$, 则称 SC_i 语义包含 SC_j , 记为 $SC_i \supseteq SC_j$; 如果有 $SC_i \supseteq SC_j$ 且 $SC_j \supseteq SC_i$, 则 $SC_i \equiv SC_j$, 此时称 SC_i 和 SC_j 语义相等。

定义 7: 服务请求。一个服务请求可以用下面的表达式来描述:

$$WSR_k(Input_k, Output_k),$$

其中, WSR_k 是服务请求的名字; $Input_k$ 是该服务请求的输入集合, $Input_k = \{I_{sk}, I_{hk}, I_{ck}\}$, I_{sk} 表示服务请求中仅计算机可以理解的输入集合, I_{hk} 表示仅人可以理解的输入集合, I_{ck} 表示计算机和人都可以理解的输入集合。 $Output_k$ 是该服务请求的输出集合, $Output_k = \{O_{sk}, O_{hk}, O_{ck}\}$, 其中 O_{sk} 表示服务请求中仅计算机可以理解的输出集合, O_{hk} 表示仅人可以理解的输出集合, O_{ck} 表示计算机和人都可以理解的输出集合。

定义 8: 语义关联。对于两个服务 $WS_i(Input_i, Output_i)$ 和 $WS_j(Input_j, Output_j)$, $Input_i = \{I_{si}, I_{hi}, I_{ci}\}$, $Input_j = \{I_{sj}, I_{hj}, I_{cj}\}$, $Output_i = \{O_{si}, O_{hi}, O_{oi}\}$, $Output_j = \{O_{sj}, O_{hj}, O_{cj}\}$ 。如果有 $O_{si} \cup O_{oi} \supseteq I_{sj} \cup I_{cj}$ 且 $O_{hi} \cup O_{oi} \supseteq I_{hj} \cup I_{cj}$, 则称从服务 WS_i 到 WS_j 语义关联, 记为 $WS_i \succ WS_j$, 并将其语义关联度记为 $AD(WS_i, WS_j)$ 。其中, WS_i 称为 WS_j 的前驱服务, WS_j 称为 WS_i 的后继服务。

对于服务请求 $WSR_k(Input_k, Output_k)$ 和服务 $WS_i(Input_i, Output_i)$, 如果 $I_{si} \cup I_{ci} \supseteq I_{sk} \cup I_{ck}$ 且 $I_{hi} \cup I_{ci} \supseteq I_{hk} \cup I_{ck}$, 则称从服务请求 WSR_k 到服务 WS_i 语义关联, 记为 $WSR_k \succ WS_i$, 并将其语义关联度记为 $AD(WSR_k, WS_i)$, 其中, WS_i 称为 WSR_k 的后继服务; 如果 $O_{si} \cup O_{oi} \supseteq O_{sk} \cup O_{ck}$ 且 $O_{hi} \cup O_{oi} \supseteq O_{hk} \cup O_{ck}$, 则称从服务 WS_i 到服务请求 WSR_k 语义关联, 记为 $WS_i \succ WSR_k$, 并将其语义关联度记为 $AD(WS_i, WSR_k)$, 其中 WS_i 称为 WSR_k 的前驱服务。

定义 9: 一个服务组合是指能够满足某个服务请求 WSR_k 的一个服务序列 WS_1, WS_2, \dots, WS_n 序列满足以下 3 个条件:

- (1) $WSR_k \succ WS_1$;
- (2) 该序列中任意两个相邻的服务 WS_i 和 WS_{i+1} 都满足 $WS_i \succ WS_{i+1}$;
- (3) $WS_n \succ WSR_k$ 。

定义 10: 领域本体的概念间语义相似度 $S(C_i, C_j)$ 。

$$S(C_i, C_j) = \frac{A}{d}$$

A 是一个可调节的参数, 由经验取值。 d 为概念间语义距离[2]。

如果概念集合 $SC_i(C'_1, C'_2, \dots, C'_m)$ 和 $SC_j(C''_1, C''_2, \dots, C''_n)$, 满足 $SC_i \supseteq SC_j$, 则用 $SS(C_i, C_j)$ 表示这两个概念集合的相似度, 计算方法为[2]:

$$SS(SC_i, SC_j) = \frac{1}{m} \left| \sum_{i=1}^m \max(S(C'_i, C''_i)) \right|$$

服务之间关联度计算公式为:

$$AD(WS_i, WS_j) = \frac{\sum_{\alpha} \alpha \cdot S(C'_i, C''_j)}{\sum_{\alpha} \alpha} \quad \alpha \text{ 为软件}$$

可理解部分相似度的权重。服务请求与服务之间的关联度可采用类似的方法定义。

服务组合满意度是指该服务组合满足某个服务请求的程度, 记为 SAT 。对于服务请求 WSR_k , 一个服务组合 WS_1, WS_2, \dots, WS_n 的满意度 SAT 用如下式计算:

$$SAT = AD(WSR_k, WS_1) \times \prod_{i=1}^{n-1} AD(WS_i, WS_{i+1}) \times AD(WS_n, WSR_k)$$

基于领域本体的服务自动组合方法的基本思想是: 在领域本体的支持下, 首先找出服务请求的后继服务, 然后找出该服务的后继服务, 以此类推, 直至找出的某个服务的后继服务是服务请求, 所得到的这个服务组合就是服务组合的结果。由于一个服务可能存在多个后继服务, 所以, 这些语义关联的服务构成了一个 Web 服务组合图($WS-Graph$)。

算法 1. $WS-Graph$ 生成算法

CreatWSGraph()//生成 $WS-Graph$

{

 for(i=0; i<n; i++)//n 为服务库中服务的数量

 for(j=0; j<n; j++)

 AD[i][j]= AD(WS_i, WS_j); //AD[n][n]为服务库中所有服务相互之间关

 //联度的存储矩阵

 for(p=0; p<n; p++)

 AD[n][p]= AD(WSR_k, WS_p); //扩展服务关联度矩阵 AD 的一行用来存

 //储服务请求的输入与各服务的关联度。

 for(q=0; q<n; q++)

 AD[q][n]= AD(WS_q, WSR_k); //扩展服务关联度矩阵 AD 的一列用来存

 //储各服务与服务请求的输出的关联度。

 AD[n][n]=0;

}

矩阵 AD[n+1][n+1]即为 $WS-Graph$ 的矩阵表示。

对于一个服务请求, 在其相应的 $WS-Graph$ 中, 从起点 (服务请求的 $Input_k$) 到终点 (服务请求的 $Output_k$) 的每条路径都是一个满足该服务请求的服务组合。其中, 最优的服务组合是指服务组合满意度最大的路径。算法 2 用于求起点到其余各服务满意度最大的路径, 结果中起点到终点满意度最大的路径为最优服务组合。

算法 2 最大满意度服务组合求解算法

MostSatisfyingServiceComposition(AD[n+1][n+1])

//假设 S 为已求得最大满意度的服务集合

//起点到其余各服务的最大满意度路径为 P[v], 最大满意度为 D[v]

{

 for(v=0; v<n+1; v++)

 {

 final[v]=FALSE;

 D[v]=AD[n][v];

 for(w=0; w<n+1; w++) P[v][w]=FALSE; //设空路径

 if(D[v]>INFINITY) {P[v][n+1]=TRUE; p[v][v]=TRUE;}

 }

```

D[n+1]=0;final[n+1]=TRUE;//初始化,  $WSR_k(Input_k, Output_k)$  属于 S 集
for(i=0;i<n+1;i++)//
{
    max=INFINITY;//当前所知离起点满意度最大的值
    for(w=0;w<n+1;w++)
        if(!final[w])//w 顶点在 V-S 中
            if(D[w]>max){v=w;max=D[w];} //从起点开始到 w 的满意度更大
    final[w]=TRUE;//w 加入 S
for(w=0;w<n+1;w++)//更新当前最大满意度路径及最大满意度
    if(!final[w]&&max*AD[v][w]>D[w])
    {
        D[w]=max*AD[v][w];
        P[w]=P[v];P[w][w]=TRUE;
    }
}
}
}
}

```

定理 1: 算法 2 求出的是满意度最大的路径。

证明: 假设此路径上有一个顶点不在 S 中, 则说明存在一条终点不在 S 而满意度更大的路径。但是这并不可能。因为算法 2 是按满意度递减的次序产生各满意度最大的路径, 故此比此路径满意度更大的所有路径均已产生, 它们终点必定在 S 中。

3 案例分析

本文以时敏目标打击为例, 验证本文方法的有效性。

时敏目标通常是指战场上随机出现的、打击机会受目标的时间窗口严格限制的高价值目标。它们一般具有状态变化频繁、定位难、跟踪难、可打击时间短等特征。时敏目标打击既涉及到复杂的计算, 又涉及到需要人的智能才能完成的任务, 如识别目标、制定方案。因此需要快速地构建包含计算机智能和人的智能的求解方案。

采用本文提出的基于服务的决策问题求解方法可很好的解决此问题。人和计算机擅长求解不同类型的问题, 因此擅长求解问题的人可发布自己的人件服务; 而针对计算机擅长求解的问题可开发相应的软件服务求解。

假设服务库中有以下服务:

表 1 服务库中的服务

服务名称	服务功能	服务类别	输入			输出		
			仅人可理解	仅计算机可理解	人和计算机都可理解	仅人可理解	仅计算机可理解	人和计算机都可理解
WS_1	识别目标	人件服务	目标图像、位置、速度			目标意图		目标类别
WS_2	制定方案	人件服务	目标意图		目标类别			打击方案
WS_3	目标打击	软件服务		打击方案			作战时间、弹药消耗	
WS_4	效能评估	软件服务		作战时间、弹药消耗				评价结果
WS_5	产生预案	软件服务		目标意图	目标类别			打击预案
WS_6	调整方案	人件服务	打击预案				打击方案	

领域本体如图 3 所示:

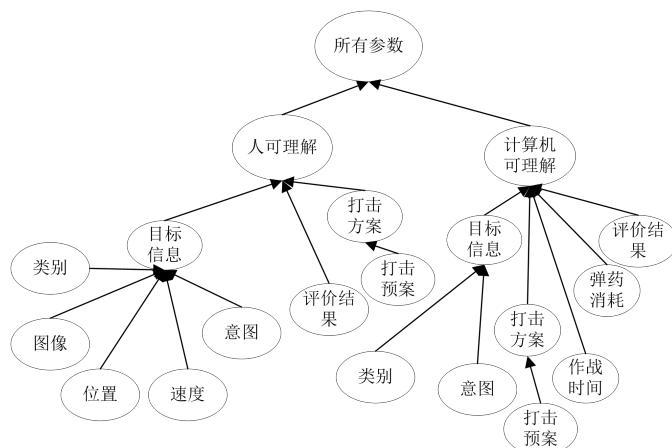


图 3 领域本体

当有服务请求 WSR_k 时,可采用软件服务与人件服务自动组装的方法完成。利用算法 1,概念间语义相似度定义中 A 的取值不影响概念间语义相似度的大小顺序,因此也不影响最终服务组合的结果,为计算方便,此处取 4,得到 WS-Graph 为:

表 2 WS-Graph

	Input	Input	Input	Input	Input	Input	Output
Output	0	1	0	0	0.5	0	0
Output	0	0	1	0	0	0.8	0
Output	0	0	0	1	0	0	0
Output	0	0	0	0	0	0	1
Output	0	0	0	0	0	1	0
Output	0	0	1	0	0	0	0
Input	1	0	0	0	0	0	0

利用算法 2,求得的满意度最大的服务组合序列是 WS_1, WS_2, WS_3, WS_4 。第 3 节提出的算法有效避免了出现 WS_5 成为 WS_1 的情况,可成功找出满意度最大的服务组合。

4 结论

本文阐述了含有人件服务的决策问题求解流程,提出了这软件服务与人件服务的自动组合算法。引入人件服务后问题求解流程可融入人的智能,并具有 SOA 的优点。案例验证了设计方法的可行性与服务组合算法的有效性。

参考文献

[1] FITTS, P. M. (1951). Human engineering for an effective air navigation and traffic control system. Ohio state University Foundation Report, Columbus, OH.
 [2] 李曼,王大治,杜小勇,王珊.基于领域本体的 Web 服务动态组合.计算机学报,2005 年 4 月第 28 卷第 4 期.
 [3] Kuhlmann T, Lamping R, Massow C. Intelligent Decision Support[J]. J of Materials Processing Technology, 1998, 76(1-3) : 257-260.
 [4] 路甬祥,陈鹰.人机一体化系统科学体系 and 关键技术[J].机械工程学报,1995,31(1):1-7.
 [5] 曾广平,涂序彦,王洪泊.“软件人”研究及应用[M].北京:科学出版社,2007.
 [6] 王宝祥,周献中,盛寅,陆晓明,毛可.面向人件服务的决策支持系统接口研究.计算机科学与探索.