

基于粒度计算和面向使命的组织适应性设计方法^{*})

修保新 张维明 刘 忠 阳东升

(国防科技大学信息系统与管理学院 C⁴ISR 国防科技发展重点实验室 长沙 410073)

摘 要 介绍了面向使命的组织设计,讨论了组织的适应性设计问题和组织适应性的代价。在此基础上提出了基于粒度计算的组织适应性设计方法,设计过程包括3个阶段:任务集和平台集粒化阶段、粒内规划阶段和决策分层阶段;并给出了平台集粒化的具体方法。案例分析表明本文的方法具有较好的性能。

关键词 使命,组织设计,粒度计算,适应性设计

Mission-based Organizational Adaptive Design Methodology Based on Granular Computing

XIU Bao-Xin ZHANG Wei-Ming LIU Zhong YANG Dong-Sheng

(School of Information System & Management, National University of Defense Technology, Changsha 410073)

Abstract In this paper, mission-based organization and organizational design are introduced, and the cost of organizational adaptation is discussed. Then we propose an organizational adaptive design methodology based on granular computing. The three phases of this design process are proposed. And the approach to the first phase is investigated. The numerical example shows the better performance of our design methodology.

Keywords Mission, Organizational design, Granular computing, Adaptive design

随着信息和网络技术的发展,现代组织正面临前所未有的挑战,这种挑战包括组织所处环境的不稳定性、组织使命的复杂性以及组织功能的多样性、组织结构上的松散性。在不确定使命(任务)环境下设计组织时,设计者面对的使命环境参数(比如任务对资源的需求)往往不可能准确获得。一旦任务开始执行,使命环境参数值可能会发生变化。另外,在任务执行过程中,会有突发事件发生,比如资源平台失效、决策节点故障、任务临时改变等等,这些突发事件会改变使命环境或者组织约束。这些都会使得静态环境下设计的优化组织不能与当前实际的使命环境相匹配。为了解决这种动态的、不确定使命环境下的组织设计问题,可以对特定的使命建立优化的组织,并能够随着环境变化进行动态重构,从而始终保持很好的性能,这种方法称为组织的适应性设计。

对于这一问题的研究,可以应用计算组织理论和 Petri 网交互模型建立适应性组织模型^[13~16]。近几年通过对团队决策的研究表明:组织的优化设计完全在于实际的任务结构和组织运作环境的关键属性^[1,17],这种结论导致应用系统工程技术来设计人类组织。Levchuk 等^[2,3]采用系统工程技术设计面向任务的组织,提出了组织的3阶段设计方法,但是由于该法效果并不理想^[5],而且方法本身无法扩展来进行组织的适应性设计,因此只能根据环境变化在事先设计好的几个组织之间进行适应性选择^[4]。近来,我们应用系统工程技术提出了基于粒度计算的组织设计方法^[5],在一定程度上提高了组织性能,而且该方法具有对环境参数敏感度小的特点。本文将扩展该方法,研究面向使命的组织适应性设计问题。

1 面向使命的组织

根据组织权变理论,没有公认一致的、普遍适用的最优组

织结构和形式,只有在某一具体使命下适用于这一具体条件的最优组织^[6]。建立组织的实体描述和任务模型,提出组织与使命的适应性测度,是组织优化设计的前提。

1.1 组织的实体与任务

Levchuk 等^[2]根据兵力组织的特点提出了决策者、平台和任务的概念。

定义 1 平台是物理资源载体,拥有资源能力,执行特定的功能。平台集记为 P ,对每一个平台 $P_k(k=1,2,\dots,K)$,定义它的最大速度 v_k 和它的能力向量 $R(P_k)=[r_{k1}, r_{k2}, \dots, r_{kL}]$,其中 r_{kl} 规定了平台 P_k 拥有的功能类型 l 的数量($l=1,2,\dots,L,L$ 是功能类型的数目)。

定义 2 任务是一种由一个平台或者一组平台执行的行动,它由相关功能完成。任务集记为 T ,通过下列基本属性定义任务 $T_i(i=1,2,\dots,N,N$ 是任务的数目):估计的处理时间 t_i ;地理限制向量(例如,位置 (x_i, y_i) 限定了任务 T_i 和 T_j 的距离 d_{ij});资源需求向量 $R(T_i)=[R_{i1}, R_{i2}, \dots, R_{iL}]$,这里 R_{il} 表示成功地处理任务 T_i 所需的功能类型 l 的能力。

定义 3 任务图定性描述任务间的依赖关系,如任务的优先顺序、数据流程以及任务间的输入输出关系等。一个有向非循环的任务图描绘了执行使命的一个计划,具体例子见案例分析中的图 1。本文以后也将任务集和任务图一起称为使命。

定义 4 决策者 DM(Decision-maker)是信息处理的实体。其能力可控制必要的平台资源来执行任务,如果任务的执行不会对其能力造成损害。根据需要可定义决策实体的知识和能力。决策者可以是作战指挥中的指挥官,或者企业组织中的经理人员。组织中可用的决策者集合记为 $DM=\{DM_m | m=1,2,\dots,M\}$ 。

^{*})国家自然科学基金资助项目(批准号:60504036)、高等学校博士学科点专项科研基金资助课题(批准号:20049998021)。修保新 博士研究生,主要研究方向:组织设计、组织适应性。

组织包括决策者集、平台到决策者的分配、决策者间的协作结构。组织用来处理一个使命，目标是决策者的内部工作负载和外部工作负载最小化，以及整个使命的完成时间最小化^[2]。

组织设计参数包括任务-决策者分配矩阵 TD 、平台-决策者分配矩阵 PD 、任务-平台分配矩阵 TP 和决策者之间的决策层次链接矩阵 DD ，分别定义如下：

$$TD(i,m) = \begin{cases} 1, & \text{如果任务 } T_i \text{ 分配到 } DM_m, \\ 0, & \text{否则.} \end{cases}$$

$$PD(k,m) = \begin{cases} 1, & \text{如果平台 } P_k \text{ 分配到 } DM_m, \\ 0, & \text{否则.} \end{cases}$$

$$TP(i,k) = \begin{cases} 1, & \text{如果平台 } P_k \text{ 分配到任务 } T_i, \\ 0, & \text{否则.} \end{cases}$$

$$DD(m,n) = \begin{cases} 1, & \text{如果从 } DM_m \text{ 到 } DM_n \text{ 存在链接,} \\ 0, & \text{否则.} \end{cases}$$

1.2 组织与使命的适应性测度

组织与使命环境的匹配或适应是对组织完成使命的性能度量，是组织最佳运作的关键^[7]。经过大量的实验验证^[8]，组织匹配主要的测度参数包括：工作负载、交流和个体间的依赖关系，而 C2 组织结构的 4 种关系就确定了决策个体的工作负载、交流和依赖关系。

本文考虑组织完成使命的两个主要的测度参数：决策者的内部工作负载和外部协作负载^[3]。 DM_m 的内部工作负载 $I(m)$ 是分配到该决策者的平台的累积负载， DM_m 的外部协作负载 $E(m)$ 是该决策者与其他决策者的协作的和， DM_m 的总工作负载是它的内部工作负载和外部协作负载的加权和^[3]：

$$W(m) = W^I \cdot I(m) + W^E \cdot E(m) \quad (1)$$

其中 W^I 和 W^E 是权重。

定义组织 O 执行使命 M 的适应性测度为各决策者总工作负载的均方根(RMS)：

$$W_{RMS}(O, M) = \sqrt{\frac{1}{M} \sum_{m=1}^M W^2(m)} \quad (2)$$

该度量同时最小化团队中各成员负载的均值和方差^[4]。

2 基于粒度计算的组织设计方法

近年来，Zadeh^[9] 讨论的粒度计算理论已经成为计算科学、逻辑、人工智能、哲学和其他领域的研究热点之一。所谓粒是指一些元素的集合，这些元素具有不可分辨性、相似性或函数和功能的内聚性等等^[9]。粒度计算是模糊信息粒度理论、粗糙集理论和区间计算理论的超集，是粒度数学的子集，它像一把大伞覆盖了所有有关粒度的理论、方法论、技术和工具的研究。在国内，张钹院士和张铃教授在研究问题求解时，独立地提出了商空间理论^[10]和模糊商空间理论^[11]，也属于粒度计算的范畴。虽然粒度计算在理论与应用方法上还存在着许多尚待解决的问题，但已经在许多领域取得了很好的应用^[12]。粒度计算的目的是在问题论域的新的层次上尽量找到计算复杂度最小的足够满意的可行近似解。因此，可以认为粒度计算是降低计算复杂度的有效工具之一。

面向任务的组织设计问题由于涉及到大量的平台和任务，是典型的复杂问题求解。通过引入粒度计算，可以克服这种计算复杂性。我们在文[5]中提出了基于粒度计算的组织设计方法，设计过程包括 3 个阶段，将组织设计问题分成了 3 个不同的优化子问题。

阶段 I (任务集和平台集粒化)：在这个阶段，分别将平台集和任务集粒化成 M 个信息粒，通过平台粒和任务粒的对应关系决定平台粒到任务粒的配置，以满足每一个任务粒的资源需求。这个阶段决定了每个决策者拥有哪些平台和需要执行哪些任务。该阶段的目标是最小化组织与使命的适应性测度。

阶段 II (粒内规划)：这个阶段决定了平台到任务的具体配置，目标是最小化使命完成的总时间。需要考虑任务序列限制、同步延迟、任务资源需求、资源能力和地理位置等约束。由于在第一个阶段已经指定了平台粒到任务粒的分配，这一阶段可以将具有大量平台和任务的规划问题分解为一些独立的具有少量平台和任务的子规划问题。

阶段 III (决策分层)：这一阶段通过指定通讯结构和决策分层来优化决策者的责任分配和决策者间的控制协作，同时平衡所有决策者的控制工作负载。这个阶段决定了决策者之间的决策交流矩阵。

以上 3 个阶段的具体问题描述及求解方法可参见文[5]。

3 组织的适应性设计

对于初始的使命环境 M ，通过前面提出的组织设计方法可以得到与之匹配的最优组织 O 。当使命环境变化为 M^* 时，根据同样的方法可以得到与之匹配的最优组织 O^* 。组织适应性设计的目的是，找到介于 O 和 O^* 之间的组织 O' ，使得组织从 O 重构为 O' 时付出的适应性代价最小。这里需要解决两个问题，一个是组织适应性的代价的计算；另一个是最优组织 O' 的求解方法。本节余下部分就这两个问题分别进行讨论(这里我们假设组织可利用的平台资源和决策者不变，对于更一般的情况可同样方法讨论)。

3.1 组织适应性的代价

组织适应性的代价主要由两个方面决定：一个是组织实体的重新配置即组织重构的代价；另一个是组织适应性变化后造成的组织性能代价。

3.1.1 组织重构代价的计算

使命环境变化时的组织重构是指平台-决策者分配和决策实体间的决策层次链接的重构^[4]。对于现代组织来说，通常决策实体间的决策层次链接的重构代价可以忽略不计(例如通信链接变化的代价很小或没有)，因此可以假设组织重构的代价由平台-决策者分配的变化决定。

假设组织 O 和 O' 的平台-决策者分配矩阵分别是 PD 和 PD' ，并不妨假设每个决策者都是不同的，这样不同的平台-决策者分配矩阵就决定了不同的组织(这种假设不影响问题的求解)。组织从 O 变化到 O' 时，决策者 m 移出和移入的平台数量为 $\sum_{k=1}^K |PD(k,m) - PD'(k,m)|$ ，假设移出和移入平台的花费是相同的并记为 W_P ，那么组织从 O 到 O' 的重构代价为

$$C^R(O, O') = \frac{W_P}{2} \sum_{m=1}^M \sum_{k=1}^K |PD(k,m) - PD'(k,m)| \quad (3)$$

3.1.2 组织性能代价的计算

组织适应性变化后的性能代价通过组织与使命的适应性测度来度量。组织 O' 执行使命 M^* 的性能代价定义如下：

$$C^P(O', M^*) = W_{RMS}(O', M^*) - W_{RMS}(O^*, M^*) \quad (4)$$

由于 O' 是与 M^* 相匹配的最优组织，因此 $C^P(O', M^*) \geq 0$ 。

3.1.3 组织适应性代价的计算

令 W^R 和 W^P 分别表示组织重构代价和组织性能代价的

权重,则当使命环境变为 M^* 时,组织从 O 到 O' 的适应性代价可定义为:

$$C^A(O, O', M^*) = W^R \cdot C^R(O, O') + W^P \cdot C^P(O', M^*) \quad (5)$$

3.2 基于粒度计算的组织适应性设计方法

组织适应性设计的目标是,找到介于 O 和 O^* 之间的组织 O' ,使得组织从 O 重构为 O' 时付出的代价最小。本节通过扩展我们提出的基于粒度计算的组织设计方法^[5]来实现这一目标。只需将阶段 I 的目标调整为最小化组织的适应性代价即可,相应地需要调整阶段 I 的平台集粒化方法。阶段 II 和阶段 III 的方法没有变化,具体可参见文^[5]。

3.2.1 阶段 I: 平台集粒化

在基于粒度计算的组织设计方法中,我们采用了遗传算法求解阶段 I 的平台集粒化问题^[5]。这里我们仍然采用这一方法,变化之处在于适应度函数的构造。

(1) 编码

采用直接基于划分结果的整数编码。如果将 K 个平台划分为 M 个子集(对应于 M 个决策者),则一条染色体就是一个长度为 K 的由 1 到 M 的整数组成的串。例如, $K=4, M=2$, 则染色体 $(1, 2, 2, 1)$ 表示平台 P_1 和 P_4 被划分到集合 1, 平台 P_2 和 P_3 被划分到集合 2。

(2) 生成初始种群

初始种群在取值域上尽量均匀地分布,种群的数量通常取 50~200。

(3) 适应度函数的构造

如果某个染色体是不可行的(例如 M 个子集中存在空集),则这个染色体的适应度设为最小值 0。否则,假设一个染色体对应了一个平台集粒化划分 $\{G_i^p\}_{i \in \tau}$ (此时对应的组织用 O 表示), $P = \bigcup_{i \in \tau} G_i^p$, $\tau = \{1, 2, \dots, M\}$, 并假设将平台粒 G_i^p 分配给决策者 DM_i 。

当前染色体的适应度值 f 定义为

$$f = 1/C^A(O, O', M^*) \quad (6)$$

组织适应性代价 $C^A(O, O', M^*)$ 越小,相应染色体的适应度越高,被选中的概率也就越大;如果 $C^A(O, O', M^*) = 0$, 那么当前染色体对应了最优解,此时当前组织 O 不需要变化。

计算 $C^A(O, O', M^*)$ 时需要的 $C^R(O, O')$ 通过式(3)得到; $W_{RMS}(O^*, M^*)$ 由基于粒度计算的组织设计方法^[5]得到; $W_{RMS}(OM^*)$ 通过下面的方法计算得到:

假设使命 M^* 的任务集为 T^* , 令 $T_0 \subset T^*$ 表示由那些不能由某一个平台粒单独完成的任务所组成的集合。由 1.3 节可知,当前组织 O 执行使命 M^* 的适应性测度 $W_{RMS}(O, M^*)$ 由 T_0 中的任务分配决定。我们采用贪婪策略进行 T_0 的分配。所谓贪婪策略是指逐步构造最优解的策略,在每个阶段都做出一个看上去最优的决策。具体步骤如下:

初始参数: $I(i) = |G_i^p|$, $E(i) = 0, i = 1, 2, \dots, M$, 其中 $||$ 表示集合的基数。初始集合: T_0 ;

第一步 任取任务 $T_j \in T_0$, 选取 T_j 到 $\{DM_i\}$ 的最优分配,使得当前的 $W_{RMS}(O, M^*)$ 最小;

第二步 根据第一步的分配结果,调整相应的 $E(i)$ 值;

第三步 $T_0 \leftarrow T_0 / \{T_j\}$, 如果 $T_0 \neq \emptyset$ 转第一步, 否则算法终止。

(4) 选择、交叉和变异

选择(或称为复制)的目的是把当前群体中适应度较高的染色体按照某种规则遗传到下一代群体中。本文采用轮盘法

选择优良个体,同时采用最优保存策略,即群体中适应度最大的染色体不参与交叉、变异操作,用来替换掉下一代群体中经过交叉、变异操作后所产生的适应度最低的染色体。

交叉运算是遗传算法中产生新染色体的主要操作过程,以某一概率相互交换两个染色体之间的部分基因。本文采用两点交叉的方法,交叉概率 pc 的取值范围是 0.5~1.0。

变异运算是染色体的某一个或几个基因座上的基因按某一较小的概率(pm 通常取 0.0001~0.1)进行改变,也是产生新染色体的一种方法。本文采用基本位变异的方法来产生变异运算,首先确定各个染色体的基因变异位置,然后依照某一概率将变异点随机选择一个与原来不同的划分结果作为新的基因值。

(5) 算法终止条件

采用规定最大迭代次数(通常取 300~500),当遗传算法的操作次数达到最大迭代次数时,即终止运算,输出结果。

4 案例分析

本节以一次联合作战的登陆战役为例说明我们的方法,其使命是登陆抢占机场和港口,为后续部队向纵深推进扫清障碍。初始使命的任务图如图 1 所示。假设组织有 5 个可利用的决策者和 20 个平台资源。初始使命的任务参数和平台参数分别见表 1 和表 2。关于案例的详细描述以及任务和平台的描述可参见文^[4]。

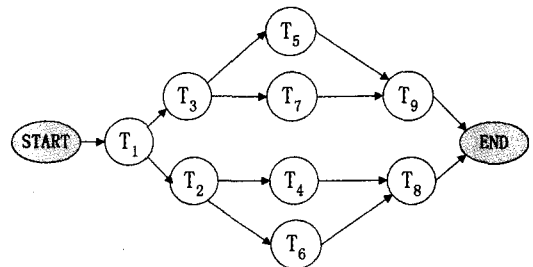


图 1 初始使命 M 的任务图

由于初始使命参数一般是事先估计的,一旦使命展开,这些参数会发生变化。本案例中我们假设使命环境参数的变化范围设计如下^[4]:

- 1) 任务资源需求向量误差: $e_R = [2, 1, 1, 4, 2, 2, 4, 0]$;
- 2) 任务位置误差: $e_L = [5, 10]$;
- 3) 任务处理时间误差: $e_t = 5$;
- 4) 最多新增 10 个新任务(参数见表 3), 这些新任务可以出现在任务图中的任何位置。

表 1 初始使命 M 的任务参数

任务	资源需求								位置	处理时间
	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈		
T ₁	0	0	0	10	14	12	0	0	[24, 60]	10
T ₂	0	0	0	10	14	12	0	0	[28, 73]	10
T ₃	0	0	0	10	14	12	0	0	[28, 83]	10
T ₄	0	0	0	0	0	5	0	0	[28, 73]	10
T ₅	0	0	0	0	0	5	0	0	[28, 83]	10
T ₆	0	0	0	0	0	10	5	0	[25, 45]	10
T ₇	0	0	0	0	0	10	5	0	[5, 95]	10
T ₈	0	0	0	20	10	4	0	0	[25, 45]	10
T ₉	0	0	0	20	10	4	0	0	[5, 95]	10

表 2 平台参数

平台	资源需求								速度
	r ₁	r ₂	r ₃	r ₄	r ₅	t ₆	r ₇	r ₈	
P ₁	10	10	1	0	9	5	0	0	2
P ₂	1	4	10	0	4	3	0	0	2
P ₃	10	10	1	0	9	5	0	0	2
P ₄	0	0	0	2	0	0	5	0	4
P ₅	1	0	0	10	2	2	1	0	1.35
P ₆	5	0	0	0	0	0	0	0	4
P ₇	3	4	0	0	6	10	1	0	4
P ₈	1	3	0	0	10	8	1	0	4
P ₉	1	3	0	0	10	8	1	0	4
P ₁₀	1	3	0	0	10	8	1	0	4
P ₁₁	6	1	0	0	1	1	0	0	4.5
P ₁₂	6	1	0	0	1	1	0	0	4.5
P ₁₃	6	1	0	0	1	1	0	0	4.5
P ₁₄	0	0	0	0	0	0	10	0	2
P ₁₅	0	0	0	0	0	0	0	6	5
P ₁₆	0	0	0	0	0	0	0	6	7
P ₁₇	0	0	0	6	6	0	1	0	2.5
P ₁₈	1	0	0	10	2	2	1	0	1.35
P ₁₉	1	0	0	10	2	2	1	0	1.35
P ₂₀	1	0	0	10	2	2	1	0	1.35

表 3 随机任务参数

随机任务	资源需求								位置	处理时间
	R ₁	R ₂	R ₃	R ₄	R ₅	R ₆	R ₇	R ₈		
min	0	2	2	0	0	0	0	0	[5,15]	7
max	5	7	7	0	0	0	1	10	[30,60]	14

4.1 组织适应性的必要性

由初始使命参数和上面设定的 4 种参数变化范围,随机生成一个使命 M₁,它可以视为是初始使命 M 的一种可能的具体实现。使命 M₁ 的任务图如图 2 所示,任务参数见文[4]。应用我们的基于粒度计算的组织设计方法^[5](假设内部

和外部工作负载权重相等,即 $W^I = W^E = 1$)设计与使命 M₁ 相匹配的最优组织为 O₁,O₁ 执行使命 M₁ 的适应性测度为 $W_{RMS}(O_1, M_1) = 4.05$ 。

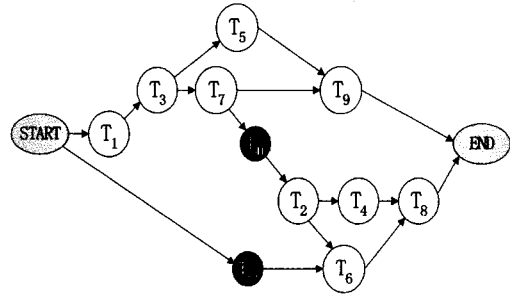


图 2 使命 M₁ 的任务图

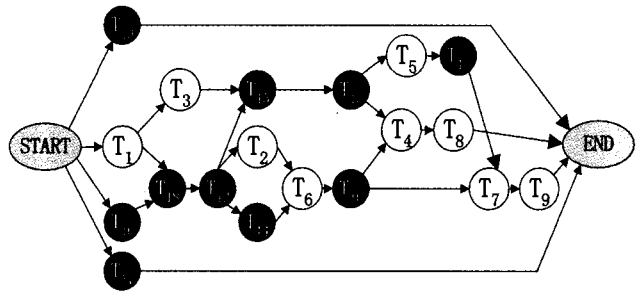


图 3 使命 M₂ 的任务图

再随机生成一个使命 M₂,可以视为是初始使命 M 的另一种可能的场景实现。使命 M₂ 的任务图如图 3 所示,任务参数同样见文[4]。如果维持原有的组织 O₁ 不变,那么 O₁ 执行使命 M₂ 的适应性测度为 $W_{RMS}(O_1, M_2) = 7.46$;如果重新设计与使命 M₂ 相匹配的最优组织记为 O₂,那么 O₂ 执行使命 M₂ 的适应性测度为 $W_{RMS}(O_2, M_2) = 4.43$ 。O₁ 和 O₂ 的平台隶属关系以及执行使命 M₂ 的任务协作见图 4。

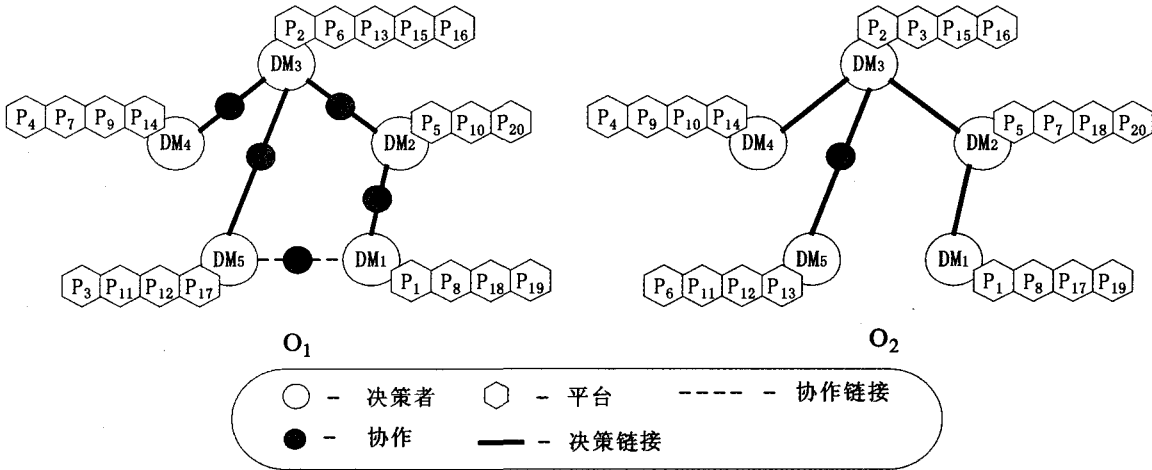


图 4 O₁ 和 O₂ 的平台隶属关系以及执行使命 M₂ 的任务协作

可以看出,当使命展开时,由于使命的真实场景会与计划中的有一定变化,根据原有计划设计的组织执行真实使命场景时,可能会导致组织执行性能的严重降低。此时组织需要进行适应性变化,从而以最小的代价来执行当前使命。

4.2 组织适应性的性能

已知原有的组织 O₁ 和当前的使命 M₂,应用我们提出的

组织适应性设计方法,得到适应性代价最小的组织为 \tilde{O} (各参数设为 $W^P = 1, W^R = 1, W^I = 3$)。 \tilde{O} 的平台隶属关系以及执行使命 M₂ 的任务协作见图 5。 \tilde{O} 执行使命 M₂ 的适应性测度为 $W_{RMS}(\tilde{O}, M_2) = 4.52$,O₁ 到 \tilde{O} 的适应性代价为 $C^A(O_1, \tilde{O}, M_2) = 6.27$ 。可以看出,当使命环境变化时,组织适应性设计方法能够找到具有最小代价的组织来执行当前的使命。

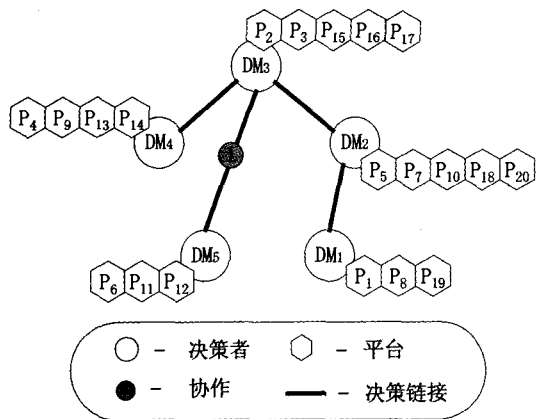


图5 \bar{O} 的平台隶属关系以及执行使命 M_2 的任务协作

如果原有的组织为 O_2 , 当使命环境变为 M_1 时, 应用我们的方法得到适应性代价最小的组织仍为 O_2 , O_2 执行使命 M_1 的适应性测度为 $W_{rms}(O_2, M_1) = 4.43$, 组织 O_2 不变的适应性代价为 $C^A(O_2, O_2, M_1) = 1.14$ 。可以看出, 组织 O_2 相比于 O_1 具有更好的适应性, 它对于环境的变化敏感度较低, 因此也可以认为是一个鲁棒的组织。如果一个组织是鲁棒的, 就可以避免因环境的不断变化带来的结构上的频繁的适应性调整, 并减少由此带来的代价。

结论 本文讨论了面向使命的组织适应性设计问题, 通过扩展已有的基于粒度计算的组织设计方法, 提出了组织的适应性设计方法。设计过程主要是对阶段 I 的平台集粒化方法进行了相应的改变。通过案例分析说明了组织适应性的必要性和本文方法的良好性能。

在以后的工作中, 我们将对组织的鲁棒性设计进行研究, 它是解决不确定环境下的组织设计问题的另一种方法。一个设计良好的、鲁棒的组织可以很好地执行一定范围内的使命, 而不是仅对某一特定使命有效。

参考文献

1 Pete A, Kleinman D L, Pattipati K R. Structural congruence of

tasks and organizations. In: Proceedings of the 1994 Symp on Command and Control Research and Decision Aids, NPS, Monterey, CA, 1994. 168~175

2 Levchuk G M, Levchuk Y N, Luo Jie, et al. Normative Design of Organizations - Part I: Mission Planning. IEEE Transactions on Systems, Man, and Cybernetics, 2002, 32(3): 346~359

3 Levchuk G M, Levchuk Y N, Luo Jie, et al. Normative Design of Organizations - Part II: Organizational Structure. IEEE Transactions on SMC, 2002, 32(3): 360~375

4 Levchuk G M, Levchuk Y N, Meirina C, et al. Normative Design of Project-Based Organizations - Part III: Modeling Congruent, Robust, and Adaptive Organizations. IEEE Transactions on SMC, 2004, 34(3): 337~350

5 Xiu Bao-Xin, Zhang Wei-Ming, Liu Zhong, et al. A Novel Organizational Design Methodology Based on the Theory of Information Granulation. In: The Fourth International Conference on Machine Learning and Cybernetics, GuangZhou, August 2005. 1~6

6 Pete A, Kleinman D L, Pattipati K R. Structural congruence of tasks and organizations. In: Proceedings of the 1994 Symp on Command and Control Research and Decision Aids, NPS, Monterey, CA, 1994. 168~175

7 Diedrich F, Entin E, Hutchins S, et al. When Do Organizations Need to Change - Part I: Coping with Organizational Incongruence [A]. In: International Command and Control Research and Technology Symposium [C], Washington, DC, June 2003

8 Kleinman D L, Levchuk G M, Hutchins S G, et al. Scenario Design for the Empirical Testing of Organizational Congruence [A]. In: International Command and Control Research and Technology Symposium [C], Washington, DC, June 2003. 324

9 Zadeh L A. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic. Fuzzy Sets and Systems, 1997(19): 111~127

10 张钹, 张铃. 问题求解理论及应用. 清华大学出版社, 1990

11 张钹, 张铃. 模糊商空间理论(模糊粒度计算方法). 软件学报, 2003, 14(4): 770

12 李道国, 苗夺谦, 张东星, 等. 粒度计算研究综述. 计算机科学, 2005, 32(9): 1~12

13 Carley K M, Svoboda D M. Modeling Organizational Adaptation as a Simulated Annealing Process. Sociological Methods and Research, 1996, 25(1): 138~168

14 Carley K M. Organizational Adaptation. Annals of Operations Research, 1998, 75: 25~47

15 Levis A H. Quantitative Models of Organizational Information Structures. In: Sage A P, ed. Concise Encyclopedia of Information Processing in Systems and Organizations, Oxford: Pergamon Books Ltd, 1988

16 Levis A H. A Colored Petri net Model of Intelligent Nodes. In: Gentina J C, Tzafestas S G, eds. Robotics and Flexible Manufacturing Systems, The Netherlands: Elsevier Science Publishers, 1992

17 Kemple W G, Drake J, Kleinman D L, et al. Experimental Evaluation of Alternative and Adaptive Architectures in Command and Control. In: Proceedings of the 1997 Command and Control Research and Technology Symposium, Washington, DC, June 1997

(上接第 161 页)

过深入的考察才能够将其提炼出来, 主要用来规定本体中其他的公理。公理一般使用一阶逻辑表达式表示, 在此基础上建立本体的公理库。公理库的建立是储存和管理公理的, 公理基本上是在进行推理的基础。

UML 是一种通用的模型描述语言, 缺少精确的语义, 因此在对本体进行描述的时候, UML 在分析和推理方面表现出的能力有限, 针对 UML 在推理方面的不足, 一些研究人员也提出了一些解决办法, 如选择通过采用基于图形变换的推理方法, 但是很多方法也只能使这个问题得到部分的解决。

第五步: 生成类的实例。根据上面建立的类的等级体系和公理库, 生成类的实例。

4.3 应急系统本体模型的作用

应急系统 Ontology 模型在应急领域中有 3 方面的作用:

- 1) 解决各子系统间的术语通信。能让不同领域的信息进行语义层的交换, 即实现语义互操作。
- 2) 实现各应用的集成。即实现同一事件的跨区跨警种的处理, 即实现应急联动。
- 3) 进行多源信息查询。包括基于时空的事件查询检索, 或事件细

节查询, 或进行事件间关系的查询。

结束语 本文对本体的定义和本体的理论方法做了深入研究分析, 阐述了使用 UML 为本体建模的可行性, 提出使用 RUP 支持的 UML 对应急系统中本体建模的指导性原则。该应急本体建模的混合方法, 突破了传统的建立本体方法的局限。采用应急系统中的本体建模的方法, 在实际项目中证明了它的可行性。

参考文献

1 Noy F N, McGuinness D L. Ontology Development 101: A Guide to Creating Your First Ontology. [Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880]. March 2001

2 Guarino N. Formal Ontology and Information Systems. In: Proceedings of FOIS'98, Trento, Italy, Amsterdam, IOS Press: 5, 1998

3 高颖. 领域本体的形式分析[D]. 北京: 中国科学院, 2003. 6~7

4 Uschold M, Grüninger M. Ontologies-Principles, Methods and Applications. Knowledge Engineering Review [J], 1996, 11(2)

5 王姣. 本体驱动的信息系统开发中的本体建模研究[D]. 吉林: 吉林大学, 2005. 59~62