计算机科学2001Vol. 28№. 10

计划驱动的合作

A Kind of Cooperation Driven by Plan

李 颖 陈跃新

(国防科技大学计算机学院 长沙410073)

Abstract Cooperation is an important mechanism for MultiAgent systems to work effectively. To realize it, a distributed plan is introduced in this paper. The constrained relations between subgoals, which have AND/OR relations, can be described in a plan. After negotiating, a steady community will be formed, then the goal will be finished after distributively performing a plan. This kind of cooperation has been realized in the project MasBuilder(MultiAgent Systems Builder).

Keywords Cooperation, Driven by plan, MultiAgent System

1. 引言

在分布异构的网络环境中,多个 Agent 往往需要通过合作来综合各自有限的资源、知识、能力以合理、有效地解决较为复杂的问题。

当多个 Agent 进行合作时,将涉及到如何依据当前 Agent 团体对一个复杂的问题进行恰当的分解,并寻找到合适的合作者分担各个子目标。此外在合作中当各个 Agent 并发实现各个子目标如何保证正确的执行顺序,怎样实现各个 Agent 之间结果的交换和综合,都将是实现合作时所需解决的问题。

在 MasBuilder(MultiAgent System Builder)项目中,我们提出了一种分布式计划来驱动 Agent 之间的合作,多个 Agent 对计划进行逐层协商,形成联合承诺,从而构成一个层次性的稳定的合作团体,并通过合作团体对计划进行分布执行来最终实现目标。

本文将基于一种思想状态的方法来描述 Agent 以及 Agent 之间的合作。每个 Agent 都具有信念、愿望、意图这样三种心智状态。其中

1)信念体现 Agent 对外部世界和自身的认识,包括对自己的能力和其它 Agent 能力的认识;

2)愿望反映 Agent 希望实现的目标;

3)意图则是 Agent 承诺将实现的目标。

在合作时,每个参与合作的 Agent 将对合作进行联合承诺。同时在合作过程中,各个 Agent 应是诚信的,即严格遵守以下社会约定

约定1

if 可以对其它 Agent 的协商请求进行承诺

then 为请求形成新的意图

通过这一约定要求 Agent 不应是欺诈的,既不能可以提供能力而不予承诺,也不能没有能力而予以承诺。

约定2[7]

if 该意图已无法实现

或该意图已实现

或实现该意图的动机已消失

then 放弃该意图

这一约定意味着 Agent 将是信守承诺的。

约定3

if 不能形成某个意图或放弃某个意图

then 应通知团体中其它成员

这里的意图,是每个 Agent 为实现合作而形成的,如果某个 Agent 只是独自放弃这种意图,将意味着其它 Agent 还将继续着这一次失败的合作。

本文下面将阐述在分布式计划中应提供怎样的机制来描述一种分布式目标搜索的问题解决过程;然后描述如何通过分布式计划驱动 Agent 之间的合作;最后是结语。

2. 分布式计划

为实现多 Agent 之间的合作,本文提出了一种分布式计划。与传统计划不同,在此所提出的分布式计划是由将要实现的目标分解出的子目标构成,各个子目标之间具有 AND 关系或 OR 关系,即在计划中将反映出对问题的分解方式。并提供必要的成分来描述目标之间的依赖关系和对分解出的子目标如何分配,以及为实现这一目标所应满足的其它约束条件,通过这样一种计划将规定在未来的某个时刻由哪个 Agent 实现哪个目标,以及它们之间的交互,并通过多个 Agent 逐层制定计划来充分实现多个 Agent 能力、知识和资

源的综合。

2.1 分布式计划下的问题求解过程

借鉴 Lesser 的基本描述机制[1]，在此我们将多 Agent 为实现目标 G 而逐层制定计划的过程表示为一棵 AND/OR 结构的分布式目标搜索树，并扩充经典的图结构以表示目标间的依赖关系，如图1所示。

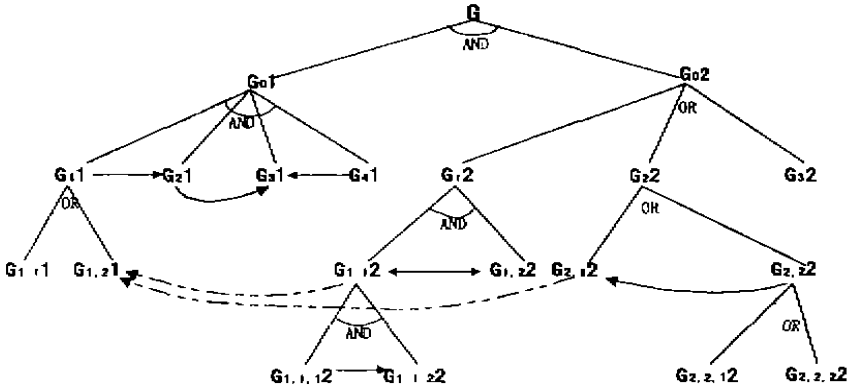

图1 分布式目标搜索树

在图1中，每一次对目标的分解即意味着一个计划的制定。通过制定计划将一个目标分解成各个子目标后，如果这些子目标为 AND 关系，意味着所有这些子目标完成，才能实现总的目标；如果子目标之间为 OR 关系，则说明只需完成其中某个子目标，即可实现目标。在实际应用时，也可根据实际领域的需要和实现策略来决定对 OR 分枝的选取，例如实现某个分枝可能所需的代价更小，或为实现冗余以更利于问题的解决，可能需同时选取多个 OR 分枝。

如果计划中的某一子目标仍然较为复杂，就需要对其进行进一步分解，即为实现这一子目标制定一个计划，这样通过逐层制定计划，而形成上述分布式目标搜索树。

目标树中的叶结点称为原子目标，即对这种目标 Agent 无需或不能再进行分解，但已能够由 Agent 通过一个动作予以实现。

各个子目标的依赖关系定义为两种：数据依赖关系和时序依赖关系，在图中分别用直线箭头和弧线箭头表示，此外子目标间还可能存在数据的交互，即有一种双向的数据依赖关系。数据依赖关系还可进一步划分为强数据依赖关系和弱数据依赖关系。

·强数据依赖关系：若该关系不能满足，则依赖目标不能成功；

·弱数据依赖关系：这种依赖关系有利于问题求解，但不必一定满足。

例如有目标 G 和依赖目标 DG，以及信息 I，DG 的实现需要 I 的输入，如果输入 I 必须由 G 产生，则 G 和 DG 之间是一种强数据依赖关系；如果输入 I 还可由其它目标提供，则 G 和 DG 之间是一种弱数据依赖关系。或者是信息 I 只是有利于 DG 的实现，但并非必需。

时序依赖关系则用于反映各个子目标在实现时序上的依赖关系。例如某几个子目标之间是否存在一种同步关系，或者在对某个公共资源的使用上是否会存在读写相关、写写相关。

虚线箭头则代表一种远程的依赖关系，即非兄弟结点之间的依赖。

虽然在图1中，只在一个层次上反映了目标 $G_{1,1}1$ 与目标 $G_{1,1}2$、目标 $G_{2,1}2$ 之间存在远程依赖关系，目标 $G_{1,1}2$ 与目标 $G_{1,2}2$ 之间存在着依赖关系，而实际上这种目标与目标之间的依赖关系应是向上传递和向下传播的，即目标 $G_{1,2}1$ 与目标 $G_{1,1}2$、目标 $G_{2,1}2$ 之间的依赖关系必然会反映到相应的父目标 $G_0 1$ 和 $G_0 2$ 上，而目标 $G_{1,1}2$ 与目标 $G_{1,2}2$ 之间的依赖关系也意味着 $G_{1,2}2$ 与 $G_{1,1,1}2$ 或 $G_{1,1,2}2$ 存在着直接的依赖关系。

通过这样逐层计划来组织 MultiAgent 系统的合作，要求实现以下几点[1]：1）定义目标图（包含对依赖关系的鉴别和分类）；2）将图的特定领域赋给合适的 Agent；3）决定图的展开区域；4）遍历图的结构；5）获取每一次成功的遍历。

在对 AND/OR 图进行遍历时，对 OR 分枝只需要择其一而行。依据图结构进行的一次成功的遍历即表示对总目标的一次实现。

获得和维护一棵与客观世界（外部环境和 Agent 团体）相符的实际目标树是多 Agent 依据分布式计划进行合作最根本的保障。

2.2 对计划的描述

每一个计划都具有计划说明和计划实现两部分，通过这两部分提供充分的机制来较好地描述上述问题求解过程。

2.2.1 计划说明 是对计划所对应的目标进行

整体说明。它主要应包含有以下几个部分：

·计划标识：用于标识一个计划。在 Agent 对计划进行协商和执行的过程中、以及运用"手段-结果"方式进行推理和决策过程中、计划标识实质上就对应所需要实现的目标。

·数据通道：在上述问题求解过程中，分解出的各个子目标之间难免存在一些数据依赖关系，所以应对此加以描述。

·执行上下文：用于说明计划启动执行时所需要满足的条件，通过这一部分既可说明目标之间的时序依赖关系，也可进一步说明数据依赖关系的类型。执行上下文是否满足可能有三种情况：不可能满足、暂未满足和已经满足。如果执行上下文不可能满足则表示该计划不可能启动执行、只能放弃；如果是暂未满足，则表示要启动该计划的执行尚需要等待；如果已经满足、则意味着可以开始执行计划。

·最大执行时间：最大执行时间用于说明计划正常执行最多所需要的时间。若计划的执行超时，则可认为该计划的执行发生了意外，应该终止，以免浪费资源，同时最大执行时间还可以用于判断执行该计划将会占用资源的时间。

·Agent 的运行个数：用于说明需要多少个实现当前计划对应的目标。当图中的一个目标分配给某个 Agent 后，将由该 Agent 依据自己的信念、愿望、意图来判断这一子目标能否实现、如何实现，从而决定能否为实现这一目标进行承诺。当某个 Agent 不能承诺实现某一目标时，是否还有其它 Agent 可供协商。

2.2.2 计划的实现 通过计划的实现部分来说明实现目标的具体方案。它包含对目标进行分解所得的 OR 和 AND 分枝，在这些分枝中也包含对每一个子目标的说明、即对应每一子目标都有相应的计划说明部分，同时还应说明各个子目标之间的 AND 和 OR 关系。为实现目标，Agent 将根据子目标的 AND 和 OR 关系选取当前子目标的一个最小项。以图2这样一个部分目标树为例：

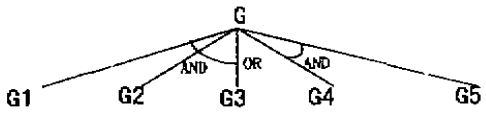

图2 计划中计划体的示意图

在这个示意图中，G 代表计划将要实现的目标，由计划说明部分进行描述，而对计划的实现，或称之为对目标的展开，则对应计划的实现部分。由上图可知、在 Agent 所制定的这个计划中：

计划实现＝{$G1 \wedge G2 \wedge G3$、$G4 \wedge G5$}

这样意味着为实现目标 G，只需其中一个合取式予以实现即可。

3. 计划驱动的合作

多 Agent 通过分布式计划进行合作的过程可分为三个阶段：个体计划过程；计划的协商过程；计划的执行过程。个体计划过程即 Agent 为实现目标（包括因其它 Agent 的协商请求而产生的目标）而制定以上所描述的计划的过程。计划协商过程则是通过对计划进行协商来形成合作团体的过程、最终通过计划的分布执行来使目标得以实现。

在多个 Agent 通过分布式计划进行合作的过程中、这三个子过程往往需要不断反馈、重复进行的。

3.1 个体计划过程

当 Agent 感知到外部输入时（包括其它 Agent 对某一目标的协商请求），将会导致 Agent 产生相应的目标。Agent 需要通过自己的信念、愿望、意图来判断能否对目标进行承诺，即能否形成新的意图。如果 Agent 能为实现这一目标制定出相应的计划、并落实到计划实现部分中的某一最小项可以予以实现，即意味着 Agent 能为实现这一目标进行承诺。

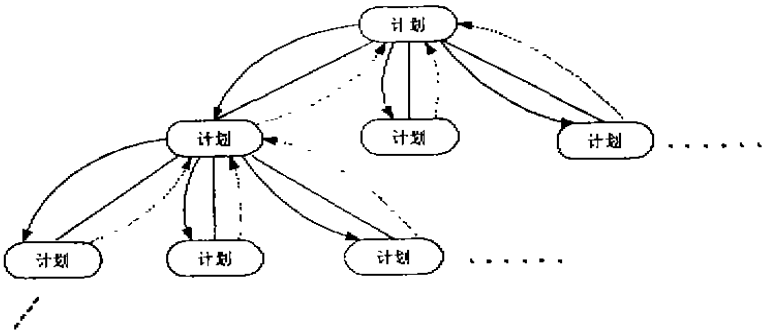

图3 计划的协商过程示意

在这个过程中，Agent 将判断该目标是否能够实现，是否需要对目标进行进一步分解，根据当前的外部环境和 Agent 团体如何对目标进行分解，在实现某一子目标时，有哪些 Agent 可供协商。此外，还需确立计划说明和计划实现部分中的其它一些约束。

3.2 计划的协商

当所选取的最小项中的子目标涉及其它 Agent 时，就需要通过计划协商过程。协商过程是 Agent 就各个子目标与相应的 Agent 进行的一个任务委托过程，同时通过协商请求来触发其它 Agent 参与合作。通过这样逐层协商的过程，最终确立对目标树的一次成功遍历。

3.2.1 协商过程 Agent 协商一个计划的过程，就是一个根据问题的复杂程度和自己对问题的解决能力而进行的一个不断对问题进行分解和委托的过程，其表示如图3所示。

在图3中，实践弧形箭头代表协商请求，虚线弧形箭头代表协商应答。这样一个计划图，实际上就对应 MultiAgent 系统对分布式目标树的一个搜索过程。最上层的计划即对应最终要实现的目标。在图3中，每个计划都有相应的 Agent 来对应，这样通过协商就形成了一个层次结构的 Agent 团体(图4)。其中每个 Agent 相对于其下层的 Agent 是组织者，而相对于其上层的 Agent 则是执行者。组织者进行计划的制定，并依据制定出的计划发起协商。执行者一次协商所请求的对象。

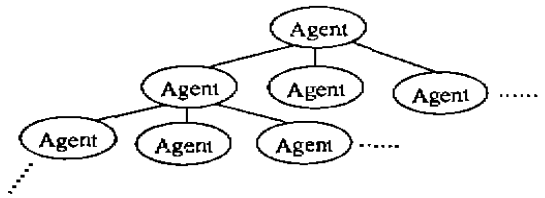

图4 Agent 团体的层次结构

当组织者协商某一子目标时，将依据子目标的计划说明部分选取规定个数的相应 Agent 进行协商，组织者将力图使当前的计划协商成功，以为合作形成相应的意图。当某个执行者不能承诺一个子目标时，组织者会为这一子目标寻找另一个执行者进行协商，而经过以上步骤，仍无法使某一子目标协商成功，组织者将会选取新的不包含失败子目标的最小项进行协商。只有经过这一系列步骤仍无法协商成功，才意味着不能为实现这一目标进行承诺。此时依照约定3，组织者将向它的组织者反馈无法承诺的应答。

3.2.2 协商算法 每一次协商，都有组织者和执行者两种角色的 Agent 参与，这二者对同一次协商有着不同的处理策略。

组织者的协商算法：

```
Process OrganizerNeg
1)根据计划说明选取某个最小项；
2)根据每一子目标 Agent 执行个数的要求，选取相应的 A-
  gent 协商每一个子目标；
3)等待协商应答；
4)if 某个 Agent 不能对协商的子目标进行承诺；
5)  then if 为实现该子目标还有新的候选者；
6)       then 选取新的候选者进行协商；
7)     else 该子目标协商失败；
8)       if 修改计划，并有新的不包含该子目标的最小项，转
         1)；
9)       else 无法对实现计划所对应的目标承诺；
10)           通知已承诺的 Agent 放弃承诺；
11)else 判断是否所有协商的 Agent 都已进行成功应答；
12)     if 所有协商的 Agent 都已成功应答，则对计划对应的
        目标进行承诺；
13)     else 转3)；
```

执行者的协商算法：

```
Process PerformerNeg
1)根据信念、愿望、意图对协商请求进行判断；
2)if 没有冲突；
3)  if 可以对协商请求的目标进行承诺；
4)    向组织者返回成功应答；
5)  else 向组织者返回失败应答；
6)向组织者返回失败应答；
```

当执行者确定是否能对协商请求的目标进行承诺时，也可能会作为组织者为自己所制定的计划发起协商。

3.2.3 协商的作用 由上述过程可知，通过协商，最终形成了一个通过多层计划来建立联系的稳定的合作团体。

未经协商之前，每个 Agent 制定计划时，都只能依据 Agent 当前对环境和团体中各个成员的认识来分解目标、分配任务，但在一个动态变化的环境中，具有能力完成某个子目标的 Agent 在当前的状态下可能已无法承担这个任务，或者该 Agent 已不在 Agent 团体中了，或者团体中又增加了几个新成员可以为实现目标提供某种能力。此外还因为每个 Agent 不可能做到全知全能，一个目标是否能够实现、当前对目标的分解方式和对任务的分配是否合理，以及是否充分发掘出计划中所应包含的约束，都必须要由处于不同环境、对问题有着不同认识侧面的 Agent 依据自己的环境、能力和知识进行判断推理，这也正是分布式计划的用意所在。只有通过协商，使得 Agent 团体得以相互交换和更新对环境和团体的认识，才能形成一棵实际的目标搜索树。

其次通过协商，使得多个 Agent 为共同实现某一目标进行承诺，并且通过信守合作时的约定，使合作建立于一个稳定的合作团体之上。

当合作团体确定后，就可将数据通道对应到具体的 Agent，当计划的执行开始时，每个 Agent 都将会拥有这一部分知识，每个执行者都可以直接与相应的 Agent 交互，从而较大程度地提高了计划执行时的并发性。

3.3 计划的执行

这样经过逐层向下协商和逐层向上承诺，使得 Agent 可以对最终的目标进行承诺时，组织者则开始启动计划的执行，同时将协商时建立好的相关数据通道传送给承担各个子目标的 Agent，此时就进入到了计划的执行阶段。

在多个 Agent 并发执行计划时，首先必须判断执行上下文中的约束是否已满足，从而保证各个子目标之间正确的实现顺序。

由于在执行开始阶段，各个合作的 Agent 都已拥有了关于数据通道的知识，这样在执行过程中，Agent 之间则可以通过建立好的通道，直接与相应的 Agent 进行交互，而无需经过上层的组织 Agent，

通过保证各个子目标正确的并发秩序，并通过建立好的数据通道进行数据交互，这样逐层向上进行结果的综合，最终自然地实现目标。

3.3.1 执行算法

```
Process Performing
1)判断执行上下文是否满足；
2)if 暂未满足
3) then 转1)；
4)else if 不可能满足；
5)  通知所有已承诺的 Agent 放弃承诺；
6)  向组织者返回执行失败应答；
7) else 满足；
8)  if 某个子目标为原子目标；
9)  使用相应的能力予以实现；
10)  else 通知所有执行者实现所承诺的子目标
11)等待执行应答；
12)if 某个 Agent 执行成功；
13) if 所有的 Agent 都执行成功，
14)  向组织者返回执行成功应答；
15) else 转11)；
16)else 通知其它 Agent 放弃承诺；
17) 向组织者返回执行失败应答；
```

参 考 文 献

1 Lesser V R. A retrospective view of FA/C distributed problem solving. IEEE Trans on Systems, Man, and Cyberntics, Special Issue on Distrbuted Artificial Intelligence, 1991, 21(6)

2 Cohen P R, Levesque H J. Intention is Choice with Commitment. Artificial Intelligence, 42

3 Nick Jennings. Cooperation in Industrial Multi-Agent Systems. World Scientific Series in Computer Science, 43

4 Doran J E, Franklin S, Jennings N R. On cooperation in Multi-Agent systems The Knowledge Engineering Review, 1997, 12(3)

5 Woodridge M, Jennings N R. Towards a theory of cooperative problem solving. In: Proc of Modeling Autonomous Agent in a Multi-Agent World(MAAMAW-94). Odense, Denmark, 1994 101

6 焦文品，史忠植. 多主体间的协作过程研究. 计算机研究与发展，2000，37(8)

（上接第64页）

高图像的处理效率，为此选用如下的模糊分布求 p_{ij}：

$$p_{ij}=1+\frac{x_{ij}+x_{max}}{(x_{max}-x_{min})/2} \quad (6)$$

其中：x_{min}为图像的最小灰度值。

三、结果讨论

应用模糊理论对图像进行预处理是图像的一个新的应用方面，它将图像像素抽象为某种模糊隶属分布，再对这种分布进行有关分析，从而完成图像的处理工作。

(a) (b) (c)

(a)原始图像 (b)pal 法模糊增强后图像 (c)改进 pal 法模糊增强后图像

图1 图像模糊增强示意图

本例中 Pal 等根据模糊理论提出图像增强处理的模糊算法，通过选择幂指形式的模糊隶属函数以叠代方式实现相关的数字处理，达到了预期的效果。该种算法的缺点是模糊隶属函数的范围为：$\beta \leqslant p_{ij} \leqslant 1$，且对于 $p_{ij} \leqslant \beta$ 时反变换无解，导致这些部分受到切削，因此它削弱了图像的处理效果，此外它的处理速度不是太理想。

改进的图像增强处理的 Pal 算法使用线性的模糊隶属函数，减少了图像处理的难度，保持了图像处理的效果，提高了图像处理的速度，因此改进的算法比原算法更有效，从处理的速度上来看它比原算法提高数十倍。图1为基于模糊算法的图像增强示意图。

参 考 文 献

1 郭桂蓉，庄钊文 信息处理中的模糊技术. 长沙：国防科技大学出版社，1996

2 Choi Y S, Kurshnapuram R. A robust approach to image enhancement based on fuzzy logic. IEEE Trans Image Processing, 1997, 6(6)

3 Chatzis V, Pitas I. Nonlinear location and scale estimators of fuzzy numbers. IEEE Trans Signal Processing, 1998, 46(1)

4 Pal S K, King R A On edge detection of X-ray images using fuzzy sets. IEEE Trans. Pattern Analysis and Machine Intelligence, 1983, 5(1): 69～77

5 Pal S K, King R A Image enhancement using smoothing with fuzzy sets. IEEE Trans Syst., Man, Cybern., 1981, 11(7): 494～501