

# 一种基于规则推理的 BDI 模型实现<sup>\*</sup>

唐文彬 朱淼良

(浙江大学人工智能研究所 杭州310027)

## An Implementation of BDI Model Based on Rule Reasoning

TANG Wen-Bin ZHU Miao-Liang

(Institute of Artificial Intelligence, Zhejiang University, Hangzhou 310027)

**Abstract** The common idea is to consider agents whose mental state comprises the three attitudes of belief, desire and intention. The relationships among these entities have received considerable attention. There still remains a large gap between theory and practice. In this paper, the authors implement a BDI agent system based on rule reasoning and show the procedure of planning. The experiment shows the system can handle application problem.

**Keywords** Agent, BDI, Mental states, Rule reasoning

## 1 引言

对于 Agent 来说,控制它的行为和内部状态的是它的思维状态,因此研究 Agent 的思维状态模型对当前 Agent 的理论和实践都有极其重要的影响,其中应用最为广泛的当推 BDI 模型。BDI 模型源于 Bratman 哲学分析的“意图中心论”,从作为思维状态研究方法论基础的“意识立场”出发,试图借鉴人类思维属性的概念来解释复杂系统的行为。一般我们把信念、愿望和意图当作基本的思维属性(BDI)。在现有的 BDI 的研究中,较多的是侧重于 Agent 模型的研究, Cohen 和 Levesque<sup>[1]</sup>基于正规模式逻辑的可能世界模型,系统地讨论了理性平衡,行为意图的形式表达和演化规则。他们的工作是基于线性逻辑和可能世界模型的,定义了包括信念,目标和意图等基本的概念及它们之间的相互关系。Rao 和 Georgeff<sup>[2,3]</sup>采用计算树理论逻辑,可能世界的时间结构由线性扩充为分支,进一步地阐述了 BDI 的概念和相互关系。Konolige 和 Pollack 提出了认知结构和意图关系图的概念,由此而定义的意图不仅包含对目标世界的期待,也包含了对不希望世界的状态的描述,在一定程度上解决了副作用的问题,但是这些研究都是仅从理论上给予讨论,缺乏可计算的模型,难以应用到实际中的 Agent 设计、实现和运行。Shoham 从系统实现的角度讨论了 Agent 的结构和行为特性,提出了面向 Agent 的程序设计,为 BDI 理论运用到实际的工程中作出了突出的贡献,许多面向 Agent 的程序设计都以此为基础。BDI 模型被提出以后,虽然还存在例如意图副作用等问题,但是仍被用来对一些实际的系统进行描述和验证,并取得一定的成功,应该说是一种比较好的思维状态描述模型。但是现在的很多相关的研究过于看中对人类理性的探究,以及对所描述系统的逻辑行为进行推理,而偏废了对 Agent 思维状态模型面向实际的应用的研究。本文中提出了一种基于规则推理系统的 BDI 模型的实现,并对 BDI 模型的各个主要的心理要素的实现进行了具体的描述。

## 2 规则推理系统

规则推理系统采用基于规则的知识库系统来实现精确的推理,该过程重复执行直到得到新的当前状态。由于传统的符号推理系统难以实时地实现,我们设计了基于 OPS83 系统的 Rete 网结构和知识编码相结合的快速推理机 FRete,推理速度大大提高,通常的推理输出可以在毫秒级完成。规则推理系统不同于传统的黑板系统,虽然规则推理系统中有中心黑板作为全局共享内存的布告板,但是没有传统黑板系统中的知识源,用规则系统执行推理,因而更是一个产生式系统。推理过程基于规则实现心智转化模型,该模型很容易用规则系统表示,且推理速度可以用知识编码技术提高,对于有大量推理运算的模型的实时推理是有巨大的好处。规则推理系统主要由知识库(规则库),事实库和推理机三部分组成。

推理机采用改进的 Rete 快速匹配算法,运用知识编码技术,极大地提高了推理速度,每一个推理平均保持在百分之一秒内,此外还扩充了获取事实的方法,在推理机中设置了中断功能,当新的事实产生引发中断后,如果推理机处于空闲状态,则立即启动进入“识别-动作”环的运行,因此新的事实能得到及时的响应,外部环境也能影响推理机的工作过程,使它处于可干预状态,从而满足实时开放系统的要求。

外部事件是 Agent 得到的外部消息,是 Agent 的心智状态转化的激发动力,也是转化规则的输入。

规则库和事实库的编写完全符合 OPS83 语法,规则库可以有用户自定义节,其中包括用户自定义的函数。每条规则由规则左部和规则右部组成,规则左部是规则的事实部分,而规则右部是该规则应该执行的动作。

由于规则库和推理机相互独立,给 Agent 的行为选择带来了极大的灵活性,对路径规划策略的修改和完善只需修改或增加知识库中的规则,不必对推理机做任何改动。

事实类定义如下:

```
<事实类> ::= class <事实类名>
{
  ((<域类型><域名>);)
}
```

<sup>\*</sup> 本课题得到国防科工委“十五”攻关项目基金“智能机器人”资助。唐文彬 博士研究生,主要研究方向为分布式智能系统、智能机器人。朱淼良 教授,博士生导师,主要研究方向为人工智能、计算机视觉、智能机器人。

规则描述如下:

```

<规则> ::= rule( )
{
  <规则左部>
  →
  <规则右部>
}

```

对于用规则不易表示的计算等,我们还可以利用自定义的函数来完成,比如对当前状态和目标状态之间距离的计算。

### 3 BDI 的思维状态及其表示

在 Agent 使用的 BDI 模型,一般都是在 Cohen 和 Levesque 的正规模态逻辑的意图模型以及 Rao 和 Georgeff 的 BDI 计算树逻辑模型基础上发展起来的。其核心最重要的就是信念(Belief)、愿望(Desire)、意图(Intention)等几个思维属性以及它们的演化,其本质上要解决的问题是如何确定 Agent 的目标和 Agent 如何去实现这个目标。

信念(Belief)是 Agent 对世界的认知,包含描述环境特性的数据和描述自身功能的数据,是 Agent 进行思维活动的基础,属于思维状态的认知方面。从 Belief 出发可以做出对环境的期待和判断以及自身状态的了解,提出达到一定目标的行为规划。

记  $Bel_i^s(\phi)$  表示 Agent<sub>i</sub> 在状态 S 的条件下存在信念  $\phi$ , 我们称  $Bel_i^s(\phi)$  为原子信念, 则  $Bel_i^{s_1}(\phi) \wedge Bel_i^{s_2}(\phi)$  和  $\neg Bel_i^s(\phi)$  为信念。比如机器人车辆 Agent 在时刻 t, 处在位置(x, y), 记为 Pos(x, y, t), 用规则推理系统的事实类可以表示为:

```

Class Pos
{
  int x, y;
  int t;
}

```

愿望(Desire)是 Agent 希望达到的状态,由此可以激发系统的规划和行动。一般来说,可以表达为 Agent 对环境状态的一种期待和判断,也就是通过判断该状态是否成立作为 Desire 是否实现的标志。Agent 可以拥有互不相容的愿望,而且也不需要相信它的愿望是绝对可以实现的,作为 Agent 的愿望,实际上就是车辆控制者愿望的体现。Agent 可以把某个 Desire 作为目标,加入到目标集中,在产生了一个目标后,Agent 会保持它,不会轻易放弃,直到该目标已经实现或确信该目标无法实现或者是由于超时继续该目标已经没有什么意义为止。

记  $Desire_i^s(\phi)$  表示 Agent<sub>i</sub> 期望在状态 S 的条件下存在  $\phi$ , 比如机器人车辆 Agent 期望在时刻 t, 处在位置(x, y), 记为 Desire(x, y, t), 用规则推理系统的事实类可以表示为:

```

Class Desire
{
  int x, y;
  int t;
}

```

这是一个特殊类,在推理的过程中,每一个 Desire(x, y, t) 都被作为一个结点(node), 每个节点,被链表的形式连接起来,可以进行搜索、添加和删除等操作。

意图(Intention)是承诺实现的愿望中选取的当前最迫切完成或最适合完成的一个,是当前 Agent 将要或正在实现的目标,它是属于思维状态的意向方面,其作用是直接指导 Agent 的行动,是 Agent 对自己未来时间行动的预先安排。意图可以被一步步细化为子意图,由此而得到实现该意图的规划。当前意图对 Agent 的当前动作具有指导性的作用,指出了大方向上 Agent 的行进路线。当前意图的实现是靠子意图

的实现,子意图的实现是靠原子动作的完成,如果由于某种原因原子动作不能完成,则会直接导致当前意图的失败,从而导致 Agent 放弃这个当前意图,但是,并不是说 Agent 要放弃这个目标,相反 Agent 将由愿望激发产生新的意图来完成这个目标。意图对目标有依赖性,如果目标不存在了,那么,Agent 也应该取消相应的意图和由它所引起的规划。意图是特殊的愿望,如果把愿望集做为一个堆栈,那么推理机可以把每次把栈顶弹出的愿望作为当前意图的一个选择。

事件(Event)是 Agent 获得环境的一个变化,将触发对自身的状态和信念进行修正,对目标进行增减的信息。比如 Agent 发现道路在时刻 t, 位置(x, y)处出现障碍,记为 Obs(x, y, t), 用规则推理系统的事件类可以表示为:

```
fact(Obs x=a, y=b, t=c)
```

通过对事实的激发(factmake), 能够把事件送入推理系统,触发思维状态的转化。同样的是通过 fact(Pos x=a, y=b, t=c)的激发可以把车辆当前的位置为(a, b, c)这个事实加入到信念集。

规划(Plan)是对当前意图的具体实现,由一个个原子动作组成动作序列,但是需要注意的是,这里的原子动作也是全局性的,是在全局环境中的,与局部环境中的规划是不同的。同样,规划对意图具有依赖性,一旦意图失败或不存在,就应该取消相应的所有的规划动作。记  $Act_i^s(\phi)$  表示 Agent<sub>i</sub> 在状态 S 的条件下执行动作  $\phi$ , 比如机器人车辆 Agent 在位置(x, y)的情况下,行进到(x', y'), 记为 Act(x, y, x', y'), 可以用规则推理系统的规则类可以表示原子动作,一条规则可以对应一个原子动作。

```

rule Move
{
  &1(Pos. x=x, Pos. y=y);
  &2(Obs. x<>x');
  &3(Obs. y<>y');
  →
  remove &1, &2, &3;
  make(Pos. x=x', Pos. y=y');
}

```

如果记 Desire(x, y, t) 为当前意图, Bel(a), ..., Bel(n), 为当前的相关信念, 由当前意图激发的动作序列为 rule(a), ..., rule(m), 则 Desire(x, y): Bel(a), ..., Bel(n) + rule(a), ..., rule(m) → Bel(b), ..., Bel(t) 称为对意图 Desire(x, y, t) 的一个实现规划, 就是 Agent 在某个状态下, 由于某个意图的激发, 做出某个规划, 这个规划是由多个动作序列组成的, 通过动作序列的完成, 到达另外的状态, 从而是对意图的一个实现。

### 4 Agent 运行流程

BDI 逻辑的计算树模型如图1, 由初始状态到目标状态的过程是一个规划的过程, 表示的是一个信念可达关系。每个世界的的时间结构都是一个时间树, Agent 拥有一个线性的历史和分支的未来, 分支代表的是 Agent 在相应时间的动作选择, 选择不同的动作将导致不同的发展去向, 刻画了世界发展的可能轨迹。因此, 如果把整个世界看作一个离散状态空间, 也就是说世界可能出现的状态的类型是有限的, 同时把 Agent 可能有的动作看作一个动作空间, 那么在每个时刻, 对于采取的相关动作, 世界可能出现的状态我们都能够预测。

基于规则推理机的巨大计算能力, 我们可以通过对 Agent 在每个时刻状态下的相关动作进行推理, 搜索全部的时间树, 一定可以找到一条正确的规划路径, 在每一步的动作

选择过程中,原子动作的选择不是随机的,而是按照一定的判别规则的.其根本的规则就是目标接近原则,也就是通过动作的执行可以使未来状态与目标状态的距离小于当前状态与目标状态的距离.记  $B_i = (Bel(i,0), \dots, Bel(i,n))$  是 Agent 当前的状态,  $B_o = (Bel(o,0), \dots, Bel(o,n))$  是 Agent 的目标状态,则两个状态间的距离可以表示成为  $B_i \times B_o$ , 可以用距离判别函数  $B: B_i \times B_o \rightarrow R$  来表示.如图1中, Agent 在  $t_2$  时刻,搜索动作空间计算动作 a, b, c 带来的未来状态与目标状态的距离,最后选择动作 b.同理, Agent 最终得出规划的轨迹(用黑粗线表示).在 Agent 的运行过程中,修正心智元素的信息来源就是事件,在每一步的运行过程中对新事件的激发推理不可少,因为由于对新事件的推理导致整个规划的改变,甚至意图的改变都是可能的. Agent 运行流程如下:

1. 初始化
2. 激发事件
3. 更新状态空间和目标以及意图
4. 搜索全部状态空间和动作空间,制定完成意图的规划
5. 执行原子动作
6. 判别目标是否完成
7. 到2

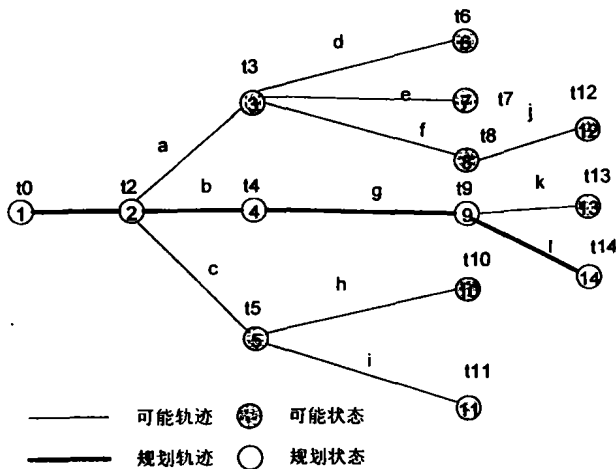


图1 计算时间树示意图

## 5 实验举例

为了说明基于规则推理的规划过程,我们建立如图2的实验环境,图中每一个方格在环境中代表一个 Agent 车辆能处的位置状态.

(1,1)	(2,1)	(3,1)	(4,1)
(1,2)	(2,2)	(3,2)	(4,2)
(1,3)	(2,3)	(3,3)	(4,3)
(1,4)	(2,4)	(3,4)	(4,4)

图2 网格环境示意图

在机器人车辆 Agent 中,状态定义为事实类 class Pos,表示了 Agent 在时刻 t 的位置.可以定义原子动作为

```
rule MoveDown
{
    &1(Pos. x=x, Pos. y=y);
    &2(Obs. x<>x);
    &3(Obs. y<>y+1);
    →
    remove &1, &2, &3;
}
```

```
make(Pos. x=x, Pos. y=y+1);
}
```

同理可以定义动作 rule MoveUP, rule MoveLeft, rule MoveRight

定义事件激活为:

fact(Obs x=a, y=b, t=c) 使用 factmake 激活,可以有 (a,b) 点障碍的信念被加入;

fact(Pos x=a, y=b, t=c) 使用 factmake 激活,可以有 Agent 到达 (a,b) 点的信念被加入.

距离函数,我们简单地用  $|x_1-x_2| + |y_1-y_2|$  来计算,用自定义函数 interval() 计算

推理过程如下:

```
main()
{
    kb kb1();
    kb1.rete-proc();
    kb1.fact-proc();
    kb1.run(); //静态初始化
    ....
    while(1)
    {
        kb1.factmake(); //事件激发
        kb1.run(); //动态推理的
    }
}
```

假设 Agent 的初始状态是  $Pos(1,1,t_0)$ , 目的状态是  $Desire(4,3,t)$ , 障碍出现在  $(3,3,tm)$ . 那么经过规则推理得出的规划为  $Desire(4,3,t): Pos(1,1,t_0) + MoveLeft, MoveLeft, MoveLeft, MoveDown, MoveDown \rightarrow Pos(4,3,t)$ .

小结 在本文中我们提出了一种基于规则推理实现的 BDI 模型实现,按照时间树搜索动作空间和状态空间,得出达到目标的规划.由于基于规则的推理机高速的推理能力,在动作空间和状态空间不是很大的时候,该模型还是能够得到比较好的效果,是 BDI 模型在实用上的一次有意义的尝试.下一步,我们要研究的是在状态空间和动作空间都比较复杂的时候,如何提高其实时规划的能力.

## 参考文献

- 1 Cohen P R, Levesque H J. Intention is choice with commitment. Artificial Intelligence, 1990, 42(2~3): 213~261
- 2 Rao A S, Georgeff M P. Modeling rational agents within a BDI-architecture. In: Fikes R, Sandewall E, eds. Proc. of the Principles of Knowledge Representation and Reasoning. San Mateo, CA: Morgan Kaufmann Publishers, 1991. 473~484
- 3 Rao A S, Georgeff M P. An abstract architecture for rational agents. In: Nebel B, Rich C, Swartout W, eds. Proc. of the Principles of Knowledge Representation and Reasoning. San Mateo, CA: Morgan Kaufmann Publishers, 1992. 439~449
- 4 Wooldridge M J, Jennings N R. Intelligent agents: Theory and practice. Knowledge Engineering, 1995, 10(2): 115~152
- 5 Haddadi A, Sundermeyer K. Belief-Desire-Intention agent architectures. In: O'Hare G M P, Jennings N R, eds. Foundations of distributed artificial intelligence. New York: John Wiley & Sons, Inc., 1996. 169~185
- 6 Haddadi A. Communication and Cooperation in Agent Systems: A Pragmatic Theory. Berlin: Springer-Verlag, 1996
- 7 Werner E. Cooperating agents: a unified theory of communication and social structure. In: Gasser L, Huhns M N, eds. Distributed Artificial Intelligence, Vol I. Los Altos, CA/Pitman: Morgan Kaufmann Publishers, 1989. 3~36
- 8 Singh M P. Multiagent Systems: A Theoretical Framework for Intentions, Know-how, and Communication (LANI Volume 799). Berlin: Springer-Verlag, 1994
- 9 Gao Peng. Study on Model and Language for Multiagent Systems: [Ph D dissertation]. (in Chinese). Zhejiang University, Hangzhou, 1998
- 10 朱淼良, 张新晖, 吴春明, 钱徽. 自主机器人自组织结构 IRASO 的仿真研究. 计算机研究与进展, 1999, 36(7): 776~782