

动态不确定环境下的决策:一种分层决策模型^{*}

杨 洋 陈小平

(中国科学技术大学计算机系多智能体系统实验室 合肥230027)

摘 要 本文提出一种智能体分层决策结构模型,试图通过分层决策技术有效地解决动态、不确定环境中的智能体的实时决策问题。本模型的高层采用 BDI 结构,以便为较长期任务的规划和推理提供充分的支持;模型的底层采用反应式结构,以保证对短期实时任务的及时响应。实验结果表明了这种分层模型在某些复杂任务领域中的有效性。

关键词 智能主体,动态不确定环境,信念-愿望-意图(BDI),RoboCup

Making Decision under Real-Time, Unpredicted Environments: a Multi-Layer Model

YANG Yang CHEN Xiao-Ping

(Multi-Agent System Lab. CS Dept. of Univ. of Sci. & Tech. of China, Hefei 230027)

Abstract This paper describes our work in developing an agent architecture based on a multi-layer model which will help to solve the problem of real time decision-making in dynamic, unpredicted environments. The high levels of this architecture use BDI model to support long-term planning and reasoning while use reactive model in the lower level which ensures the system to react timely in real-time environments. Experiments show that such a multi-layer architecture is effective in some complicated real-time domains.

Keywords Intelligent agent, Dynamic unpredicted environment, Belief-desire-intention (BDI), RoboCup

1 引言

随着信息自动化科技发展,各种硬、软自主机器人的应用领域正在不断扩大,从而对机器人的能力提出了越来越高的要求。一方面,自主机器人的应用环境正由传统的静态、确定环境向实时动态、不确定的环境转化;另一方面,随着任务复杂程度的提高,单个机器人往往难以独立完成任务,需要多个机器人协调合作共同完成任务。因此,面向动态不确定环境的多主体系统的研究已经引起了国内外学术界的高度重视。

智能体设计者所面临的一个主要问题是如何使智能体在动态、不确定的环境中对系统环境的发展变化及时地做出合理的决策。对于多主体系统而言,如何使智能体与同伴进行协调合作从而有效地完成任务,也是设计者所应重点关心的问题。在这样的应用领域中,传统的搜索和规划方法往往需要比较长的时间,很难满足实时性的需求;并且实时环境中不断发生的突发事件容易造成系统不断的重新规划,导致整个决策系统产生抖动。因此需要设计一种新的决策模型来满足实践的需要。

本文提出一种分层决策的智能体决策体系结构模型,试图通过分层决策技术有效地解决上述问题,较好地满足动态不确定环境中的应用要求。在该模型结构中,我们把智能体的决策模型分为两大层次:一个层次处理那些对实时性要求不高,但却对智能体的行为具有方向性指导作用的慎思式推理和规划任务。在这个层次中,我们主要采用一个层次处理实时性要求很强,但所需处理的问题相对比较局部、简单的实时性任务。本文将主要围绕在 RoboCup 仿真机器人足球比赛^[4, 5]这个试验平台上如何利用 BDI 模型构建智能体的慎思决策模型来介绍我们的工作。

本文第2节介绍模型的整体体系结构;第3和第4节给出模型中两大层次的具体实现;第5节给出在 RoboCup 仿真机器人足球比赛试验平台上的实验结果;最后是总结和未来工作展望。

2 智能体体系结构模型

长期以来,大量的研究工作都致力于构建于可用于真实机器人的规划推理系统。早期的 SHAKY 机器人的研究工作表明^[6],现实世界中大量的动态不确定性或不可预测因素对于机器人设计者提出了重大的技术挑战。传统的知识表示和推理模型在提供了良好的推理规划机制和目标驱动的行为模式的同时,由于未考虑计算资源限制,存在实时性较差的缺陷,而且也不够灵活。和传统的规划推理系统不同,另一种称为 Subsumption 模型的体系结构则完全抛弃了知识表示和推理规划^[7, 8]。该模型通过简单、低级动作的组合来实现复杂的行为模式。与传统的推理规划系统相比,尽管 Subsumption 模型缺乏对于中长期目标的处理能力,但却展现了良好的在实时环境下的快速反应能力。

最近的研究工作表明^[9],在动态、不确定环境中,尤其是存在多主体的情况下,智能体需要具备中长期规划的能力以作出尽可能合理的决策;但同时又存在很多任务,要求智能体能够及时迅速地作出反应而并不需要太复杂的思考。对于这样的领域,将上述两种模型的优点结合起来,构建出分层次的混和式模型,是一个值得探讨的方向。

为了将两种模型的优点结合起来,我们采用分层式的决策技术。其基本原理是根据决策粒度将决策过程化分为若干层次,每个层次在不同抽象粒度上处理决策问题。层与层之间有机结合的关键是“愿望”(Desires)。在我们的模型中,愿望定

^{*})本文得到国家自然科学基金(60275024)和863计划(2001AA4222)支持。

义为智能体所期待达到的状态,也就是智能体的任务。愿望既可以是外部的环境状态,也可以是系统内部的状态。比如说智能体可能会期望把球踢到某个位置,也可能希望了解自己现在还没有获取到的一些信息。需要指出的是,在同一时刻,智能体可能同时拥有多个不同的愿望,它也没有必要一定要完成每个愿望。它可以根据环境的实际情况以及给定的策略进行取舍,选择其中的一个或多个最紧迫的 Desire 作为自己的目标。

在层次决策的模型中,每个层次从更高级的层次获取可能的任务,产生决策后送往下层。我们称这种愿望为外部愿望。与此同时,每个决策层次也拥有自己的环境监视器,根据所获取的信息对环境产生的变化作出及时的反应,产生相应的愿望,称为内部愿望。越高的层次决策周期越长,实时性越低。通过层次化的结构,系统可以在不同的抽象层次上处理实时性要求不同的事件,从而既可以对环境变化作出及时的反应,又可以使非预期事件造成系统重新规划的影响得到有效的控制。除此之外,层次化的规划推理还提供了更加清晰的功能规范和更高的开发效率。具体模型见图1。

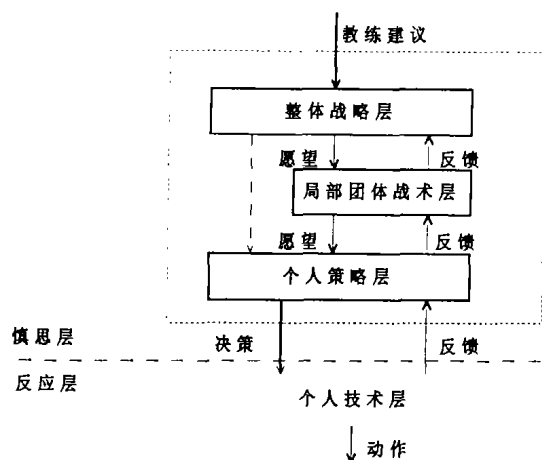


图1 智能体的体系结构

从图中我们可以看出,在我们的实现模型当中,我们根据直观把决策划分为4个层次。其中上3层采用了慎思式结构的BDI决策模型,自顶向下分别为整体战略层、局部团队战术层和个人策略层。其中整体战略层从球队整体的层面上对球队每次进攻或防守所应采用的战术策略进行规划;局部团队战术层处理局部团体(可以是临时产生的)的目标形成、合作策略等相关问题;个人策略层则负责处理仅和个人行为相关的目标和策略。整体战略层和局部团队战术层都涉及到社会因素,因此在这两层中我们可以看到其他 agent 的表示和相关推导——它们的目标、信念等等。在这里需要指出的是整体战略层和局部团队战术层并不意味着全局的中央控制主体的存在。事实上,团队中的每一个智能体都对整个球队或是某个局部团体所要采取的策略有着自己的看法。尽管有时它们之间会由于观察的局部性或是信息的不完整性而造成策略上的冲突,但这种冲突可以通过智能体相互之间的通讯交流和观察来加以协调解决,从而达成共同目标。

在采用了慎思式结构的3个层次之间,愿望是它们有机结合的关键。它代表了智能体所期望达到的某种状态,也就是智能体所希望完成的任务。从图上我们可以看到各个层之间通过传递愿望来达成协调一致。在我们的模型当中,每个层除了从上一层获取愿望外,它们本身也拥有自己的愿望发生器。这就意味着从上层传递下来的愿望有时候不会被采纳——如果

有更好的选择的话。这个特点使得该决策模型更加灵活多变,大大增加了在不确定环境中的健壮性。

模型的最下一层——个人技术层则采用了类似于 Subsumption 体系的结构,以处理实时性要求很强,但所需处理的问题相对比较局部、简单的实时反应式部分,如机器人球员的盘带、传球等实时性很强的技术动作。

3 个人技术层

在动态实时的环境当中,很多任务处理的问题相对比较局部,并不需要太多的推导和计算,但却对实时要求比较强,如球员的盘带、传球等实时性很强的技术动作的实现等等。因此在这个层次中我们采用了类似于 Brooks 的 Subsumption 的结构。具体结构如图2所示。

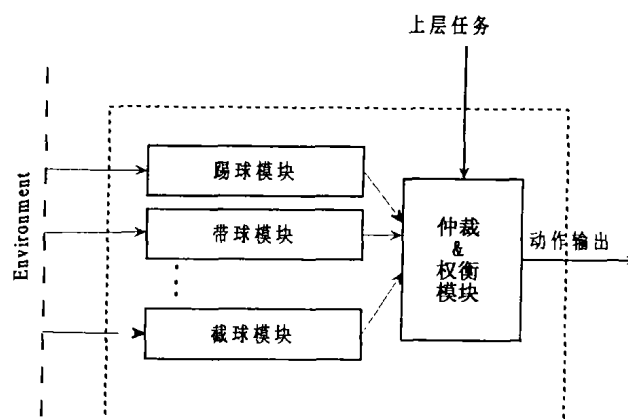


图2 反应层结构图

如图2所示,我们利用神经网络或强化学习等方式把球员常用的技术动作实现为一个可以独立运行的反应式模块,即它们的内部没有符号级推理。各个模块之间相互独立,各自根据环境情况作出相应的决策并送往仲裁模块。仲裁模块将根据当前环境的状态和上层传入的任务作出在竞争的动作之中选出其中最合适的作为动作输出。通过这种模式,系统可以对许多实时要求较高但相对比较简单任务作出及时的反应。

4 基于 BDI 模型的规划推理

4.1 基于 BDI 的决策模型

在我们的实现中,上3层采用了相同的结构。各层之间的区别仅仅在于它们工作在不同的抽象层次上。在每一个推理规划层次的内部,我们采用了 BDI 模型结构。BDI 的主要思想是依靠一组精神状态来描述智能体的内部状态和控制结构,包括信念(Belief)、愿望(Desire)和意图(Intention)。其中信念为智能体对所处环境的认识,即智能体的知识;愿望为智能体所希望达到的状态,也就是智能体所希望完成的任务;而意图则是基于信念的维系 Desire 和规划之间的特殊精神状态,它实质上是作出了一定承诺的 Desire,也就是智能体所承诺要完成的任务。在动态、不确定的环境中,理性主体的行为是受制于意图的。智能主体不会无理地随意改动自己的意图,也不会坚持不合时宜的意图,这些都取决于智能体所采取的承诺策略。

图3中所示的是其中某一个决策层次的具体实现结构。由于在类似于 RoboCup 的应用领域之中,智能体不但在计算资源上受到限制,而且在环境感知上也受到很大的限制,也就是

说智能体既不是全能的,也不是无所不知的,因此对于智能体来说,要制定过于详细的计划不但难以实现,而且通常没有太大的意义。所以在我们的推理模型中采用的是 Means-End 推理。这就意味着智能体在制定计划时常先决定应该达到哪

一个目标,至于如何达到这个目标则留到将来再继续决定。例如,当机器人球员要传球时,他可以先决定把球传给哪一个队友。而如何传球,是直接传球还是先盘带一会则可以留到将来合适的时机再进一步确定。

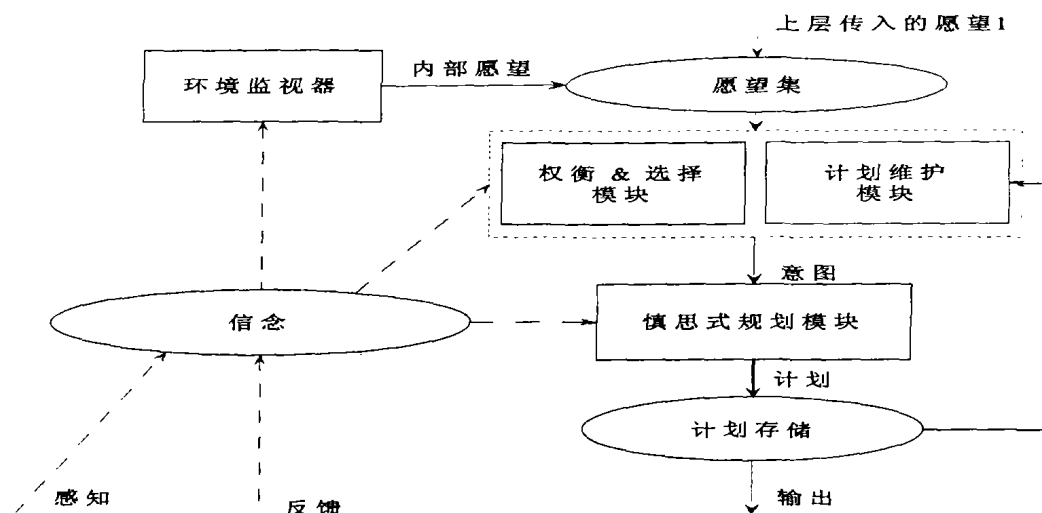


图3 基于 BDI 的决策模型结构图

因此在任意运行时刻,每一层将得到上层传入的外部愿望,以及由于环境监视器所产生的对潜在的机会或威胁作出反应而产生的内部愿望。除此之外,本层还将产生别的内部愿望,其来源是当前正在执行的部分计划(Partial Plan)。所有愿望都将存放在愿望集中。当愿望产生后,有两个主要问题需要解决,一个是如何权衡这些意图并选择其中一个作为智能体的意图;另一个则可归结为规划问题,即如何形成合理计划以达到最终目的。在这里,第一个问题是由两个模块——权衡选择模块和计划维护模块解决的。其中计划维护模块将检测这些愿望是否和当前正在执行的计划相容,否则将触发权衡选择模块在它们之中进行选择,当存在更好的选择时则中止当前计划。最后被选中的愿望将作为智能体的意图传递给推理模块,推理模块将根据意图制定合理的计划或进一步完善当前的计划并将其存在计划存储模块中。

在上面的结构中,意图的存在保证了智能体的决策总是围绕着特定的目标进行,这使得推理系统在推理过程中不至于盲目扩展子目标,从而有助于系统在有限的计算资源下更快地进行决策。而且 Means-End 的推理方式也使得在某些条件发生变化的时候,只需要修改计划中受到影响的部分即可,而不必完全推翻整个计划。这就在一定程度上保证了动态的环境中决策过程的持续性,而不会像经典的规划那样由于环境的不断变化而不断地重新规划。与此同时,计划维护模块的存在则保证了在环境产生了重大变化而导致计划不可继续进行时,系统也能够作出及时的反应。

4.2 Means-End 推理机

上面提到,智能体的规划问题是通过 Means-End 的推理机制实现的。为此,我们设计了一个 Means-End 推理机。其核心是一个反向推理的产生式系统。

1. 知识表示:采用符号化的表示, $S=S_1 \times S_2 \times \dots \times S_n$,其中 S_i 为系统中环境变量。

2. 规则的表示:每一条规则都包括两个主要部分,触发条件 C 和动作算子 α 。

Rule: if C then α

触发条件 C 可以是系统的某个状态,也可以是要达到的

目标。

$C; S \mid \text{Goal}$

动作算子 α 定义了规则的执行过程。它由一系列的步骤组成。步骤不仅可以是原子动作,也可以是子目标。

$\alpha: \langle \text{Step1}, \text{Step2}, \dots \rangle$

Step: Action | SubGoal

3. 推理机的基本工作过程。

推理机制采用反向推理,其原理是从目标出发,根据规则库找到一条路径到达起始状态。其工作流程如下:

- (1) 初始化
- (2) 更新状态空间及目标意图 I
- (3) 搜索规则库,制定完成意图 I 的计划
- (4) 存储当前计划并输出原子动作

5 实验分析

为了验证结构的有效性,我们在仿真机器人足球^[4,5]这个平台上进行了实验分析。

系统运行情况:同时运行11名仿真球员

测试环境: Athlon 1.2GHz, 768MB RAM, Mandrake 9.0 Linux

实验1:测试系统中长期规划以及动态执行能力。

该试验将通过让智能体完成特定目标来测试系统的规划及动态执行能力。实验设定让一名机器人球员 A 从中场开球,他的目标固定设置为把将球射入对方球门。同时设定2名拦截对手,他们的初始位置随机放置在球员 A 正前方15米外。当球被拦截对手截住或是完成目标时实验结束。我们通过观测球员 A 生成的计划过程以及执行情况来评测系统的中长期规划以及动态执行能力。实验反复进行100次。

从实验中我们可以观测到机器人球员的计划及其执行过程。在绝大多数情况下,系统能够产生出正确的计划以支持其完成目标,并且能够在一定程度上对执行期间所遇到非预期的情况进行计划调整(产生出的计划体列表略)。

实验中我们通过球被拦截时,球距离球门中点的距离来作为对球员 A 完成目标情况的评估。我们可以看到,在绝大

多数情况下,球员 A 可以根据计划将球带到球门25米以内的位置去(可以射门的距离)。具体数据如表1所示。我们可以看到系统具备相当的推理规划能力。

实验2:测试系统在实时环境下的执行效率情况。

该实验通过采样和统计在完整比赛中智能体决策时平均消耗的时间来评估系统在实时环境下的执行效率。实验结果如表2所示。

我们可以看到系统在70%的情况下能够在40ms 以下作出决策,97.5%的情况下能够在100ms 以下。这保证了系统能够在实时环境下有效地工作(RoboCup 仿真平台 Soccer Server 设定的决策周期为100ms)。

实验3:测试系统在动态不确定环境中的整体性能。

我们从去年参加 RoboCup 2002 仿真机器人足球比赛的球队中挑选了一些不同风格球队作为我们的对手,以此来评估该模型在动态、不确定环境中的表现。表3列出了该部分的实验结果。

由实验数据可以看出,该系统模型能够较好地适应像 RoboCup 这样的动态实时领域。

表1

最终距球门距离	目标	0-15m	15-25m	25-35m	35m 以上
达成次数	5	36	28	19	12

表2

平均时间(ms)	0-10	10-40	40-70	70-90	100以上
占决策的百分比	23.7%	46.8%	17.2%	9.8%	2.5%

表3

对手	实验场次	每均平均进球数	每场平均被进球数
Robolog2002	20	6.8	0.0
UvA trilearn 2002	20	1.2	0.8
YowAI2002	20	7.9	0.0
HelliRespina 2002	20	2.5	0.1

(上接第119页)

法,关于这种方法的测试数据自动生成工具的研究具有很重要的意义。

参考文献

- Heller E. Using Design of Experiment Structures to Generate Test Cases. In: Proc. 12th Int'l Conf. Testing Computer Software, ACM, New York, 1995. 33~41
- Mandl R. Orthogonal Latin Squares: An Application of Experimental Design to Compiler Testing. Communications of the ACM, 1985, 28(10): 1054~1058
- Brownlie R, Prowse J, Phadke M. Robust Testing of AT&T PMX/StarMail using OATS. AT&T Technical Journal, 1992, 71(3): 41~47
- Phadke M S. Quality Engineering Using Robust Design. Prentice Hall, Englewood Cliffs, NJ, 1989
- Taguchi G. System of Experimental Design, Quality Resources, 1987. Translation of Jikken keikakuho, Maurzen Co., Tokyo, 1976
- Cohen D M, et al. The AETG System: An Approach to Testing Based on Combinatorial Design. IEEE Trans. on Software Engineering, 1997, 23(7): 437~444
- Cohen D M, et al. The Combinatorial Design Approach to Auto-

总结以及工作展望 实验表明,本文所提出的混和式的智能体结构模型在一定程度上能够在实时反应和规划推理之间取得较好的折衷,因而可以满足动态不确定环境下的智能体决策要求。然而当前我们的工作主要还是集中于单个智能体的决策推理,在未来的工作中,应该进一步完善在这个结构上的多智能体主体的合作问题。

参考文献

- Bratman M E. Intentions, Plans, and Practical Reasons. Harvard University Press, 1987
- Rao, Georgeff M. BDI Agents from Theory to Practice: [Technical Note 56]. AAIL, April 1995
- Rao A S, Georgeff M P. Modeling rational agents within a BDI-architecture. In: Proc. of the Second Intl. Conf. on Principles of Knowledge Representation and Reasoning (KR'91), Morgan Kaufmann, 1991. 473~484
- Kitano H, et al. oboCup: The Robot World Cup Initiative. In: Proc. of the IJCAI-95 Workshop on Entertainment and AI/ALife, 1995
- Kitano H, et al. The Robocup synthetic agent challenge 97. In: Proc. of the Fifteenth Intl. Joint Conf. on Artificial Intelligence, San Francisco, CA, Morgan Kaufman, 1997
- Nilsson N J. Shakey the robot: [Technical Report 223]. SRI International, 1984
- Wooldridge M, Jennings N R. Intelligent agents: Theory and practice. Knowledge Engineering Review, 1995, 10(2)
- Brooks R A. Intelligence Without Reason. In: Proc. IJCAI-91', Sydney, Australia, 1991
- Stuart R. Rationality and intelligence. In: Proc. of the 14th Intl. Joint Conf. on Artificial Intelligence, 1995. 950~957
- Chen Xiaoping. On Deliberative Adaptability. In: Proc. of 2002' National Conf. on Theoretical Computer Science. Also in: Computer Science, 29(9): 49~51

matic Test Generation. In: IEEE Software Sep. 1996. 83~87

- Lei Y, Tai K C. In-Parameter-Oder: A Test Generation Strategy for Pairwise Testing: [Technical Report TR-2001-03]. Dept. of Computer Science, North Carolina State Univ, Raleigh, North Carolina, Mar. 2001
- Tai K C, Lei Y. A Test Generation Strategy for Pairwise Testing. IEEE Trans. on Software Engineering, 2002, 28(1): 109~111
- Kobayashi N, Tsuchiya T, Kikuno T. A New Method for Constructing Pair-wise Covering Designs for Software Testing. Information Processing Letters, 2002, 81(2): 85~91
- Cohen D M, et al. New Techniques for Designing Qualitatively Independent Systems. J. Combin. Designs, 1998, 6(6): 411~416
- Williams A W, Probert R L. A Practical Strategy for Testing Pair-wise coverage of Network Interfaces. In: Proc. 7th Internat. Symp. Software Reliability Engineer. Nov. 1997. 246~254
- Cohen D M, et al. The Automatic Efficient Tests Generator. In: Fifth Int'l Symposium on Software Reliability Engineering, IEEE, 1994. 303 ~ 309
- Xu Baowen, Nie Changhai, Shi Qunfeng, Lu Hong. An Algorithm for Automatically Generating Black-Box Test Cases. Journal of Electronics, 2003, 20(1): 74~77
- 聂长海, 徐宝文. 基于接口参数的黑箱测试用例自动生成算法. 计算机学报, 2004, 27(3): 382~388