

一种 BDI Agent 的多线程实现方法^{*})

刘芳 姚莉 张维明 王长纓

(国防科技大学信息系统与管理学院 长沙 410073)

摘要 Agent BDI 结构的实现是多 Agent 系统开发中的难点之一。本文研究了理性 BDI 模型,提出并实现了一种 AGENTFRAME 结构,其中明确表示了 Agent 的信念、愿望和意图,以该结构为模板所生成的 Agent 能够长期持续地自主运行,完成推理和协作等多种智能行为。

关键词 Agent, BDI 结构,多线程

A Multi-Thread Implementation Method of BDI Agent

LIU Fang YAO Li ZHANG Wei-Ming WANG Chang-Ying

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

Abstract The implementation of BDI Structure Agent is one of the puzzles in Multi-agent system development. This paper has researched the BDI theory model, and presented an AGENTFRAME structure. In this structure, the belief, desire and intention have been expressed explicitly. And the Agents based AGENTFRAME can run persistently and accomplish many intelligent behaviors such as reasoning and cooperation.

Keywords Agent, BDI structure, Multi-thread

1 概述

近几年来,主体(Agent)的研究主要集中在 Agent 的应用与性能,而忽略了其实现及内部特性。要想建立能够在一定环境中长期稳定运行,并能自主、智能地完成用户的设计目标的 Agent 和多 Agent 系统,必须研究 Agent 的程序实现^[1]。Agent 的信念-愿望-意图(belief-desire-intention, BDI)^[2]结构被认为是一种灵活有效、能够很好地适应动态环境变化的 Agent 结构,它能直观、清晰地表示 Agent“决定做什么,以及决定怎样做的过程”,已经开始从形式化理论研究逐渐转入实现研究^[3]。同时,Agent 开发平台的焦点问题之一也在于为开发者提供一个 Agent 清晰可靠的 BDI 框架,使得开发者能够方便地生成、调试和监控 Agent 和多 Agent 系统。

文[4]给出了 BDI Agent 的 UML 图,进而研究了 Agent 的行为和活动管理,但是没有进一步给出多任务处理的实现方案。JADE^[5]是现在较为流行的多 Agent 系统开发平台,它提供了 Agent 运行的容器、Agent 通信的基本设施和 Agent 基类等,用户可以方便添加以 FSM 组装的 Agent 行为。但是,利用该平台生成的 Agent 并不完全遵循 BDI 结构,该平台所实现的 Agent 基类中采用非优先循环策略来进行行为调度,没有体现 Agent 利用 BDI 来对自身和协作 Agent 的行为进行推理的过程;同时,Agent 任务完成的情况、Agent 状态都隐含在行为的执行参数中,需要开发者在行为的实例变量和与它相关的 Agent 中必须保存所有的计算状态。这样,希望能够开发具有清晰 BDI 结构的 Agent 的用户来说,JADE 就存在着缺陷。

本文提供了一种 BDI Agent 的多线程实现框架 AGENTFRAME 结构。AGENTFRAME 结构已经嵌入多

Agent 系统开发平台 MBOS^[6]的内核中,平台所生成的 Agent 就是基于此模板结构自动生成的。

2 BDI 理论模型

一个 Agent 体的信念是对环境和其自身的了解。信念不同于知识,一般说来,知识是为真的信念。愿望和意图都是关于一个 Agent 希望发生的事件的状态,一般认为它们的区别在于:意图是衡量承诺的一个尺度,它将引导和控制 Agent 未来的活动,意图在实际推理过程中起着至关重要的作用;而愿望指 Agent 希望达到的目标,但可能有机会实现,也可能永远不去实现该目标。一旦 Agent 根据承诺或协作知识产生了某种意图,该意图将驱使 Agent 寻求合适的手段达到这一意图,直到这个意图结束为止,也即意图是 Agent 行为的控制器,引导着一个 Agent 未来的活动和当前行为的选择。意图最明显的性质是它将导致行为。概括起来,意图的主要作用包括:①意图驱动手段目的推理;②意图约束未来的慎思过程(或目标选择);③意图持续地控制着 Agent 的行为;④意图影响未来实际推理所基于的信念^[7]。

智能 Agent 的实际推理过程涉及到 3 个重要环节:①选择目标;②确定怎样获得这些目标;③当环境发生变化时,如何调整所追求的目标,以适应变化了的环境。过程①称作慎思,过程②称作手段-目的推理,过程③称作反应性。信念-愿望-意图 Agent 模型的一般结构就是针对上述 3 个环节而建立的。基于 Wooldridge 的 BDI 结构,本文将 BDI Agent 的一般结构形式化如下:

设 Bel 为所有可能信念的集合,Des 为所有可能愿望的集合,Int 为所有可能的意图的集合。这 3 个集合必须保持概念的一致性。

^{*})国家自然科学基金(70371008)资助项目。刘芳 博士研究生,主要研究方向为分布式人工智能、多 Agent 系统;姚莉 博士,教授,博士生导师;张维明 博士,教授,博士生导师;王长纓 博士,主要研究方向为多 Agent 学习。

在任意给定时刻,一个 BDIAgent 的状态表示为如下三元组: (B, D, I) , 这里 $B \subseteq Bel, D \subseteq Des, I \subseteq Int$ 。

Agent 的信念求精函数 brf 是一个映射:

$$brf: \rho(Bel) \times P \rightarrow \rho(Bel)$$

该函数依据当前的感知和当前的信念确定一个新的信念集合。

意见产生函数 $options$ 定义为:

$$options: \rho(Bel) \times \rho(Int) \rightarrow \rho(Des)$$

该函数起着如下作用:

①我们可以把 DIAGENT 的意见产生过程看作循环求精一个层次计划结构的过程,不断地考虑和承诺逐步实现的意图,直到最后获得目标。

②意见函数驱动手段-目的推理,同时满足几个其它的约束:所产生的意见必须与 Agent 当前的信念和当前的意图相一致;其次,应当能识别出环境变化的优势,提供 Agent 获得意图的新方法,或新的可能性。

一个 BDIAgent 慎思过程表示为 $filter$ 函数:

$$filter: \rho(Bel) \times \rho(Des) \times \rho(Int) \rightarrow \rho(Int)$$

并且, $filter$ 满足下列约束:

$$\forall B \in \rho(Bel), \forall D \in \rho(Des), \forall I \in \rho(Int), filter(B, D, I) \subseteq I \cup D$$

$filter$ 函数依据 Agent 当前的意图、当前的信念和愿望刷新 Agent 的意图。 $filter$ 函数需要丢弃无法完成、或已无意义的意图,为无法实现的意图选择新实现方法以及选择新的意图。

执行函数是返回一个相应于可执行意图的直接可执行行为:

$$execute: \rho(Int) \rightarrow A$$

Agent 的决策函数表示为: $action: P \rightarrow A$

可由下列伪代码定义:

```
function action(p: P): A
```

```
begin
```

```
B := brf(B, p)
```

```
D := options(B, D)
```

```
I := filter(B, D, I)
```

```
return execute(I)
```

```
end
```

综上所述, BDI 体系结构是一个实际的推理结构,其基本成分是表示 Agent 信念、愿望和意图的数据结构,以及表示慎思过程的功能和手段-目的推理。

3 BDI Agent 的多线程实现

基于 BDI 理论模型,我们将 BDI Agent 的设计分为两部分:一部分为独立 Agent 运行时程序的“外部”部分,一部分为 Agent 程序结构——AGENTFRAME。

我们设计了信念库、愿望集、目标集和计划库,以及 Agent 的原子操作模板,其中计划是使用有限状态自动机(FSM),将 Agent 的原子操作组装起来刻画的,并且可以嵌套。它们独立于 Agent 程序代码,是 Agent 的开发者根据实际需要设定的。MBOS 平台为开发者提供了基于开发者所建立本体的信念库模板和非基于本体的信念库模板,提供了定义计划的 FSM 的图形化工具,开发者只需要编写原子操作的具体函数。我们将信念库、愿望集、目标集、计划库和原子操作定义为概念上一致的独立对象,是为了:①能够显示地表示

Agent 的信念,使得 Agent 和多 Agent 系统的用户能够通过观察 Agent 的信念,获得 Agent 的任务完成情况;②在运行时将 Agent 的原子操作以统一的接口提供给 Agent,使得领域知识与控制知识分离,以实现更好的重用;③可嵌套的计划层次地刻画了 Agent 的能力,使得 Agent 的行为调度更加灵活。

在 AGENTFRAME 结构中,对应于理性 BDI 模型的信念模型、目标模型和计划模型的数据结构分别为信念集、目标队列和计划集。在 Agent 运行时,它将根据我们设计的控制机制来对这些静态结构进行推理、修改和控制,并根据设定的策略对信念库进行修改。其主要过程是:

- 基于信念库中的信念确定一个与愿望相吻合的任务作为目标,置入目标队列;

- 从目标队列中取出一个目标,基于计划库中的计划对此目标进行规划,规划的结果即为意图,Agent 将承诺完成此意图;如果该目标不能独立完成,需要通知其他 Agent 与之协作完成,此时规划的结果即为联合意图,愿意协作完成该任务的其他 Agent 都将承诺完成此联合意图;

- 在任务执行过程,保留未完成的计划的状态参数,便于发生错误之后回溯;

- 信念库中的信念需要随着环境的变化、接收到的其它 Agent 的消息和任务完成与否进行实时更新。

AGENTFRAME 结构中的控制机制是一个多线程的并发控制机制,它不仅能够使得理性 BDI 模型中的知识表示结构进行逻辑推理,有效地完成 Agent 的决策和协作任务,支持 Agent 的反应性、自主性和社会性这 3 项基本特性,而且能够支持 Agent 的时态延续性和多任务的并发执行,以及规划能力和学习能力。

一旦用户启动 Agent 执行,Agent 就被系统赋予一个进程空间,开始持续自主地运行,其所有的行为都由该控制机制管理,直至用户发出终止指令才结束其生命,将进程空间归还系统。在 Agent 的进程空间中有 3 个常驻线程和若干临时线程,其中的常驻线程为主控线程、消息接收线程和消息发送线程,临时线程是当前正在执行中的任务执行线程,它是在任务开始执行时被创建,在任务执行完毕时从进程空间中消失。

图 1 所描述的是 AGENTFRAME 结构及其多线程并发控制机制,其中的主要部件包括:

感知器,负责感知外部环境,接收其他 Agent 和用户发送的消息;

效应器,负责作用于外部环境,向其他 Agent 发送消息和为用户提供输出信息;

决策模块,负责对任务进行选择,诸如哪些任务拒绝完成、哪些任务需要优先完成,以及哪些任务需要串行执行、哪些任务需要并行执行等等;

规划模块,负责采用手段-目的推理方法来制定任务实施的规划方案。必要的时候需要请求学习模块执行,帮助获取最优的规划方案;

学习模块,负责在运行期间执行学习行为,获取新知识和新技能,从而提高 Agent 的工作性能。学习过程中需要对计划库和信念库进行存取,在已有知识和技能的基础上执行推理和归纳等学习行为,并保存新知识和新技能;

计划库,可以提供 Agent 能够辨认的关于计划操作符的逻辑描述,每个计划和子计划用有限状态自动机表示;

信念库,存储 Agent 所了解的关于自身和环境状态的信息,这些信息在进行逻辑推理时要用到;

调度模块,负责对任务进行调度,诸如何时开始执行何种任务、何时挂起何种任务、何时中止何种任务等等;

协作控制模块,负责控制与其它 Agent 的交互作用的时间和决定采用何种交互协议,以及依据相关的交互协议,控制与协作者的会话过程;

相识者数据库,用于描述与其它 Agent 之间的组织关系,以及该 Agent 关于其它 Agent 的功能的信念。协作控制模块在处理与其它 Agent 的协作关系时,会对之进行存取;

执行监控模块,负责维护 Agent 内部正在执行的任务,并对其进行监视。

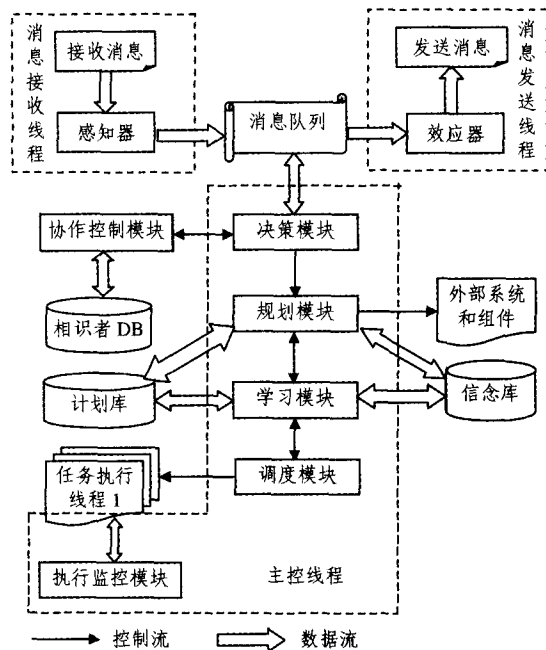


图 1 AGENTFRAME 结构及其多线程并发控制机制

设想一个一般的控制过程:感知器持续地监听网络消息,当接收到一个来自其它 Agent 的消息时,它将消息传递给消息队列。决策模块从消息队列中依次取出一条消息,将它解释成为一个实现目标的请求,并传递给规划模块,由它采用手段一目的推理方法制定出实现该目标的规划方案。规划过程中需要存取计划库和信念库,以及调用外部系统和组件,必要的时候还需要执行学习模块,以便更快地制定出最优的规划方案。如果是自身无法实现的复杂目标,则由协作控制模块通过存取相识者数据库,选择合适的相识者,将协作请求经由效应器发送给相识者。规划方案制定完毕,由调度模块创建用于实现该目标的新的任务执行线程,多个任务执行线程可以并发执行。在任务执行线程的运行过程中,由执行监控线程对其进行实时监控。

Agent 内部的数据交换可以通过共享内存和在线程间收发自定义消息的形式来实现。线程之间的数据流如图 2 所示,图中①是指消息接收线程将从网络上接收的来自其他 Agent 的协作请求传送到主控线程,由主控线程进行规划并创建新的任务执行线程;②是指主控线程将 Agent 的协作请求

传送到消息发送线程,由消息发送线程传送给其它 Agent;③是指消息接收线程将从网络上接收的来自其他 Agent 的关于某个任务执行线程的回复消息传送给该任务执行线程;④是指任务执行线程之间的数据交换;⑤是指任务执行线程将执行结果作为对于其它 Agent 的回复,交由消息发送线程传送给相应的 Agent;⑥是指主控线程对任务执行线程进行监控,将指令性信息传送给相应的任务执行线程,任务执行线程在任务执行过程中需要主动将执行情况的信息提供给主控线程,二者之间存在双向的数据流。

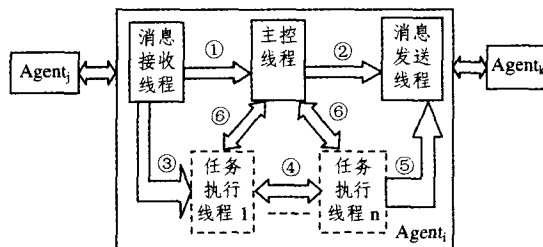


图 2 AGENTFRAME 结构的多线程的数据流

结论 基于理性 BDI 模型,本文实现了一种 AGENT-FRAME 结构,其中明确表示了 Agent 的信念、愿望和意图。以该结构为模板所生成的 Agent 能够长期持续地自主运行,完成推理和协作等多种智能行为。MBOS 已经通过了多项应用实例的测试和评估,结果表明 AGENTFRAME 结构是可行的,能够长期持续地自主运行,并有效地完成各种推理、协作等智能行为。

为了给 Agent 系统开发者提供更大的自由,我们目前正在进行试图将 Agent 的慎思过程显示刻画出来的研究,也就是将决策和规划模块的控制机制标准化,给出统一的封装接口,以使得开发出来的多 Agent 系统能适应更复杂的应用环境。

参考文献

- 1 Luck M,McBurney P,Shehory O, et al. Agent Technology Roadmap-A Roadmap for Agent Based Computing. 2005, AgentLink III
- 2 Michael W. Agent-based Software Engineering. IEEE Proc Software Engineering,1997,144(1):26~37
- 3 Zambonelli F,Omicini A. Challenges and Research Directions in Agent-Oriented Software Engineering. Autonomous Agents and Multi-Agent Systems, 2004,9:253~283
- 4 Vidal J M. Inside an Agent. In:IEEE Internet Computing January, 2001,February 2001. 84~86
- 5 JADE. http://Jade.tilab.com
- 6 姚莉,张维明. 多智能体系统开发环境(MBOS). 计算机世界, 2001,7:26~28
- 7 姚莉,张维明,等. 智能协作信息技术. 北京:电子工业出版社, 2002