自主 Agent 结构 AASC 及其应用研究*

李　斌[1,2]，唐小燕[1]，吴梅丽[1]

(1. 扬州大学 计算机科学与工程系，江苏 扬州 225009；2. 南京大学 计算机软件新技术国家重点实验室，江苏 南京 210093)

摘　要: Agent 结构研究是 Agent 理论与技术研究的重要内容。基于 BDI 结构和情境演算理论，提出了一个能刻画 Agent 的多种特征(如自主性、主动性、反应性、社会性)，尤其是自主性的 Agent 结构 AASC。AASC 是一种具有广泛应用范围的 Agent 结构，它为不同类型 Agent(如主动 Agent、反应 Agent、社会 Agent、合作 Agent)的构造与表示提供了统一的框架。基于联合意向合作模型，详细讨论了合作 Agent 的构造方法。

关键词: Agent 结构；心智状态；情境演算；合作 Agent

中图法分类号: TP311；TP18　　**文献标识码**: A　　**文章编号**: 1001-3695(2005)07-0021-03

Research on Autonomous Agent Architecture AASC and Its Application

LI Bin[1,2]，TANG Xiao-yan[1]，WU Mei-li[1]

(1. *Dept. of Computer Science & Engineering, Yangzhou University, Yangzhou Jiangsu* 225009, *China*; 2. *State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing Jiangsu* 210093, *China*)

Abstract: Agent architecture is an important topic in the current research of Agent theory and technology. Based on BDI structure and situation calculus, an autonomous Agent architecture called AASC, which can depict various features (e.g. autonomy, proactivity, reactivity and social ability), especially autonomy of Agent, is presented in this paper. Since AASC is general Agent architecture, it can serve as a uniform framework for constructing and representing various types of Agents such as proactive Agent, reactive Agent, social Agent and cooperative Agent. Based on joint intentions cooperation model, a method for user to construct cooperative Agent is discussed in detail.

Key words: Agent Architecture; Mental States; Situation Calculus; Cooperative Agent

1　引言

Agent 结构研究是 Agent 理论与技术研究的重要内容[1]。目前研究表明，Agent 结构可分为三类[2]：①基于行为(Behavior-based)的结构，典型工作是 MIT 的 Brooks 提出的所谓包容结构(Subsumption Architecture)[3]。其优点是 Agent 能及时而快速地响应外来信息和环境的变化；缺点是其智能程度较低，缺乏足够的灵活性。②层次结构(Layered Architecture)，典型工作是 INTERRAP[4]。其优点是 Agent 具有较高的智能、较强的灵活性以及快速响应性；缺点是各层需要不同的规范语言，如反应层语言、规划层语言以及协调层语言等，另外与 BDI 结构相比，缺乏完善的理论基础。③BDI 结构，典型工作有 PRS，JAM 等[5]。其优点是具有完善的理论基础，较强的灵活性以及快速响应性；缺点是缺乏行动本身的推理及面向目标的规划。由于 Agent 往往处于动态的环境之中，要体现 Agent 的根本特征——自主性，仅有上述研究是不够的，还必须刻画各种心智成分如何随环境的变化而变化，这就需要 Agent 具有对行为的推理能力。目前，情境演算(Situation Calculus)[6]是广为采用的行动推理形式框架，在其中不但能讨论行动对系统状态的影响，而且可以研究规划问题。但现有的情境演算理论由于缺乏心智状态的表示，难以在其中探讨心智成分与 Agent 行为之间的相互关系。

结合 BDI 模型和情境演算的优点，构造一个自主 Agent 结构 AASC[7~9]，使之既能表示 Agent 的心智状态又能进行行动推理和规划，能够实现 Agent 不同程度的自主性以及刻画 Agent 多种特征，从而为实际应用中根据特定的应用背景建构 Agent 提供必要的理论基础。AASC 是一种具有广泛应用范围的 Agent 结构，它为不同类型 Agent 的构造与表示提供了统一的框架。利用 AASC，可以构造主动 Agent(Proactive Agent)、反应 Agent(Reactive Agent)、社会 Agent(Social Agent)、合作 Agent(Cooperative Agent)。基于联合意向合作模型，本文将详细讨论合作 Agent 的构造方法。

2　自主 Agent 的结构 AASC

目前人工智能领域广泛接受的一种观点认为，Agent 自主行为取决于 Agent 的心智状态，即 Agent 的信念、期望、意向、能力等心智要素是控制 Agent 智能行为的关键因素。因此设计并构造自主 Agent 的关键在于如何刻画 Agent 的各种心智成分以及这些心智成分如何影响 Agent 的行为。但对哪些心智成分是刻画 Agent 的自主性是必需的与基本的尚无定论，各种以不同心智成分为要素的 Agent 模型也是层出不穷。BDI 学派认为 Agent 的行为由其信念(Belief)、期望(Desire)及意向(In-

收稿日期: 2004-07-10；修返日期: 2004-09-15

基金项目: 国家“973”计划资助项目(2002CB312002)；江苏省自然科学基金资助项目(BK2004052)

tention) 所决定,因此信念、期望及意向成为 Agent 模型的主要成分。这种模型能成功地解释 Agent 在静态环境中的行为,但不能有效地解释其在动态环境中的行为,其主要原因在于这种 Agent 模型不能表示 Agent 行动的效果及 Agent 执行行动的能力,因此不能预测其行为对环境的影响。本文将考虑一些新的心智成分用于反映 Agent 对自身行为的知识以及对外界环境的作用效应、Agent 完成复杂行动的经验等。基于上述要素,定义的 AASC Agent 心智状态如下。

定义 1 AASC Agent 的心智状态是一个七元组 $M = (Bel, Act, Eff, Qual, Stra, Goals, Evt)$,其中,*Bel* 是状态信念集合,其元素为一阶语句; *Act* 是原子行动集; *Eff* 是效应集,其中每一原子行动、流对最多对应一条效应规则; *Qual* 是预决条件集,其中每一原子行动最多对应一条预决条件规则; *Stra* 是策略集; *Goals* 是目标集; *Evt* 是事件集。

关于 Agent 的心智成分的详细说明及其表示方法参见文献[7]。进一步,基于情境演算理论,提出了一个能刻画自主 Agent 的结构 AASC,其结构定义如下。

定义 2 自主 Agent 是一六元组 $(M, Rea, Upd, Rev, Sel_{ge}, Sel_a)$。其中,$M$ 表示 Agent 的非空心智状态集; *Rea* 表示推理算子; $Upd: M \times A \rightarrow M$,称为心智状态更新函数,$A$ 表示 Agent 的封闭原子行动项的集合; $Rev: M \times \text{Form}(L) \rightarrow M$,称为心智状态修正函数; $Sel_{ge}: M \rightarrow Goals \cup Evt$,称为目标与事件选择函数; $Sel_a: M \times (Goals \cup Evt) \rightarrow A$,称为行动选择函数。函数 *Upd*, *Rev*, Sel_{ge} 和 Sel_a 形式化地刻画 Agent 的自主性,决定了 Agent 的运行。

3 基于联合意向的合作模型在 AASC 中的实现

合作式问题求解是 MAS 系统研究的重要内容,而合作模型是合作式问题求解的关键。文献[10]提出了一个基于联合意向的合作模型,下面给出基于联合意向的合作模型在 AASC 中的实现。

3.1 基于联合意向的合作模型

Jennings 认为,在合作式问题求解中,多个 Agent 仅仅具有公共目标是不够的,它们还必须具有实现目标的公共处方(Common Recipe)。根据公共处方以及 Agent 之间协商、劝说和冲突消解等,建立各 Agent 的联合行动,因此,联合意向在多个 Agent 合作完成公共目标中起着重要的作用。联合意向定义为多个 Agent 在具有公共目标的情形下,执行联合行动的承诺。在文献[10]中,Jennings 提出了合作式问题求解的联合意向模型,即多个 Agent(用 $\alpha_1, \cdots, \alpha_n$ 表示)具有公共处方 $\sum$ 时,如何合作完成公共目标 σ。合作式问题求解的联合意向模型如下:

Joint_Responsibility({ $\alpha_1, \cdots, \alpha_n$} , σ, $\sum$)
BEGIN
FORALL $\alpha_i \in$ { $\alpha_1, \cdots, \alpha_n$}
WHILE α_i 具有公共目标 σ 且 α_i 有意向执行根据公共处方 $\sum$ 建立的联合行动 DO
PARALLEL
执行根据公共处方 $\sum$ 建立的联合行动以实现公共目标 σ;
监控公共目标 σ 的实现情况;
监控根据公共处方 $\sum$ 建立的联合行动的执行情况;
END_PARALLEL;
挂起根据公共处方 $\sum$ 建立的联合行动;
CASE 不能承诺的原因 OF
α_i 不具有公共目标 σ
(1) 放弃实现公共目标 σ 的承诺;
(2) 放弃根据公共处方 $\sum$ 建立的联合行动;
α_i 不能执行根据公共处方 $\sum$ 建立的联合行动
IF 有执行联合行动的其他方法
THEN 选择新的方法执行联合行动
ELSE 寻找新的公共处方 $\sum$ 重新建立联合行动
END_CASE;
通知其他成员承诺失败、原因以及执行联合行动的新方法;
END;

3.2 基于联合意向的合作模型在 AASC 中的设计与实现

设计基于联合意向的合作模型需考虑三个方面的问题:

(1) 协商,其作用是确定参与合作的 Agent,建立各 Agent 的联合行动;

(2) 规划,其作用是确保联合行动能在适当的时机由适当的 Agent 执行;

(3) 监控,其作用是对异常情况(如环境突变、资源冲突等)进行处理。

在具体实现时,用称为 MA(Manager Agent)的 Agent 进行协商、规划和监控。MA 根据熟人模型(即关于其他 Agent 的能力知识)、公共处方来进行协商和规划,其协商和规划的算法如下:

negotiation_and_planning(ma, Ω, G) ;
//Ω 为熟人集; G 为公共目标
BEGIN
MA 检查为实现目标 G 需建立的联合行动;
MA 确定处方 R 能实现目标 G;
FORALL 行动 θ R
选择 Agent A Ω 执行行动 θ;
计算执行行动 θ 的起始时间 t_θ;
发送请求(θ t_θ) 给 A;
发送请求与原有的承诺进行不一致检查;
IF 不存在冲突 THEN A 建立承诺(θ t_θ);
IF conflicts((θ, t_θ) , C) $\wedge$ priority(θ) > priority(C)
THEN 1) A 建立承诺(θ, t_θ) ;
(2) 重新规划 C;
IF conflicts((θ, t_θ) , C) $\wedge$ priority(θ) < priority(C)
THEN 1) 寻找能执行行动 θ 的起始时间($t_\theta + \triangle t_\theta$) ;
(2) A 建立承诺(θ, $t_\theta + \triangle t_\theta$) ;
(3) 返回起始时间($t_\theta + \triangle t_\theta$) ;
RETURN 接受承诺或修改时间给 ma;
IF 起始时间已修改 THEN 更新 $\triangle t_\theta$ 修改后序行动;
END_FORALL
END;

MA 对异常情况进行处理的监控算法如下:

monitor(MA)
BEGIN
IF 实现目标 G 的动机不存在
THEN 1) 放弃所有的联合行动;
(2) 放弃实现公共目标 σ 的承诺;
IF 参与合作的某 Agent 的联合行动不能执行
THEN 重新协商规划,确定另一 Agent 执行此联合行动;
IF R1 的执行结果不能实现目标且存在另一处方 R2
THEN 1) 放弃根据 R1 建立的联合行动;
(2) 根据 R2 重新建立新的联合行动;
IF R1 的执行结果不能实现目标且不存在另一处方
THEN 1) 放弃根据 R1 建立的联合行动;
(2) 放弃实现公共目标 σ 的承诺;
END;

因此,基于 AASC,MA 和参与合作的 Agent 的心智状态的初始信念集应增加熟人模型(即关于其他 Agent 的能力知识),

策略规则集中应增加协商规划策略及监控策略。

3.3　实例分析

现用积木世界来说明上述方法。假设有三个 Agent, 分别用 Agent1, Agent2 和 Agent3 表示, 其中 Agent1 为 MA, Agent2 和 Agent3 组成 MA 的熟人集。Agent1 的目标是将图 1 所示的初始状态中的积木 C 置于积木 B 上, 即满足 on(C, B) 。Agent1 不能执行原子行动 pickup(*x*) 和 stack(*x*, *y*) ; Agent2 能执行原子行动 pickup(A) 和 stack(A, *y*) ; Agent3 能执行原子行动 pickup(*x*) 和 stack(*x*, *y*) 。因此 Agent1 必须与 Agent2 和 Agent3 合作共同完成目标。

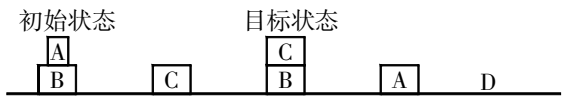

图 1　积木世界示例图

Agent1, Agent2 和 Agent3 的心智状态构造如下:

(1) Agent1 的心智状态

①环境信念集

block(A) ; block(B) ; block(C) ; table(D)

//A, B, C 为积木, D 为桌子

on(A, B) ; on(B, D) ; on(C, D)

//A 在 B 上; B 在 D 上; C 在 D 上

clear(A) ; clear(C)　//A 和 C 上无其他积木

②原子行动

pickup(*x*)　//抓起积木 *x*;

stack(*x*, *y*)　//置积木 *x* 于积木 *y* 上

③预决条件规则

(pickup(*x*) , false)　//不能执行行动 pickup(*x*)

(stack(*x*, *y*) , false)　//不能执行行动 stack(*x*, *y*)

④目标集

(move(C, B) , on(C, B) , 1)　/* 达到目标应满足条件 on(C, B) , 实现此目标的行动表达式为 move(C, B) , 目标的权值为 1* /

⑤策略规则集

(clearing(*x*) , clear(*x*) ? ∪(- clear(*x*) ?; πyπz((on(*y*, *x*) ∧(table (*z*)) ?; move(*y*, *z*))))　//移去积木 *x* 上的其他积木

(move(*x*, *y*) , clearing(*x*) ; clearing(*y*) ; pickup(*x*) ; stack(*x*, *y*))

//将积木 *x* 置于积木 *y* 上

(2) Agent2 的心智状态

①环境信念集、原子行动同 Agent1 的环境信念集、原子行动。

②效应规则

(clear(*x*) , (⊥, pickup(*y*) , *T*, on(*y*, *x*))

/* 如果积木 *y* 在积木 *x* 上, 则执行行动 pickup 后积木 *x* 上无其他积木* /

(on(*x*, *y*) , *T*, pickup(*x*) , ⊥, T)

//执行行动 pickup 后积木 *x* 不在积木 *y* 上

(clear(*x*) , *T*, stack(*y*, *x*) , ⊥, - table(*x*))

//如果 *x* 为积木, 则执行行动 stack 后积木 *x* 上有其他积木

(on(*x*, *y*) , ⊥, stack(*x*, *y*) , *T*, *T*)

//执行行动 stack 后积木 *x* 在积木 *y* 上

③预决条件规则

(pickup(*x*) , block(*x*) ∧clear(*x*) ∧ *x* = A)

//pickup(*x*) 的执行条件是 *x* 为积木 A 且 *x* 上无其他积木

(stack(*x*, *y*) , block(*x*) ∧clear(*y*) ∧ *x* ≠ *y* ∧ *x* = A)

/* stack(*x*, *y*) 的执行条件是 *x* 为积木 A 且另一积木 *y* 上无其他积木* /

(3) Agent3 的心智状态

①环境信念集、原子行动同 Agent1 的环境信念集、原子行动; 效应规则同 Agent2 的效应规则。

②预决条件规则

(pickup(*x*) , block(*x*) ∧clear(*x*))

//pickup(*x*) 的执行条件是积木 *x* 上无其他积木

(stack(*x*, *y*) , block(*x*) ∧clear(*y*) ∧ *x* ≠ *y*)

//stack(*x*, *y*) 的执行条件是 *x* 为积木且另一积木 *y* 上无其他积木

Agent1 为实现其目标 on(C, B) , 执行行动表达式 move(C, B) , 即行动序列为 { pick(A) ; stack(A, D) ; pickup(C) ; stack (C, B) } , 因为 Agent1 不能执行原子行动 pickup(*x*) 和 stack(*x*, *y*) , 它必须与 Agent2 和 Agent3 合作以完成其目标, 具体过程如下:

(1) Agent1 与 Agent2、Agent3 协商

①Agent1 发服务请求消息给 Agent2 和 Agent3

request(Agent2, pickup(A)) ; request(Agent2, stack(A, D)) ; request(Agent3, pickup(A)) ;

request(Agent3, stack(A, D)) ; request(Agent3, pickup (C)) ; request(Agent3, stack(C, B)) 。

②Agent2 和 Agent3 处理服务请求消息, 将结果回复给 Agent1

Agent2 承诺执行行动 pickup(A) 和 stack(A, D) ;

Agent3 拒绝执行行动 pickup(A) 和 stack(A, D) ;

Agent3 承诺执行行动 pickup(C) 和 stack(C, B) 。

③Agent1 确定 Agent2 和 Agent3 的联合行动

Agent2 执行联合行动 pickup(A) 和 stack(A, D) ;

Agent3 执行联合行动 pickup(C) 和 stack(C, B)) 。

(2) Agent1 监控联合行动的执行

Agent2 拒绝承诺执行行动 pickup(A) 和 stack(A, D) 。

(3) Agent1 与 Agent3 重新协商建立联合行动

request(Agent3, pickup(A)) ; request(Agent3, stack(A, D)) ;

Agent3 承诺执行行动 pickup(A) 和 stack(A, D) 。

(4) Agent3 执行重新建立的联合行动

(5) Agent1 监控联合行动的执行, 目标完成后, 发消息给各合作 Agent 解除承诺

4　总结

基于 BDI 结构和情境演算理论, 本文提出了一个能刻画 Agent 的多种特征(如自主性、主动性、反应性、社会性) , 尤其是自主性的 Agent 结构 AASC, 它是一种具有广泛应用范围的 Agent 结构, 为不同类型 Agent 的构造与表示提供了统一的框架。基于联合意向合作模型, 本文详细讨论了合作 Agent 的构造方法。

参考文献:

[1]　ennings NR. An Agent-based Approach for Building Complex Software Systems[J] . Communications of the ACM, 2001, 44(4) : 35-41.

[2]　Wooldridge MJ, Jennings NR. Intelligent Agent: Theory and Practice [J] . Knowledge Engineering Review, 1995, 10(2) : 115-152.

[3]　Brooks BA. A Robust Layered Control System for a Mobile Robot [J] . IEEE Journal of Robotics and Automation, 1986, 2(1) : 14-23.

(下转第 25 页)