◎学术探讨◎

一种基于社会承诺的 Agent 组织模型 OMSC

焦玉玺,李洪涛,张 伟
JIAO Yu-xi,LI Hong-tao,ZHANG Wei

烟台大学 计算机学院 智能信息处理实验室,山东 烟台 264005
Laboratory of Intelligent Information Processing,School of Computer Science,Yantai University,Yantai,Shandong 264005,China
E-mail:jyxyt2001@163.com

JIAO Yu-xi,LI Hong-tao,ZHANG Wei.Social commitment based agent organization model OMSC.Computer Engineering and Applications,2008,44(1):35-38.

Abstract: One of the main challenges faced by the multi-agent community is to ensure the coordination of autonomous agents in open heterogeneous multi-agent systems.In order to coordinate their activity,agents should be able to interact with each other. As a communication and interaction mechanism,social commitment provides an approach for agents to coordinate.However,multi-agent coordination only by means of interaction models is difficult to achieve.Agent organizations can control the way agents interact and cooperate effectively.In this paper we bring together the two mechanisms of coordinating agents:social commitment and agent organizations,propose a model based on social commitment for agent organizations,analyze how to reason using social commitment and commitment-based multi-agent systems,and give an example for our model.This model provides a new approach for agents to coordinate.

Key words: agent organizations;social commitment;multi-agent coordination

摘 要: 多 Agent 领域所面临的一个重大的挑战是解决开放异质的多 Agent 系统中自治 Agent 间的协调问题。多 Agent 为了协调它们之间的活动,需要进行交互。社会承诺作为一种通信和交互机制,为自治的多 Agent 提供了一种协调的途径。然而,仅靠交互难以实现多 Agent 间的协调。Agent 组织作为一种协调模型可以有效地控制多 Agent 间的交互与合作。论文将社会承诺和 Agent 组织两种协调机制相结合,提出一种基于社会承诺的 Agent 组织模型 OMSC,分析了 Agent 如何用社会承诺进行推理以及基于社会承诺的多 Agent 系统并给出了一个实例,为多 Agent 间的协调提供了一种新的方法。

关键词: Agent 组织;社会承诺;多 Agent 协调
文章编号: 1002-8331(2008)01-0035-04 **文献标识码:** A **中图分类号:** TP18

多 Agent 协调是 Agent 领域所面临的主要挑战之一。开放异质的多 Agent 系统中,每个 Agent 有不同的内部结构,它们可以按照自己的意愿加入和离开系统。因此,在不对 Agent 的内部模型做出要求的情况下很难描述系统所期望的行为。以往对 Agent 的内部模型大都做出某种假设,如假设 Agent 顺从组织的协调。由于 Agent 的自治性,不能认为 Agent 对协调机制的顺从理所当然,自利的 Agent 很可能会违反协调机制的要求。

为了协调它们之间的活动,多 Agent 需要进行交互。社会承诺作为对 Agent 行为约束的外部机制没有对 Agent 的内部模型做出任何假设,换言之,使用社会承诺不必考虑 Agent 的内部结构,仅关心承诺本身和承诺的执行状况。社会承诺不同于个体承诺和集体承诺,不必考虑 Agent 的内部推理,从而避免逻辑推理的复杂性。社会承诺作为一种通信语言已经成功应用于 Dialog Game[12,13],在通信的基础上实现多 Agent 间的交互。

然而,仅靠交互很难实现多 Agent 间的协调。原因之一是恶意的 Agent 很可能会违反它们所做出的承诺,这就需要采取措施避免这种情况的发生。系统的设计者应当描述系统对 Agent 行为的期望,对违反期望行为的 Agent 实施必要的惩罚。Agent 组织是解决这一问题的有效方法之一。已有的工作对各种组织模型和方法进行了研究,这里希望与社会承诺相结合,为开放异质的多 Agent 系统提供一种新的协调方法。将 Agent 组织和社会承诺结合需要回答的问题是:

(1)如何使用社会承诺来描述 Agent 扮演角色过程中期望的行为,即如何使用社会承诺来描述角色对 Agent 的约束?

(2)如何使用社会承诺来描述 Agent 对社会承诺的创建过程?

本文首先给出了社会承诺的定义、对社会承诺的操作以及社会策略(元承诺)的定义。第 2 章介绍了如何使用社会承诺来定义角色和组织结构,给出了一种基于社会承诺的 Agent 组织模型 OMSC。第 3 章分析了 Agent 如何用社会承诺进行推理并

基金项目: 国家自然科学基金重大项目(the National Natural Science Foundation of China under Grant No.60496323)。

作者简介: 焦玉玺(1980-)男,硕士研究生,主要研究方向为分布式人工智能、多 Agent 技术;李洪涛(1982-)男,硕士研究生,主要研究方向为分布式人工智能、网格计算;张伟(1961-),男,博士,教授,主要方向为分布式人工智能、多 Agent 技术。

给出了一种基于社会承诺的多 Agent 系统框架。第 4 章是基于 OMSC 的举例分析。最后给出结论并指出今后的研究方向。

1 社会承诺及其相关操作

文献[1]把承诺分为个体承诺、集体承诺和社会承诺三种类型。前两种承诺类似于 BDI 模型中的个体意图和联合意图,社会承诺是指一个 Agent 向另一个 Agent 保证执行一个特定的行为序列(下文中提到的承诺未作明确区分的均指社会承诺)。本文把社会承诺定义为一个谓词。

定义 1 社会承诺是谓词:

$SC(id,debtor,creditor,object,IA,status,condition,S_d)$

其中,id 是社会承诺的标识,debtor 表示承诺方,creditor 表示接受承诺方,object 表示承诺的内容,IA 是制度 Agent,status 表示承诺的状态,condition 表示承诺处于活动状态的必要条件,S_d 表示对 debtor 的奖惩。

一旦社会承诺形成,debtor 向 creditor 承诺在条件 condition 规定的时间内完成 object。object 可以是一个可实现目标、一个可完成任务或是一系列可执行动作。creditor 可以是一个 Agent 也可以是一个 Agent 群体。社会承诺被创建后放在一个承诺库中,并由 IA 负责统一管理和维护。IA 是承诺的见证者和维护者,负责对承诺进行正当合法的操作。通常情况下 condition 是 object 的最终期限。status 可以是以下几种状态之一:inactive(非活动状态),active(活动状态),violated(违反状态),fulfilled(实现状态),cancelled(取消状态)。社会承诺一旦被创建,只要满足条件就会处于 active 状态,处于 active 状态的社会承诺在最终期限内没有被完成就会进入 violated 状态。可以对承诺的状态的修改操作由 IA 来完成。

对承诺的操作有 3 种:承诺的创建、承诺的取消、承诺状态的修改。承诺的创建操作如图 1 所示:SCB 是系统中的承诺库。Creditor 向 debtor 提出对自己承诺的请求,debtor 同意后通知 IA 创建对 creditor 的承诺。对承诺的取消操作由 creditor 发起,并通过 IA 来完成。IA 不间断地监测 Agent 的活动,根据承诺的执行状况更新承诺状态。例如:对在最终期限内完成的承诺置以“fulfilled”状态,否则置“violated”状态。

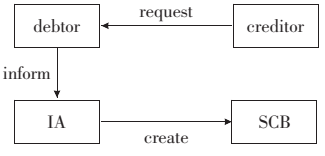

图 1 承诺的创建操作

Agent 对承诺的创建可用一个谓词来表示:$Create(x,SC(x,y,obj))$。创建承诺需要多 Agent 之间进行 Request 和 Inform 等交互通信操作。$Request(y,x,SC(x,y,obj))$表示 y 向 x 提出承诺的请求,$Inform(y,x,SC(x,y,obj))$表示 y 通知 x 创建承诺。基于承诺的多 Agent 系统中,为了便于对社会承诺进行描述和管理需要对承诺分类,$SC(x,y,obj)$表示以上三种属性相同的承诺集。

定义 2 承诺的归类:

$SC(x,y,obj)\equiv\forall id\,\forall IA\,\forall st\,\forall condSC(id,x,y,LA,obj,st,cond,S_x)$

由于 Agent 的自治性,Agent 在收到承诺请求时需要判断是否创建该承诺。同时,Agent 在没有收到承诺请求的条件下也可能会主动创建承诺。文献[5]引入了社会策略(Social Policy)的概念。社会策略是一个元承诺(meta-commitment),它的 object 域是一个包含承诺及其操作的条件表达式。本文仅关注社会策略的一个特殊形式:承诺的创建策略,它描述了对一个承诺请求的可能回答。

定义 3 社会策略是一个元承诺:

$SP(x,y,IA,\forall z\in A\,\forall obj\in T:Request(z,x,SC(x,z,obj))\Rightarrow Create(x,SC(x,z,obj)),status,condition,S_x)$

其中,A 表示系统中所有 Agent 的集合,T 表示任务及子任务的集合。

以上定义描述了在 IA 的监督下,x 对于某种请求要向 y 做出相应的承诺。社会策略规定了 Agent 的决策,描述了组织对 Agent 行为的期望。

2 Agent 组织模型 OMSC

Agent 组织对应一种社会结构和 Agent 间交互的模式,可以保证多 Agent 全局一致的行为。结构是组织的核心,无论哪种模型都不可避免地需要对组织结构进行描述。组织结构是对 Agent 组织的抽象,Agent 组织是对组织结构的实例化[16]。组织结构的核心概念是角色,角色是 Agent 功能的抽象描述,它规定了 Agent 的行为、权利、义务和目标等。为了协调系统全局的行为,需要描述 Agent 扮演角色对 Agent 的约束。包括当 Agent 扮演角色时必须做什么(义务),可以做什么(允许),不可以做什(禁止)以及与其它 Agent 的权利关系(层次性)。

Agent 组织中角色对 Agent 的约束:

(1)角色的目标:Agent 扮演角色就要完成角色的目标。

(2)角色的义务:Agent 扮演组织的角色要履行对组织的义务。

(3)角色间的权利关系:角色与角色有一定的权利关系。

(4)允许和禁止:Agent 扮演角色时可以做某些任务和不可以做某些任务。

2.1 角色的目标承诺

角色的描述包含对角色目标的描述,Agent 扮演某一角色必须完成该角色的目标。以往对角色目标的描述以一种 Agent 所能直接理解的方式进行,无形中对 Agent 的内部模型做出了假设。本文不对 Agent 的内部模型做出任何假设,以一种所有 Agent 都能理解的方式—社会承诺—对角色目标进行描述。从 Agent 角度来讲,Agent 要清楚系统对自己行为的期望,因此就要理解什么是社会承诺。Agent 对社会承诺进行理解并转化为内部概念(例如:目标、意图)用于内部推理。

定义 4 角色的目标承诺是一个社会承诺:

$SC(x,OA,IA,object,status,x:R,S_x)$

$x:R$ 表示 x 扮演角色 R 这一事实,它出现在承诺的 condition 域,表示只要 Agent 扮演角色这一事实成立角色目标就有效。为了与前面对社会承诺的定义取得一致,将角色的目标承诺定义为扮演组织中角色的 Agent 向代表组织的 Agent OA 做出的承诺。

2.2 角色的义务策略

Agent 在扮演角色的过程中有义务完成特定的任务,这些任务与上下文有关。Agent 在某一上下文中的义务由确定的条件来触发;这一触发过程不能用一个社会承诺来静态地表示,需要一个承诺的创建策略来详细描述创建承诺的条件,这个承

诺的创建策略就是角色的义务策略。在 Agent 扮演角色的过程中，角色的义务策略动态地约束 Agent 使其创建在不同上下文中的义务承诺(义务承诺是指 Agent 对要履行的义务所作的承诺)。

定义 5 角色的义务策略是一个元承诺:

$SP(x,OA,IA,\forall obj\in T:Request(OA,x,SC(x,OA,obj))\Rightarrow Create(x,SC(x,OA,obj)),status,x:R,S_x)$

2.3 角色的权利关系策略

层次化的 Agent 组织中角色和角色存在着权利关系,这种上下级的关系使得下级对于上级委托的任务要无条件接受。角色间的权利关系可以用元承诺来表示,它是一种特殊的承诺创建策略,这里称为角色的权利关系策略。

定义 6 角色 R_2 对角色 R_1 的权利关系策略是一个元承诺:

$SP(x,OA,IA,\forall y:R_2\forall obj\in T:Request(y,x,SC(x,y,obj))\Rightarrow Create(x,SC(x,y,obj)),status,x:R_1,S_x)$

Agent 扮演某一角色就要采纳与该角色有关的权利关系策略,这些策略描述了当 Agent 收到请求时创建承诺的决策。角色的权利关系可以归结为角色对角色的义务,前者是一种层次结构,而后者是一种强加给角色的规范,比如一个角色对另一个角色的权利可以归结为一个角色有义务接受来自另一角色的请求。

2.4 使用社会承诺定义角色和 OMSC 组织结构

对角色的定义不仅要用社会承诺来描述期望的行为,还要用义务的允许和禁止来描述系统可能的任务,另外还应包含角色所使用的资源。对于允许和禁止的任务,只要描述允许的任务即可。通常情况下,任何不能在允许中清楚描述的任务就是禁止的任务。

定义 7 角色定义为一个六元组:

$Role=<name,RGC,ROP,RAP,res,perm>$

Name:角色的标识。

RGC:角色的目标承诺集。

ROP:角色的义务策略集。

RAP:角色的权利关系策略集。

res:角色拥有的资源集。

perm:允许的任务集。

OMSC 组织模型的核心是组织结构,基于以上对角色的定义,可以把组织结构定义为角色和组织目标的集合。

定义 8 组织结构定义为一个二元组

$O_{stru}=<G,R>$

G:目标及子目标的集合。

R:角色的集合。

以上定义的组织结构不但具有角色的层次结构,而且包含约束 Agent 的行为规范。

定义 9 角色承诺可表示为下式:

$x:R\equiv RC(x,OA,IA,rolename,active,\text{true},S_x)$

当角色承诺变为活动状态,就表示 Agent x 扮演角色 $x:R$ 成为事实,定义该角色的所有承诺和策略都相应地变为活动状态。表达式 $x:R$ 为真当且仅当 RC 的状态为 active。RC 的条件域为 true 表示只要承诺没有被取消就一直处于活动状态。当 Agent 不再扮演角色时,角色承诺就被取消,相应地表达式 $x:R$ 变为假。

3 模型分析

3.1 使用社会承诺和策略进行推理

当 Agent 扮演某一角色时,它向组织承诺以期望的行为进行活动,这些期望的行为用社会承诺和社会策略来描述。社会承诺描述了 Agent 必需完成的角色目标；社会策略描述了 Agent 创建承诺的上下文和在上下文中期望的行为。社会承诺约束了 Agent 的行为;社会策略限制了 Agent 的决策。

社会承诺的定义中,不但描述了 Agent 的期望行为,还给出了违反期望行为应受到的惩罚。如果 Agent 决定要违反所做出的承诺,需要 Agent 对决策进行评估以确保自己利益的最大化。社会策略描述了 Agent 做出承诺的条件,它对 Agent 的决策有直接地影响,因为社会策略描述了组织对 Agent 在一定上下文中的行为期望。因此,社会策略使 Agent 决定什么时候做出承诺,以及承诺的对象和内容。

以社会承诺和社会策略的方式进行推理,Agent 可以判断扮演某一角色的得失。Agent 这种组织意识的社会推理使其能够自治地活动于开放的系统中。

3.2 制度 Agent

制度 Agent(IA)是承诺的见证者和监督者,并通过对承诺的执行情况进行奖惩来实现对承诺执行情况的监督。IA 完成如下任务:

(1)检测承诺的执行状况并置相应的状态。

(2)选择合适的奖惩方案。

(3)对 Agent 实施合理的奖励和惩罚。

随着环境的变化,IA 不断核实承诺的执行状况并对承诺置以相应的状态。选择一个合理的奖惩方案需要考虑的力度和公平性。文献[2]给出了人类社会奖惩的本体,提出了人工制度应该使用“严格责任(strict liability)”的假说:违反承诺的 Agent 应当支付所承诺任务的价值。另外，为了公平起见应该考虑 Agent 对承诺的违反是故意的还是非故意的，对非故意违反承诺的 Agent 可以从轻处罚甚至不予处罚。

为了确保 Agent 对惩罚的支付,文献[2][15]提出了一种相同的解决方案，建议 Agent 内部应当包含一个特殊的部件,称作:Dialog Manager 或 Governor。这个部件根据承诺的 status 域和 S_x 域对 Agent 进行自动地惩罚,不需要 Agent 的干预。当然,还有其它的解决方案,比如使用声誉的概念。Agent 的声誉分布于其它 Agent 中,因此 Agent 无法拒绝这种惩罚。另外一种解决方案类似于软件工程的瀑布模型，当 Agent 违反承诺时,另一承诺将会自动被创建用来对 Agent 进行惩罚;如果 Agent 继续违反这一承诺,那么下一个用于惩罚的承诺会被创建直至把 Agent 逐出组织为止。

3.3 基于社会承诺的多 Agent 系统

基于社会承诺的多 Agent 系统至少包含一个承诺库,用来储存 Agent 所创建的承诺以及定义角色的承诺和策略。系统在运行过程中承诺库由 IA 负责管理和维护。当 Agent 想要创建承诺时向 IA 提出请求,IA 把请求创建的承诺加入承诺库。同样对承诺的取消也是相同的操作,只有被承诺方才有资格提出申请。IA 监视着组织运行环境中所有的变化,确认承诺的任务是否被完成并相应地修改承诺的状态。例如:IA 检测到某一承诺已经过了最终期限并且所承诺的任务没有完成,从而确认该承诺处于违反状态。IA 根据承诺的状态,对相关 Agent 实施合理必要的奖惩。IA 还能够回答 Agent 对承诺的查询，使得 A-

gent 清楚与自己有关的承诺。

系统运行之初,承诺库仅包含定义角色的承诺和策略。这些反映组织结构的承诺和策略都处于 inactive 状态，等待 Agent 加入组织对它们进行激活。当 Agent 加入一个组织,它与代表组织的 Agent 进行交互来决定扮演组织的角色,因此它向 IA 请求创建角色承诺。IA 收到请求时创建该角色承诺并激活角色的所有承诺和策略,随后 Agent 扮演该角色来履行角色的承诺和策略。当由于某种原因 Agent 不再扮演角色时,Agent 与 OA 交互并由 OA 向 IA 提出取消承诺的请求,IA 收到请求后把该角色承诺从承诺库中删除,相应地把定义该角色的所有承诺和策略置为 inactive 状态。IA 负责承诺的创建、删除、承诺状态的修改以及对 Agent 实施必要的奖惩。对 Agent 的惩罚可以使用瀑布模型,在瀑布模型中社会策略描述了对 Agent 违反承诺的惩罚，这也是角色描述的一部分。社会策略确保了 Agent 对惩罚的承诺。

4 举例

在一个简单的拍卖组织 AO 中,有两个角色:拍卖者 RA、参与者 RP。拍卖者作为组织的管理者负责创建拍卖,接受来自参与者的投标,决定投标的胜者,以及对参与者进行奖惩。参与者仅负责投标,并且投标是秘密进行的,各参与者彼此不知道投标的内容。AO 的目标 AO_goal 是拍卖的成功。RA 的目标是:start(),informbid(),judgewinner()。这三个目标是顺序的,start():创建一个拍卖 Auid;informbid():通知 PA 进行投标;judgewinner():确定胜者。PA 的目标是:bid()。

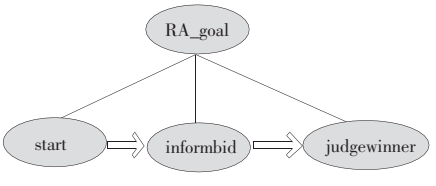

图 2 RA 的目标分解图

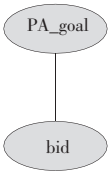

图 3 PA 的目标分解图

对 AO 的描述如下:

AO_{stru}=<R,AO_goal> //组织结构

R={RA,PA,IA} //角色的集合

$Role\_RA$=<RA,RA_RGC,RA_ROP,RA_RAP,RA_res,RA_perm> //RA 的定义

RA_RGC={$SC(x,OA\equiv RA,IA,RA\_goal,inactive,x:RA)$;} //$RA$ 的目标承诺集

RA_goal={$start()$;$informbid()$;$judgewinner()$;} //RA 的目标集

RA_ROP={$SP(x,OA\equiv RA,IA,Request(IA,x,SC(x,IA,Remove(Ag)))$ //RA 的义务策略集

$\Rightarrow Create(x,RA,Remove(Ag)),inactive,x:RA)$} //$Remove(Ag)$:把 Ag 驱逐出组织

RA_RAP=Φ

RA_res={ $Aution\_goods$;}//RA 的资源集

RA_perm={$start()\rightarrow informbid()\rightarrow judgewinner()$;} //$RA$ 的允许集

$Role\_PA$ =<PA , PA_RGC , PA_ROP , PA_RAP , PA_res , PA_perm>

RP_RGC={$SC(x,OA\equiv RA,IA,RP\_goal,inactive,x:PA)$;}

PA_goal={$bid()$;}

RP_ROP=Φ

RP_RAP={SP $(x,OA\equiv RA,IA,\forall y:RA\in obj\in T:Request(y,x,SC(x,y,obj))\Rightarrow Create(x,SC(x,y,obj)),inactive,x:PA)$;}

PA_res=Φ

PA_perm={$bid()$;}

以上是对 AO 的描述,当 Ag1 加入组织,它与 RA 进行交互决定扮演组织的角色 RP，因此它向 IA 请求创建角色承诺。IA 收到请求时创建该角色承诺并激活角色的所有承诺和策略,随后 Ag1 扮演该角色来履行角色的承诺和策略。当由于某种原因 Ag1 不再扮演角色时,Ag1 与 RA 交互并由 RA 向 IA 提出取消承诺的请求,IA 收到请求后把该角色承诺从承诺库中删除,相应地把定义该角色的所有承诺和策略置为 inactive 状态。在 Ag1 扮演角色 RP 的过程中,要履行对角色承诺和策略,IA 不断检测承诺履行状况,对承诺置以相应的状态,对处于 violated 的承诺,IA 向组织的管理者 RA 提出请求 $Request(IA,RA,SC(RA,IA,Remove(Ag1))$,把 Ag1 从组织中驱逐出去。

5 结束语

社会承诺已成功应用于多 Agent 间的交互和通信,在一定程度上实现了多 Agent 间的协调。本文针对社会承诺和 Agent 组织两种协调模型的特点，提出了一种基于社会承诺的 Agent 组织模型 OMSC，为多 Agent 间的协调提供了一种新的方法。模型最大的特点是没有对 Agent 的内部模型做出任何假设,真正适合应用于开放异质的环境中。本文使用社会承诺来定义角色和组织结构,清楚地描述了组织对 Agent 的约束。使得 Agent 可以容易地进行社会推理,判断扮演某一角色的得失,来决定是否扮演角色。本文引入 IA 来监视承诺的履行状况,结合奖惩机制保证了 Agent 对承诺的履行。

本文提出的基于社会承诺的 Agent 组织模型还仅是一种概念性的模型,下一步工作可以从以下三个方面进行。第一,在本文的基础上研究基于社会承诺的多 Agent 交互模型并基于时态逻辑给出形式化描述;第二,深入研究对 Agent 履行承诺的奖惩机制;第三,承诺库的设计与维护问题。

(收稿日期:2007 年 9 月)

参考文献:

[1] Castelfranchi C.Commitments:from individual intentions to groups and organizations[C]//Proceedings of the Int Conf on Multi-Agent Systems ICMAS95,1995:41-48.

[2] Pasquier Ph,Flores R,Chaib-draa B.Modeling flexible social commitments and their enforcement[C]//Gleizes M-P.Proc of the 5th Int Workshop EngineeringSocieties in Agents World ESAW05,2004:111-116.

[3] Singh M.An ontology for commitments in multiagent systems:towards a unification of normative concepts[J].AI and Law,1999,7:97-113.

[4] Dignum F.Autonomous agents with norms[J].Artificial Intelligence and Law,1999,7:69-79.

[5] Singh M.An ontology for commitments in multiagent systems:towards a unification of normative concepts[J].AI and Law,1999,7:97-113.

[6] Ferber J,Gutkenecht O,Michel F.From agents to organizations:an organizational view of multi-agent systems.AAMAS,2003.

(下转 41 页)