多 Agent 系统中基于认知的信任框架研究

王家昉 冯志勇 徐 超 许光全
(天津大学计算机科学与技术学院 天津 300072)

摘 要 基于认知角度提出了一种 Agent 间以预动(proactive)的方式建立信任的形式化框架.在框架中首先区分了代理与非代理情形下的信任并分别给出定义.从信任的定义出发,施信方(the trustor)针对信任建立基于认知的推理过程,并根据推理需要主动向受信方请求信息.在获得所需信息后,考虑到交互信息的可靠性问题,施信方在认知推理的基础上进行关于可靠性的模糊推理,并决定是否建立信任.通过这个框架,Agent 间可以在缺乏直接交互经验或者第三方证言的情况下,以预动的方式动态地建立信任,并且在信任建立的过程中,可以纳入复杂的上下文约束.同时,通过认知推理与模糊推理的结合,可以根据场景的需要采用不同的规则,给信任的建立带来更大的灵活性.

关键词 信任;多 Agent 系统;预动
中图法分类号 TP18 **DOI 号**: 10.3724/SP.J.1016.2010.00139

Research on Trust Framework in Multi-Agent System from the Cognitive View

WANG Jia-Fang FENG Zhi-Yong XU Chao XU Guang-Quan
(*School of Computer Science and Technology, Tianjin University, Tianjin* 300072)

Abstract From the cognitive view, this paper proposes a framework of establishing trust in a proactive manner. In the framework the definition of trust in the situation of delegation and non-delegation are distinguished, the agent as trustor establishes a reasoning process of trust based on the definition of trust. And then he takes initiative to request information which is needed in the reasoning process from the trustee. Considering the reliability of information in communication, the trustor conduct on a fuzzy reasoning process on the basis of cognitive reasoning after information received. The framework especially aims at the situation that the trustor has no experience of interaction with the trustee, and has no witness information about the trustee. With this framework, trust between agents can be established in a proactive manner. Context change can also be taken into consideration in the establishment of trust. Furthermore, the framework brings more flexibility on establishment of trust, as agent can use different inference rules in different context.

Keywords trust; multi-agent system; proactive

1 引 言

信任在多 Agent 交互中扮演着重要角色.目前在多 Agent 系统中,很多信任模型都是从博弈论的角度出发[1],基于 Agent 间的交互历史或第三方证言,根据预先定义的效用函数进行博弈,从而确定是否建立信任.但是在一些临时系统[2]中(例如在一些

收稿日期:2009-07-15;最终修改稿收到日期:2009-09-16.本课题得到国家"八六三"高技术研究发展计划项目基金(2007AA01Z137189)资助.**王家昉**,男,1982 年生,博士研究生,主要研究方向为智能 Agent、多 Agent 系统.E-mail: wjf@tju.edu.cn.**冯志勇**,男,1964 年生,教授,博士生导师,主要研究领域为知识工程、服务计算、应用中间件.**徐 超**,男,1982 年生,博士研究生,主要研究方向为服务计算、情感计算.**许光全**,男,1979 年生,博士,主要研究方向为智能 Agent、信任模型.

基于多 Agent 系统的虚拟组织中，参与的 Agent 往往是为了适应快速变化的环境、寻找最好的合作机会组织在一起，如现在出现的一些发布任务的网站，任务中国、猪八戒等），Agent 之间可能缺乏交互历史，也不能从第三方那里获取彼此间的信息. 此外，基于博弈论的模型往往采用预先定义的效用函数，难以反映 Agent 在信任建立过程中的动态上下文变化.

这种情况下可以考虑施信方（the trustor）与受信方（the trustee）间通过直接交互过程来建立初步的信任. 本文从认知的角度出发提出了一个施信方通过与受信方的预动交互建立信任的形式化框架. 这个框架从认知的角度入手，将 Agent 视为具有内部状态、输入以及输出的系统. 首先利用 Agent 的信念、意图给出了代理情形与非代理情形下的信任定义，施信方从信任定义出发建立关于信任的推理过程，并根据推理过程的需要向受信方 Agent 请求信息. 考虑到信息的可靠性问题，施信方接收到信息后，在基于认知推理的基础上进行关于信息可靠性的模糊推理，最终根据信任的可靠性决定信任能否建立. 这种针对信任建立的交互过程一方面通过认知推理反映了信任建立时动态变化的上下文，另一方面利用模糊推理使得信任建立更具灵活性.

本文第 2 节介绍相关的研究，主要是在多 Agent 领域针对信任的研究；第 3 节主要介绍多 Agent 系统（Multi-Agent System，MAS）中预动的信任框架，包括基于认知的信任的定义以及基于控制论的信任建立的过程；第 4 节通过一个场景说明如何根据本文所提出的框架在 Agent 间建立信任；第 5 节总结本文，并对进一步的工作进行讨论.

2 相关研究

在可计算的信任与信誉方面，研究人员提出了很多模型[1]. Marsh 的模型从博弈论的角度出发，根据信任关系可能带来的效用及其重要性权值以及直接交互的经历计算信任[1,3]. ReGerT 系统也是从博弈论的角度出发，考虑直接经验、第三方证言和社会结构三种因素，通过不同的模块组成一个完整的信任模型，同时用户可以决定采用或不采用特定的模块，为信任的建立带来灵活性[1,4]. 此外，Histos、Abdul-Rahman 与 Hailes、Yu 与 Singh 等很多模型，一般均考虑从博弈论的角度出发，根据交互经验或第三方证言来计算信任[1]，计算过程较为简单，方法容易利用. 但是由于多采用预先定义的效用函数，在计算过程中难以体现 Agent 运行中上下文的动态变化，此外，在很多基于博弈论的模型、方法中[5-7]，一般需要直接交互的经验以及第三方证言. 但是对于处于临时系统[2]中的 Agent，往往缺乏这些信息，采用如上的方法有一定困难.

Castelfranchi 与 Falcone 的模型则从认知的角度为信任提供了一个新的视角[1,8]. 他们的模型主要讨论了 Agent 间发生任务代理关系时的信任. 他们将信任定义为一些相关的信念，比如关于能力、依赖性、倾向性的信念等. 通过这样的定义，信任的建立可以通过基于认知逻辑的推理来实现，在推理的过程中可以纳入 Agent 运行的上下文约束，提高 Agent 间建立信任的灵活性. 本文所采用的信任的定义主要基于他们的模型. 但是，他们的模型在信任建立的过程方面依然借助于博弈论的方法，此外他们没有对非任务代理情形下的信任作进一步讨论.

从认知的角度来看，Agent 拥有其自身的心智部件（mental attitudes），如信念、意图等[9-10]. 在 MAS 中，通信在 Agent 间的协作中发挥着重要作用. Agent 接收消息，而后更新自身的信念以及意图等，并发出信息作为输出. 这样一个过程可以通过控制论中的状态空间方程来描述. 此外，对于 Agent 间的行为可以通过采用预动的方式来实现，以提升 Agent 间协作的效率. 如 SharedPlans 和 Joint Intention 中关于 Agent 帮助行为的研究[11-14]以及 Fan 等关于 Agent 间进行消息预动传递的研究等[15]. 那么在信任建立的过程中，特别是在一些临时系统[2]中，Agent 在缺乏与其它 Agent 的交互经历以及第三方证言的情况下，施信方与受信方就信任的建立展开预动的通信，可以有效地促进 Agent 间信任的建立.

3 Agent 间预动的信任建立框架

3.1 框架中信任的定义

通过相关工作的介绍可以看出，针对基于博弈论的方式建立信任所存在的问题，Castelfranchi 与 Falcone 提出的基于认知的信任模型提供了一种解决办法. 本文提出的框架也采用基于认知的信任定义，所采用的形式化表达基于 Fan 等在相关工作中[15]采用的意图语义和相关的公理、假设. 本文涉及的表达式如下：信念算子 Bel（相信）与 MB（双方均相信）（参见文献[9-10]）；意图算子 $Int.To$（意图

于执行,参见文献[9-10,12])、$Int.Th$(意图于命题成立,参见文献[9-10,12])、$Pot.Int.To$(潜在地意图于执行,参见文献[9-10,12])、$Pot.Int.Th$(潜在地意图于命题成立,参见文献[9-10,12]);动作 $Attempt$、$Inform$、$Request$(试图、通知,请求等,参见文献[15]);表达式 $CONF$(命题、动作间的冲突,参见文献[15])、$constr(C)$(上下文约束 C 中的要素集合,参见文献[15])、$Hold(p,t)$(客观成立,参见文献[15])、CBA(有执行动作的能力,参见文献[15])、$recipe_X(\alpha)$(Agent X 用以执行 α 的 $recipe$ 的集合,参见文献[15]). $prop(X)$表示与 Agent X 的信念、意图有关的命题表达式的集合.

3.1.1　代理关系中的信任

首先讨论关于代理过程中的信任,即施信方将某些动作代理给受信方的过程中的信任关系. 在 Castelfranchi 等的工作中提到信任是代理行为在精神(心智)上的对应物[8]. 本文对信任的定义的思想来源于他们给出的信任定义. 在信任的定义中首先应该包括施信方与受信方. 其次需要包括双方就什么问题建立信任. Castelfranchi 等认为代理关系是指,施信方 Agent 需要或者希望受信方 Agent 能够将特定的行为纳入到自己的规划当中[8]. 因而在任务代理关系中的信任需要包括施信方希望受信方产生的关于在恰当时间执行代理动作的意图. 此外还包括信任所处的特定上下文以及信任存在的时间. 综合这些因素,关于代理过程中存在的信任通过如下算子表达:$Trust(X,Y,Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t)$. 其中 X、Y 分别代表信任关系中的施信方与受信方,g 是 X 代理给 Y 的动作,$Int.To(Y,g,t_{Int},t_g,C_g)$表明在 X 对于 Y 的信任在于:在动作应该执行的时刻 t_g 之前的时刻 t_{Int},Y 形成执行关于 g 的意图. C_{Trust} 是 X 对于 Y 的信任存在的上下文约束,这些约束来自于关于信任的推理过程中涉及到的信念、意图的一些约束. t 是信任成立的时刻. 下面的讨论中,以 X 代表施信方 Agent,Y 代表受信方 Agent.

下面给出信任的语义. Castelfranchi 等提出的信任定义主要是用心智部件表达了 3 个方面的内容:关于 Y 执行代理动作的能力、倾向以及 X 对于 Y 的依赖性. 这里采用如下的定义给出信任的语义定义.

定义 1.　X 建立关于 Y"能够意图于执行特定动作"的信任,是指 X 相信在恰当的时刻,Y 要形成关于特定动作的潜在意图,X 也要相信自己意图于让"Y 能够意图于执行特定动作"成立,此外,X 还要相信 Y 有能够用于执行特定动作的具体步骤:

$$Trust(X,Y,Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t)=$$
$$Bel(X,\exists t'_{Int}<t_{Int}Pot.Int.To(Y,g,t'_{Int},t_g,C_g\wedge C'),t)\wedge$$
$$Bel(X,\exists t''_{Int}<t_{Int}Int.Th(X,Int.To(Y,g,t_{Int},t_g,C_g),t''_{Int},t_{Int},C_g\wedge C'),t)\wedge$$
$$Bel(X,\exists R_g\in recipe_Y(g)(CBA(Y,g,R_g,t_g,C_g)\wedge\forall\alpha\in R_g,\exists t'<t_\alpha Int.To(Y,\alpha,t',t_\alpha,C_\alpha)),t),$$
$$C'=\forall Z\neq Y\,\forall t''<t_g\rightarrow Int.Th(X,Int.To(Z,g,t_{Int},t_g,C_g),t'',t_{Int},C_g),$$

记:

$$Bel_{Trust\_Int\_Th}(t)=Bel(X,\exists t''_{Int}<t_{Int}Int.Th(X,Int.To(Y,g,t_{Int},t_g,C_g),t''_{Int},t_{Int},C_g\wedge C'),t),$$
$$Bel_{Trust\_Int\_To}(t)=Bel(X,\exists t'_{Int}<t_{Int}Pot.Int.To(Y,g,t'_{Int},t_g,C_g\wedge C'),t),$$
$$Bel_{Trust\_CBA}(t)=Bel(X,\exists R_g\in recipe_Y(g)(CBA(Y,g,R_g,t_g,C_g)\wedge\forall\alpha\in R_g,\exists t'<t_\alpha Int.To(Y,\alpha,t',t_\alpha,C_\alpha)),t).$$

信任的定义主要是通过 Agent 的 3 个信念来表达的. 信念 $Bel_{Trust_Int_To}(t)$是说 X 相信在 t_{Int} 之前的某个时刻 t'_{Int},Y 能够形成关于在 t_g 时刻在上下文约束 $C_g\wedge C'$ 下执行 g 的潜在意图. C'是说在 t_g 前的任意时刻 t'',除了 Y 以外,X 不会意图于让其它 Agent在 t_g 时刻完成动作 g. 信念 $Bel_{Trust_Int_Th}(t)$是说 X 相信在 t_g 前的某个时刻 t''_{Int},X 意图于在 t_{Int} 时刻使得 Y 产生执行 g 的意图. 信念 $Bel_{Trust_CBA}(t)$则反映了 X 对于 Y 完成 g 的能力的信念. 即 X 相信 Y 有能够用来完成动作 g 的相关的方法,并且对于这些方法中的动作,Y 能够在恰当的时候形成关于这些动作的意图.

关于信任如何在 Agent 交互中发挥作用需要进一步讨论. 根据 Shared Plan 理论以及 Fan 等关于预动地传递信息的讨论,当一方 Agent X 要求另一Agent Y 执行某个动作 β,或将自己规划中的某个动作 β 交给 Y 完成,在这个过程中,X 应该意图于让 Y 意图于执行 β. 假定 X 想要完成动作 α,R_α 是 X 用以完成 α 的某个 $recipe$,即 $R_\alpha\in recipe_X(\alpha)$. 而动作 $\beta\in R_\alpha$,即 β 是 R_α 中的一个动作,此时 X 因为某种原因希望 β 由 Y 来完成. $Pot.Int.Th(X,Int.To(Y,\beta,t_{Int},t_\beta,C_\beta),t,t_{Int},C_\beta)$代表 X 在 t 时刻有潜在的意图,让 Y 在 t_{Int} 时刻意图于在 t_β 时刻执行 β. 考虑到 Agent 往往代表不同的利益实体,因而 X 意图于让 Y 执行 β,Y 可能出于利益的考虑使得 X 的意图失败. 这时就需要 X 通过信任,特别是信任涉及的上下文约束来约束自己的选择. 也就是说,X 希望通

过 R_α 来完成 α，那么对于 β 不仅要有意图 $Pot.Int.Th(X,Int.To(Y,\beta,t_{Int},t_\beta,C_\beta),t,t_{Int},C_\beta)$，还需要建立起关于 Y 执行 β 的信任，才能将潜在意图转化为意图（$Pot.Int.Th$ 转化为 $Int.Th$）. 这时信任应该纳入 $Int.Th(X,Int.To(Y,\beta,t_{Int},t_\beta,C_\beta),t,t_{Int},C_\beta)$ 的上下文约束，保证在 X 意图成立的时候，信任也是成立的.

3.1.2 非代理关系中的信任

除了代理关系中的信任，本文认为在 Agent 协作中某些非代理情形下也需要信任来约束 Agent 的决策. 考虑在协作的过程中，其中一个 Agent X 的某个动作 α 可能会受到另一个 Agent Y 的某个动作 g 的干扰而不能顺利完成. 在他们协作的过程中，出于种种原因 X 需要完成 α，并且出于协作的考虑，X 需要建立关于 Y 不会执行 g 的信任，在此情况下 X 才选择执行 α. 此时 X 是施信方，Y 是受信方. 这里称这种情况为非代理情形下的信任. 这种情形与代理过程中信任的不同在于：代理过程中 X 相信 Y 有能力执行 g 并且有意图执行 g 时 X 信任 Y，而非代理情形下 X 相信 Y 有能力执行 g 的情况下 Y 不会执行 g 时 X 信任 Y. 这样，非代理情形下的信任通过如下算子来表达：

$$Trust(X,Y,\forall t_{Int}\leq t_g\rightarrow Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t),$$

其中 X、Y 分别代表信任关系中的施信方与受信方，g 是导致 Y 与 X 发生冲突的动作，$\forall t_{Int}<t_g\rightarrow Int.To(Y,g,t_{Int},t_g,C_g)$ 表明在 X 对于 Y 的信任在于：在动作 g 应该执行时刻 t_g 前的任意时刻 t_{Int}，Y 不会形成执行关于 g 的意图. C_{Trust} 是 X 对于 Y 的信任存在的上下文约束. t 是信任存在的时刻. 这里采用如下的定义给出信任的语义.

定义 2. X 建立关于 Y“不会意图于执行特定动作”的信任，是指 X 相信存在某个执行该特定动作的具体步骤，能让 Y 用来执行特定动作，但是 X 相信对于任何 Y 用来执行特定动作的具体步骤，都有某些步骤 Y 不会形成执行这些步骤的意图.

$$\begin{aligned}&Trust(X,Y,\forall t_{Int}\leq t_g\rightarrow Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t)=\\&Bel(X,\exists R_g\in recipe_Y(g)CBA(Y,g,R_g,t_g,C_g),t)\wedge\\&Bel(X,\forall R_g\in recipe_Y(g)(CBA(Y,g,R_g,t_g,C_g)\rightarrow\\&\quad\forall\alpha\in R_g,\forall t'\leq t_\alpha\rightarrow Int.To(Y,\alpha,t',t_\alpha,C_\alpha)),t).\end{aligned}$$

这里记

$$\begin{aligned}Bel_{Trust\_CBA\_N}(t)=&Bel(X,\exists R_g\in recipe_Y(g)CBA(Y,g,\\&R_g,t_g,C_g),t),\\Bel_{Trust\_Int\_N}(t)=&Bel(X,\forall R_g\in recipe_Y(g)(CBA(Y,g,R_g,\\&t_g,C_g)\rightarrow\exists\alpha\in R_g,\forall t'<t_\alpha\rightarrow Int.To(Y,\alpha,\\&t',t_\alpha,C_\alpha)),t).\end{aligned}$$

非代理情形下的信任通过信念 $Bel_{Trust\_Int\_N}(t)$ 和 $Bel_{Trust\_CBA\_N}(t)$ 来表达. $Bel_{Trust\_CBA\_N}(t)$ 是说 X 相信 Y 有一些用来执行 g 的 $recipe$，能够在 t_g 时刻用来执行 g. $Bel_{Trust\_Int\_N}(t)$ 信念是说对于任何能够用来在 t_g 时刻执行 g 的 $recipe$ R_g，其中会存在某些动作 $\alpha\in R_g$，在 α 应该执行的 t_α 时刻，Y 不会形成关于 α 的意图，从而阻止 g 的执行.

首先需要说明的是，在非代理情形下的信任中，施信方因该相信受信方有能力对自己的目标造成威胁，因而需要在信任的定义中包括信念 $Bel_{Trust\_CBA\_N}(t)$. 如果施信方 X 认为受信方没有能力发起能够威胁自己的行动，那么 X 与 Y 之间没有必要就 g 建立信任. 其次是关于非代理情形下的信任如何在 Agent 交互中发挥作用. 考虑如下情形：

$$\begin{aligned}&Bel(X,\forall t_{Int}\leq t_gCONF(Int.Tx(X,prop,t',t_{prop},C_{prop}\\&\quad Int.To(Y,g,t_{Int},t_g,C_g),t_{prop},t_g,C_{prop},C_g),t),\\&\qquad\qquad t<t',t<t_g.\end{aligned}$$

其中 $Int.Tx$ 代表 $Int.To$ 或者 $Int.Th$. 上面的信念是说，在时刻 t，X 相信 X 在 t' 时刻意图于在时刻 t_{prop} 执行动作 $prop$，或者在 t' 时刻意图于命题 $prop$ 在 t_{prop} 时刻成立，会与 Y 在 t_g 前任意时刻意图于在 t_g 时刻执行 g 相冲突. 在这种情况下，X 想要形成关于 $prop$ 的意图，应该将 $Trust(X,Y,\forall t_{Int}\leq t_g\rightarrow Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t)$ 纳入到关于 $prop$ 的意图或潜在意图的上下文约束当中，特别是在 X、Y 之间存在协作，但是各自又有不同利益的情况下.

3.2 基于控制论的信任建立过程

上下文约束在信任中发挥着重要的作用. 在文献[16]中谈到在决定是否信任的过程中，是伴随着不确定的行为的. 因为如果双方行为都确定的情况下，也就不需要所谓信任的约束了. 随着上下文的变化，这些不确定的行为在特定的上下文下就会给信任带来影响，因而这些纳入到信任上下文当中的因素真正反映了 Agent 在建立信任过程中经历的动态变化.

上面所给出的信任定义考虑到了这样的上下文约束问题. 以代理关系中的信任为例，Agent 通过推理能够得到信任的相关信念，假定有如下规则：

$$\begin{aligned}&p_1\wedge p_2\wedge\cdots\wedge p_i\wedge\cdots\wedge p_n\rightarrow Bel_{Trust\_Int\_To}(t),\\&p'_1\wedge p'_2\wedge\cdots\wedge p'_i\wedge\cdots\wedge p'_{n'}\rightarrow Bel_{Trust\_Int\_Th}(t),\\&p''_1\wedge p''_2\wedge\cdots\wedge p''_i\wedge\cdots\wedge p''_{n''}\rightarrow Bel_{Trust\_CBA}(t).\end{aligned}$$

其中 $p_i(1\leq i\leq n)$ 与 $p'_i(1\leq i\leq n')$ 以及 $p''_i(1\leq i\leq n'')$ 是 Agent 的命题. 当 Agent 知道 p_i、p'_i 以及 p''_i 的

真值后,就能得出相关的信念从而建立信任.此外鉴于伴随信任出现的不确定性,根据 Agent 运行环境的变化,可能存在着某些规则影响 p_i、p'_i、p''_i的真值,比如存在如下规则:

$p'_{i1} \wedge p'_{i2} \wedge \cdots \wedge p'_{ij} \wedge \cdots \wedge p'_{ik'} \rightarrow q'_i$并且此时 $Bel(X, CONF(p'_i, q'_i, t_{p'_i}, t_{q'_i}, C_{p'_i}, C_{q'_i}), t)$(施信方 X 在 t 时刻认为 p'_i与命题 q'_i存在冲突);或者 $p'_{i1} \wedge p'_{i2} \wedge \cdots \wedge p'_{ij} \wedge \cdots \wedge p'_{ik'} \rightarrow \neg p'_i$.当 Agent 在建立信任的过程中如上规则在特定的上下文下被满足的时候,可能导致信念 $Bel_{Trust\_Int\_To}(t)$(或 $Bel_{Trust\_Int\_Th}(t)$、$Bel_{Trust\_CBA}(t)$)不成立.这些规则在信任建立中发挥约束作用.

鉴于信任建立过程中存在不确定的因素,因而不论在上面与 $Bel_{Trust\_Int\_To}(t)$、$Bel_{Trust_Int_Th}(t)$和 $Bel_{Trust_CBA}(t)$相关的规则,还是发挥约束作用的规则,往往存在一些命题,Agent 在建立信任的时候还不了解它们的真值,这些命题往往与受信方的信念、意图相关,表示受信方当前的状态.此时施信方 Agent想要建立信任,就应该采取预动的方式,从受信方以及其它 Agent 那里寻求这些信息,特别是在缺乏与受信方的交互历史或第三方证言的情况下.这里采用控制论当中状态空间方程来说明这样一个预动的信任建立过程.非代理情形下的信任建立的过程与代理关系下的过程类似,这里主要以代理过程中信任为例来说明信任的建立过程.

施信方与受信方间建立信任的过程可以通过如下的状态空间方程来表达.

方程 1.

对于施信方(trustor)的 Agent:

$$X_{\text{trustor}}(t+1) = F(X_{\text{trustor}}(t), U_{\text{trustor}}(t), t),$$

$$Y_{\text{trustor}}(t) = G(X_{\text{trustor}}(t), U_{\text{trustor}}(t), t).$$

对于受信方(trustee)的 Agent:

$$X_{\text{trustee}}(t+1) = F'(X_{\text{trustee}}(t), Y_{\text{trustor}}(t), t),$$

$$Y_{\text{trustee}}(t) = G'(X_{\text{trustee}}(t), Y_{\text{trustor}}(t), t),$$

$$U_{\text{trustor}}(t+1) \subseteq Y_{\text{trustee}}(t).$$

这里下标中的 trustor 与 trustee 分别代表施信方与受信方两个 Agent.$X(t)$代表着 Agent 的内部状态,是一组 Agent 心智部件的集合.按照状态空间方程的定义,将 $U(t)$与 $Y(t)$则分别称为输入、输出变量的集合.这里输入变量 $U_{\text{trustor}}(t)$是与 Y 的信念有关的表达式 *prop* 构成的集合,此时 X 没有"Y 关于 *prop* 信念"的信念.为了形成针对 Y 的信任,X 需要"Y 关于*prop*信念"作为输入.而输出变量 $X_{\text{trustor}}(t+1)$则是 X 希望告知 Y 的、表达 X 自身的某些信念的表达式,在与 Y 的交互中发送给 Y,表达自己的意图、信念.由于 X 需要 Y 对于输入变量所代表的信念进行回复,因而输入变量也要发送给 Y,在后面的讨论中假定 $Y(t)$集合中包含有 $U(t)$,这里输入、输出变量采用二元组(*prop*,*true_value*)的形式,真值包括 unknown(针对输入变量)、true 和 false(针对输出变量),具体定义如下:

假定 X 为施信方,Y 为受信方

X 的输入变量 *var* 定义为如下集合中的元素

$\{(prop_i, true\_value_i) \mid$

$1 \leq i \leq n, prop_i \in prop(X), true\_value = \text{unknown}\}$

$true\_value_i = \text{unknown}$,

iff $Bel(X, Bel(Y, prop_i, t), t) = \text{false}$

and $Bel(X, Bel(Y, \neg prop_i, t), t) = \text{false}$.

X 的输出变量 *var* 定义为如下集合中的元素

$\{(prop_i, true\_value_i) \mid$

$1 \leq i \leq n, prop_i \in prop(X), true\_value = \text{true} \mid \text{false}\}$

$true\_value_i = \text{true}$, iff $Bel(X, prop_i, t) = \text{true}$,

$true\_value_i = \text{false}$, iff $Bel(X, \neg prop_i, t) = \text{true}$.

其中 $prop_i$是符合意图语义[9]的命题.开始,施信方在构建关于信任的推理的过程中,找出那些与受信方有关并且自己不能确定其真值的命题,将它们放入到 $U_{\text{trustor}}(t)$中作为输入变量,同时将其包含在输出变量 $Y_{\text{trustor}}(t)$中发送给受信方来寻求答案.此外施信方还可以找出可能有助于受信方得出关于输入变量的真值的信念与意图,将它们放在 $Y_{\text{trustor}}(t)$中发送给受信方.受信方接受到 $Y_{\text{trustor}}(t)$后,首先根据施信方发送过来的内容更新自己的内部状态,并且通过推理得出输入变量中命题的真值,放到输出变量 $Y_{\text{trustee}}(t)$发送回施信方,施信方再更新自身的内部状态,完成对于信任的推理并建立信任.这里,$t+1$ 表示状态或变量真值被更新后的时间.$U_{\text{trustor}}(t+1) \subseteq Y_{\text{trustee}}(t)$是说在施信方 trustor 所接收到的输入变量 $Y_{\text{trustee}}(t)$中,应该包括施信方所要求的输入变量 $U_{\text{trustor}}(t)$被受信方所更新后的值 $U_{\text{trustor}}(t+1)$.

F、G、F'、G'分别代表 X 与 Y 为了建立信任进行的推理或计算过程.

F:对于施信方 Agent X,过程 F 表示根据内部状态 $X_{\text{trustor}}(t)$集合建立推理关于建立信任的推理过程,确定输入变量集合 $U_{\text{trustor}}(t)$的成员,并且在接收到受信方 Agent Y 发送过来的输入变量后,将内部状态 $X_{\text{trustor}}(t)$更新为 $X_{\text{trustor}}(t+1)$.

G:对于 X,过程 G 根据内部状态和输入变量 $U_{\text{trustor}}(t)$确定输出变量 $Y_{\text{trustor}}(t)$,这里输出变量包括输入变量以及根据场景需要 X 认为需要告知 Y

的信息.

F'、G'的作用与 F 和 G 类似,这里主要讨论施信方针对受信方信任的建立,因而不对 F'、G'表示的受信方的行为与推理过程进行讨论,假定受信方 Y 能够按照施信方 X 的要求给出相应信息,可以通过基于认知的推理,也可基于博弈的方式来计算.

3.3 信任建立过程的描述

根据上面提到的步骤,首先要考虑的是建立信任的通信过程的启动.通过上一节的介绍可以看出,信任在 Agent 协作和交互的过程中,对于意图和行为的选择发挥着约束作用.对于代理情形下的信任而言,如果施信方 Agent 有潜在的意图让受信方 Agent代理完成特定的动作,在这种潜在意图转化为关于代理的意图之前,应该将信任纳入到潜在意图的上下文约束当中,因此这里给出假定.

假定 1(代理情形). 在代理情形下,X 有让"Y 意图执行特定动作"成立的潜在意图,那么 X 意图于让针对"Y 意图执行特定动作"的信任成立:

$$Pot.Int.Th(X,Int.To(Y,g,t_{Pot.Int.Th},t_g,C_g),t,t_{Pot.Int.Th},C_{Pot.Int.Th})\rightarrow$$
$$\exists t_{Trust}<t_g Int.Th(X,Trust(X,Y,Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t_{Trust}),t,t_{Trust},C_{Int.Th}),$$

其中 $t_{Pot.Int.Th}<t_g$ 且 $t<t_{Trust}<t_g$.

假定 2(非代理情形). 在非代理情形下,X 有执行某个动作或让某个表达式成立的潜在意图,并且 X 相信另一个动作的执行与 X 要执行的动作或希望成立的表达式有冲突,如果 X 相信 Y 有执行另一个动作的意图,那么 X 意图于让针对"Y 不会意图执行特定动作"的信任成立.

$$Pot.Int.Tx(X,\alpha,t,t_\alpha,C_\alpha)\wedge Bel(X,CONF(\alpha,g,t_\alpha,t_g,C_\alpha,C_g),t)\wedge$$
$$Bel(X,\exists t'<t_\alpha Pot.Int.Tx(Y,g,t',t_g,C_g),t)\rightarrow$$
$$\exists t_{Trust}<t_g Int.Th(X,Trust(X,Y,\forall t_{Int}\le t_g\rightarrow Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t_{Trust}),t,t_{Trust},C_{Int.Th})$$

对于代理的情形,这个假定是说在当前时刻 t,施信方 Agent X 有潜在意图想要受信方 Agent Y 在时刻 $t_{Pot.Int.Th}$ 形成关于在 t_g 执行特定的行为 g 的意图,那么 t_g 前的特定时刻 t_{Trust},X 应该产生一个关于在 X 和 Y 之间建立针对所代理动作 g 的意图.对于非代理情形,这个假定是说在当前时刻 t,X 有关于 α 的意图(意图于在时刻 t_α 执行 α 或让 α 成立).此时 X 又相信在 t_α 之前的某个时刻 t' Y 会形成关于 g 的意图(意图于在时刻 t_g 执行 g 或让 g 成立),并且 X 相信 α 与 g 之间存在冲突,那么 X 应该产生一个关于 g 的非代理情形下信任成立的意图.其中 t_{Trust} 根据具体需要而定.

在当前时刻 t,施信方 Agent X 形成关于信任的意图后,按照前面介绍的基于控制论状态空间的框架描述,就需要建立针对 Agent 的推理过程并与受信方 Agent 交互.

规则 1. 当施信方 Agent X 意图于让针对于 Y 的代理或非代理情形下的信任成立,那么 X 形成关于执行 *TrustAct* 动作的潜在意图.其中动作 *TrustAct* 用于建立围绕施信方于受信方交互的推理过程.

代理情形

$$Int.Th(X,Trust(X,Y,Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t_{Trust}),t,t_{Trust},C_{Int.Th})\rightarrow$$
$$Pot.Int.To(X,TrustAct(X,Y,g,inputVar,outputVar,t,t_{Res},t_{Trust}),t,t_{Trust\_Act},C_{Trust}\wedge C_{Int.Th})$$

非代理情形

$$Int.Th(X,Trust(X,Y,\forall t_{Int}\le t_g\rightarrow Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t_{Trust}),t,t_{Trust},C_{Int.Th})\rightarrow$$
$$Pot.Int.To(X,TrustAct(X,Y,inputVar,outputVar,t,t_{Res},t_{Trust},g),t,t_{Trust\_Act},C_{Trust}\wedge C_{Int.Th}),$$

其中 $t<t_{Trust\_Act}<t_{Res}<t_{Trust}$.

规则 1 是说当施信方 X 意图于在他与受信方 Y 之间建立信任(代理情形和非代理情形下)之后,X 随之形成一个在时刻 t_{Trust_Act} 执行动作 *TrustAct* 的意图. *TrustAct* 动作负责建立关于信任的推理过程,查找输入输出变量,并请求 Y 对与输入变量中命题的真值进行回答,其定义如下.

定义 3.

$$TrustAct(X,Y,g,inputVar,outputVar,t,t_{Res},t_{Trust})=((t<t_{Res})?;ConstructTrust(Y,inputVar,outputVar,g,t))?;Request(X,Y,\varepsilon,TrustResponse(Y,X,inputVar',outputVar',g,t,t_{Trust}),t_{Res},t_{Trust},C_{TrustRes})$$

X 在 t 时刻执行动作 *TrustAct*,要求 X 在时刻 t 之前完成关于信任的状态空间构建,包括输入输出变量的查找,查找所得到的输入输出变量分别放在 *inputVar* 与 *outputVar* 中.这里 *inputVar* 相当于状态空间中的 $U_{\text{trustor}}(t)$,而 *outputVar* 相当于 $Y_{\text{trustor}}(t)$.在状态空间构建成功之后在时刻 t_{Res} X 要求 Y 在 t_{Trust} 时刻之前执行 *TrustResponse* 动作,这个动作要求 Y 根据接收到的输入变量 *inputVar'*,得到 X 所要求的输入变量的值并放在 *outputVar'* 中,然后返回给 X.

首先来看与构建状态空间相关的 *ConstructTrust* 动作.这里将 *ConstructTrust* 作为原子动作.以代理过程中建立信任为例.由于 X 形成关于

$Trust(X,Y,Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t_{Trust})$的意图,根据 Trust 的定义 1,$X$ 要通过规则得到信念 $Bel_{Trust\_Int\_To}(t)$、$Bel_{Trust_Int_Th}(t)$和 $Bel_{Trust_CBA}(t)$以及相应的上下文约束.通过这些规则,可以构建一个树状推理过程.

这里假定 Agent 采用的规则均采用 Horn 子句的形式.在树形推理结构中根节点是信任对应的表达式,其子节点是信任定义中的信念.每个节点对应一个命题 $prop$,父节点对应的命题 $prop_i$ 与子节点对应的命题 $prop_{ij}$ 之间有如下关系:$prop_{i1}\wedge prop_{i2}\wedge\cdots\wedge prop_{ij}\wedge\cdots\wedge prop_{in}\rightarrow prop_i$.对于 $prop_{ij}$ 所对应的子节点同样可以根据规则 $prop_{ij1}\wedge prop_{ij2}\wedge\cdots\wedge prop_{ijn}\rightarrow prop_{ij}$ 进一步扩展,不断递归这个过程直至没有新的规则可以加入到树状结构中.这样就建立起一个关于信任的推理过程.建立 $Trust(X,Y,Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t_{Trust})$的过程中可能对应多个树状推理结构.这个过程的建立可以考虑采用反向链接算法来实现.

但是前面谈到过,信任伴随着不确定的因素.在针对上面所建立的树状推理结构中,往往会因为上下文的变化,影响到某些节点上所对应的命题 $prop_i$ 的真值.针对 $prop_i$,在 X 的推理规则中可能存在如下规则:$prop_{i1}\wedge prop_{i2}\wedge\cdots\wedge prop_{ij}\wedge\cdots\wedge prop_{in}\wedge prop_i\rightarrow\perp$,或者 $prop'_{i1}\wedge prop'_{i2}\wedge\cdots\wedge prop'_{ij}\wedge\cdots\wedge prop'_{in}\rightarrow prop'_i$,而此时 X 相信 $CONF(prop_i,prop'_i,t_i,t'_i,C_i,C'_i)$.这里元谓词 $CONF$[15] 意味着在时刻 t_i 成立的 $prop_i$ 与时刻 t'_i 成立的 $prop'_i$ 存在冲突.这些因素的出现会导致关于信任的推理不成立.因而这些因素可以看作是对与各个节点所对应命题的上下文约束,进而可以看作是对于信任是否成立的上下文约束.在针对信任的推理过程中纳入更多这样的约束,可以使得信任的建立更加可靠.发挥约束作用的规则在这里称为约束性规则.因而,需要对上面介绍的树状推理过程做进一步扩充.首先,按照如上过程建立树状推理结构,而后让树状结构中对应的各个节点均对应一个二元组$\langle prop,C_{prop}\rangle$,其中 $prop$ 是节点对应的命题,C_{prop} 则是对于 $prop$ 成立与否的一些约束:

$prop_1\wedge prop_2\wedge\cdots\wedge prop_i\wedge\cdots\wedge prop_n\in C_{prop}$,if

$prop_1\wedge prop_2\wedge\cdots\wedge prop_i\wedge\cdots\wedge prop_n\wedge prop\rightarrow\perp$

or

$(prop'_1\wedge prop'_2\wedge\cdots\wedge prop'_i\wedge\cdots\wedge prop'_n\rightarrow prop'$, and $Bel(X,CONF(prop,prop',t,t',C,C'),t)$.

对于纳入到 C_{prop} 中的 $prop_1\wedge prop_2\wedge\cdots\wedge prop_i\wedge\cdots\wedge prop_n$,每个命题 $prop_i$ 通过虚线与 $prop$ 联系起来,最终构成如图 1 所示的推理过程.

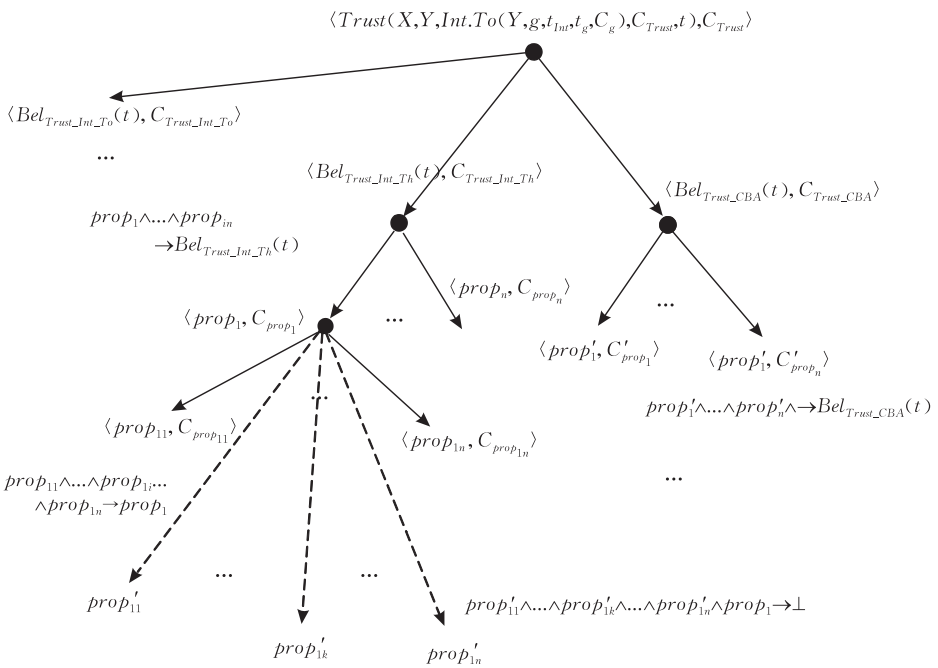

图 1　包含约束规则的信任推理过程

对于存在约束的节点,通过约束得到的真值可能与通过实线连接对应的规则得到的真值发生冲突,此时 Agent 需要进行一致化(reconciliation)的过程,即定义规则决定何种情况下选择何种真值.比如施信方偏向于冒险,则选择节点对应命题为真的情形,如果偏向保守,则倾向于选择对应命题为假的

情形.

通过上面的过程,*ConstructTrust* 动作完成了关于信任 $Trust(X,Y,Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t_{Trust})$的推理过程的构建,这个推理过程即基于控制论的信任框架中的状态空间.此时,对于树状推理结构中的每个命题,X 均了解其真值,那么是否与 Y 建立信任就可以通过推理得出.然而,很多情况下由于信任中涉及的不确定性因素,推理中往往存在一些命题,特别是关于受信方 Y 的一些情况,X 不了解其真值(不知道真值,或怀疑与 Y 的信念不一致,即 X 持有某些信念,这些信念与 X 相信"Y 持有某信念"有冲突,故而 X 没有"Y 持有某信念"的信念."Y 持有某信念"可以考虑作为输入变量,真值为 unknown),需要从受信方那里收集情况,根据收集到的情况完成针对信任的推理.这些 X 所不了解真值的命题可以考虑作为输入变量.此外 X 在推理过程中也可以选择一些 X 相信有助于 Y 得到输入变量相关信息的命题作为输出变量.这里需要假定的是,所确定的输出变量是 X 认为 Y 能够给予回复的并且 X 认为有必要向 Y 需求答案的命题.

在 X 的输入、输出变量确定完成,并将其放在 *TrustAct* 中指定的 *inputVar*、*outputVar* 中后(*outputVar* 包含 *inputVar*,因为要发送给 Y),*ConstructTrust* 完成了主要的工作.而后 X 将请求 Y 执行 *TrustResponse* 动作,随着请求 Y 接受到 *outputVar* 作为自己的输入 $inputVar'$,其具体定义如下.

定义 4.

$TrustResponse(Y,X,inputVar',outputVar',g,t,t_{Trust})=$

$\forall var=(prop,true\_value)\in outputVar'$,

$Inform(Y,X,\varepsilon,Bel(Y,prop,t),t,t_{Trust})$, iff $true\_value=$ true

$Inform(Y,X,\varepsilon,Bel(Y,\neg prop,t),t,t_{Trust})$,

iff $true\_value=$ false

$Inform(Y,X,\varepsilon,\neg Bel(Y,prop,t)\wedge\neg Bel(Y,\neg prop,t),t,t_{Trust})$, iff $true\_value=$ unkonwn.

TrustResponse 的作用在于:Y 得到的关于输入变量集合 $inputVar'$ 后,对其中包含的 X 请求的输入变量在时刻 t_{Trust} 向 X 作出回复.

在 Y 接受到对应的请求后,形成有关于 *TrustResponse* 的意图,比如:

$Bel(Y,\exists t_{past}<t\ Int.Th(X,Int.To(Y,TrustResponse,t,t_{Res},C_{TrustRes}),t_{past},t,C_{TrustRes}),t)\rightarrow$

$Pot.Int.to(Y,\alpha,t,t_{exeR},C'_{TrustRes})$,

其中 $t<t_{exeR}<t_{Res}<t_{Trust}$.

其中 *TrustResponse* 动作作为动作 α 的一部分.此外 α 还负责获得输入变量中各个命题的真值,并将这些真值放在 $outputVar'$ 中.这些真值可以是通过查找 Y 自身的信念或者通过推理得出.当然,Y 可以选择忽视 X 的请求,但是这样的选择会影响到 X 对于 Y 信任的建立.

Y 在接受到 X 发送过来的输出变量后,需要根据输出变量中表达的 X 的信念等来更新自己的信念、意图等.此外,在 Y 发送回输入变量相关的真值后,X 首先更新自己的信念、意图等等.在双方更新自己信念的过程中会涉及到信念修正(Belief revision)的问题.这里可以采用一些已经提出的信念修正算法[17-18].由于信念修正是一个复杂的过程,将在以后的工作中进行讨论,这里假定双方对于接受到的信念直接转化为自身的信念.

3.4 模糊推理与认知推理结合的信任建立

在 X 接收到 Y 发送过来的输入变量,并更新自身的状态后,X 可以根据输入变量的相关信息,完成图 1 所示的关于信任的推理过程,最终决定是否建立信任.在 X 完成关于信任推理的过程中,完全采用真假二值来确定信任是否成立可能会缺乏一定的灵活性.例如假定在代理情形下,X 有如下两条规则:

$$prop_1\wedge prop_2\wedge\cdots\wedge prop_n\rightarrow Bel_{Trust\_Int}(t),$$

$$prop'_1\wedge prop'_2\wedge\cdots\wedge prop'_n\rightarrow Bel_{Trust\_Int}(t).$$

前一条规则的前提比较难以满足,但是得出的结论"更可靠",后一条前提相对容易满足,在前一条规则无法满足的情况下,可以考虑后一条规则,但是得出关于 $Bel_{Trust_Int}(t)$结论"不如前一条规则可靠".在基于博弈论的信任建立过程中往往存在这样有一定模糊性的描述,而完全采用二值逻辑的推理缺乏区分这些模糊表达的能力.而基于博弈论的方式通过数值表达的方式则相对灵活.因而在信任推理树的基础上,这里提出的信任建立过程中也引入数值方式表示各个表达式的可靠性.这种情况下可以考虑在如图 1 所示的基于认知的推理过程的基础上,结合模糊推理来完成信任的建立.

由于要在基于认知的信任建立过程的基础上结合模糊推理的过程,两种推理的过程之间要有一定相似性,即信任推理树中的规则与模糊推理规则之间有一定对应关系.这里采用文献[19]中利用模糊推理建立信任时采用的模糊推理过程.首先定义如下语言变量$(x,U,W(X),G,M)$来描述规则以及表达式的可靠性:其中 x 代表模糊变量的名字;$U=[0,5]$,是模糊变量的可靠性取值范围,采取已有的方法,可靠性通过 0～5 之间的实数值来表示;

$W(X)=\{weak\_reliable, reliable, strong\_reliable\}$ 是一组术语的集合,术语用以表达规则或表达式的可靠性,*weak_reliable* 用来表达“不太可靠”,*reliable* 表示“可靠”,而 *strong_reliable* 表示“非常可靠”;G 是为 x 生成语言值的语法规则的集合,这些规则取决于信任推理树中表达的规则;M 是 x 的语义规则,将可靠性取值与语言值对应起来,通过如下的隶属度函数表达,根据隶属度函数计算得到的模糊值反映可靠性在多大程度上属于“不太可靠”等语言值:

$$M(weak\_reliable)=[weak\_reliable],$$

$$[weak\_reliable](u)=\begin{cases}-\frac{1}{2.5}u+1, & u\in[0,2.5]\\ 0, & u\in[2.5,5]\end{cases},$$

$$M(reliable)=[reliable],$$

$$[reliable](u)=\begin{cases}\frac{1}{2.5}u, & u\in[0,2.5]\\ -\frac{1}{2.5}u+2, & u\in[2.5,5]\end{cases},$$

$$M(strong\_reliable)=[strong\_reliable],$$

$$[strong\_reliable](u)=\begin{cases}0, & u\in[0,2.5]\\ \frac{1}{2.5}u-1, & u\in[2.5,5]\end{cases}.$$

在 X 接收到 Y 所发送过来的输入变量后,对每个输入变量赋予一个可靠性值.按照语言变量的定义,可靠性值处于 0～5 之间,数值越大表明可靠性越高.这个赋值受到很多因素的影响(例如来自于具体场景中 X 对上下文的观察和他可能持有的经验),这时这些可靠性值就可以结合特定的博弈论模型来计算.具体的结合方式将在今后的研究中逐步加入.

模糊推理规则都是基于信任推理树中基于认知的推理规则.首先定义一些模糊规则中需要的表达.针对一些输入变量:

$\langle Bel(Y, prop, t'), weak\_reliable\rangle$ 表示输入变量$\langle prop, \text{true}\rangle$“不太可靠”;

$\langle Bel(Y, \neg prop, t'), weak\_reliable\rangle$ 表示输入变量$\langle prop, \text{false}\rangle$“不太可靠”;

$\langle \neg Bel(Y, \neg prop, t') \wedge \neg Bel(Y, prop, t'), weak_reliable\rangle$表示$\langle prop, \text{unknown}\rangle$“不太可靠”.

输入变量“可靠”与“非常可靠”有类似的表达.为了表达方便起见,这里用$\langle prop, weak_reliable\rangle$表示“*prop* 为真”的可靠性通过语言值 *weak_reliable* 来描述(“不太可靠”),$\langle \neg prop, weak_reliable\rangle$表示“$\neg prop$ 为真”的可靠性通过语言值 *weak_reliable* 来描述.*reliable* 与 *weak_reliable* 的情形类似.通过这些表示,可以对信任推理树中所包含的规则进行扩展.

以前面提到的规则 r1:$prop_1 \wedge prop_2 \wedge \cdots \wedge prop_n \rightarrow Bel_{Trust_Int_Th}(t)$为例,在这个规则的基础上可以定义如下模糊规则:

r1_f1:$\langle prop_1, strong\_reliable\rangle$ and $\langle prop_2, strong\_reliable\rangle$ and … and $\langle prop_n, strong\_reliable\rangle$ then $\langle Bel_{Trust\_Int}(t), strong\_reliable\rangle$.

模糊规则 r1_f1 是说对于规则 *r*1 而言,如果前提中每个表达式 $prop_i(1\leq i\leq n)$为真的可靠性都可以用 *strong_reliable*(“非常可靠”)来描述,那么结论 $Bel_{Trust_Int_Th}(t)$为真也可以用 *strong_reliable* 来描述.通过模糊规则的定义,不但将输入变量的可靠性引入到信任建立过程,也考虑到了信任建立中存在的模糊因素.在前面的讨论中,规则 r1 与 r2 能够得到相同的结论,但是 r1 得到的结论“更加可靠”.这种区别可以通过模糊规则的定义来体现.对于规则 r2 可以定义如下模糊规则:

r2_f1:$\langle prop'_1, strong\_reliable\rangle$ and $\langle prop'_2, strong\_reliable\rangle$ and … and $\langle prop'_m, strong\_reliable\rangle$ then $\langle Bel_{Trust\_Int}(t), reliable\rangle$.

模糊规则 r2_f1 是说如果前提中每个表达式 $prop'_i(1\leq i\leq m)$为真的可靠性都可以用 *strong_relaible*(“非常可靠”)来描述,那么结论$Bel_{Trust_Int_Th}(t)$为真可以用 *reliable*(“可靠”)来描述.这样就可以对规则 r1 与 r2 的结果加以区分.

通过这种方法,信任推理树中的每条规则被扩展为一组模糊规则,从推理树中的叶节点开始,通过模糊规则推理出父节点的可靠性值,通过自底向上的方式,在信任推理树的基础上进行模糊推理,模糊推理的过程如图 2 所示.其中,叶节点包括输入变量对应的表达式以及约束性规则的前提中的相关表达式.在图 2 中,对于形如 $prop_1 \wedge prop_2 \wedge \cdots \wedge prop_n \rightarrow prop$,模糊化的过程是指以 $prop_i(1\leq i\leq n)$的可靠值作为输入,通过隶属度函数将可靠性值映射为模糊值,这些语言值和相应的一组模糊规则一起,在模糊推理引擎中进行推理,获得描述 *prop* 可靠性的模糊值.而后 *prop* 的可靠性值还要用于进一步的模糊推理,因而需要将描述 *prop* 可靠性的模糊值再映射为描述 *prop* 可靠性值.从信任推理树的叶节点开始,对于树中的节点,都可以通过其子节点根据规则进行的模糊推理获得一个描述其可靠性的值,而这个值被用来进行针对该节点的父节点的模

糊推理，通过这样自底向上的方式，最终能够得到关于信任的可靠性值. X 为信任能否成立设定一个阈值，就可以根据信任的可靠性值决定是否信任 Y.

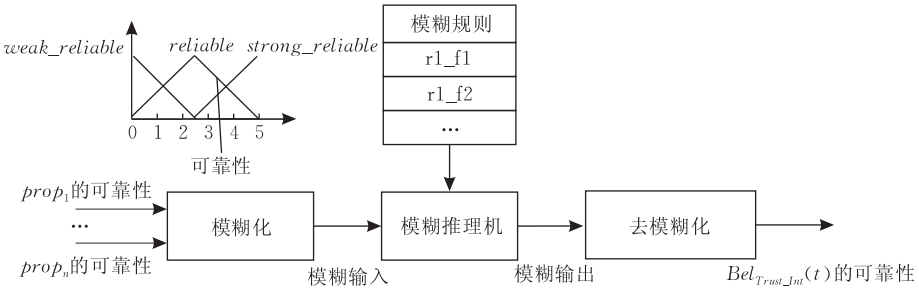

图 2 针对信任推理树中一条规则的模糊推理的过程

4 场景分析

本节中通过一个场景的分析来说明如何利用前一节所提出的框架在 Agent 间建立信任. 假定某家机构 X 提供某项科研课题等待申请，有一家公司 Y 希望能够申请对此课题的研究. 只有当 X 信任 Y 能够按照 X 期望的情况来完成该课题，X 才会同意将课题交给 Y 去研究. 而此时在 X 不熟悉 Y，而且很难获得 Y 的一些相关信息的情况下(比如 Y 是刚刚创业的公司)，那么 X 与 Y 如何建立信任?一般来说，Y 需要就 X 所提出的一些问题进行答辩. 此时，X 根据建立 trust 的需要主动提出问题要求 Y 进行回答.

首先对场景做一个详细的说明. X 所要达成的目标是“按照合同书的要求完成各项内容”，这里用 g 表示. 此时 X 要将课题交给提出申请的 Y，X 对 Y 要建立如下所示的信任：

$Trust(X,Y,Int.To(Y,g,t_{Int},t_g,C_g),C_{Trust},t)$，

即 X 信任 Y 会在时刻 t_{Int} 意图于在 t_g 时刻按照合同完成项目. 此时假定在上下文约束 C_{Trust} 中包含有两类约束：有一个比较完善的实施方案，有一个合理的人员配置.

关于实施方案的约束包括如下内容：对于目标 g，$Prob(g)=\{prob_1,prob_2,\cdots,prob_n\}$ 是所要解决的问题的集合，而 $Tech(g)=\{(prob_1,tech_1),(prob_1,tech_2),\cdots,(prob_n,tech_n)\}$ 是 Y 计划用来解决 $Prob$ 中的问题的技术的集合. 此时上下文约束中可能需要考虑如下的一些约束：对于 $prob_i$ Y 打算采用什么样的 $tech_i$ 来解决?$tech_i$ 是已有技术还是针对 $prob_i$ 提出来的新技术?如果是已有技术，那么是在什么文献中被提出来的，最初用于解决什么问题，在解决 $prob_i$ 时需要有什么改动?如果是针对 $prob_i$ 提出来的，是否存在类似的技术，他们之间有什么差别? 如果发现 $tech_i$ 不能解决 $prob_i$，有什么备选方案?

在人员安排的约束方面需要考虑的约束有：参与人员的数量以及基本信息，包括研究领域、发表的论文、以前参与的项目等等，此外还有参与人员的变动，比如退休等如何影响人事安排? 对于所采用的技术 $tech_i$，参与人员的熟悉情况如何等.

假定 X 面对 Y 的申请，有潜在的意图让 Y 承接项目并达成目标 g. 根据假定 1 以及规则 1，在 X 有意让 Y 完成项目的时候，X 需要建立对于 Y 的信任，并由此形成关于 $TrustAct$ 的意图，最终在推理得出信任成立的前提下，才能将 g 代理给 Y. 首先 X 需要建立针对信任的推理过程，并确定输入、输出变量.

根据这些上下文约束的条件，假定 X 有如下规则：

r_X1：

$Bel(X,ProperDoc(Y,g),t)\rightarrow$

$Bel(X,\exists t'_{Int}<t_{Int}Pot.Int.To(Y,g,t'_{Int},t_g,C_g\wedge C'),t)$，

$C'=\forall Z\neq Y\,\forall t''<t_g\rightarrow Int.Th(X,Int.To(Z,g,t_{Int},t_g,C_g),t'',t_{Int},C_g)$.

规则 r_X1 是说在当前时刻 t，如果 X 相信 Y 为达成目标 g 准备了完善的文档(项目申请书等)，那么 X 相信在上下文约束 C_g 与 C' 下 Y 在 t_{Int} 之前的某个时刻形成关于目标 g 的潜在意图. 谓词 $ProperDoc(Y,g)$ 代表 Y 为目标 g 准备了完善的文档. C' 是说除了 Y 以外，X 不会让其它 Agent 来代理完成目标 g. 其它规则与涉及的谓词不再详细说明，参见表 1 与表 2. 其中 r_X4、r_X5、r_X7 起着约束性规则的作用.

通过这些规则可以构建关于信任的推理过程，首先，根据规则 r_X1、r_X2、r_X3、r_X6，X 可以构建出一棵与图 1 类似的推理树：规则的结论做为父节点，与前提中 $\wedge$ 连接的各个命题之间以实线连接，

表 1　场景中涉及的规则

规则名	规则的含义
r_X2	当前时刻 t, X 相信 Y 准备了详细的申请文档,并针对课题中的一些问题提出了新的技术,参与项目的人员应该有充足的时间,这时 X 相信 Y 为目标 g 进行了充分的准备.
r_X3	当前时刻 t,X 相信 Y 有关于 g 的潜在意图,并且相信 Y 对于 g 进行了充分准备,那么 X 形成信念 $Bel_{Trust\_Int\_Th}(t)$.
r_X4	当前时刻 t,X 相信可能有某些参与 g 的研究人员 $researcher_i$ 会出国学习,那么 X 认为有研究人员没有充足时间参与 g.
r_X5	当前时刻 t,X 相信有某些 Y 提出的用以解决问题 $prob_i$ 的技术 $tech_i$ 代价比较高,那么 X 相信 Y 在达成目标 g 的过程中不能用 $tech_i$ 解决 $prob_i$. 此外 X 相信有某些 Y 提出的用以解决问题 $prob_i$ 的技术 $tech_i$ 难以掌握,那么 X 相信 Y 在达成目标 g 的过程中不能用 $tech_i$ 解决 $prob_i$.
r_X6	当前时刻 t,X 相信对于 Y 所提出的针对各个问题 $prob_i$ 的技术 $tech_i$,如果都能被 Y 用以解决对应的问题,那么 X 认为 Y 有一个用以达成 g 的 $recipe$ R_g,对于中的 R_g 的各个动作 α,Y 会在动作应该执行的时刻 t_α 之前某个时刻 t' 形成关于 α 的意图.
r_X7	当前时刻 t,X 相信有某些 Y 所提出的针对各个问题 $prob_i$ 的技术 $tech_i$ 是成熟的技术,但以前未被用以解决问题 $prob_i$,如果 Y 没有对做出修改,那么 X 相信某些问题 $prob_i$ 不能被 Y 提出的 $tech_i$ 解决.

表 2　场景中涉及的谓词

谓词	含义	涉及的规则
$ProperDoc(Y,g)$	Y 为目标 g 准备了完善的文档	r_X1, r_X2
$NewTech(Y,tech_i,prob_i)$	Y 针对问题 $prob_i$ 提出新技术 $tech_i$	r_X2, r_X5
$SloveProblem(Y,tech_i,prob_i)$	Y 利用 $tech_i$ 来解决问题 $prob_i$	r_X2, r_X5, r_X6
$Participant(Y,researcher_i,g)$	$researcher_i$ 是 Y 安排参与 g 的参与人员	r_X2, r_X4
$EnoughTime(researcher_i,g)$	参与者 $researcher_i$ 有充足时间参与 g	r_X2, r_X4
$WellPrepared(Y,g)$	Y 为 g 进行了充分准备	r_X3
$WillStudyAbroad(researcher_i)$	研究人员可能会出国学习	r_X4
$HighCost(tech_i)$	技术 $tech_i$ 代价高昂	r_X5
$HardToMastery(tech_i)$	技术 $tech_i$ 难于掌握	r_X5
$OldTech(tech_i)$	$tech_i$ 是现有技术	r_X7
$UsedBefore(tech_i,prob_i)$	$tech_i$ 以前曾被用以解决问题 $prob_i$	r_X7
$ChangeFor(Y,tech_i,prob_i)$	Y 针对 $prob_i$ 对 $tech_i$ 进行了修改	r_X7

$Trust(X,Y,Int.To(Y,g,t_{Int},t_g,C_g),t)$ 做为父节点,与 r_X1、r_X3、r_X6 的结论之间以实线连接. 此外鉴于 $Bel(X,\exists researcher_i(Participant(Y,researcher_i,g)\wedge WillStudyAbroad(researcher_i)),t)$ 与 $Bel(X,\forall research_i(Participant(Y,researcher_i,g)\wedge EnoughTime(researcher_i,g)),t)$(记为 $prop$)之间存在冲突,因而在 $prop$ 对应的节点〈$prop$,C_{prop}〉上,上下文约束 C_{prop} 中应该包括. $Bel(X,\exists researcher_i(Participant(Y,researcher_i,g)\wedge WillStudyAbroad(researcher_i)),t)$ 类似地由规则 x_R5、x_R7 得到相关信念的约束,最终得到如图 3 所示的信任推理过程(为展示方便,图中忽略了规则 x_R1 以及 $Trust$ 与 $Bel_{Trust\_Int\_To}(t)$ 的相关分支,部分节点忽略其约束上下文,部分节点忽略信念算子 Bel).

在图 3 所示的推理树当中 X 首先确定输入变量,随着 X 请求($Request$)Y 执行 $TrustResponse$ 的动作发送给 Y. Y 在接受到 X 的请求和输出变量 $Y_{trstor}(t)$(Y 自己的输入变量)后,首先根据接受到的输入变量更新自身的状态. 如果 Y 决定就 X 的请求做出回复,那么 Y 首先通过查找自身信念或通过推理,获得 X 所请求的输入变量的真值(true、false、unknown),并将其放在输出变量 $Y_{trustee}(t)$ 中,进而执行 $TrustResponse$. X 在接受到这些变量的真值后,更新图 3 所示的推理过程 ,最终决定信任是否建立. 这里模糊推理的过程鉴于篇幅不再详细给出.

5　结　语

本文首先从认知的角度出发给出代理与非代理情形下的信任定义,而后给出了一个 Agent 间预动的建立信任的信任框架. 在这个框架中,Agent 被看作是具有内部状态、输入以及输出的系统;同时,通过受控系统的状态空间方程来描述 Agent 间为了建立信任进行的通信过程:首先,施信方建立针对信任的推理过程,而后就推理过程中缺乏的信息与受信方进行交互,最终结合模糊推理完成针对信任的推理并决定是否建立信任. 该框架主要针对在 MAS 中,施信方 Agent 在缺乏与受信方的交互历史或关于受信方的第三方证言的情形下,与受信方建立信任存在困难情况. 此外,框架在推理的过程中可以纳入 Agent 运行中复杂的上下文变化,避免了基于博弈论的信任模型中采用预先定义的效用函数而缺乏灵活性的问题.

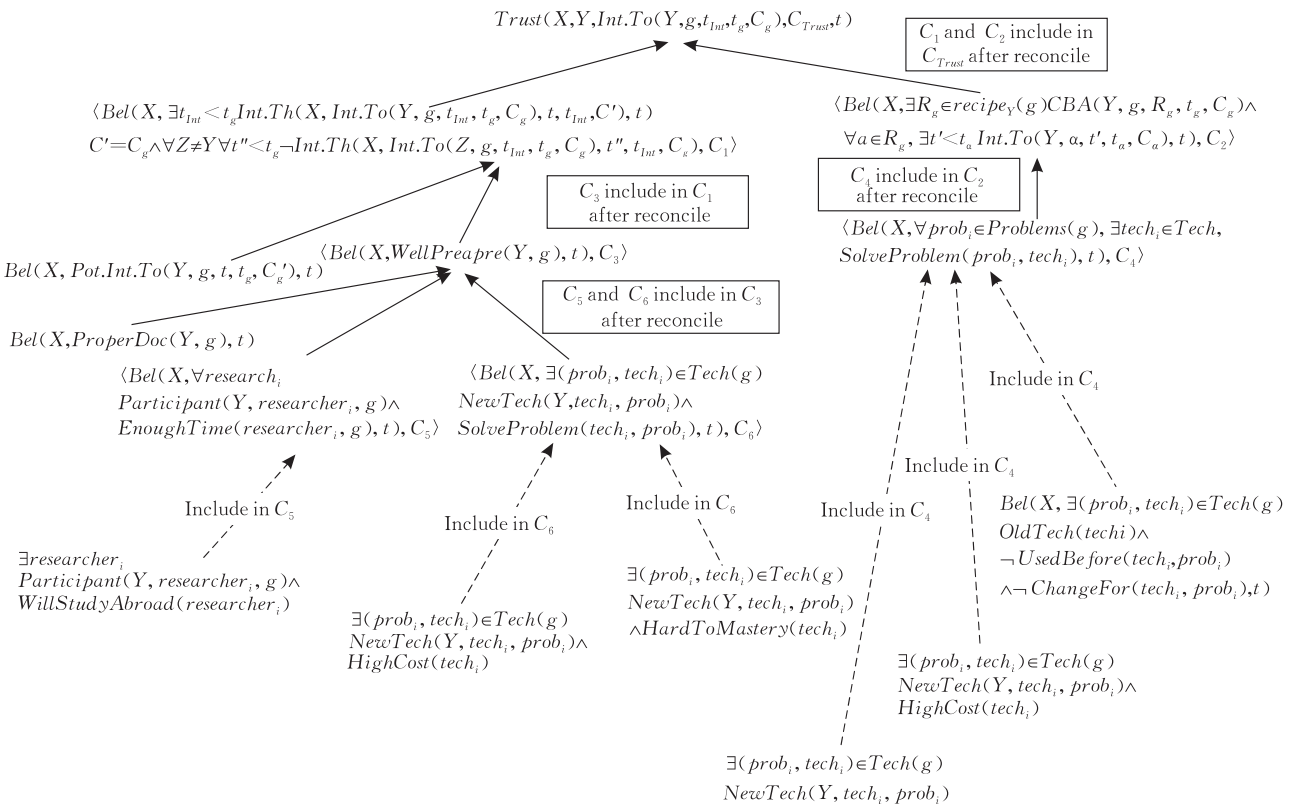

图 3 场景的信任推理过程

在本文提出的框架中还有一些问题需要进一步研究. 首先,尽管采用基于认知的推理可以对复杂的上下文变化进行推理,但是在关于信任的推理规则,以及交互中消息的可靠性方面,会涉及模糊的因素,如何通过模糊推理来与基于博弈的信任模型进行结合以及模糊推理中去模糊化的过程,对于施信方与受信方之间交互中涉及的信念更新问题,还需进一步研究.

参 考 文 献

[1] Jordi Saraber, Carles Sieraa. Review on computational trust and reputation models. Artificial Intelligence Review, 2005, 24(1): 33-60

[2] Xu Guang-Quan, Feng Zhi-Yong, Wu Hua-Bei, Zhao De-Xin. Swift trust in a virtual temporary system: A model based on the dempster-shafer theory of belief functions. International Journal of Electronic Commerce, 2007, 12(1): 93-126

[3] Marsh S. Formalizing trust as a computational concept [Ph. D. dissertation]. UK: University of Stirling, 1994

[4] Sabater J, Sierra C. Reputation and social network analysis in multi-agent systems//Proceedings of the AAMAS-02. Bologna, Italy, 2002: 475-482

[5] Wang Ping, Zhang Zi-Li. Dynamic trust processing in multi-agent system. Computer Science, 2005, 32(3): 182-185(in Chinese)
(王平,张自力. 多 Agent 系统中信任的动态性处理. 计算机科学, 2005, 32(3): 182-185)

[6] Liu Feng-Ming, Lu Xing-Jia, Ding Yong-Sheng. Computation model for trustworthiness in P2P network service environment. Computer Applications. 2008, 28(3): 623-625 (in Chinese)
(刘凤鸣, 陆星家,丁永生. 一种 P2P 网络服务环境的信任度计算模型. 计算机应用, 2008, 28(3): 623-625)

[7] Zhao Shu-Liang, Jiang Guo-Rui, Huang Ti-Yun. Trust model of Multi-agent System. Journal of Managemnt Sciences in China, 2006, 9(5): 36-43(in Chinese)
(赵书良, 蒋国瑞, 黄梯云. 一种 Multi-agent System 的信任模型. 管理科学学报, 2006, 9(5): 36-43)

[8] Rino Falcone, Cristiano Castelfranchi. Social trust: A cognitive approach//Christiano Castelfranchi, Tan Yao-Hua eds. Trust and deception in virtual societies. Norwell, MA, USA: Kluwer Academic Publishers, 2001: 55-90

[9] Wooldridge M, Jennings N R. Intelligent agents: Theory and practice. The Knowledge Engineering Review, 1995, 10(2): 115-152

[10] Wooldridge, Michael. The logical modelling of computational multi-agent systems [Ph. D. dissertation]. University of Manchester, Manchester, UK, 1992

[11] Grosz B, Kraus S. The evolution of SharedPlans//W M Rao A eds. Foundations and Theories of Rational Agency. New York: Springer, 1998: 227-262

[12] Grosz B J, Kraus S. Collaborative plans for complex group action. Artificial Intelligence, 1996, 86(2): 269-375

[13] Levesque H J, Cohen P R, Nunes J T H. On acting together//Proceedings of the AAAI-1990. Boston, Massachusetts: AAAI Press, 1990: 94-99

[14] Cohen P, Levesque H. Teamwork. Nous, 1991, 25(4): 487-512

[15] Fan Xiao-Cong, Yen J, Volz R A. A theoretical framework on proactive information exchange in agent teamwork. Artificial Intelligence, 2005, 169(1): 23-97

[16] Lewicki R J, Bunker B B. Developing and maintaining trust in work relationships//Dramer R M, Tyler T R eds. Trust in Organizations: Frontiers of Theory and Research. Thousand Oaks, CA: Sage Publications, 1996: 119-143

[17] Wassermann, Renata. An algorithm for belief revision//Proceedings of the KR2000, 2000: 345-352

[18] Jin Yin, Thielscher M, Zhang Dong-Mo. Mutual belief revision: Semantics and computation//Proceedings of the AAAI-2007. Vancouver, British Columbia, Canada: AAAI Press, 2007: 440-445

[19] Schmidt S, Steele R, Dillon T S et al. Fuzzy trust evaluation and credibility development in multi-agent systems. Applied Soft Computing, 2007, 7(2): 492-505

WANG Jia-Fang, born in 1982, Ph. D.. His research interests include intelligent agent and multi-agent system.

FENG Zhi-Yong, born in 1964, Ph. D., professor, Ph. D. supervisor. His research interests include knowledge engineering, service-oriented computing and middleware.

XU Chao, born in 1982, Ph. D. candidate. His research interests include service-oriented computing and affective computing.

XU Guang-Quan, born in 1979, Ph. D.. His research interests include trust and multi-agent system.

Background

The work presents in this paper belongs to the field of Multi-Agent System and Computational Trust in virtual society. Many people have proposed trust model between agents. Most of these models adopt the game theory view, and trust is established between agents based on the information of interaction history or witness information. These game theory based models use pre-defined utility functions to calculate trust values. Although they introduce simple ways to establishment trust, pre-defined utilities can hardly reflect the dynamic context changes that agents may face. Besides, researches on temporary system show that information of interaction history and witness are not always available. This paper proposes an approach of establishing trust in a proactive manner. The approach adopts a cognitive view of trust. Then the dynamic context change can be reflected in the reasoning process of trust establishment. And when information of interaction history and witness are not available, trust can be still established by proactive direct interaction between agents. In the direct interaction, the trustor can collect what information that he need to form the impression on the trustee. Considering the reliability of information in communication, the trustor conduct on a fuzzy reasoning process on the basis of cognitive reasoning after information received. Then the results that got in the game theory based models can be combined with the cognitive approach, and trust can be establish without interference from used.

The research proposed in this paper is part of work of project "Solution-oriented service infrastructure and support environment" which is supported by a grant from the National High Technology Research and Development Program of China under grant of 2007AA01Z130. As Service Oriented Architecture is developed and many services exist on the internet, the project is about to establish a service network as a basis structure and composite different services on the internet to provide the new service that the service consumers need. In the process of service composition, trust needed to be established between service provider and service consumers. Then agents stand for different entities can proactively established trust between them.

On the area of trust between agents, the lab has published the paper: Guangquan Xu, Zhiyong Feng, Huabei Wu, and Dexin Zhao. Swift Trust in a Virtual Temporary System: A Model Based on the Dempster-Shafer Theory of Belief Functions. International Journal of Electronic Commerce, 2007, 12(1): 93-126. In this paper, a swift trust model was proposed in which the mechanism of swift trust is realized by way of layered reasoning. This model is used to establish swift trust in the context of temporary system.