

基于本体的决策模型冲突预测及协商研究^{*})

张 波 向 阳

(同济大学电子信息工程学院 上海 201804)

(同济大学嵌入式系统与服务计算教育部重点实验室 上海 201804)

摘 要 依据模型知识的特性,定义了模型描述本体和任务求解本体,为模型提供语义支持。模型语义分为描述语义和行为语义。基于描述语义的相似性判定,模型可进行潜在冲突预测;然后根据任务求解本体的定义,模型通过行为语义交互,进行行为协商;而在执行过程中,模型需要为每一个操作申请资源,因此模型根据模型描述本体和描述语义对资源申请进行协商,从而得到互不冲突的操作执行序列,消除冲突。最后通过实验分析验证算法的有效性。

关键词 模型,本体,描述语义,行为语义,冲突预测,协商

Collision Detection and Negotiation of Decision Models Based on Ontology

ZHANG Bo XIANG Yang

(School of Electronic Information and Engineering, Tongji University, Shanghai 201804, China)

(Key Laboratory of Embedded System and Service Computing, Ministry of Education of the People's Republic of China, Tongji University, Shanghai 201804, China)

Abstract Model description ontology and task resolving ontology are defined to offer the semantic supports for models, according to the characteristics of knowledge of model. Model's semantics are defined as description semantics and action semantics. Models can detect the potential collisions through the judgment of description semantics similarity; then, according to the definition of task resolving ontology, models negotiate the actions each other based on the action semantics. In the course of implementation, model requires the resources for every action. So models negotiate for the application of resources based on the model description ontology and the description semantics, and then get the action lists without collision.

Keywords Model, Ontology, Description semantics, Action semantics, Collision detection, Negotiation

1 引言

模型是决策支持系统中问题求解的重要组成部分。当模型发生冲突时,可能会导致系统效率下降,甚至问题求解失败。模型的冲突预测以及协商机制一直是计算机执行任务时需要解决的基本问题,其主要目标是能够自动地发现将要发生或者已经发生的冲突,并且能够通过自主的交互策略进行相互协商,从而解决冲突。在实际应用过程中,模型行为一般会受到其它模型行为的影响,或者被不断变化的动态环境所限制。这些影响和限制都将对模型的执行时间、空间以及执行效率产生作用,从而产生各种复杂的问题或者难以挽回的损失。

对于模型的冲突预测以及协商,国内外过去曾经出现过很多研究方法,其中大多是基于特定的兴趣以及模型的应用领域。Lin Ming C 利用层次表示的面对面快速算法,实现动态空间中的多面体几何模型冲突预测^[1,2]; Michael Baker 通过解释“teacher-student”和“learner-learner”之间的对话关系,利用一组通信行为集和定义好的一组陈述关系规则来进行模型协商^[3,9,10]。R. Kakehi 等提出一种交互式的协商协议,用来在多 Agent 环境中进行资源重用协商^[4]。Madhav Ponamgi 提出一种增量算法,用来在动态环境中进行模型冲突预测,该增量算法结合一种层次描述方法,在框架到框架的增量中计算冲突^[5]。协商研究方面, Peyman Faratin 和 Carles Sierra

提出了一个形式化模型在自治 Agent 之间进行协商。该模型在机器学习的基础上进行假设计算,从而评估 Agent 的计划,达到为 Agent 提供策略以及方法的目的^[6]。在国内,何炎祥和陈莘萌提出多 Agent 系统中拥有不同目标的 Agent 进行交互式资源协商,在基于系统局部知识的情况下达到解决资源冲突问题的目的^[11,12]。

传统方法无论是冲突预测算法还是协商策略,要么是严格地面向某一类特定的模型,要么缺乏足够的领域知识支持或者缺乏形式化的知识描述手段。冲突预测以及协商过程中无法提供准确的语义描述。本文通过定义模型的语义基础,解决模型之间无法“理解”对方的行为和意图等问题,利用本体对模型进行形式化描述,使之实现机器可理解。进而在模型的整个执行过程中通过冲突预测和协商算法的研究,使模型之间在语义层次上解决冲突成为可能。

2 本体及语义

2.1 本体

本体概念最初起源于哲学领域,目前已经被广泛应用在知识工程、自然语言处理、协作式信息系统和知识管理的多个领域。对于本体, Gruber 将其定义为“本体是概念模型的明确的规范说明”^[7]。本体可以为某一领域描述指定领域知识,给出领域内对象、实体和概念,以及它们之间关系和活动的规范性、明确性和一致性的形式化描述。

^{*})国家自然科学基金资助项目(70371054、70771077)。张 波 博士研究生,研究方向为语义网与本体论;向 阳 教授,博士生导师,研究方向为智能决策支持系统。

为了叙述方便,本文进行下列一些定义,并给出相关假设。

定义 1(模型描述本体) 模型描述本体定义为 $DO=(C^M, R, A, P)$ 。其中 C^M 表示模型概念(类), R 表示各类模型之间的关系, A 为各类模型概念之间的规则集合, P 为描述模型的属性集合。模型描述本体用来为每个模型提供描述层次上的语义支持。

模型描述本体将模型分为不同的概念(类),每个具体的模型均为某个概念(类)中的实例。通过本体描述概念之间的关系,为不同的模型之间建立关联。如图 1 所示,每个圆圈代表一个模型概念,各自带有属性集合。图中箭头表示本体中概念之间的关系。

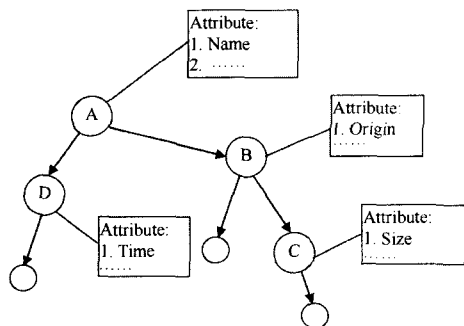


图 1 模型描述本体示例

相似地,我们可以得到任务求解本体的定义:

定义 2(任务求解本体) 任务求解本体 $SO=(C^T, R, A, P)$ 。 C^T 为任务概念; R 表示每个任务之间模型实施的动作,体现任务之间的状态转换; A 表示与动作 R 所关联的语义规则; P 为描述该类任务所处状态的属性集合。

2.2 模型语义

模型在任务执行过程中,语义环境可以反应模型自身的特征、所执行任务的特征、所处的环境、所处行为以及相关行为。模型的语义分为两类:描述语义和行为语义。模型描述语义用来表示与模型本身相关的知识,如模型的类别、属性(包括结构属性、静态方法等)、状态等;行为语义可以表示模型在所采取的行为的动作的类别以及该动作可能引起的关联动作。一般来说,模型的行为可以包括顺序、循环、分支以及调用等几个行为类型。

定义 3(描述语义) 模型描述语义的定义为 $DS=(Mc, Mp, Ms)$ 。 Mc 为模型所属描述本体中的概念(类); Mp 为描述模型的属性集合; Ms 表示模型所处的状态集合。由上述定义可知: $Mc \in DO, C^M, Mp \in DO, P, Ms \in SO, P$ 。

定义 4(行为语义) 模型的行为语义定义为 $BS=(Bc, Br)$ 。 Bc 表示模型采取行为的非空集合; Br 表示关联行为,即该行为所处状态可能导致与其相关的行为集合。由此可知: $Bc \in SO, R, Br \in SO, R$ 。

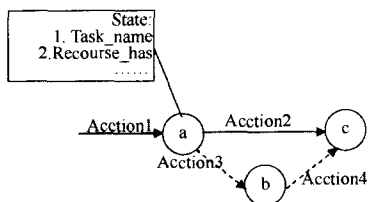


图 2 行为语义

图 2 表示了一个行为语义。行为 $action1$ 的关联行为是

$(action2, action3)$, 而 $(action3, action4)$ 是行为 $action2$ 的可替代行为。

从行为角度而言,任务执行过程是模型由一个任务阶段通过执行某种行为到达另一个任务阶段的求解过程。由图 3 中可以看到模型 A 进行任务求解的一个片断。图中显示了本体与模型之间的对应关系。模型 A 状态在模型描述本体中对应其概念(类),同时任务求解本体对应其任务状态等。通过任务求解本体中一个任务到另一个任务的关系获知其行为。执行该行为后,模型 A 状态转变为任务求解本体中下一个任务状态。

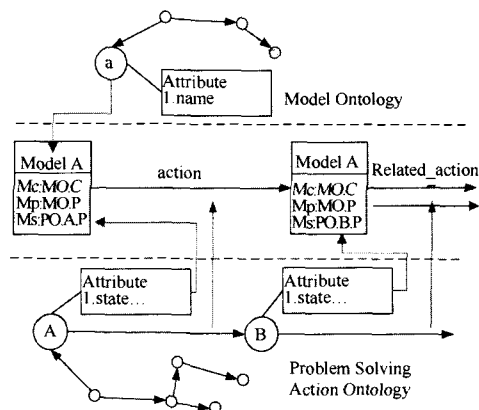


图 3 模型求解过程示例

定义 5(模型冲突) 模型的冲突是指在出现下列任意一种情况的时候所导致的模型任务无法继续执行:

- 多个模型同时执行相同的行为而导致对有限资源的拥挤申请和占用;
- 模型申请其他模型占有的资源,而占有资源的模型无法释放,导致死锁;
- 只可有限次使用的资源被消耗完后,模型无法获取替代资源而导致无限期待。

模型的任务执行质量和效果直接与时间、效率等相联系。一旦出现冲突再转而进行处理,可能导致死锁等情况的出现。这时模型不可能恢复到初始状态重新对任务求解。在执行任务以前,模型对将要进行的任务求解过程进行预测,并且进行冲突协调,这种方法显然能够更加可靠地避免出现不可恢复状态。

本文假设以下的前提:

假设 1 每个模型均由模型描述本体描述;执行的任务都按照任务求解本体进行操作;

假设 2 每一个模型可以根据情况动态地决定其执行计划的过程;

假设 3 每一个模型在开始时不一定了解其他模型的目标和执行任务的过程;

假设 4 所有的模型经过冲突预测以及协商后,得到规划好的操作序列后执行任务;

假设 5 同类模型可能存在相同行为,申请同一资源。

模型之间基于上面所设定的假设进行冲突预测以及计划协商,通过交换和评估彼此之间的任务执行过程,模型之间达成协议,制定出互不干扰且具有相对最高效率的计划。

3 基于描述语义的模型相似性

模型的结构、任务状态、求解行为、求解所需的资源等各方面存在相似的可能性。相似的模型执行任务的时候,很可

能由于求解行为相同而对同一类型的同一资源进行申请。当这一资源的访问权限具有独占性时,潜在的冲突就可能发生。因此,模型的冲突很大一部分可能是因为两个结构相似的模型同时执行任务时所导致的资源使用、权限访问等方面的冲突引起的。基于上述原因,我们认为需要对相似模型进行判别,以便对它们进行处理。

模型描述本体为模型提供了良好的知识表示,这就使描述语义基础上的相似性检测成为可能性。在此基础上,我们引入多层递归算法^[8],实现对模型描述语义的相似性进行判定。

在多层递归相似性判定算法中,被预测模型 A 和 B 的描述语义分别为

$$DS^A = (Mc^A, (Mp_0^A, Mp_1^A, \dots, Mp_m^A), (Ms_0^A, Ms_1^A, \dots, Ms_n^A))$$

$$DS^B = (Mc^B, (Mp_0^B, Mp_1^B, \dots, Mp_p^B), (Ms_0^B, Ms_1^B, \dots, Ms_k^B))$$

其中, Mc^X 表示描述模型 X 在模型本体中所属的类; 向量 Mp_i^X 表示模型 X 所拥有的属性集合; 向量 Ms_j^X 表示模型 X 所拥有的状态集合。

$Code(DS, A, DS, B)$ 用来计算两个模型之间的语义向量相似度。模型语义向量进行排序, 当两个模型的语义向量在选定的阈值下相同时, 我们就认为两个模型相似, 否则两个模型不相似。

多层递归相似性判定算法接受按照模型描述本体排序的属性序列, 其相似性计算是在多个层次上针对不同阈值 e_i 以及相似性权重 $Weight_i$ 进行递归计算。该算法试图在两个属性序列中找到最高层次的匹配对。此时输入序列被分为两部分。然后算法继续进行匹配处理, 直到一个部分的序列在较低层次和较小的阈值下递归匹配结果为空。一般来说, 算法中阈值和权重是通过经验或者启发式规则获取的。

算法 1 多层递归相似性判定算法 (MLRM)

输入: 模型 A 和 B 描述语义属性序列

输出: 模型相似性

(1) 对于模型 A 和模型 B 的描述语义中 DS, Mp 序列 $(Mp_0^A, Mp_1^A, \dots, Mp_m^A)$ 和 $(Mp_0^B, Mp_1^B, \dots, Mp_p^B)$ 按照模型描述本体中 DO, P 进行排序, 获得位置对应的排序结果;

(2) 选定层次值 e_i ;

(3) 如果属性描述序列中任何一个为空, 则相似度为 0, 表示模型不相似, 算法结束; 否则继续下一步;

(4) 设置参数 $P=0, Q=0$; 当 $P \leq m$ 且 $Q \leq k$ 时, 执行下列计算

(a) 对于所有 x 从 P 到 m 以及 y 从 Q 到 k , 计算 $Code(Mp_x^A, Mp_y^B)$ 。如果

$Code(Mp_x^A, Mp_y^B) > e_i$, 则相似度 $Sim(A, B) += Code(Mp_x^A, Mp_y^B) \times Weight_i$, 其中 $Weight_i$ 为权重, 在算法开始前选定;

(b) 计算

$$Sim(A, B) +=$$

$$MLRM(Mp_0^A, Mp_1^A, \dots, Mp_{m-1}^A), (Mp_0^B, Mp_1^B, \dots, Mp_{p-1}^B), e_{i+1});$$

(c) 递归计算

$$Sim(A, B) +=$$

$$MLRM(Mp_0^A, Mp_1^A, \dots, Mp_{m-1}^A), (Mp_0^B, Mp_1^B, \dots, Mp_{p-1}^B), e_{i+1});$$

(d) $P=x+1; Q=y+1$;

(5) 再次递归计算

$$Sim(A, B) +=$$

$$MLRM((Mp_0^A, Mp_1^A, \dots, Mp_{m-1}^A), (Mp_0^B, Mp_1^B, \dots, Mp_{p-1}^B), e_{i+1})$$

(6) 输出相似度 $Sim(A, B)$, 算法结束。

4 基于行为语义的协商算法

4.1 符号定义

模型 A 从开始到任务求解完成假设需要 n 个步骤, 那么该计划的操作序列为 $action_0 action_1 action_2 \dots action_n$, 其中 $action_i (i=0 \dots n)$ 代表模型一个动作, 也就是任务求解本体中的一个行为; 其每行动的语义描述为

$$BS = (action_i, (action_j \dots action_k))$$

模型之间通过由本体定义的行为语义展示其计划集来进行协商。本文使用了如下 6 种消息元语^[11]。

$Send_plan(Plan)$: 向另一模型发送其计划集; $Reject(Plan)$: 向另一模型表示其计划集不可接受, 并将自己的计划集发送给它; $Send_value(Plan, EvaluateValue)$: 向另一模型发送一计划集及评估值; $Accept(Plan)$: 向另一模型发送可接受的计划集; $Query_information(State(resource))$: 询问是否存在如此状态的资源; $Inform(Resource)$: 当模型收到一个 $query_information$ 消息时, 如果存在如此状态的资源, 则返回资源名, 否则返回 NIL。

在操作执行的过程中, 模型 A 的状态描述 $Ms(A, n)$ 随着改变为

$$Ms(A, 0) \xrightarrow{action_0} Ms(A, 1) \xrightarrow{action_1} Ms(A, 2) \dots \xrightarrow{action_{n-1}} Ms(A, n)$$

因此模型 A 对于某一个任务的操作序列集记为 $Plan(A)$ 。在协调的时候, $Plan(A)$ 被提供出去进行协商。

4.2 行为协商算法

我们给出如下的协调算法(下面定义 A 为预计到冲突的那个模型, B 为另一个与之协调的模型)。我们首先定义一个行为比较函数 $compare(Plan(A), Plan(B))$, 用来查找行为序列中相同行为。此函数将两个操作序列中的行为进行顺序比较, 找出其中在相同位置上等价的部分。

算法 2 模型行为协商算法 (MANA)

输入: $Plan(A), Plan(B)$

输出: 调整后的行为序列

(1) A 和 B 各自选择随机步数 x, y , 先分别制定步数深度为 x 和 y 的执行计划 $Plan(A)_1$ 和 $Plan(B)_1$;

(2) A 向 B 发送一个 $send_plan(Plan(A)_1)$, 将其计划集提供给 B;

(3) B 用 $compare$ 函数对 $Plan(A)_1$ 和 $Plan(B)_1$ 进行比较, 匹配行为的集合。若 $compare$ 函数不为空, 则存在冲突行为, 继续执行下一步; 否则, 表明行为语义上不存在冲突, 转 (8);

(4) B 得到冲突行为的集合, 向 A 发送消息 $reject(action)$, 表示拒绝包含此类行为序列的计划。A 收到后, 退回到 $action_{i-1}$ 。此时, A 通过模型描述本体定义的描述语义 $DS, Ms(A, i)$ 查找行为语义中 $action_{i-1}$ 的关联行为 $action_{i-1}$ (reaction) 中可以到达目标状态 $DS, Ms(A, n)$ 的所有可替代行为序列, 制定可替代 $action_i, action_{i+1} \dots action_n$ 的新计划。

(5) 执行新计划后模型所处状态称为后状态 $PostState$ 。A 在制定一个从 $action_{i-1}$ 后状态到其目标的新计划, 并比较这个新计划与其原计划所包含的操作数, 从而对新计划评估。本文运用文献^[8]中定义的评估函数 $evaluate(Psk)$;

$evaluate(PSk) = \text{Minimum}(\text{PostState}(PSk), S_{b,g}) - \text{Minimum}(S_{b,t}, S_{b,g})$

其中 $\text{Minimum}(S_0, S_n)$ 表示返回从 S_0 到 S_n 的最少操作数; $\text{PostStat}(\text{action})$ 表示返回 action 执行后模型的状态;

PSk 是 $\text{Plan}(A)$ 中的元素, $S_{b,t}$ 是 B 当前的状态, $S_{b,g}$ 是 B 实现其目标时的状态。

将 $evaluate(PSk)$ 的结果分为下列两种情况:

(a) $evaluate(PSk) \leq 0$

模型接受此计划。如果存在多个计划的评估值小于零, 则选择值最小的那个计划。算法结束。

(b) $evaluate(PSk) > 0$

继续执行下一步。

(6) A 的所有计划都不能为 B 所接受。此时, B 向 A $\text{send-plan}(\text{Plan}(B)_1)$ 发送自己的计划集。 A 在接受到此消息后, 也同样用本算法第(3)、(4)步选择 B 的计划集中可接受的计划。

(7) 如果存在可接受的计划, A 向 B 发送 $\text{accept}(\text{Plan}(B)_1)$ 消息, 接受计划 $\text{Plan}(B)_1$, 算法结束; 否则 A 选择其中评估值最小的计划, 通过 send-value 将此计划及其评估值发送给 B , B 在收到此消息后将接收到的值与其在算法第(5)步中的最小值进行比较, 并接受与较小的值相关的计划。

(8) 模型各自确定协商后的 x 步和 y 步, 然后分别从 $x+1$ 步和 $y+1$ 步开始选择随机步数, 直到所有行为协商完成。

4.3 资源协商算法

虽然模型的行为经过协商已经取得了一致理解, 但是不同行为仍然可能由于需要对同一资源进行使用而导致冲突。因此, 对于这样的情况也需要处理。模型描述语义 DS 提供了模型在某一时刻的资源占用情况说明, 因此可以在此基础上进行协商。

算法3 模型资源协商算法(MRNA)

输入: 模型行为序列

输出: 无可能导致资源使用冲突的行为序列

(1) 模型 A 在行为序列 $\text{Plan}(A)$ 中根据任务本 SO 对于执行每一个 action 时给出状态描述语义 $Ms(A, i)$, 并提交给 B ;

(2) B 将自己的状态描述语义 $Ms(B, j)$ 与 $Ms(A, i)$ 进行匹配, 得到 $\text{Plan}(B)$ 中在某一行为 action_i 时可能 A 存在资源使用冲突的情况, 继续执行下一步; 否则, 不存在冲突, 算法结束。

(3) B 向 A 发送一个 $\text{query-information}(\text{State}(\text{resource}))$ 消息, 询问是否存在可用的替代资源;

(4) 如果存在这样的替代资源, A 向 B 发送一个 $\text{inform}(\text{Resource})$ 消息, 告知替代资源名; 否则 A 发送一个 query-information 消息, 询问是否有符合 A 的要求的替代资源。

(5) 如果 B 收到的是 A 发来的 inform 消息, 则 B 制定一个使用该替代资源的新计划。否则 B 搜寻满足 A 要求的资源。如果存在, 则 B 向 A 发送一个 $\text{inform}(\text{Resource})$ 消息, 通知存在其所需的资源; 否则向 A 发送一个 $\text{inform}(\text{NIL})$ 消息。

(6) 如果 A 收到了替代资源名, 则制定一个使用该替代资源的新计划。否则 B 在 action_{j-1} 处等待 A 使用完毕该资源, 并释放后再进行。

5 基于本体的模型冲突预测协商算法

本文通过对模型描述语义层次上的相似性判定来进行冲突预测, 通过行为语义层次上的行为协商以及资源申请协商来获得无冲突的行为序列。因此, 对于两个相互独立的模

型的执行过程而言, 必须要包含上述两个层次的算法。

算法4 模型冲突预测及协商(MCDN)

输入: 模型 A , 模型 B

输出: 无冲突执行序列

(1) 检测两个模型的本体类别。分为3个类型: (i) 若存在预测相似记录, 则直接进入算法第3步; (ii) 若存在预测不相似记录且不存在不冲突纪录, 则算法推出, 直接求解; (iii) 若不存在两个模型所属类别的预测记录, 则根据描述语义 DS , 利用算法1预测相似度;

(2) 分两种情况:

① 当描述相似度低于预设的阈值 N 时, 模型被认为不相似, 则记录预测结果, 然后直接并发执行。若并发过程中出现冲突, 则记录该两类模型为不相似但仍需协商类型, 并转入第(3)步;

② 当描述相似度高于预设的阈值 N 时, 模型被认为相似, 需要直接进行协商;

(3) 利用行为语义 BS , 根据算法2进行行为协商, 得到无冲突行为序列;

(4) 利用描述语义 DS , 根据算法3进行状态协商, 得到无重复资源使用行为序列;

(5) 算法结束。

6 测试与分析

6.1 本体的开发

本体是本文算法研究的基础。模型的语义通过本体表达出来。我们运用 Stanford 大学开发的本体开发工具 Protégé 2003 开发完成模型描述本体和任务求解本体。图4中, 我们给出了一个简单的一元线性回归预测模型任务求解本体例子。

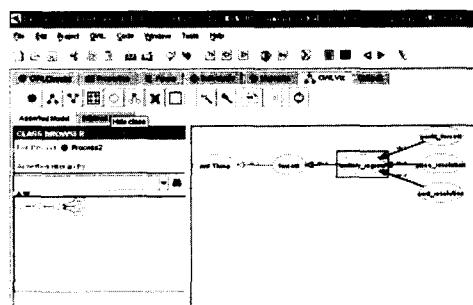


图4 一元线性回归预测模型任务求解本体

6.2 算法效率分析

本节主要通过仿真实验, 随机验证算法的执行效率, 并且对出现的实验数据进行相关分析。实验的平台为我们自主开发的“煤炭企业经济控制决策支持系统”原型系统。

6.2.1 冲突预测对模型的影响分析

我们从模型库中随机抽取模型进行冲突预测。模型任务完成后将被释放, 这些模型可能重新被别的任务调用再次运行。每一组抽取测试完毕以后, 重新进行随机抽取。如图5中显示, 当模型不采用本文的算法进行冲突预测的时候, 将出现显著的冲突次数; 而采用了本文中的算法时, 冲突出现的次数明显降低。我们注意到, 即便采用了本文中提出的算法, 模型仍然存在一定的冲突发生几率, 主要原因是因为系统本身的资源有限。经过冲突预测, 并通过协商使其尽量避免冲突, 仍然会由于无法获得足够的资源等等其他原因出现冲突。

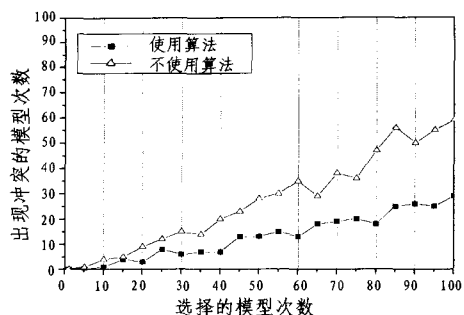


图5 模型冲突预测效率分析

6.2.2 协商算法对模型执行结果的影响

与前面一样,我们也随机选择20个模型用两种方法进行任务求解。一种是不采用本文的算法直接进行任务求解,若出现冲突则模型进行等待,看是否能够有一方释放资源或者交出权限,从而使任务得以顺利进行;另一种是采用本文提出的协商算法,通过求出无冲突行为序列进行任务求解。如图6所示,采用本算法的模型拥有较高的模型求解成功率。

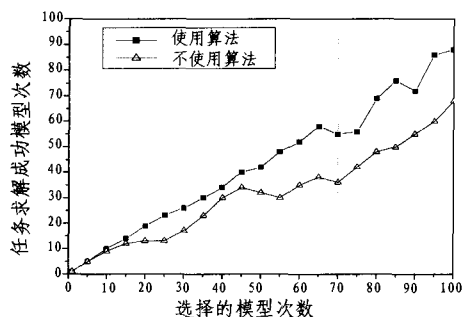


图6 模型协商效率分析

在协商算法下,模型可以通过交互方式解决行为、资源申请等冲突。一旦出现如上一节所出现的那种冲突时,模型仍然可以再次通过协商等待资源的释放,因此在牺牲一部分时间的基础上,能够获得最终的任务求解成功。而直接并发执行的模型,在出现类似于死锁的冲突情况下,将会无法完成任务。值得注意的是,协商算法在实验过程中出现的时间损耗,可能会给任务求解带来一些损失。

6.2.3 模型求解时间效率分析

求解时间是衡量决策质量的一个重要指标。我们进行了50次对比实验。每次对比都是实行同一个任务求解,并且分为两组,一组不采用本文所提出的预测协商算法,另一组则使用本文的算法。我们采用DSS内部的时间记录方式来记录每次求解所使用的时间(单位为秒)。

图7中,灰色线条表示未使用MCDN算法的一组模型,而黑色线条代表采用了MCDN算法的一组模型。通过记录时间分析,在刚开始的几组数据中,由于采用MCDN算法模型组需要首先预测相似性,然后做相关记录,并且进行协商,因此时间明显多于未采用算法的模型组。随着MCDN算法中对模型相似性记录的增加,采用了MCDN算法的模型组显示出协商算法的有效性,花费的求解时间逐步降低。这意味着未采用算法的模型组在并发求解的过程中,花费的等待时间远远大于采用MCDN算法的模型组。由于我们的DSS系统中,本体定义的模型类别相对较少,降低了需要预测的次

数;同时,模型的数量并未达到较多的数量,因此MCDN算法的时间效率还有待更为庞大的系统来检验。

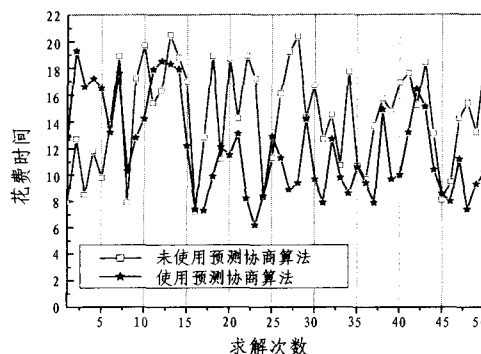


图7 MCDN算法时间效率分析

结束语 本文以本体语义为基础,研究了一种模型冲突预测以及行为协商的方法。本文研究有如下特点:模型的语义被分为描述语义和行为语义两个方面,清晰地定义了模型所处的动态环境和动态行为;通过模型描述本体和任务求解本体的定义,为模型提供可在动态环境中进行自我调整消除冲突的能力;强调语义描述,以及模型的自主交互。下一步的工作主要将集中于研究两个模型以上的情况下多对多的模型协商方法以及如何优化算法效果等方面。

参考文献

- [1] Lin M C, Manocha D, Cohen J, et al. Collision Detection; Algorithms and Applications//Proc. of Algorithms for Robotics Motion and Manipulation. 1997; 129-141
- [2] Clin M, Gottschalk S. Collision detection between geometric models; a survey//Proc. of the IMA Conf. on Mathematics of Surfaces. 1998; 37-56
- [3] Baker M. A Model for Negotiation in Teaching-Learning Dialogues. Journal of Artificial Intelligence in Education, 1994; 199-254
- [4] Kakehi R, Tokoro M. A negotiation protocol for conflict resolution in multi-agent environments. Intelligent and Cooperative Information Systems, 1993; 185-196
- [5] Ponamg M, Manocha D, Lin M C. Incremental algorithms for collision detection between solid models//the third ACM symposium on solid modeling and applications. 1995; 293-304
- [6] Faratin P, Sierra C, Jennings N R. Negotiation Decision Functions for Autonomous Agents. Robotics and Autonomous Systems, 1998, 24(3/4); 159-182
- [7] Gruber T R. A translation approach to portable ontology specification. Knowledge Acquisition, 1993, 5(2); 199-220
- [8] Soo V-W, Lee C-Y, Li C-C, et al. Automated semantic annotation and retrieval based on sharable ontology and case-based learning techniques// Digital Libraries, 2003. Proceedings. 2003 Joint Conference. May 2003; 61-72
- [9] Baker M J. A model for negotiation in teaching - learning dialogues. Journal of Artificial Intelligence in Education, 5 (2); 299-254
- [10] Baker M J, Lund K. Promoting reflective interactions in a CSCL environment. Journal of Computer Assisted Learning, 1997, 13; 175-193
- [11] 何炎祥,陈萃萌. Agent 和多 Agent 系统的设计与应用. 湖北:武汉大学出版社, 2001; 80-92
- [12] 何炎祥,陈燕涛,吴思,等. MDOCEM 中可重用资源的管理策略. 小型微型计算机系统, 1998, 19(9); 13-18