基于有色网的多 Agent 计划建模*)

杜卓敏　何炎祥
(武汉大学计算机学院　武汉 430072)

摘　要　有色网能够描述资源和操作的具体语义。首先，由于计划中的操作和状态的个数的有限性，与有色网的元素个数有限性约束完全一致。另外，计划中的动作与有色网中的变迁语义类似，以及计划中的操作和状态和有色网中的库所语义非常类似。因此，有色网应用到计划的形式化中，有其独特的优势。本文根据约定的前提条件，计划建模从操作、状态和交互 3 个方面来具体实现，并给出了建模方法。计划的规范描述、有效性验证以及计划的模拟都可以直接应用经典 Petri 网或有色网的理论技术。

关键词　计划，有色网，多 agent

Modeling for Multi-Agent Plan Based on Coloured Petri Nets

DU Zhuo-Min　HE Yan-Xiang
(School of Computer, Wuhan University, Wuhan 430072)

Abstract　Coloured Petri nets can describe the concrete semantics of resources and operations. Firstly, the number of operations and states of a plan in multi-agent system must be finite, and the finite number of elements of Coloured Petri nets really satisfies the restriction of plan. And action of plan is similar to transition of Coloured Petri nets and state and resource of plan is similar to place of Coloured Petri nets. Therefore it is effective to model plan by Coloured Petri nets. Based on some premises, the action, the state and the interaction components of plan are modelled in detailed. Moreover, the description, validity and simulation of plan can be implemented by the theoretical techniques of Petri net or Coloured Petri nets.

Keywords　Plan, Coloured petri nets, Multi-agent

agent 是具有自治能力的程序实体[1~3]。一般说来，agent 有 3 种类型的活动：观察、计划和执行。观察为 agent 提供决策必要的信息，包括环境中事件、状态以及其它 agent 相关信息。计划规定 agent 将要完成的动作序列。执行为实现目标触发相应的动作序列。显然，计划是 agent 体现自治性的一个重要特征，因此对计划建模是 agent 系统建模的一个重要组成部分。

agent 建模一直是研究热点之一[4~7]。文[4]和文[8]分别采用 Petri 网和公理系统对 agent 动作的动态语义进行形式化。文[9]描述多 agent 状态，但是对 agent 动作的描述则显得不足，更缺乏计划的语义描述。文[10～12]从形式化和非形式化角度分析和比较了多 agent 计划，但是作者关注系统状态的变化，而忽略了动作的描述。实际上，不同的动作语义可能导致不同的状态，因此忽略动作导致了某些状态的丢失。另外，由于 agent 知识是独立感知的，因此知识处于不断地更新和优化，而基于这种知识的计划，也具有分布不多样性，且不断地更新。由于 agent 受到环境和认知能力的限制，计划具有非确定性[13~15]。非确定性不仅表现在 agent 动作执行序列的不确定，而且表现在对其它 agent 的认知也是不确定的[16, 17]。从而对计划的建模，不仅对计划进行形式化，能够对计划规范进行验证、证明或者优化，同时应该体现这种不确定性。

由于有色网(Coloured Petri Net, CPNs)[18]良好的理论基础和对非确定性的刻画，本文利用有色网系统来对多 agent 计划进行建模、对计划进行验证、证明或者优化。本文内容是这样安排的：第 1 节详细提出了应用有色网对计划建模应该遵守的必要前提，以及如何应用有色网概念对计划的基本构件动作、状态和交互进行描述。在第 2 节中根据所提出的建模方法具体讨论一个计划模型。第 3 节主要针对计划模型中特殊边界特点在有色网中的约束进行讨论。这些约束必须在有色网的验证和优化予以保证。最后总结有色网在实际计划建模的必要性、可行性，以及我们以后采用有色网技术来验证、优化、证明相关工作。

1　有色网模型

传统的 Petri 网虽然可以很好地描述系统的并行非确定性语义，但是由于高度抽象特性，因此并不适合描述具体资源和操作的语义。有色网是对经典 Petri 网的扩展，聚合了数据操作控制和同步，可以使环境、动作的发生条件以及动作发生结果同时展现在同一张网中。另外，有色网模型发生变化时，不会影响有色网的结构，即当系统的某一局部发生变化时，不会影响到网的其他部分，表现出良好的稳定性。其它形式化工具，如有穷自动机描述的系统，当局部结构发生变化时，涉及到的变化是不可预料的。

为简洁起见，本文直接应用 Petri 网和有色网的相关概念，具体定义参见文[19]。设 S 为非空集合，IN_0 是非负整数集，则从 S 到 IN_0 的函数叫做 S 上的多重集(bag 或 multi-set)。多重集和集合的区别在于前者允许同一个元素可以出

*)国家自然科学基金重大研究计划(90104005)资助、武汉大学科技创新基金资助。杜卓敏　博士，讲师；何炎祥　博士，教授，博士生导师。

现多次。例如集合{1,1,2}是集合{1,2}上的多重集,其中元素1出现了两次。根据以上定义,可知函数 $f:\{1,2\}\to IN_0$,其中 $f(1)=2, f(2)=1$。在有色网建模中,如没有特别说明,集合均是有限多重集。S_{MS} 表示集合 S 上的所有有限多重集所成之集合。以下用 $[S\to R]_L$ 表示从集合 S 到集合 R 的所有线性函数集合。

定义 $\Sigma=(N, C, I_-, I_+, M_0)$称为有色网系统的充分必要条件是:

1)$N=(P, T; F)$为 P/T 网[19],称为 Σ 的基网。

2)$C: P\cup T\to \Pi(D)$,D 是颜色集合,$\Pi(D)$为颜色集 D 之幂集合,使得:

对 $p\in P, C(p)$是库所 p 上所有可能的托肯色之集合。

对 $t\in T, C(t)$是变迁 t 上所有可能出现的色之集合。

3)I_- 和 I_+ 分别是 $P\times T$ 上的负函数和正函数,使得对所有 $(p,t)\in P\times T$;

$I_-(p,t)\in[C(t)_{MS}\to C(p)_{MS}]_L$,且 $I_-(p,t)=0$ 的充分必要条件是 $(p,t)\notin F$;

$I_+(p,t)\in[C(t)_{MS}\to C(p)_{MS}]_L$,且 $I_+(p,t)=0$ 的充分必要条件是 $(t,p)\notin F$;

4)$M_0: P\to D_{MS}$,称为 Σ 的初始标识,它必须满足条件 $\forall p\in P: M_0(p)\in C(p)_{MS}$,即 $M_0(p)$是 p 的托肯色集合上的多重集。

为实现有色网应用于计划建模,本文遵循以下前提:agent 对系统和其它 agent 所认知的知识是有限的,并且对知识是封闭的。每个 agent 所提供的动作数目有限,并且在一定时期内是稳定的,从而保证每个计划中需要的动作是相对有限并可控制。每个 agent 具有观察并保存其它 agent 信息的知识库,计划的制定是基于该知识库的,计划个数是有限的。每个计划只需要有限的 agent 参加,并且只需要有限的动作执行便可以得到结果。计划的有限性保证了计划的可管理性。任务必须在有限的动作内完成,否则认为是不可完成的任务。本文中所讨论的计划是原子的,即计划不包含子计划。下面从计划所包含的动作、状态和交互3个方面详细讨论计划的建模。

1.1 动作建模

agent 通过接受消息来启动相应的动作。如果所需资源满足时,该动作就可以运行。这里资源是一个广义的概念,包括动作发生所需的一切条件。动作语义与有色网变迁语义类似。因此计划模型中,动作由变迁描述,即任意变迁 $t, t\in T$,T 是 agent 已认知的所有动作集合。动作的具体语义由颜色来区别,即不同颜色区别不同动作。每个动作消耗或产生资源的具体数目由函数 I_- 和 I_+ 来决定。

动作的语义就是有色网某一标识下变迁获得授权(enabling),对变迁相关联的弧表达式求值,即:1)根据变迁的输入弧所关联库所中的托肯数,对该弧表达式 I_- 求值;2)变迁的发生从输入弧所关联的库所中移去托肯,同时向输出弧所关联的库所中增加 I_+ 个托肯。移去的托肯数由输入弧表达式 I_- 决定;同样地,增加的托肯数由输出弧表达式 I_+ 决定。计划的动态行为由有色网的标识变化来描述。

1.2 状态建模

agent 对状态、变量、消息、资源一旦识别,该认知的知识在一时期内相对稳定。本文中对具有被动性、没有主动运行能力的状态、变量、消息以及系统资源等都称为资源。资源可抽象为状态。在计划模型中,状态用库所描述,库所的特性由颜色确定。有色网中,库所关联一个类型(颜色集),该类型决定了库所所能包含数据的合法性。

托肯(token)表示属于库所类型的一个具体值。用前导数字加上 * 表示具有相同值的托肯数目,如库所中有两个相同值的托肯,表示为:2 * (1, "Modellin"),其中2表示托肯的个数,1表示所属数据类型(颜色),"Modellin"是托肯的具体值。托肯的颜色描述了托肯的个性。库所中托肯表示库所的当前状态。计划状态由一系列分布在各个不同库所中的托肯描述。有色网系统初始标识描述计划的初始状态。

计划所涉及的声明由两部分组成:第一部分定义颜色;第二部分定义变量。声明可以标注在网的任意显著位置。图1是颜色类型及变量的声明片断。

```
Declaration
Color INT = integer;
Color DATA = string;
Color INT×DATA = (INT, DATA);
Var n, k: INT;
Var p, str: DATA;
Var stop = "###";
```

图1 颜色和变量声明

1.3 交互建模

在计划模型中,动作之间交互通过状态作为中介。状态和动作之间的交互由弧来描述。弧有两种类型:输入和输出。双向弧是两个相反方向的有向弧的简化。弧描述了动作需要的资源和产生的资源数量。

动作和状态由弧连接,弧关联弧表达式。动作的发生根据 I_- 和 I_+ 移去由输入弧连接(输入库所)的库所中的托肯。变迁发生后,向输出弧所连接的库所(输出库所)加入托肯。这个过程改变计划的标识(状态)。动作的发生移去或者增加的托肯数目由弧表达式(I_- 和 I_+)确定。有色网有两种弧表达式类型:变量、颜色,如图2,其中 $I_+: (n, p)\in$ INT×DATA。

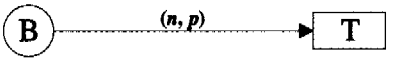

图2 弧表达式1

另外一个弧类型是 if-then-else 表达式(如图3),else 可以包含 if-then-else 子句。其中 if 和 else 条件不可相交,如果条件相交,则相对应的 then 后面的子句必相同。实际上,if-then-else 与 Dijkstra 的哨命令[20]有相似之处,控制着变迁发生的控制转移。

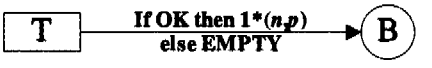

图3 弧表达式2

图3中,OK 表示变迁 T 成功发生,EMPTY 表示空数据。任何复杂数据类型的值都可以是空,表示不产生托肯。

有色网中的弧表达式约束了变迁发生的条件以及产生的结果。条件或结果由库所包含的托肯及其颜色(值)来确定。变迁关联的弧表达式实际上是对有色的托肯求(色)值。如果

弧表达式不能得到具体带色值，那么该变迁不可能发生。

2 一个例子

本节用例子来说明计划的有色网建模。设多 agent 环境下，agent R 对一个字符串中小写字母转换成大写字母，如 abc 转换成 ABC。该任务的完成需要其它 agent 的合作。环境知识从 agent 的知识库中获取。Agent R 自主地从其知识库中选取动作。本例子的计划由有色网系统描述(图 4)。参与这个任务的有 3 个 agent：请求服务 agent R，网络传输 agent T，处理 agent P。

agent R 中，状态 Send 保存需要处理的字符串，并且对将要处理的子字符串进行编号，每个子字符串在 Send 中以一个带有颜色的 token 出现。用(1，“Modelli”)表示颜色是 1 的 token，该 token 表示字符串“Modelli”。Send 中的托肯都是以不同的颜色出现。SendPk 向其它的 agent 提供字符串。字符串的顺序是有意义的，库所 NextSend 控制着下一个向 agent T 提供的字符串。变迁 RevAck 是从 agent T 得到的消息：下一个合法的字符串是哪一个颜色 token。A，C 是缓冲区。RsltRT 保存转换的结果。

Agent T 负责把字符串传至 agent P，其中的动作 TransPk 传送字符串，TransAck 传送确认消息并告知下一个合法的字符串标识符。TransRslt 则传送处理后的结果。

Agent P 对字符串进行处理，即把其中的小写字符转换成大写字符。B 和 D 是传输缓冲区。动作 TransAck 确认顺序是否正确的符号串。库所 NextRec 控制着所需的下一个合法字符串。操作 CapStr 把字符串进行转换。库所 RsltPT 保存转换的结果，其长度随着新的字符串的加入而不断更新。每个 agent 都有缓冲区，为讨论方便，把相邻 agent 的缓冲区合并一起加以考虑。如 agent R 有缓冲区 A_R，同时 agent T 有缓冲区 A_T，则合并记为 A_{RT}。对于 agent P 和 agent T 有缓冲区 A_{PT}。缓冲区 B，C，D，Rlst 类似。

假设 agent R 有数据：Modelling Means of Coloured Petri Nets，已经分割成 5 个数据包：Modelli，ng Means，of Colou，red Petr，I Nets。因此 sned 的初始标识就是这 5 个数据包，其颜色分别是 1，2，3，4，5。OK 表示网络传输成功。Agent R 中 NextSend 指定下一个传输的数据包，其初始标识 M0(NextSend)为 1。Agent P 中 NextRec 指定下一个接受的数据包，其初始标识 M0(NextRec)为 1。如果到达的数据包不是指定的，则要求发送方继续传送。其它初始标识均为空值，在计划运行过程中，标识都会发生变化。

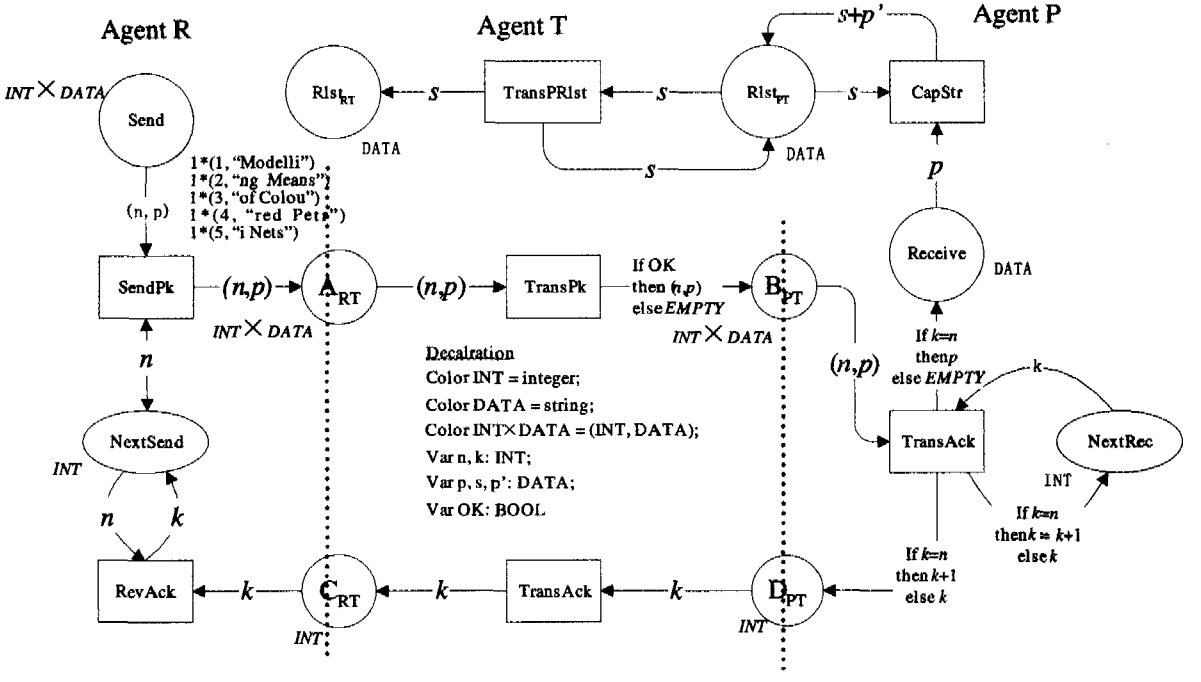

图 4 字符串处理计划

3 有色网分析

在对计划进行建模后，一项重要的工作就是对计划模型的验证。根据传统 Petri 网和有色网理论，模型的验证可以应用可达图、进程或不变量等技术。为实现计划模型的验证，计划模型必须满足以下网特征。

定义 设 $x\in X$ 为 Petri 网 N 中的任一元素，$^{\cdot}x=\{y|(y,x)\in F\}$称为 x 的前集或者输入集；

$x^{\cdot}=\{z|(x,z)(F\}$称为 x 的后集或者输出集；

定理 计划中，有且仅有一个库所 s，使得 $^{\cdot}s=\varnothing$。

多 agent 环境中，计划的最终目标是该 agent 提出任务计算请求，同时 agent 必须提供任务的基本初始条件。该 agent 不能提供的操作或者条件则由合作 agent 提供。如果有使得 $^{\cdot}s=\varnothing$ 的库所个数多于 1 个，那么表示任务的初始条件由多于一个的 agent 给出，因此 s 不是计划的唯一提出者。从而计划由多个 agent 启动，显然这是与实际的计划制定者唯一性相矛盾。

定理 计划中，如果 $s^{\cdot}=\varnothing$，则称 s 为计划完成态。

当计划执行到某些状态时，该状态不再发生变化，也就是到达了不动点。

以上两个定理规定了计划的两个边界特性。在计划的中间运行状态，可以应用有色网标识等概念描述。

设 S 为有色网系统 Σ 在标识 M 下的步(step)[19]，则 S 发生后的后继标识是 M'。对于 $\forall p\in P$，P 是 agent 所认知的所有状态集合。

$$M'(p)=M(p)+\sum_{t\in T}I_+(p,t)(X(t))-\sum_{t\in T}I_-(p,t)(X(t))$$

其中，$X(t)$表示在 X 步中，变迁 t 按颜色出现的次数。如 $X(t)=2m+w$ 表示在 X 步中，变迁 t 按颜色 m 出现两次，颜色 w 出现一次。那么，M 与 M' 关系称为后继关系，记为：$M[X>M'$，即 Σ 在标识 M 经过变迁集合 X 之后，到达标识 M'。

根据后继关系及步概念，有色网从一个状态到下一个状态，有两个基本特点：1）在同一步内，动作可以并发地发生。2）根据 I_- 和 I_+ 以及 M 就可以决定下一个状态 M' 的标识。虽然有色网没有明确 Σ 中托肯的个数守恒，但是实际上 token 数守恒对于研究计划性质非常重要。如图 4 中，因为 agent 的初始状态有几个字符串，该计划的运行必须保证在最终状态托肯个数保持不变，否则表示异常情况的出现。另外在图 4 中，路径 $A\rightarrow$TransPk$\rightarrow B$ 或者路径 $C\rightarrow$AckPK$\rightarrow D$ 之中，只能同时传递一个托肯，其语义是这条路径上，每次只能传送一个字符串。这种托肯数在网中的不变性质称为不变量。不变量（包括 T 不变量和 S 不变量）是有范围限制的，即局部不变量和系统不变量。有色网不变量的计算可以参考文[19]中矩阵表示及其计算。

有色网中的发生序列（库所变迁交替出现的序列）描述计划中动作的执行序列以及相应的状态变化序列集合。因此，发生序列描述了计划为实现目标所要采取的步骤以及每个步骤之后所处的状态。如果不考虑库所（所处的状态），那么发生序列就是 agent 为完成目标所执行的动作序列。同样，不考虑变迁（动作），那么发生序列就是 agent 为完成目标所必须经历的状态序列。上节例子中，从 agent 发生请求到返回结果，整个计划的完成，可能需要 n 步完成，而步 $S_i(1\leqslant i\leqslant n)$ 在整个计划中具有顺序关系（步内动作可以并发），即发生序列：

$$M_0[S_1>M_1[S_2>\cdots M_n[S_n>M$$

对于这个序列，可以从两个角度进行研究：1）状态的变化序列：$M_0M_1M_2\cdots M_nM$；2）变迁的变化序列：$S_1S_2\cdots S_n$。

定理 计划中，任意标识 M_i，都是可以由初始状态 M_0 可达的，即存在一个发生序列：$M_0[S_1>M_1[S_2>\cdots M_j[S_j>M_i$。

证明：假设该序列不存在，则有色网中存在两种情况：网是不连通的，即计划模型包含有至少两个不连通子网，而此时子网各自的操作没有交互。因此，计划的目标结果是独立的，这与计划的结果唯一性和计划内操作关联性相矛盾。因此该计划不是合法的计划。

当有色网是连通的，而存在状态 M_k 在计划内是不可达的，库所 p_k 在标识 M_k 下不含托肯。p_k 不是孤立的，故 p_k 存在前驱变迁 t_k，并且存在函数 I_+。因为 p_k 不含托肯，故 t_k 没有授权，即计划中动作 t_k 是不发生的。显然在制定计划的时候，选取了一个动作，但是该动作不参加运行。由此可知，该计划不是最优的，甚至是错误的。

因此，计划中，任意标识 M_i 都是可以由初始状态 M_0 可达的。

小结 多 agent 环境决定了计划是分布并行的、计划制定是非唯一的和运行是不确定的。因此，计划的形式化必须描述这些分布并行特征和交叉假设问题[21]。正如我们所知，无论是 Z，CSP，CSS 有穷自动机等形式化工具，虽然是受到广泛接受的形式化工具，但是由于它们都是基于交叉语义假设[22]，因此无法描述计划的非确定性。当前流行的 UML 建模工具[23]，由于缺乏理论支持，无法对计划进行有理论意义的形式化。

Petri 网作为直观的、具有强大理论支持的形式化技术，已经得到广泛的应用。Petri 网适宜描述分布并行计算语义，并且有效地克服了交叉假设。然而正如我们所提到的，经典 Petri 网不能描述系统中资源以及操作的具体语义[22]。有色网正是基于这个缺憾提出的。有色网的元素个数的有限性和多 agent 环境下的计划以及操作、状态和交互的个数的有限制恰好一致。实际上，计划参与的动作以及相应的状态必然是有限的，否则计划是不可制定和不可完成的。计划中的动作与有色网中的变迁、计划中的状态和资源与有色网中的库所，有非常类似的特性，因此用有色网来描述计划的语义非常有效。另外，计划的有色网描述、有效性以及模拟可以直接应用有色网和经典 Petri 网的理论工具。

本文正是基于有色网这一优势，应用到多 agent 环境下的计划建模。我们已经讨论了有色网对计划建模的可行性和建模所面对的特殊要求、前提以及计划模型所要满足的性质。在以后的工作中，我们将陆续介绍如何对计划进行验证、优化、模拟等理论和技术问题。

参 考 文 献

1 Jacques F. Multiagent Systems: An Introduction to Distributed Artificial Intelligence. NewYork: Addison-Wesley, 1999
2 Barreteau O, Bousquet F. SHADOC: *a multi-agent model to* tackle viability of irrigated systems, 2000, 94: 139～162
3 Cohen P R, Levesque H J. Rational interaction as the basis for communication. In: MIT Press, 1990. 221～255
4 Innocent B, et al. Formalization of a Spatialized Multiagent Model Using Coloured Petri Nets for the Study of an Hunting Management System. 2000, 1871: 123～132
5 Munindar P S, Anand S R, Michael P G. Formal methods in DAI: logic-based representation and reasoning. In: Cambridge MA, USA: MIT Press, 1990. 331～376
6 Bernhard B, Müller J o P, James O. Agent UML: A Formalism for Specifying Multiagent Software Systems. 2001, 11: 270～230
7 Markus L, Franz A, Oscar N. A Formal Language for Composition. Cambridge University Press, 2000. 69～90
8 Ardissono L, Boella G, Lesmo L. A plan-based formalism to express knowledge about actions. In: Proc. 4th ModelAge Workshop: Formal Models of Agents. Pontignano, Italy, 1997. 255～268
9 Perini A, et al. Agent-oriented modeling by interleaving formal and informal specification. In: Proceedings of the 4th International Workshop on AgentOriented Software Engineering. Vol. 2935. Melbourne, Australia: Springer, 2003
10 Michael, E B, David I, Martha P. Plans and Resource-Bounded Practical Reasoning. In: Philosophy and AI: Essays at the Interface. Cummins and John. Cambridge, Massachusetts: The MIT Press, 1991. 1～22
11 Cimatti A, et al. Weak, strong, and strong cyclic planning via symbolic model checking. 2003, 147: 35～84
12 Rune M J, Manuela M V. OBDD-based Universal Planning for Synchronized Agents in Non-Deterministic Domains. 2000, 13: 189～226
13 Philip R C, Jerry M, Martha E P. Intentions in Communication. ENGLAND, MA, USA: the MIT Press, 1990
14 Sandra C. Plan Recognition in Natural Language Dialogue. Cambridge, MA, USA: MIT Press, 1990
15 Philip R C, Hector J L. Intention is choice with commitment. 1990, 42: 213～261
16 Weigand H, Frank D. Formalization and rationalization of communication. In: the Second International Workshop on Communication Modeling (LAP-97), Dietz F D a J. ed. Veldhoven, The Netherlands, 1997. 71～86
17 Verharen E, Dignum F, Weigand H. A Language/Action Perspective on cooperative information agents. 1998, 8: 39～59
18 Jensen K. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Germany: Springer Verlag, 1992
19 Chongyi Y. Petri nets Theory. Beijing: Publishing House of Electronics Industry, 1998
20 Dijkstra E W. A Discipline of Programming. NewYork: Prentice Hall, 1976
21 Chandy K M, Jayadev M. Parallel Program Design: A Foundation. NewYork: Addison- Wesley, 1989
22 Guofu Z. Research on Modelling Technology of Software Architecture based on UniNet. Beijing, 2003
23 Philippe K. The Rational Unified Process: An Introduction. Boston, MA: Addison-Wesley Professional, 2000