模拟的概念、计算及在自动机最小化上的应用*)

徐正权　袁志斌　王能超

（华中科技大学计算机科学与技术学院　武汉 430074）

摘　要　对 Büchi 自动机进行优化是提高基于自动机的模型检测效率的重要手段。本文对直接模拟关系，延迟模拟关系和公平模拟关系的概念，算法进行了比较，并探讨了基于这些模拟关系的自动机优化方法。最后对未来的研究方向作了简要的介绍。

关键词　模拟，自动机，最小化

Simulations for Minimizing Automata

XU Zheng-Quan　YUAN Zhi-Bin　WANG Neng-Chao

（Department of Computer Science and Technology, Huazhong University of Science Technology, Wuhan 430074）

Abstract　Optimizing Büchi automata is an important step in efficient model checking for linear-time specification. In this paper we compare among three notions of simulation: direct, delay, and fair. We study simulation relations as a tool for minimizing automata. Finally, some research directions for future work are also discussed.

Keywords　Simulation, Automata, Minimization

1　引言

自 Robin Milner 提出模拟[1]的概念以来，其已被广泛地应用于计算机科学的诸多领域，特别是自动验证技术，如模型检测[2]中。模型检测是一种关于系统性质验证的算法方法，它通过状态空间搜索的方法来检测一个给定的计算模型是否满足某个用时序逻辑公式表示的特定的性质。

基于线性时序逻辑的模型检测通常需要规约转换为等价的 Büchi 自动机，现在有很多种转换算法[3~5]，但一般结果自动机与原时序逻辑公式的规模是指数关系。为提高模型检测的效率，需要尽可能的对自动机进行优化。自动机优化的一个重要的方法就是借助模拟的概念计算其等价自动机。

为了避免将不需要的系统行为引入模型，一般的做法是增加公平性条件，借此来区分需要和不需要的系统行为。模拟关系没有考虑公平性问题，为此需要对模拟的概念加以修改，以满足区分公平性行为的需要。本文探讨了多种考虑了公平性条件的模拟概念，计算方法及在自动机最小化上的应用。

2　基本概念

本节给出了本文涉及的重要概念：

定义 1　给定无穷序列 π，$\inf(\pi)=\{s \mid \forall i \exists j>i \pi(j)=s\}$。

定义 2（Büchi 自动机）　有穷字符集 Σ 上的 Büchi 自动机 $A=(S,R,S_0,L,F)$。其中 S 为有穷状态集合，$S_0\subseteq S$ 为初始状态集合，$R\subseteq S\times S$ 为迁移关系，$L: S\to AP$ 为标注函数，$F\subseteq S$ 为接受条件。

以下除特殊说明，自动机均指 Büchi 自动机。自动机 A 上的路径是状态 S 上的无穷序列 $\pi=a_0,a_1,\cdots$，其中 $a_0\in S_0$，而且对于所有的 $i\geqslant 0, a_{i+1}\in R(a_i)$。一个路径 n 如果满足 $\inf(\pi)\cap F\neq\phi$，则该路径为自动机所接受。

定义 3（博弈）　博弈为六元组 $G=(V,V_0,V_1,v_I,E,W)$ 其中：V 为位置的集合；V_0,V_1 为选手 0，1 的位置，$V_0\neq\phi$ 或 $V_1\neq\phi$，$v_I\in V$ 为初始位置，$E\subseteq V\times V$ 为迁移关系，$W\subseteq V^\omega$ 为选手 0 获胜的赛局的集合。有向图 (V,E) 为 G 的博弈图，在不发生混淆的情况下也用 G 来表示。赛局指的是 G 中从 V_I 出发的最大路径；如果赛局 $\pi=v_0v_1v_2\cdots$ 是无穷序列，而且 $\pi\in W$，则选手 0 获胜，否则选手 1 获胜。如果 π 为有穷序列，如果 π 的最后一个位置属于选手 1(0)，则选手 0(1)获胜。

奇偶博弈是博弈的一种，其特点是博弈图中每一个节点都有一个优先级，其赛局 π 为无穷序列时胜负判定方法是如果赛局中无穷出现的最小优先级为偶数，则选手 0 获胜，否则选手 1 获胜；π 为有穷序列时的判定方法与一般博弈相同。

3　模拟的定义

设 Büchi 自动机 $A_1=(S_1,R_1,S_{01},L_1,F_1)$ 和自动机 $A_2=(S_2,R_2,S_{02},L_2,F_2)$ 有相同的原子命题。

定义 4（模拟关系(simulation relation)[1]）　关系 $H\subseteq S_1\times S_2$ 是两个自动机 A_1 和 A_2 之间的模拟关系当且仅当任意的 s_1 和 s_2，如果 $H(s_1,s_2)$ 满足以下条件：

(1) $L_1(s_1)=L_2(s_2)$；

(2) 对于所有的满足 $R_1(s_1,s_1')$ 的 s_1' 都存在 s_2' 满足 $R_2(s_2,s_2')$ 和 $H(s_1',s_2')$。

即：$\forall s_1'[(s_1,s_1')\in R_1\to\exists s_2'[(s_2,s_2')\in R_2\wedge(s_1',s_2')\in H]]$

如果自动机 A_1 的任意初始状态 $s_{01}\in S_{01}$ 在自动机 A_2 中存在一个初始状态 $s_{02}\in S_{02}$ 满足 $H(s_{01},s_{02})$，则称自动机 A_2 模拟 A_1，记作 $A_1\preceq A_2$。如果 A_2 模拟 A_1，而且 A_1 模拟 A_2，则称 A_1 和 A_2 模拟等价(simulation equivalent)，记作 $A_1\approx$

*)本文工作得到国家自然科学基金(70271069)的支持。**徐正权**　教授，博士，主要研究方向：软件工程、形式化方法；**袁志斌**　博士生，主要研究方向：程序正确性、形式化方法；**王能超**　教授，博士生导师，主要研究方向：并行计算。

A_2。

以上模拟(需要强调时称之为普通模拟)的定义没有考虑公平性的问题,不能直接用来讨论对 Büchi 自动机的优化。以下介绍几种考虑了公平性问题的模拟关系,它们分别是直接模拟关系[6],延迟模拟关系[7]和公平模拟关系[8]。

定义 5(直接模拟关系) 关系 $H\subseteq S_1\times S_2$ 是两个自动机 A_1 和 A_2 之间的直接模拟关系当且仅当任意的 s_1 和 s_2,如果 $H(s_1,s_2)$满足以下条件:

(1)$L_1(s_1)=L_2(s_2)$;而且如果 $s_1\in F_1$ 则 $s_2\in F_2$。

(2)对于所有的满足 $R_1(s_1,s_1')$的 s_1' 都存在 s_2' 满足 $R_2(s_2,s_2')$和 $H(s_1',s_2')$。

即:$\forall s_1'[(s_1,s_1')\in R_1\rightarrow\exists s_2'[(s_2,s_2')\in R_2\wedge(s_1',s_2')\in H]]$。

博弈是计算机理论中一个应用广泛的工具,它不仅可以深刻的刻画模拟的概念,而且还可以为模拟的计算提供高效的算法。设有自动机 A_1 和 A_2,两个选手称为反对者和拥护者。反对者运行在系统 A_1 上,拥护者运行在 A_2 上。

定义 6(模拟的博弈表达) 给定自动机 A_1 和 A_2,模拟博弈由有穷或无穷轮组成。开始时反对者从 A_1 的初始状态中挑选一个状态 s_{01} 作为出发点,拥护者则从结构 A_2 的初始状态集中挑选一个状态与 s_{01} 相对应,而且满足 $L_1(s_{01})=L_2(s_{02})$。在以后的任一轮中设反对者在状态 s_i 上,拥护者在状态 s_j 上,此时,反对者移动到 s_i 的后续状态 s_{i+1} 上,与此相对应,拥护者要从 s_j 的后续状态中挑选一个状态 s_{j+1},使得 $L_1(s_{i+1})=L_2(j+1)$,并且移动到该状态。如果拥护者找不到相匹配的状态则拥护者输,否则该博弈就会产生两个无穷路径,此时拥护者胜出。

定义 7(策略) 给定自动机 A_1,A_2,拥护者的策略 $\tau\subseteq(S_1\times S_2\rightarrow S_2)\cup(S_{01}\times\{\perp\}\rightarrow S_{02})$为满足以下要求的函数:如果 $s_2'=\tau(s_1',s_2)$则$(s_2,s_2')\in R_2$。

如果拥护者始终按策略 τ 来选择状态而且在博弈中拥护者获胜则称 τ 是拥护者的获胜策略。

根据以上讨论可知对于给定的自动机 A_1, A_2, A_2 模拟 A_1($A_1\leqslant A_2$) 当且仅当在由 A_1 和 A_2 构成的模拟博弈中拥护者有获胜的策略。

定义 8(延迟模拟的获胜策略) 在由 A_1 和 A_2 构成的模拟博弈中,拥护者有延迟获胜的策略当且仅当博弈进行无穷多轮,而且如果反对者到达一个公平状态,则拥护者在之后的有穷轮内也进入自己的公平状态。

定义 9(公平模拟的获胜策略) 在由 A_1 和 A_2 构成的模拟博弈中,拥护者有公平获胜的策略当且仅当博弈进行无穷多轮,而且如果反对者沿着公平路径移动,则拥护者也能沿着公平路径移动。

定义 10(延迟/公平模拟) 给定自动机 A_1, A_2, A_2 延迟/公平模拟 A_1 当且仅当在由 A_1 和 A_2 构成的模拟博弈中拥护者有延迟/公平获胜策略。

根据以上定义可以得出以下关系:$(A_1\leqslant_{di}A_2)\subseteq(A_1\leqslant_{de}A_2)\subseteq(A_1\leqslant_f A_2)$

4 模拟的计算

模拟关系不仅可以用博弈来表达,而且博弈理论还为模拟关系的计算提供了高效算法。给定 Büchi 自动机 A,及初始状态 s_0,s_0',定义如下四种博弈图[7]:

(1)普通模拟博弈图,记作:$G^0_{A,A}(s_0,s_0')$,

(2)直接模拟博弈图,记作:$G^{di}_{A,A}(s_0,s_0')$,

(3)公平模拟博弈图,记作:$G^f_{A,A}(s_0,s_0')$,

(4)延迟模拟博弈图,记作:$G^{de}_{A,A}(s_0,s_0')$。

设博弈图中的状态(s,s')属于选手 0 的获胜域,则在自动机 A 中 s' 模拟 s,即 $s\leq s'$。该博弈图的规模为 $O(|S||R|)$,而且其节点的优先级为 3 级即可。不同模拟对应的博弈图的构造模式基本相同,具体构造如下:

设公平博弈图为 $G^f_{A,A}=(V_0^f,V_1^f,E_A^f,p_A^f)$,其中优先级函数为 $p_A^f:V\rightarrow\{0,1,2\}$,

$$V_0^f=\{v_{(s,s',a)}\mid s,s'\in S\wedge\exists s''((s'',a,s)\in R)\}\tag{1}$$

$$V_1^f=\{v_{(s,s')}\mid s,s'\in S\}\tag{2}$$

$$E_A^f=\{(v_{(s_1,s_1',a)},v_{(s_1,s_2')})\mid(s_1',a,s_2')\in R\}\cup\{(v_{(s_1,s_1')},(v_{s_2,s_1',a})\mid(s_1,a,s_2)\in R\}\tag{3}$$

$$p_A^f(v)=\begin{cases}0, & \text{如果}(v=v_{(s,s',a)}\text{或}v=v_{(s,s')})\text{而且}s'\in F,\\ 1, & \text{如果}v=v_{(s,s')},s\in F,\text{而且}s'\notin F,\\ 2, & \text{其他。}\end{cases}\tag{4}$$

普通模拟博弈图 $G^0_{A,A}$ 的构造只需要在 $G^f_{A,A}$ 的基础上作如下变更:所有的节点的优先级均改为 0。

直接模拟博弈图 $G^{di}_{A,A}$ 的构造只需要在 $G^f_{A,A}$ 的基础上作如下变更:所有的节点的优先级均改为 0,并且去掉一些边,使得边的定义如下:

$$E_A^{di}=E_A^f\setminus(\{v,v_{(s_1,s_1')})\mid s_1\in F\wedge s_1'\notin F\}\cup\{(v_{(s_1,s_1')},w)\mid s_1\notin F\wedge s_1'\in F\})\tag{5}$$

延迟模拟博弈图 $G^{de}_{A,A}$ 的定义如下:

$$V_0^{de}=\{v_{(b,s,s',a)}\mid s,s'\in S\wedge b\in\{0,1\}\wedge\exists s''((s'',a,s)\in R)\}\tag{6}$$

$$V_1^{de}=\{v_{(b,s,s')}\mid s,s'\in S\wedge b\in\{0,1\}\wedge(s'\in F\rightarrow b=0)\}\tag{7}$$

$$p_A^{de}(v)=\begin{cases}b, & \text{如果}v=v_{(b,s,s')}\\ 2 & \text{如果}v\in V_0\end{cases}\tag{8}$$

$$\begin{aligned}E_A^{de}=&\{(v_{(b,s_1,s_1',a)},v_{(b,s_1,s_2')}\mid(s_1',a,s_2')\in R\wedge s_2'\notin F\}\\&\cup\{(v_{(b,s_1,s_1',a)},v_{(0,s_1,s_2')})\mid(s_1',a,s_2')\in R\wedge s_2'\in F\}\\&\cup\{(v_{(b,s_1,s_1')},v_{(b,s_2,s_1',a)})\mid(s_1,a,s_2)\in R\wedge s_2\notin F\}\\&\cup\{(v_{(b,s_1,s_1')},v_{(1,s_2,s_1',a)})\mid(s_1',a,s_2')\in R\wedge s_2\in F\}\end{aligned}\tag{9}$$

因为 $G^0_{A,A}$ 和 $G^{di}_{A,A}$ 的所有节点的优先级都相同,只有一个优先级的博弈其选手 0 的获胜域可以在线性时间内计算[9],因为博弈图的规模为 $O(|S||R|)$,因此:

定理 1[10] 给定 Büchi 自动机 A,设其状态数为 n,迁移的数目为 m,则普通模拟和直接模拟的计算复杂度为 $O(mn)$。

因为 $G^{de}_{A,A}$ 和 $G^f_{A,A}$ 为奇偶博弈,Jurdzinski 应用进展度[11]的概念为奇偶博弈获胜域的计算提出了一个高效算法[12]。由于 $G^{de}_{A,A}$ 和 $G^f_{A,A}$ 的优先级只有 3 级,Etessami 等人对 Jurdzinski 的算法进行了优化,提出了一个针对公平模拟及延迟模拟的快速算法[7]。

定理 2[7] 给定 Büchi 自动机 A,设其状态数为 n,迁移的数目为 m,则计算公平模拟和延迟模拟的时间复杂度为 $O(mn^3)$。

5 基于模拟的自动机最小化

自动机最小化在基于自动机的模型检测中具有极为重要的意义。在模型检测中,两个自动机可以相互替代的条件是

这两个自动机所识别的语言相同，因为当自动机识别的语言相同时，它们的时序特征也是相同的。因为自动机识别语言相同性的计算十分困难，为此借助模拟关系及模拟等价来计算。其方法是合并模拟等价的状态即构造商自动机，及去掉一些冗余的边即小兄弟失连。

定义 11(商自动机) 给定 Büchi 自动机 A 和 A 上的等价关系$\approx$，设$[s]$为关于等价关系$\approx$的状态 s 的等价类。则自动机 A 关于等价关系$\approx$的商自动机为 $A/\approx=(S/\approx,R_{\approx},[s_0],L,F/\approx)$，其中 $R_{\approx}=\{([s],a,[s'])\mid \exists s_0\in[s],s_0'\in[s']$ 满足$(s_0,a,s_0')\in R\}$。

定义 12(小兄弟失连) 给定 Büchi 自动机 A 和 A 上的模拟关系 H，状态 s_1 是状态 s_2 的小兄弟当且仅当存在一个状态 s_3 满足下列条件：

(1)$(s_3,s_2)\in R$ 而且$(s_3,s_1)\in R$

(2)$(s_1,s_2)\in H$ 而且$(s_2,s_1)\notin H$

小兄弟失连就是从 R 中去掉迁移(s_3,s_1)。

由于各个模拟关系对公平性的处理各不相同，所以其化简自动机的能力也各不相同。直接模拟是要求最严格的模拟，但是其对自动机的简化程度最小。公平模拟取消了有模拟关系模拟的计算同时进入接受状态的要求，而只要求公平计算能得到模拟。虽然有公平模拟关系的状态更多，但是它们却不能直接用来对自动机进行化简[7]，延迟模拟的要求和简化能力都介于直接模拟和公平模拟之间。以下三个定理揭示了各自简化自动机的能力。

定理 3 设 $A_{\approx}^{de}$ 为 Büchi 自动机 A 关于直接模拟的商自动机，则 $A\equiv A_{\approx}^{de}$，而且对于直接模拟关系，小兄弟失连处理后的自动机与原自动机等价[13]。

定理 4 设 $A_{\approx}^{de}$ 为 Büchi 自动机 A 关于延迟模拟的商自动机，则 $A\equiv A_{\approx}^{de}$[7]，而且对于延迟模拟关系，小兄弟失连处理后的自动机与原自动机不一定等价[14]。

定理 5 设 $A_{\approx}^{f}$ 为 Büchi 自动机 A 关于公平模拟的商自动机不一定与原自动机等价[7]，而且对于公平模拟关系，小兄弟失连处理后的自动机与原自动机不一定等价[14]。

5.1 基于直接模拟/延迟模拟的自动机最小化

根据直接/延迟博弈图的定义构造博弈图 G_A^{di}/G_A^{de}，计算最大模拟关系和模拟等价关系，然后计算商自动机。对于直接模拟关系还要进一步完成小兄弟失连。

5.2 基于公平模拟的自动机最小化

公平模拟等价关系的点的合并不能保证商自动机与原自动机等价，但是大部分的模拟等价点是可以合并的，所以基于公平模拟的自动机优化必须在节点合并和小兄弟失连后判定结果自动机与原自动机是否等价。

定理 6[15] 设 Büchi 自动机 $A=(S,S_0,R,F,L)$ 满足以下条件：$v,w\in S$ 而且 $R(v)=R(w)$，$R^{-1}(v)=R^{-1}(w)$，$L(v)=L(w)$，如果 $v\in F$ 则 $w\in F$。设有自动机 A' 其中 $S'=S\backslash\{v\}$，如果 $v\in S_0$ 则 $S_0'=S_0\cup\{w\}\backslash\{v\}$，否则 $S_0'=S_0$，$R'=R\cap(S'\times S')$，$L'=L\backslash\{v\}$ 则 $L(A)=L(A')$。

根据以上定理，自动机中点的合并可以通过增加边来完成。合并自动机 A 中的状态 v,w，所产生的自动机 A'' 就等价于在原自动机 A 上增加迁移 $\Delta R=(\{v,w\}\times R(\{v,w\}))\cup(R^{-1}(\{v,w\})\times\{v,w\})$ 所产生的自动机 A'。因为自动机 A' 是在自动机 A 上增加了一些边而得到的自动机，所以 $A\leq A'$，即 $A\leq A''$，因此，要证明商自动机 A'' 与 A 等价，只需要另外证明 $A'\leq A$ 即可。

设去掉自动机 A 中边的子集 ΔR 得到自动机为 $rem(A,\Delta R)$，显然 $rem(A,\Delta R)\leq A$，所以要证明小兄弟失连后的自动机 $rem(A,\Delta R)$ 与 A 等价，只需要另外证明 $A\leq rem(A,\Delta R)$ 即可。

设对给定自动机 A 构造对应的博弈图为 $G_{A,A}^{f}=(V_0,V_1,E)$，设 $\Delta R\subseteq S\times S$，定义一个新的自动机 $rem(A,\Delta R)=(S,S_0,R\backslash\Delta R,F,L)$，该自动机是去掉了自动机 A 的迁移的子集 ΔR 而得到的自动机。设博弈图 $rem(G_{A,A}^{f},\Delta R)=(V_0,V_1,E')$，其中 $E'=E\backslash\{((s_1,s,a),(s_1,s'))\mid(s_1,s)\in V_1,(s_1,s',a)\in V_0,(s,s')\in\Delta R\}$。

类似的我们可以定义另外的自动机 $add(A,\Delta R)=(S,S_0,R\cup\Delta R,F,L)$，该自动机是在 A 上增加了边的子集合 ΔR 而得到的自动机。设博弈图 $add(G_{A,A}^{f},\Delta R)=(V_0,V_1,E'')$，其中 $E''=E\cup\{((s,s_2),(s',s_2,a))\mid(s,s_2)\in V_1,(s',s_2,a)\in V_0,(s,s')\in\Delta R\}$。不难证明以下定理：

定理 7[15] 设有 Büchi 自动机 $A,\Delta R\subseteq S\times S$ 为 A 上迁移的子集，$G_{A,rem(A,\Delta R)}^{f}=rem(G_{A,A}^{f},\Delta R)$，而且 $rem(rem(G_{A,A}^{f},\Delta R),\Delta R')=rem(G_{A,A}^{f},\Delta R\cup\Delta R')$。与此相类似，$G_{add(A,\Delta R),A}^{f}=add(G_{A,A}^{f},\Delta R)$，而且 $add(add(G_{A,A}^{f},\Delta R),\Delta R')=add(G_{A,A}^{f},\Delta R\cup\Delta R')$。

根据以上分析，判定商自动机与原自动机是否等价只需要证明博弈图 $add(G_{A,A}^{f},\Delta R)$ 从初始节点出发有获胜策略即可，而且博弈图 $add(G_{A,A}^{f},\Delta R)$ 只需要在博弈图 $G_{A,A}^{f}$ 增加相应的边即可得到。判定小兄弟失连后的自动机与原自动机是否等价只需要证明博弈图 $rem(G_{A,A}^{f},\Delta R)$ 从初始节点出发有获胜策略即可，而且博弈图 $rem(G_{A,A}^{f},\Delta R)$ 只需要在博弈图 $G_{A,A}^{f}$ 上去掉相应的边即可得到。

基于公平模拟的自动机最小化方法分为两步[15]：第一：构造公平模拟博弈图 $G_{A,A}^{f}$，计算最大模拟等价关系；依次对每一对模拟等价的节点进行合并，并判定合并后的自动机与原自动机是否等价，如果不等价则撤销这对节点的合并，如果等价则保留合并。第二：在第一步的结果博弈图上，对所有的小兄弟失连逐一判定，如果与原自动机等价则删除对应的迁移，否则就保留该迁移。

结束语 自动机优化对提高基于自动机的模型检测的效率是一个十分重要的环节，对这一问题研究人员进行了大量研究，取得了丰硕的成果。但是由于自动机优化本身所固有的复杂性使得这一领域尚有大量的课题期待进一步的研究。

因为时序逻辑公式可以更为自然的转化为广义 Büchi 自动机，而且其状态空间较小，方便符号化[16]，所以将现在基于模拟的自动机优化方法应用到广义 Büchi 自动机上是一个极具研究价值的方向。

虽然针对自动机优化已经提出了不少模拟的概念，但是依旧不能最大程度的满足优化的需要，因此寻找更加合适的模拟类型也是今后的一个重点研究方向。

参 考 文 献

1 Milner R. An algebraic definition of simulation between programs. In IJCAI, 1971. 81～489

2 Clarke E M, Grumberg O, Peled D. Model Checking. Massachusetts: MIT Press, 1999

3 Gerth R, Peled D, Vardi M Y, Wolper P. Simple on-the-fly automatic verificationof linear temporal logic. In: Proceedings of the Fifteenth IFIP WG6. 1 International Symposium on Protocol Specification, Testing and Verification XV, London, UK, 1996. 3～18

4 Somenzi F, Bloem R. Effecient Büchi automata from ltl formu-

lae. In CAV, 2000. 248～263
5 Gastin P, Oddoux D. Fast ltl to bautomata translation. In: CAV '01: Proceedings of the 13th International Conference on Computer Aided Verification, London, UK, 2001. 53～65
6 Dill D L, Hu A J, Wong-Toi H. Checking for language inclusion using simulation preorders. In: CAV '91 Proceedings of the 3rd International Workshop on Computer Aided Verification, London, UK, 1992. 255～265
7 Etessami K, Wilke T, Schuller R A. Fair simulation relations, parity games, and state space reduction for Büchi automata. In: ICALP '01 Proceedings of the 28th International Colloquium on Automata, Languages and Programming, London, UK, 2001. 694～707
8 Henzinger T A, Kupferman O, Rajamani S K. Fair simulation. Inf. Comput., 2002, 173(1): 64～81
9 Andersen H R. Model checking and boolean graphs. Theor. Comput. Sci, 1994, 126(1): 3～30
10 Henzinger M R, Henzinger T A, Kopke P W. Computing simulations on finite and infinite graphs. In: FOCS '95 Proceedings of the 36th Annual Symposium on Foundations of Computer Science (FOCS'95), Washington, DC, USA, 1995. 453～462
11 Klarlund N, Kozen D. Rabin measures and their applications to fairness and automata theory. In LICS, IEEE Computer Society, 1991. 256～265
12 Jurdzinski M. Small progress measures for solving parity games. In: Horst Reichel and Sophie Tison, eds. STACS 2000, 17th Annual Symposium on Theoretical Aspects of Computer Science. Proceedings, volume 1770 of Lecture Notes in Computer Science, Lille, France, February 2000. 290～301
13 Bustan D, Grumberg O. Simulation-based minimization. ACM Trans. Comput. Logic, 2003, 4(2): 181～206
14 Bustan D, Grumerg O. Applicability of fair simulation. Inf. Comput, 2004, 194(1): 1～18
15 Gurumurthy S, Bloem R, Somenzi F. Fair simulation minimization. In: CAV '02 Proceedings of the 14th International Conference on Computer Aided Verification, London, UK, 2002. 610～624
16 Juvekar S, Piterman N. Minimizing generalized Büchi automata. In: Thomas Ball, Robert B. Jones, eds. CAV, volume 4144 of Lecture Notes in Computer Science, Springer, 2006. 45～58

(上接第 259 页)

UML 元模型，目的是方便建立互转换机制，以实现 MDA。它与 UML 之间有以下区别：

(1)裁减。取消了许多 Java 中不需要的 UML 元类，而保留了 Java 必需的元类：Element、Classifier、Type、Class、Interface、Feature、BehavioralFeature、Parameter 等。

(2)改变与扩展。把 UML 中类似元类改变为 Java 名称，如 Field、Method、Constructor；根据 Java SE5 的新语言成分，扩展了枚举 Enum、诠释 Annotation、可诠释元素 AnnotatableElement、诠释实例 AnnotationInstance 等，图 2 中也扩展了类属。注意本文仅对诠释部分进行了详细描述。

本文提出的三个诠释元类及两个图符对于 UML 扩展诠释具有指导意义。UML 允许自定义构造型来扩展已有元类，以支持新的建模元素。关于如何扩展 UML 来支持诠释建模，作为另一种技术途径另文再述。

结论与进一步工作 诠释相对于注释，其特征是基于类型、静态实例化、关联目标的实例化、实例不变性。为达到诠释的可视化、规范化建模的目的，本文以 MOF 为规范，参照 UML 元模型，扩展已有 Java 元模型，提出一个新的 Java SE5 元模型，并确定了诠释的图符以支持可视化建模。该元模型能反映诠释的特征，可支持诠释的建模和编程。该元模型具有一致性、规范性、简单性。

进一步的工作包括扩展该元模型以支持 AOP(如 AspectJ5)及 AOM(Aspect-Oriented Modeling)，研究动态诠释及应用，实现元模型以开发 MDA 及相关建模工具等。

参 考 文 献

1 Gosling J, Joy B, Steele G, Bracha G. The Java Language Specification (Third Edition)[M]. Addison-Wesley Professional; 2005 ISBN 0321246780
2 许满武，严悍，张琨，李千目. Java 程序设计[M]. 北京：高等教育出版社，2006，ISBN 704019645X，普通高等教育"十五"国家级规划教材
3 Harold E. An Early Look at JUnit 4 [OL]. http://www-128.ibm.com/developerworks/java/library/j-junit4.html, 2005. 9. 13
4 Hibernate Annotations Reference Guide, v3. 2. 0 [OL]. http://www.hibernate.org/hib_docs/annotations/reference/en/html/, 2006. 4
5 Kiczales G, Mezini M. Separation of Concerns with Procedures, Annotations, Advice and Pointcuts [C]. In: proceedings of ECOOP 2005. Glasgow, UK, July 2005
6 Yan H, Kniesel G, Cremers A B. A Meta Model for AspectJ [R]. Technical Report IAI-TR-2004-3, October 2004, Computer Science Department III, University of Bonn, Germany, ISSN 0944-8535
7 Java Document. Getting Started with the Annotation Processing Tool (apt)[OL]. http://java.sun.com/j2se/1.5.0/docs/guide/apt/GettingStarted.html, 2005
8 OMG(Object Management Group), Unified Modeling Language: Specification: Superstructure [EB]. version 2.0, 2005. 7. 4
9 OMG(Object Management Group), Unified Modeling Language Specification: Infrastructure [EB]. version 2.0, 2004. 10. 16
10 Kleppe A, Warmer J, Bast W. MDA Explained: The Model Driven Architecture——Practice and Promise [M]. Addison-Wesley Professional; 1st edition (April 25, 2003) ISBN: 032119442X
11 Java Document. Mirror API [OL]. http://java.sun.com/j2se/1.5.0/docs/guide/apt/mirror/index.html, 2005
12 Java Document. Reflection API [OL]. http://java.sun.com/j2se/1.5.0/docs/guide/reflection/index.html, 2003

(上接第 263 页)

参 考 文 献

1 Maes P. Concepts and experiments in computational reflection. ACM SIGPLAN Notices, 1987, 22(12): 147～155
2 Bates J. The role of emotion in believable agents. Communication of the ACM, 1994, 37(7): 122～125
3 Maes P. Agents the reduce work and information overload. Communication of the ACM, 1994, 37(7): 31～40
4 Kruchten P B. The 4+1 view model of architecture. IEEE software, 1995, 12(6): 42～50
5 Nwana H S. Software Agents: An Overview. Knowledge Engineering Review, 1996, 11(3): 205～244
6 Magee J, Kramer J. Dynamic structure in software architectures. ACM SIGSOFT Software Engineering Notes, 1996, 21(6): 63～83
7 Richard N T. A Component and Message-Based Architectural Style for GUI Software. IEEE Transaction on Software Engineering, 1996, 22(8): 390～406
8 Wooldridge M. Agent-based software engineering. IEEE Proc. on Software Engineering, 1997, 144(1): 26～37
9 Perkowitz M, Etzioni O. Adaptive web pages: Automatically synthesizing web pages. In: Proceedings of AAAI/IAAI' 98, 1998. 727～732
10 Wermelinger M. Towards a chemical model for software architecture reconfiguration. IEEE Proc-Softw, 1998, 145(5): 130～136
11 Oreizy P, Taylor R N. On the role of software architectures in runtime system reconfiguration. IEEE Proc-Softw, 1998, 154(5): 137～144