开放式环境下基于多 Agent 角色协作的虚拟植物建模方法

梁茹冰[1] 李吉桂[2]

(华南农业大学理学院 广州 510642)[1] (华南师范大学计算机学院 广州 510642)[2]

摘 要 在计算机中模拟植物生长发育情况在农学、林学、遥感和生态学上都有很广泛的应用。静态环境下的模拟工作已做得很好,但在动态环境下模拟工作由于难度较大而成果较少。基于多智能体角色协作机制,提出一种在开放式环境下虚拟植物的建模方法。模型考虑了角色分配、角色回收、基于 L-系统的树形模型算法、各种随机因素影响、剪枝算法和光照影响算法。对于 Agent 技术在虚拟植物建模上的研究提出了一些应用方法和技术路线。

关键词 角色,多智能体,虚拟植物,L-系统,协作

Method for Modeling Virtual Plant System Based on Multi-agent and Role-cooperation in Opened Environment

LIANG Ru-bing[1] LI Ji-gui[2]

(College of Science, South China Agricultural University, Guangzhou 510642, China)[1]

(School of Computer, South China Normal University, Guangzhou 510642, China)[2]

Abstract Simulating the development and growth of living organisms like plants, cells and especially trees has been used widely in agriculture, forestry, bionomics and remote sense. However, most of the models are based on static environment, because of the deficulties in dynamic states. This paper proposed a new method for modeling virtual plant system based on multi-agent and role-cooperation in opened environment. The proposed model considers role assignment and withdrawal, L-systems algorithm, some random factors, pruning algorithm and illuminating influence algorithm. Our work is beneficial to researching on the method of applying agent technology to developing virtual plant model.

Keywords Role, Multi-agent, Virtual plant, L-systems, Cooperation

代理或主体(Agent)是驻留于某一环境中,具有灵活的自主性、反应性、主动性、社会性和移动性等行为特征的计算机软件系统[1]。因此,对于 Agent 和其所在的环境,可以这样描述它们之间的关系(图 1):Agent 通过接收来自传感器(包括硬件传感器和软件传感器)的数据感受环境,同时做出相应的动作来反作用于环境,并影响环境。

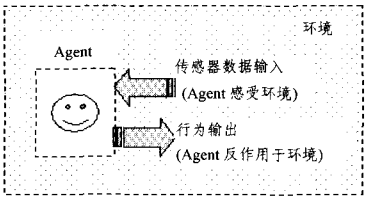

图 1 Agent 与环境的相互作用图

在多代理系统(MAS: Multi-Agent System)中,如果遇到复杂问题、分布式环境、Agent 之间有共同的任务,就需要考虑协作的问题。协作总是发生在 Agent 间有一致的目标,并且通过合作能够求解复杂问题的情况。因此,对于协作的研究,重点是如何协调各 Agent 的行为,以实现系统全局利益的最大化。

多 Agent 马尔可夫决策过程(MMDPs: Multi-Agent Markov Decision Processes)是表示多 Agent 动态协作问题的一种工具。马尔可夫决策过程理论是强化学习的基础,强化学习(如 Q 方法,TD 方法,Dyna 方法)是 Agent 常用的学习算法,其实质是一个搜索和记忆的结合。但在 MMDPs 中协作关系的实现面临系统的联合行为空间和联合效用空间的指数问题,以及 Agent 之间频繁通信问题,这使得该模型在大多数情况下并不适用。另外,现有的基于强化学习方法(如 Q 学习方法)也同样受到计算上高维问题和 Agent 之间频繁通信问题的困扰。

目前有关多 Agent 系统的研究中,主要涉及如何消除多 Agent 系统中协作的冲突;如何解决对立情况下的竞争问题;如何实现多 Agent 的友好合作。基于角色的多 Agent 协作模型将不同角色的能力、知识、任务、要求等内容赋予相应的 Agent,使各 Agent 承担起所要求角色的任务。已有的基于动态角色分配机制的多智能体应用:使命演习仿真[2]、机器人足球赛[3]和机器人协作运输[4]、电子商务协作联盟系统[5]等。

为求解某一目标,系统初始阶段需要为参与者主体分配不同的角色,称此为"角色分配"。当参与者主体集合发生变化,或系统约束发生变化等,系统需调整初始阶段的角色分配,称此为"角色再分配"。角色分配和再分配的不断重复过程称为"动态角色分配"。动态角色分配能够很好地适应开放式环境下的多智能体系统的自主性、适应性、动态性和智能性要求。

到稿日期:2008-04-29 课题研究受到华南农业大学校长科学基金支持(项目编号为 2007K025)。

梁茹冰(1980—),女,讲师,研究方向为智能算法、图像工程;李吉桂(1942—),男,教授,研究方向为 Agent 技术、计算机网络。

在目前的研究中还有一个共同的问题，那就是群体协作机制支持不够，通过研究比较，我们定义了角色在多 Agent 系统协作模型中的涵义和作用，抽象了角色、职责等群体特性及协作关系，并将其应用于模拟虚拟植物在动态环境的建模问题中。

1 模型的全局流程图

模型的全局流程如图 2 所示。

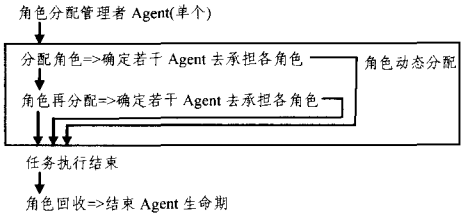

图 2 模型的全局流程图

模型中的 Agent 有：角色分配 Agent、植物生长 Agent(包括叶、茎、花、果实、根等)和环境 Agent。其中环境 Agent 包括：光照、风向、水分分布、土壤肥力、园丁(修剪枝条)等等。

每个 Agent 对应一个或多个角色，角色实际上就是一个权限，或者说功能。例如，角色分配 Agent 具有管理员角色，可以完成角色分配、角色继承、角色再分配等工作；植物生长 Agent 是模拟植物生长的角色，包括叶、茎、花、果实、根等的动态生长过程，及对环境的适应；环境 Agent 是模拟植物外围环境的角色，包括光照、风向、水分分布、土壤肥力、园丁(修剪枝条)等等，该角色会频繁与植物生长 Agent 进行通信，来模拟环境对植物生长的影响。

本模型初步考虑植物生长模型中的形态发生模型(即拓扑结构和几何结构)，不考虑器官造型(如各种纹理、渲染等等)；在生长模拟中，不考虑地下部分和花、果实的模拟；在环境交互方面，考虑了光照和剪枝因素的模拟算法。

2 L-系统方法

L-系统方法是一种基于产生式的规则重写机制，是植物生长 Agent 的主要算法。这里涉及到生长规则的制定。由于对生长规则的界定目前还没有通用的方法，这里根据植物生长模式，将它们分为单轴型生长模式、二分型生长模式和三分型生长模式[6]。图 3 显示了 3 种生长模式下模拟的树型图。

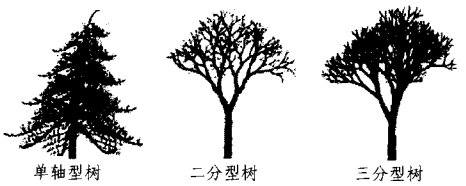

图 3 植物生长模式

进一步抽象，可以得出单轴型、二分型和三分型生长模式的子树枝生长类型[6]，如图 4 所示。

图 4 植物生长类型

L-系统建模算法：

一个 L-系统是一个三元组 (V,W,P)，其中 V 表示一个字符集，$V*$ 是 V 上所有字的集合；$W\in V*$ 为公理的非空字，一般用 W 代表初始元素；$P\subset V\times V*$ 为产生规则的确定集合，用 P 代表系统中的生长规律。

对于单轴型的植物生长模式，其 L-系统规则可写为：

W:X;

P1:X→Y[X];

P2:Y→YY;

对于二分型的植物生长模式，其 L-系统规则可写为：

W:X;

P1:X→Y[X][X];

P2:Y→YY;

对于三分型的植物生长模式，其 L-系统规则可写为：

W:X;

P1:X→Y[X][X][X];

P2:Y→YY;

其中，X 和 Y 构成 V；初始状态 W 均是 X；P1 和 P2 是模拟植物生长的两条规则；“[ ]”是按照 L 系统文法规定表示的枝条方向的偏转。程序运行过程中，如果当前状态是 X，则按 P1 规则生长，如果当前状态是 Y，则按 P2 规则生长。这就是基于字符重写的 L 系统对植物生长过程模拟的基本思想。

为了更好、更真实地对动态环境下植物生长过程进行仿真，我们的系统对上述规则进行了综合及随机化处理：

第一，将 3 种生长模式进行了组合，每次迭代计算机都会随机选择一种生长模式进行模拟；

第二，针对单纯应用规则重写导致的呆板效应，我们添加了随机效果；

第三，运用 Agent 技术对上述过程进行了实现，并且设计了环境 Agent 和角色分配 Agent 来实现动态环境的模拟，体现了植物生长 Agent 对外围环境的自主性、适应性和智能性。

下面，具体介绍模型中的各 Agent，以及各 Agent 协作实现植物形态随机模拟和环境适应性方面的关键技术。

3 植物生长 Agent (PA:Plant Grow Agent)

植物生长 Agent 运用设计好的规则，利用 L-系统方法来生成一棵树。在这里，我们把规则进行了封装，用面向对象的方法进行实现。在植物生长 Agent 这个“黑盒”中，隐藏着各种能够适应环境改变的规则，规则之间的通信是内部流，是私有或保护的，其它 Agent 对规则的参数更改(例如调整分叉角度的随机范围、高速分支成活的概率值等等)设计为一些接口，是公有的，它们是基于消息传递机制的。

基于上面的思想，用 VC++和 OpenGL 编程实现植物生长 Agent 的算法(Plant Growing Algorithm Based on Agent)如下：

(1) 相关参数初始化：$L,n,P_1,\theta_{11},\mu_{11},P_2,\theta_{21},\theta_{22},\mu_{21},\mu_{22},P_3,\theta_{31},\theta_{32},\theta_{32},\mu_{31},\mu_{32},\mu_{33}$；

(2) 从原点处生成一根长度为 L 的竖直的枝；

(3) 判断当前树形层数是否超出规定层数 n，如没有转向(4)，否则转向(9)；

(4) 按概率 P_1,P_2,P_3 随机选择生长模式(单轴型、二分型或三分型)，如果是单轴型生长模式转(5)，如果是二分型生长模式转(6)，如果是三分型生长模式转(7)；

(5) 进入单轴型生长子算法 A_1；

(6) 进入二分型生长子算法 A_2；

(7) 进入三分型生长子算法 A_3；

(8) $n=n+1$ 转(3)；

(9) 树形模拟结束。

算法中相关参数含义说明：

L 为初始长度、n 为迭代次数；$P_1+P_2+P_3=1$；

P_1 为选中单轴型生长模式的概率，θ_{11} 为单轴生长模式下的旋转角度，μ_{11} 为单轴生长模式下的生长系数；

P_2 为选中二分型生长模式的概率，θ_{21}，θ_{22} 分别为二分型生长模式下左右子树的旋转角度，μ_{21}，μ_{22} 为二分型生长模式下各分支的生长系数，取值范围在[0,1]上；

P_3 为选中三分型生长模式的概率，θ_{31}，θ_{32}，θ_{33} 分别为三分型生长模式下三个子树的旋转角度，μ_{31}，μ_{32}，μ_{33} 为三分型生长模式下各分支的生长系数，取值范围在[0,1]上；

这些参数值由角色分配 Agent 和环境 Agent 分别进行设置。下面分别给出单轴型、二分型和三分型生长子算法：

第一，单轴型生长子算法 A1：

(1) 整体向上移动一段距离 L，即树枝长度；

(2) 以原点为圆心，逆时针或顺时针旋转一定角度 θ_{11}，缩放树枝长度为 $\mu\times L$，画出右子树或左子树，画完后顺时针或逆时针旋转 θ_{11}，还原坐标。

第二，二分型生长子算法 A2：

(1) 整体向上移动一段距离 L，即树枝长度；

(2) 以原点为圆心，逆时针旋转一定角度 θ_{21}，缩放树枝长度为 $\mu_{21}\times L$，画出左子树，画完后顺时针旋转 θ_{21}，还原坐标；

(3) 以原点为圆心，顺时针旋转一定角度 θ_{22}，缩放树枝长度为 $\mu_{22}\times L$，画出右子树，画完后逆时针旋转 θ_{22}，还原坐标。

第三，三分型生长子算法 A3：

(1) 整体向上移动一段距离 L，即树枝长度；

(2) 以原点为圆心，逆时针旋转一定角度 θ_{31}，缩放树枝长度为 $\mu_{31}\times L$，画出左子树，画完后顺时针旋转 θ_{31}，还原坐标；

(3) 以原点为圆心，逆时针或顺时针旋转一定角度 θ_{32}，缩放树枝长度为 $\mu_{32}\times L$，画出右子树或左子树，画完后顺时针或逆时针旋转 θ_{32}，还原坐标；

(4) 以原点为圆心，顺时针旋转一定角度 θ_{33}，缩放树枝长度为 $\mu_{33}\times L$，画出右子树，画完后逆时针旋转 θ_{33}，还原坐标。

添加的随机效果有：

(1) 树干长出分枝点的随机性；

(2) 分枝与主干形成角度的随机性；

(3) 同一层树枝长度的随机性；

(4) 树干分枝总数目的随机性等；

(5) 三种生长模式的随机性选择；

(6) 同时增加了树枝粗细变化效果。

从实验结果(图 5)可以看出，添加了上述 6 种随机效果后，模拟出的树形生长情况更符合自然树的生长。

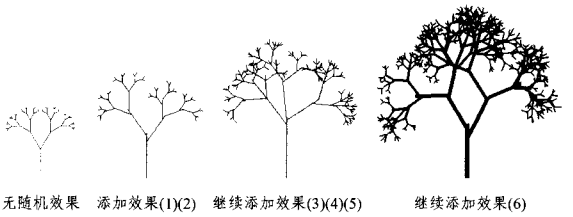

图 5 添加随机因素的模拟效果

4 环境 Agent (EA:Environment Agent)

EA 包括：光照、风向、水分分布、土壤肥力、园丁(修剪枝条)等等。

当虚拟植物的外围环境发生了改变，比如光照从左边变化到右边，这种改变会在某段时间里影响植物的生长(在现实世界中为一个月，在计算机虚拟程序中可能只是一个迭代的时间)。

EA 通过 Agent 通信语言(KQML)将这种体现环境变化的语句写成消息的形式，具体格式遵守 KQML 通信协议。例如上面对于光照方向设置的通信，可以写成下面的形式：

```
Ask-one
  :Sender    Environment_Agent
  :Content   (light = "right")
  :Language  Virtual C Plus Plus
  :Receiver  PlantGrow_Agent
```

环境因素(园丁因素和光照因素)的添加：

(1)剪枝算法

依靠前面提出的算法，可以很容易地构建一棵树的形态，不过如果仅仅依靠几条规则的迭代，而不作其它方面的控制的话，就失去了真实性。

剪枝算法就是一种抑制"顶端优势"的算法。顶端优势是指植株的顶芽在生长上占有优势，顶芽的存在控制侧芽的生长的一种现象。顶端优势现象常常出现在较高大的植物体上，在模型中体现为迭代层数的大小。可以设置一个层数判断的因子 L，当到达这个层数时，就执行剪枝算法，来抑制顶端优势。对于偏转角度过大(如图 6 所示)或过小的枝条，不符合自然界的现象，这种枝条需要剪去。至于角度设置多少为合适，或许每一层都不一样，则要根据实际模拟的植物体属性来界定。

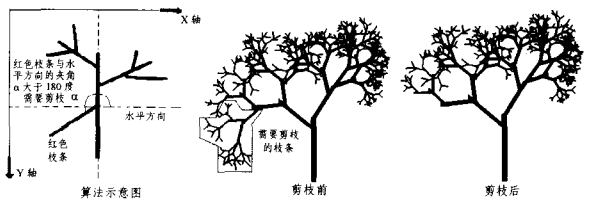

图 6 剪枝算法的模拟效果

(2)添加光照效果的算法

设迭代上限为 n，用于模拟一段时间 T 内的生长过程。根据植物的趋光性，我们在算法中加重向光源方向的生长随机性概率。例如，二分型的生长模式，其两个分支 θ_{11} 和 θ_{21} 在没有光源的影响下随机偏转角度范围均是[25°,60°]；现在添加光照，设光源在右边；我们可以将偏转角度 θ_{11} 和 θ_{21} 的取值区间分别更改为[25°,45°]和[35°,60°]；再将各分支的生长系数 μ_{21} 和 μ_{22} 由原来的均在[0.6,0.8]上取值更改为 μ_{21} 在[0.6,0.75]上取值，μ_{22} 在[0.65,0.8]上取值；对于三分型的

生长模式，同理；这样经过几次迭代后，整个树形必然的偏向于右边，即光源的方向。

图 7 所示是在右边光照下，迭代次数 n 分别取 2,3,4,5,6 时的树形模拟图。

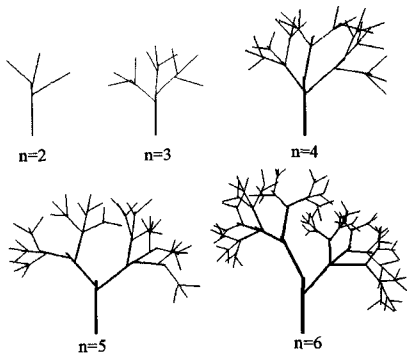

图 7　添加光照的模拟效果

不难发现，对于偏转角度范围的定义可以同时实现剪枝和添加光照效果两个目的，在该角度变化区间里随机取值，很自然地避免了植株的“顶端优势”。

要说明的是，这里参数的取值只是一个经验值，即经过多次实验后的一个能够达到较好模拟效果的值，并不是确定的，区间范围可以更改，这往往取决于具体要模拟的植物品种。

5　角色分配 Agent（RA：Roles Assignment Agent）

每个 Agent 对应一个或多个角色，角色实际上就是一个权限，或者说功能。

角色分配 Agent(RA)具有管理员角色、可以完成角色分配、角色继承、角色再分配等工作；植物生长 Agent(PA)是模拟植物生长的角色，包括叶、茎、花、果实、根等的动态生长过程，及对环境的适应；环境 Agent(EA)是模拟植物外围环境的角色，包括光照、风向、水分分布、土壤肥力、园丁(修剪枝条)等等，该角色会频繁于植物生长 Agent(PA)进行通信，来模拟环境对植物生长的影响。

6　系统的运行

该实验编程工具采用 VC++和 OpenGL 图形库，基于面向对象的思想设计各智能体的功能。

(1) 模拟过程一旦开始，角色分配 Agent(Rolls Assignment Agent：简称“RA”)就会生成两个智能体，分别为植物生长 Agent(Plant Grow Agent：简称为“PA”)和环境 Agent(Environment Agent：简称为“EA”)，PA 和 EA 在植物生长模拟的整个过程中都是运行的，直到树形模拟结束，PA 和 EA 才会由 RA 进行回收。

(2) 通过人机交互，EA 获得初始参数配置，即对前面定义的参数 $L,n,P_1,\theta_{11},\mu_{11},P_2,\theta_{21},\theta_{22},\mu_{21},\mu_{22},P_3,\theta_{31},\theta_{32},\theta_{32},\mu_{31},\mu_{32},\mu_{33}$ 进行全部或部分初始化。

(3) 基于 KQML，EA 与 PA 进行通信，将初始化的参数值以消息的方式传递给 PA，如果是部分参数初始化，将调用默认的构造函数进行参数值定义。

(4) PA 模拟树形生长过程，调用植物生长 Agent 的算法(如第 3 小节介绍)。

(5) 在 PA 运行过程中，EA 也是运行的，它不停地将外围环境的参数传递给 PA，PA 接收环境的信息，并及时调整参数来动态适应环境。

(6) 达到结束条件，树形模拟算法结束，RA 回收分配的各角色，并释放 EA 和 PA 占用的系统资源。

系统的构成如图 8 所示。

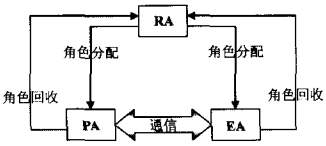

图 8　系统结构图

角色分配 Agent(RA)为系统分配两个 Agent——植物生长 Agent(PA)和环境 Agent(EA)；其中，PA 中涉及的算法有：

- 基于 L-系统方法的植物生长模拟算法；
- 生长过程中各随机量的生成，随机范围通过与 EA 通信获得；
- 剪枝算法；
- 光照影响算法。

EA 的功能有：

- PA 在模拟开始前各参数的初始化值，是由 EA 确定并传递给 PA；
- PA 执行剪枝和光照等算法时所需要的参数的值，也是由 EA 确定并传递给 PA。

上面所列各项模拟的效果图详见前各小节介绍。

结束语　本文提出并构造了一个基于角色分配的多 Agent 协作系统，用于对植物体在动态环境中的生长过程的模拟。该模型是一个多 Agent 系统，利用多智能体之间的协作关系和角色分配机制来完成复杂任务。

模型初步考虑植物生长模型中的形态发生模型，即拓扑结构和几何结构方面，不考虑器官造型(如各种纹理、渲染等等)；在生长模拟中，对植物的叶、枝、茎等形态变化进行了模拟；在环境交互方面，考虑了光照和剪枝因素的模拟算法。从实验结果上看，效果较好，并具有一定的可扩展性。

在多 Agent 系统的研究中，主要涉及如何消除多 Agent 系统中协作的冲突，以及如何解决对立情况下的竞争问题对于这种有冲突的竞争问题，具体表现如多棵植物之间对资源的竞争，以及多个智能体间的协商算法和学习算法。我们试图将这些思想引入模型中，以更好地适应开放式环境下的多智能体系统的自主性、适应性、动态性和智能性要求。

参 考 文 献

[1] Wooldridge M. 多 Agent 系统引论. 石纯一，等译. 电子工业出版社，2003

[2] Tambe M, Pynadath D, Chauvat N. Building dynamic agent organizations in cyberspace[J]. IEEE Internet Computing, 2000, 4(2): 65-73

[3] Kitano H, Tambe M, Veloso M, et al. The Robocup synthetica-gents'challenge[C]// 1997 Int'l Joint Conf. on Artificial Intelligence. Nagoya, Japan, 1997

[4] Chainiowicz L, Mario F M, Kumar V. Dynamic role assign-ment for cooperative robots[C]// 2002 IEEE Int'l Conf. on Robotics & Automation. Washington, DC, 2002

[5] 琚春华，钱阳峰. 基于角色的多 Agent 层次型协作管理模型与商务应用. 计算机工程与设计，2007，28(8)

[6] 毛卫强，耿卫东，潘云鹤. 植物真实感建模方法研究，计算机科学，2000，27(4)

[7] 张树兵，王建中. 基于 L 系统的植物建模方法改进. 中国图像图形学报，2002，7(A)：5