

文章编号:1001-9081(2008)02-0531-04

基于消息通信的多智能体系统的应用

史乐, 李辉, 原江波
(四川大学 计算机学院, 成都 610064)
(liquidshile@gmail.com)

摘要:在基于多智能体的分布式实时仿真系统中,如何实现各 Agent 之间的通信与协调是一个核心问题。采用触发器消息传送方式为多智能体系统中的 Agent 建立了高效的通信机制,并通过分析机场地面交通中面临的问题构建了多智能体系统模型。实现了基于触发器消息通信机制的多智能体系统在机场地面交通仿真中的应用。

关键词:多智能体;触发器消息;通信机制;交通仿真

中图分类号: TP391.9 **文献标志码:** A

Multi-Agent system based on the message communication

SHI Le, LI Hui, YUAN Jiang-bo

(College of Computer Science, Sichuan University, Chengdu Sichuan 610064, China)

Abstract: To implement the cooperation and communication among Agents in the distributed real-time simulation system based on the multi-Agent is a great challenge. In this paper, an effective communication mechanism of Multi-Agent System (MAS) based on trigger message communication is provided and a MAS model has been constructed by analyzing problems of airport ground traffic and put it into practice in the airport ground traffic stimulation.

Key words: multi-Agent; trigger message; communication mechanism; traffic simulation

0 引言

多智能体系统(MAS)是分布式人工智能的重要课题。一个标准的 MAS 包含一组 Agent,各个 Agent 以自治性为前提,通过各种通信环境相互协作。Agent 通信是 Multi-Agent 协作的核心技术^[1]。

基于 Multi-Agent 的分布式仿真对于解决大规模分布式实时仿真模拟系统的设计和验证具有重要的意义。在复杂的工程模型协作设计,网上交互游戏,虚拟交通仿真等领域的应用前景十分广泛^[2]。在应用 MAS 进行分布式实时仿真中,如何实现 Multi-Agent 之间的有效通信和事件同步仍然是个难点。本文研究并验证了将通用触发器消息系统^[3]作为分布式实时仿真系统中 MAS 的通信方式,并在机场塔台模拟机系统^[4]中得到实现。

1 机场地面交通仿真的问题分析

1.1 机场地面道路交通中的地图组织

地图的组织方式是整个交通仿真的基础。本文采用了星状与网状拓扑相结合的组织方式,把道路、路口、停车位、停机坪组织为图结构,方便了车辆自动寻路^[9]。通过带有回溯的深度优先搜索生成了以交叉路口为路径起始点的路径表结构,方便了对路口的查找。

交通系统的道路和交叉路口构成了一个图的结构。该结构定义如下: N 表示节点的集合, $node_i$ 表示一个节点,则集合 $N = \{node_1, \dots, node_k\}$ 。 \vec{R} 表示一条有向道路, $\vec{R}\{node_i, node_j\}$ 表示一条由 $node_i$ 到 $node_j$ 的道路。 $-\vec{R}$ 代表 \vec{R} 的反向,表示 $node_j$ 到 $node_i$ 的道路。 $r\{node_i, node_k, \dots, node_j\}$ 为一个有

序序列,表示 $\vec{R}\{node_i, node_j\}$ 经过的所有节点,即 $\vec{R}\{node_i, node_j\} = r\{node_i, node_k, \dots, node_j\}$ 。对任意 $\vec{R}\{node_i, node_j\}$ 的求解也是系统的功能之一。 $\vec{R}\{cross-node_i, cross-node_j\}$ 表示一条由交叉路口 $cross-node_i$ 到交叉路口 $cross-node_j$ 的道路。本文通过带有回溯的深度优先搜索生成 $\vec{R}\{cross-node_i, cross-node_j\}$ 表结构。 $neighbor(node_x)$ 表示点 $node_x$ 所有的相邻节点组成的集合, $neighbor(node_x) = \{node_m, \dots, node_n\}$ 。 $Width(x)$ 表示道路 x 的宽度或者车辆 x 的宽度, $pos(t)$ 表示车辆/机场的飞机在 t 时刻所在的地图位置。

1.2 Agent 的几何运算与逻辑判断

Agent 的算术和逻辑运算问题是基于 MAS 仿真系统的核心。下面列出了在机场地面交通仿真中各 Agent 遇到的关键算数与逻辑运算。

1) 运动计算

车辆运算的计算公式: $pos(t + \Delta t) = pos(t) + \vec{V}(x) \times \Delta t$; $pos(t)$ 表示车辆原来的位置, Δt 表示新的运算时间累加, $\vec{V}(x)$ 表示车辆 x 的速度。车辆运动计算的关键是位置更新,各个路口的交警 Agent 通过速度更新消息通知车辆 Agent 进行实时的速度更新。

2) 碰撞检测

车辆在 3 种基本碰撞检测中需要实施避让:

(1) 车辆 $Vehicle_i$ 和 $Vehicle_j$ 的行驶路径为 \vec{R}_m ,如果车辆行驶朝向、车辆宽度与道路宽度符合公式: $\vec{R}_m = -\vec{R}_m \cap (Width(\vec{R}_m) < Width(Vehicle_i) + Width(Vehicle_j))$,则 $Vehicle_i$ 和 $Vehicle_j$ 必然相撞。其中 $\vec{R}_m = -\vec{R}_m$ 代表车辆行驶方向相反, $Width(\vec{R}_m)$ 表示路宽, $Width(Vehicle_i)$ 代表车宽, $Width(\vec{R}_m) < Width(Vehicle_i) + Width(Vehicle_j)$ 表示道路宽

收稿日期:2007-08-16;修回日期:2007-12-16。 基金项目:国家 863 计划项目(2006AA12A104)。

作者简介:史乐(1982-),男,山东滨州人,硕士研究生,主要研究方向:图形图像、虚拟现实;李辉(1970-),男,四川成都人,副教授,主要研究方向:图形图像;原江波(1981-),男,河南焦作人,硕士研究生,主要研究方向:计算机图形学、仿真技术。

度小于两辆车的宽度和,排除了可以相向并行行驶的情况。为避免碰撞,路口交警 Agent 会发送强制寻路消息通知车辆 $Vehicle_i$ 和 $Vehicle_j$ 重新寻路。

(2) 车辆 $Vehicle_i$ 和 $Vehicle_j$ 的行驶路径为 \vec{R}_m , 如果车辆行驶朝向、车辆宽度与道路宽度符合公式: $\vec{R}_m = \vec{R}_m \cap (\vec{V}(Vehicle_i) > \vec{V}(Vehicle_j)) \cap (dis(Vehicle_i, Vehicle_j) < safeDis)$, 其中 $\vec{R}_m = \vec{R}_m$ 表示 $Vehicle_i$ 与 $Vehicle_j$ 行驶方向相同, $\vec{V}(Vehicle_i) > \vec{V}(Vehicle_j)$ 表示 $Vehicle_i$ 速度大于 $Vehicle_j$ 的速度, $dis(Vehicle_i, Vehicle_j)$ 代表 $Vehicle_i$ 与 $Vehicle_j$ 之间的距离, $safeDis$ 是一个人为规定的安全距离。这种情况就是同向追尾碰撞,路口交警 Agent 会发送速度更新消息通知 $Vehicle_i$ 减速来避免碰撞。

(3) 车辆 $Vehicle_i$ 由节点 $node_i$ 到节点 $node_j$, 确定的行驶路径为 $\vec{R}_m = r\{node_i, node_k, \dots, node_j\}$; 车辆 $Vehicle_j$ 由节点 $node_x$ 到节点 $node_y$, 确定的行驶路径为 $\vec{R}_n = r\{node_x, node_k, \dots, node_y\}$ 。 $Vehicle_i$ 在 T_1 时刻到达 $node_k$, $Vehicle_j$ 在 T_2 时刻到达 $node_k$ 节点。如果 $T_1 = T_2 \cap (\vec{R}_m \times \vec{R}_n)$, $T_1 = T_2$ 表示两车同时到达, $(\vec{R}_m \times \vec{R}_n)$ 代表 $Vehicle_i$ 和 $Vehicle_j$ 不在同一条道路, 并且道路有交叉。对于这种情况,路口交警 Agent 会发送刹车消息命令 $Vehicle_i$ 或 $Vehicle_j$ 在路口等待来避免碰撞。

2 机场地面交通仿真系统中各种 Agent 的建模

本文把 Agent 作为系统的抽象模型,为每个路口、停车位、车辆建立 Agent 模型,并使用 MAS 结构组织整个系统^[10,11]。

车辆/飞机 Agent ($Vehicle_i$ -Agent_i, VA) 建模如图 1 所示。

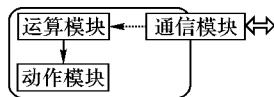


图 1 VA 建模

路口虚拟交通警察 Agent ($Cross_ \& Police_i$ -Agent_i, CA) 建模如图 2 所示。

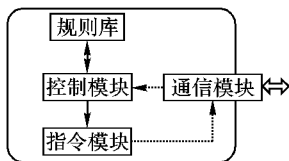


图 2 CA 建模

道路停车位或机场停机位 Agent ($Park_i$ -Agent_i, PA) 建模如图 3 所示。

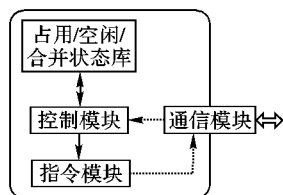


图 3 PA 建模

3 Multi-Agent 的通信方式

3.1 Agent 基本通信类型

Agent 之间的通信是实现 Multi-Agent 相互作用的基础。在 MAS 理论中,Agent 之间的通信方式通常有以下几种方式:

1) 黑板结构的通信方式^[5]: Multi-Agent 以共享数据存储的形式提供公共数据区,实现 Agent 之间的通信和数据共享。Agent 都可以访问共享数据存储区,但 Agent 之间不能直接通信。当仿真系统中的 Agent 数量较多时,共享数据存储区中的数据量会呈指数增长,导致系统效率降低^[6]。

2) 消息结构的对话方式^[6,7]: Agent 之间通过消息系统直接通信。

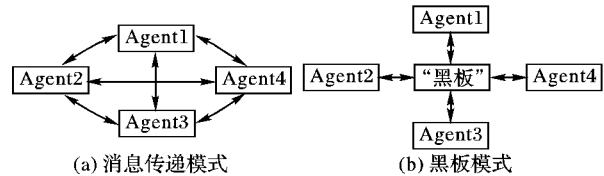


图 4 Agent 通信

3) 其他通信方式^[8]: 远程通信方式和移动 Agent 通信方式等。它们通常不会应用于分布式实时仿真系统,因此不在本文讨论范围内。

3.2 触发器消息系统

在 MAS 中,一个 Agent 不能强迫其他 Agent 执行某个动作。Agent 只能通过通信行为影响其他 Agent。本文采用了触发器消息系统^[3] (Trigger Message System, TMS) 作为 MAS 的通信系统。触发器消息是系统中使 Agent 做出反应的任何“刺激源”^[3]。触发器消息可以通过条件触发器来产生,Agent 可以确定哪些种类的触发器消息对它是有意义的。

3.2.1 触发器消息系统的优点

采用 TMS 的优势就在于避免了轮询 (polling)。而轮询存在一个显著的缺点:每个 Agent 都必须查询整个系统来寻找该 Agent 感兴趣的事件。假设有 M 个 Agent, N 个事件,每次轮询接近 $M \times N$ 次,时间复杂度为 $O(n^2)$ 。

使用 TMS 进行通信,事件发生时,相应的 Agent Message 被注册。TMS 可以根据 Message 的类型和范围来过滤触发器消息,保证每个 Agent 只得到它感兴趣的 Message。Agent 由原来的主动探索变为被动感知,只有外部环境变化时,Agent 才会被通知。这就保证了每个 Agent 只对自己感兴趣的事件做出反应,最大限度的降低了冗余信息,减少了 Agent 的额外处理。假设有 M 个 Agent, N 个事件,使用 TMS, Agent 得到的 Message 数量远小于 N 。时间复杂度为 $O(n)$ 。

TMS 也比轮询系统具有更高的通用性、可重用性和扩展性。当一个新的 Message 被引入系统时,几乎不需要编写额外的代码来处理这个新类型。

3.2.2 Agent 逻辑分组的概念

为了最大化的发挥 TMS 智能过滤的好处,本文在设计和应用中采用了对 Agent 进行空间逻辑区域分组的办法。把对所有的 Agent 进行触发器消息检查变为对 Agent 组进行检查。比如:系统中共有 M 个 Agent,未分组前,TMS 检测一条 Agent Message,需要遍历 M 次。将地图划分为 4 个象限后,TMS 检测一条同样的消息理想状态只需要检测 $[(M/4) + 4]$ 次,检测速度就提高近 4 倍。分组使得只通过一次检测就可以过滤掉大量无关的 Agent,从而最小化处理开销。

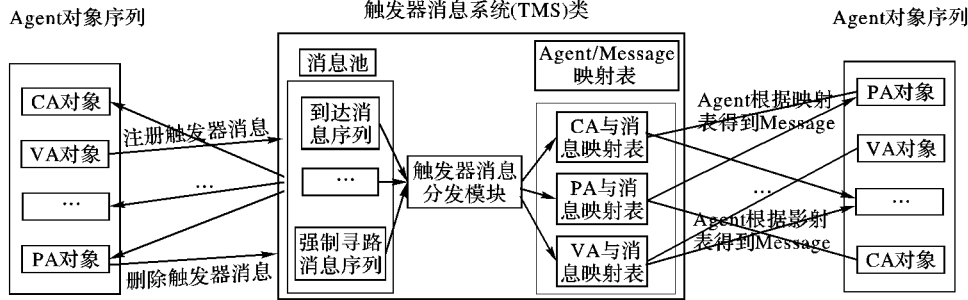
还可以考虑根据其他标准对 Agent 分组,TMS 可以循环检测这些逻辑分组。比如可以把路口 Agent 分为十字路口

组、三叉路口组,等等,都会提高 TMS 的消息过滤效率。如果 TMS 确定某个组对当前的一个触发器消息感兴趣,那么就可以进一步测试组内的每一个成员。组内的成员也可以是

Agent 组,这就形成了一个层次结构。

3.2.3 触发器消息系统设计

触发器消息系统结构与功能,如图 5 所示。



注: VA-车辆/地面飞机Agent; CA-路口虚拟交通警察Agent; PA-停车位/停机坪Agent

图 5 触发器消息系统结构与功能

TMS 是一个维护触发器消息池,过滤后分发触发器消息的类。它提供了注册、删除、更新、分发触发器消息的方法。

1) 触发器消息结构定义。

根据 Agent 之间通信的需要,定义了包含发送、接受 ID、消息类型、附加变量在内的一个结构体。附加变量主要包括最新速度、通用布尔变量、Agent 分组信息等。

2) 注册触发器消息。

消息通过 TMS 提供的注册函数 RegisterMessage() 注册到系统中。TMS 负责创建一个新的触发器消息并且设置其属性。完成后将消息实例的指针返回,将消息压入消息池。

3) 删除触发器消息。

当消息失效后, Agent 调用 TMS 提供的删除函数 RemoveMessage() 删除消息池中的消息。

4) 更新触发器消息。

触发器消息系统的核心是 Update() 函数。这个函数删除过期的触发器消息,刷新动态位置的触发器消息,并通知与某个触发器消息有关的所有智能体。Update() 函数在每个时钟周期内被调用一次。

5) TMS 负责维护的触发器消息池。

Agent message 向 TMS 注册后,压入消息池(TMS Message Pool)保存。为提高查找效率,消息池按照消息类型对 Message 进行分组管理。

6) 触发器消息分发模块。

触发器消息分发模块是 TMS 中较为独立的一个部分,它负责过滤消息和建立 Agent 与该 Agent 独有消息的映射表。是保证 Multi-Agent 高效通信与事件同步的基础。

系统在每个运算周期中调用触发器消息类中消息分发函数 DistributeMessage() 来分发消息池中的消息,建立消息映射表,提高 Agent 获得消息的速度,进而确保系统的实时性。DistributeMessage() 函数首先取得消息池中的消息,然后根据消息的接收 ID 将消息压入消息映射表。各个 Agent 在下个运算周期访问对应的消息映射表,取出消息进行操作。

消息的分发实现了每个 Agent 只用对它感兴趣的事件 Event 做出反应,同时两个运算周期的配合也确保了系统的实时性。

7) TMS 分发消息后产生的 Agent 对应消息映射表。

为避免轮询,进一步提高效率设计了映射表。每个 Agent ID 对应一组经过分发模块检测后生成的消息指针。Agent 对象只需要根据自己的类型和 ID 查询该表就可以快速的得到自己在新的时钟周期要处理的消息。

8) 触发器的逻辑概念。

触发器(Trigger_i)是负责触发器消息产生的逻辑运算机。当满足触发条件时,Agent 就会产生并注册触发器消息。在 CDZS 塔台模拟机机场地面管制仿真子系统中设计了以下几种触发器:

(1) 路口触发器—当车辆 Vehicle_i 到达,离开路口 Cross_i 的逻辑控制区域 Square_i() 时,触发该触发器。

(2) 停车位/停机坪触发器—当车辆/飞机 Vehicle_i 进入或离开停车位/停机坪 Parking_i 的逻辑区域 Square_i() 时,触发该触发器。

(3) 路口虚拟交通警察触发器—虚拟交警触发器是一个较复杂的触发器。它与路口触发器共同绑定在路口虚拟交通警察 Cross_i&Ploice_i-Agent_i 上。该触发器负责监测 Cross_i&Ploice_i-Agent_i 控制的 Vehicle_i, 在满足触发条件时控制 Vehicle_i 刹车,恢复运行,更新速度。同时控制 Vehicle_i 组完成避让。

9) CDZS 塔台模拟机机场地面交通仿真子系统消息种类简介:

(1) Vehicle_i 到达消息。Vehicle_i 到达 Trigger_i 所在 Cross_i&Ploice_i-Agent_i 时触发。

(2) Vehicle_i 离开消息。Vehicle_i 离开 Trigger_i 所在 Cross_i&Ploice_i-Agent_i 时发出。

(3) Vehicle_i 注册消息。Vehicle_i 向 Cross_i&Ploice_i-Agent_i 注册,管制权移交。注册成功后 Cross_i&Ploice_i-Agent_i 向 Vehicle_i 发送接受注册消息,更新 $\vec{V}(Vehicle_i)$ 。

(4) Cross_i&Ploice_i-Agent_i 接受注册消息。Cross_i&Ploice_i-Agent_i 发向注册成功的 Vehicle_i, 附带 Vehicle_i 最新 $\vec{V}(Vehicle_i)$ 。

(5) Vehicle_i 刹车消息。Cross_i&Ploice_i-Agent_i 向受控 Vehicle_i 发出,通知 Vehicle_i 更新 $\vec{V}(Vehicle_i) = 0$ 。

(6) Vehicle_i 恢复行驶消息。Cross_i&Ploice_i-Agent_i 向受控 Vehicle_i 发出,通知 Vehicle_i 恢复原来的 $\vec{V}(Vehicle_i)$ 。

(7) Vehicle_i 速度更新消息。Cross_i&Ploice_i-Agent_i 向受控 Vehicle_i 发出,实时更新 $\vec{V}(Vehicle_i)$ 。

(8) 强制避让消息。Cross_i&Ploice_i-Agent_i 向受控的多辆 Vehicle_i 发出,Vehicle_i 收到此消息后,将在到达最近的 Cross_i&Ploice_i-Agent_i 后重新自动寻路。

4 Agent 消息处理流程

在基于 MAS 组织的交通仿真系统中以各种 Agent 作为

运算的主体,使用消息通信为这些 Agent 传递参数或指令集。在每个运算周期执行的工作流程如图 6 所示。

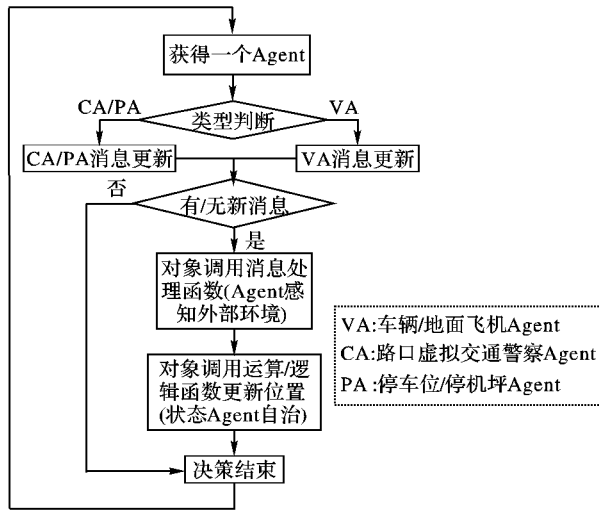


图 6 单个时钟周期内的 MAS 系统流程

5 MAS 在塔台模拟机系统中的应用

以首都机场为例,在机场二次雷达仿真界面上可以看到机场地面交通仿真系统的应用结果:飞机 SAS213 和 JAL123 在通过 s501 点进入各自的停机位时,处在 s501 点的 Agent 对他们进行了实时调度,避免了碰撞,完成了一次交通管制。实际运行状态截图如图 7 所示。

图 7 为首都机场 1 号跑道二次雷达界面,选中的飞机为 JAL123,右侧飞机为 SAS213,初始化 JAL123 启动,SAS213 等待命令。

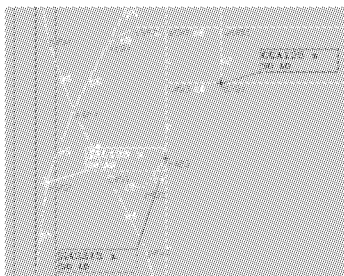


图 7 实验飞机位置初始化

图 8 为 JAL123 到达离 s501 点最近的路口交警 Agent 所在的节点 c501 时,向 s501 点所在的 Agent 发送注册消息。注册成功后,收到 s501 点所在 Agent 发出的接受注册消息。s501 点所在的 Agent 发现拥有更高权值的运动物体 SAS213 需要通过 s501 时向 JAL123 发出的刹车消息,使其保存现有运行状态,停止运行。

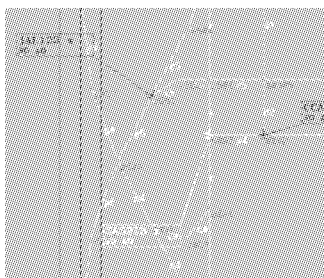


图 8 高权值飞机通过路口

图 9 为 SAS213 接到命令启动,首先向 s501 所在的 Agent 注册,然后从 s403 点出发,根据接受注册消息返回的最新速度通过 s501 节点。

图 10 为 JAL123 收到 s501 点所在的 Agent 发出的恢复行驶的消息,恢复原有运动状态。

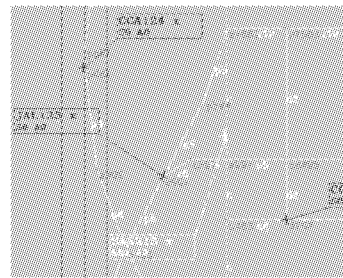


图 9 低权值飞机受控刹车

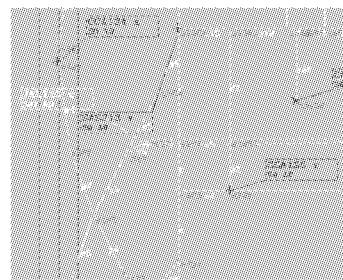


图 10 路口智能体控制成功

6 结语

在基于 Multi-Agent 的分布式仿真系统研究中,如何实现 Multi-Agent 彼此之间的通信和协作是此类系统中关键问题之一。本文采用触发器消息较好的解决了这个难题,在 MAS 分布式实时仿真系统应用中实现了 Multi-Agent 彼此之间的协调通信。并且,设计了一个消息类,通过该消息类实现了消息的注册、删除、更新和按运算周期完成分发。避免了对所有 Agent 的轮询,从而减少了每个 Agent 的冗余处理,提高了效率,具有良好的可扩展性。本文的研究工作已在塔台模拟机系统中得到了应用,并取得了满意的效果。

本文中消息的触发器规则设计是消息系统需要进一步完善的重点,需要考虑各种触发条件和触发器之间的配合,最大限度避免冲突。

参考文献:

- [1] WOOLDRIDGE M. 多 Agent 系统引论 An Introduction to Multi-Agent Systems [M]. 石纯一,译. 北京:电子工业出版社,2003.
- [2] FULLFORD D. Distributed interactive simulation: Its past, present and future [C]// Proceedings of the Winter Simulation Conference. San Diego: ACM Press, 1996: 179 - 186.
- [3] RABIN S. 人工智能游戏编程真言[M]. 庄越挺,吴飞,译. 北京:清华大学出版社,2005.
- [4] 陈凯民. 新一代机场塔台视景模拟机的研究与实现[D]. 成都:四川大学,2006.
- [5] GILMORE J F, ROTH S P, TYNOR S D. A blackboard system for distributed problem solving blackboard architectures and applications [M]. San Diego: Academic Press, 1989.
- [6] 史忠植. 智能 Agent 及其应用[M]. 北京:科学出版社,2000.
- [7] 石教英. 计算机体系结构[M]. 杭州:浙江大学出版社,1998.
- [8] NWANA H S. Software Agents: an overview [J]. Knowledge Engineering Review, 1996, 11(3): 205 - 244.
- [9] FELNER A, KRAUS S, NETANYAHU N. PHA * : Finding the shortest path with A * in an unknown physical environment [J]. Journal of Artificial Intelligence Research, 2004, 21: 631 - 670.
- [10] 曾建勤,刘琨,王国成,等. MAS 技术在交通仿真中的应用[J]. 中国科技大学学报,2006, 36(2): 193 - 197.
- [11] CICORTAS A, SOMOSI N. Multi-Agent system model for urban traffic simulation[C]// 2nd Romanian-Hungarian Joint Symposium on Applied Computational Intelligence. Timisoara: [s. n.], 2005: 5.