

面向 Agent 的车辆监控空间数据调度模型

唐健¹, 邹蓉²

(1. 武汉大学遥感信息工程学院, 武汉 430079; 2. 武汉大学卫星导航定位技术研究中心, 武汉 430079)

摘要:设计面向 Agent 的车辆监控空间数据调度模型, 通过软件 Agent 实体将静态海量地图数据与实时车辆空间数据的调度分解成原子任务进行处理并显示。该模型充分利用 Agent 的自治性、主动性和反应性特点, 自主处理地图数据和车辆信息调度, 在软件层次上实现车辆动态信息显示的实时性, 有效解决了海量地图数据显示造成的响应过慢或停顿问题, 降低了系统设计的复杂度, 为如何将 Agent 思想和方法学应用于 GIS 开发提供了参考。

关键词: 软件 Agent; 车辆监控; 地理信息系统

Agent Oriented Spatial Data Scheduling Model for Vehicle Monitor

TANG Jian¹, ZOU Rong²

(1. School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079;

2. GPS Research Center, Wuhan University, Wuhan 430079)

【Abstract】 This paper discusses an Agent oriented spatial data loading model for vehicle monitor. The model divides tasks of huge static map loading and real time vehicle spatial data loading to atomic task, then displays the data. It takes full use of the Agent's feature: autonomy, pro-activity and reactivity to process map data and vehicle data. As a result, the problem of slowing down the system response caused by displaying vehicle information in real time and huge map data can be resolved. And this model is a good usable example about how to apply the thought and method of Agent to develop GIS.

【Key words】 software Agent; vehicle monitor; Geography Information System(GIS)

1 概述

随着 GIS 技术和 GPS, RFID 等多种定位技术的普及和民用化, 车辆监控及其相应的决策服务成为现实。车辆监控服务一般需要为用户提供实时位置显示和车辆轨迹信息回放等功能, 这些动态信息通常具有较高实时性。监控系统的地图信息较全面, 包括影像、矢量数据和地形数据等, 因此, 需要显示的数据量很大, 地图数据的调度绘制将占用大量系统资源, 影响车辆信息的实时显示。硬件平台的限制, 导致无法将所有数据载入系统。现有车辆监控系统通常在成熟地图引擎上增加动态车辆数据显示功能, 例如国外的 ESRI AO, MapX, 国内的灵图电子地图系统等。这些系统以静态方式配置, 而非动态调度, 无法适应海量 GIS 数据及动态车辆信息显示的应用需求。

为了解决上述问题, 必须在软件层面上建立一个良好的海量地图数据和动态车辆空间信息的同步调度策略, 找到一种动态加载数据的方法。

Agent 的概念最早于 20 世纪 70 年代出现在人工智能领域。它是一个代表所有智能实体的抽象名词^[1]。研究者试图利用此技术开发具有更高智能特征、高度灵活并能更友好地与人类进行交互的计算机系统。20 世纪 90 年代, Agent 被引入软件工程领域, 产生了面向 Agent 的软件工程^[2], 它以 Agent 作为基本的计算抽象和概念模型, 能有效管理并控制复杂软件系统的复杂度。

Agent 可分为智能 Agent 和软件 Agent, 智能 Agent 侧重于研究 Agent 的认知、学习、理性决策、分布式问题求解等

问题; 软件 Agent 侧重于研究如何构造基于 Agent 的软件系统。Agent 通过定义自己的目标、活动、行为、规则、服务、资源等属性可完整地描述系统要承担的角色。Agent 良好的特性使静态地图数据调度、动态车辆数据调度、数据显示等主要功能在逻辑上分离, 分别交由相应的 Agent 来完成, 并可通过消息模式协调各智能体之间的行为。因此, 本文提出了一种基于软件 Agent 的 GIS 数据调度模型。

2 智能体建模方法

对 Agent 的定义仍存在很多争议, 面向 Agent 的系统工程方法和建模方法也是多种多样, 3 种主要方法如下:

(1) Gaia 建模方法。用成员结构、成员社会职责(权利、活动)和成员关系(通信、响应、输入输出等)定义系统。

(2) BDI 建模方法。通过 3 种基本精神状态, 即信念、期望和意图(信念和期望所驱动的行为)描述 Agent 的行为。

(3) MAS-commonKADS 建模方法。将系统分为概念描述阶段和需求定义阶段, 概念描述阶段使用 Use Case 获取用户初始需求, 根据 Use Case 生成消息序列图, 并从消息序列图中识别角色及其相互关系。

比较上述 3 种建模方法, Gaia 方法与面向对象的方式具有相似的定义; BDI 和 Mas-commonKADS 方法侧重于模型

基金项目: 地理空间信息工程国家测绘局重点实验室开放基金资助项目(200622)

作者简介: 唐健(1981-), 男, 博士研究生, 主研方向: 地理信息系统, 空间数据挖掘, 智能交通系统; 邹蓉, 博士研究生

收稿日期: 2007-09-24 E-mail: tangjian_win@163.com

思想、理念等概念层次,是描述思维的有效手段,Gaia 方法的可操作性和实用性较高,软件实现较容易。

按体系结构的不同,Agent 可分为以下 6 种类型:

(1)简单反应式 Agent,简单地对外部刺激产生响应,没有内部状态。

(2)包含自身状态的反应式 Agent。

(3)慎思式 Agent,又称为认知式 Agent,具有现实符号模型的基于知识的系统。

(4)目标驱动的 Agent,与反应式 Agent 不同,基于目标的 Agent 在实现目标方面更灵活,只要指定新的目标,就能产生新的作用。

(5)效用驱动的 Agent,只有目标无法产生高质量的作用,一个具有显式效果函数的 Agent 能作出理性决策。

(6)复合式 Agent,在一个 Agent 内组合多种相对独立和并行执行的智能状态,其结构包括感知、动作、反应、建模、规划、通信和决策等模块。

本文提出的数据调度模型在综合考虑可操作性、实用性及模型系统需求的基础上,采用 Gaia 系统建模思想分别定义调度模型中的几种反应式 Agent,如图 1 所示。通过各 Agent 独立完成静态地图数据调度、动态车辆数据调度、任务分配和图形显示工作。

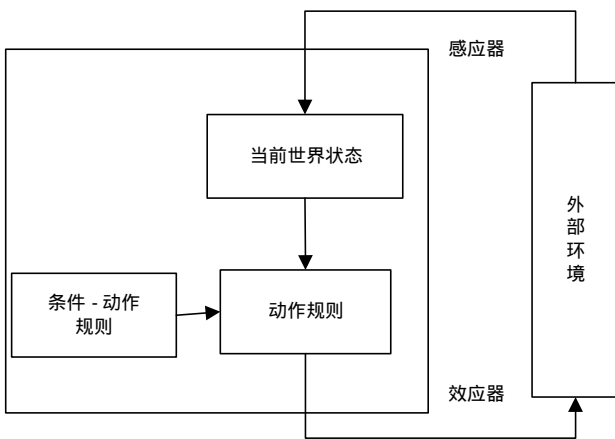


图 1 反应式 Agent

3 系统模型

3.1 车辆监控 GIS 系统

车辆监控系统是智能交通系统、车辆和运营管理服务系统的一部分。它将全球卫星定位系统(GPS)、交通地理信息系统(TGIS)和无线数字通信技术(GMS)融为一体^[3]。一个完整的车辆监控系统应提供车辆实时位置、行驶状态、紧急情况报警及车辆控制等功能。其主要工作流程如下:

(1)由安装在车辆上的 GPS 终端接收 GPS 卫星测定的当前车辆位置。

(2)车辆终端采集系统采集当前车辆的诸如行驶速度、油量等状态信息。

(3)无线通信网络将上述信息发送回监控中心和数据存储中心。

(4)监控中心服务系统将位置和其余状态属性信息匹配在电子地图上,直观地显示车辆相对位置。监控中心还可根据情况向各车辆终端发送控制命令。

监控中心需要同时管理大量地图数据和车辆位置等动态信息。因此,需要一个良好的可扩充数据调度引擎。

软件 Agent 技术为车辆监控系统的设计提供了全新模式。利用 Agent 思想将监控中心系统按功能分解为独立工作、但相互协调联系的 Agent 模块,从而降低系统设计的复杂度,让数据加载和显示分离,互不干扰。上述方法在软件层次上解决了车辆动态信息显示的实时性与海量地图数据显示造成的响应过慢、停顿问题。

3.2 Agent 划分

为了便于模型的进一步扩展,本文数据调度模型按功能模块将整个车辆监控中心划分 4 个子 Agent 模块,分别为静态地图数据调度 Agent、动态车辆信息调度 Agent、数据显示 Agent、任务管理 Agent。模型体系结构如图 2 所示。

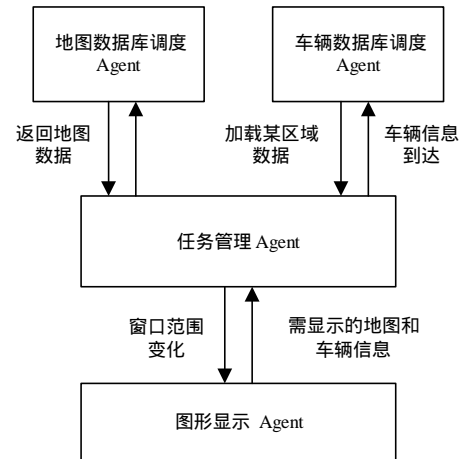


图 2 模型的体系结构

地图数据调度 Agent、车辆数据调度 Agent 和图形显示 Agent 是简单的反应式 Agent,它们接收环境信息的改变,根据变化信息做出相应动作。地图数据调度 Agent 能感知外部环境,即任务管理器发出的加载某一区域地图数据的信息,根据此消息加载该区域数据,加载完成后通知并返回数据给任务管理 Agent。

车辆数据调度 Agent 不接收环境信息,它在模型启动时主动接收车辆的位置、状态等信息并告知外部环境车辆信息的变化。图形显示 Agent 根据用户通过平移、放大、缩小设置的显示区域通知任务管理器显示区域的变化,将任务管理器传递的图层数据和车辆信息显示在图形窗口上。

任务管理 Agent 是一个具有内置状态的反应式 Agent,它记录了模型已访问过的地图区域及其访问次数。模型每次访问地图数据时,会首先检查访问的历史信息,若数据已被加载,则直接作用反馈给图形显示 Agent,若未被加载,则通知地图数据 Agent 加载数据。如果有数据区域在地图窗口变化 N 次后仍未被访问,则自动将该数据区域从内存删除。任务管理 Agent 通过上述策略保证系统内存中只有一定数量的地图数据,不会超过系统处理能力,从而实现海量地图数据的漫游显示。

3.3 模型实现

本实验系统采用 Java 编程语言实现,Java 语言与平台无关,采用虚拟机技术,是网络应用的优秀程序设计语言,可一次开发,为系统日后的分布式扩展应用奠定了基础。图形图像的绘制采用了 JOGL(Java OpenGL),JOGL 绑定了 OpenGL 的大部分功能,并具有与 OpenGL 一致的访问接口。JOGL 可以与 Java2D 很好地融合,实现在 OpenGL 的绘图中使用 Java2D 功能。

本系统按功能分解为多个独立的 Agent 模块。每个 Agent 由其相应的主类和附属功能类组成。为了使各个 Agent 拥有独立的工作能力，系统为每个 Agent 附加了线程类，Agent 由此线程类派生而来。各个 Agent 类的具体实现如表 1 所示，AgentThread 模块结构如图 3 所示。

表 1 数据调度模型各个 Agent 类

主类、包名	实现功能
class com.wuhan.agent.AgentThread	线程类,为各 Agent 提出主动工作的动力
class com.wuhan.agent.MapAgent	地图数据加载 Agent
class com.wuhan.agent.GPSAgent	车辆信息加载 Agent
class com.wuhan.agent.MapViewAgent	图形显示 Agent
class com.wuhan.agent.TaskAgent	任务管理 Agent
package com.wuhan.event	各个 Agent 之间的通信规则实现包

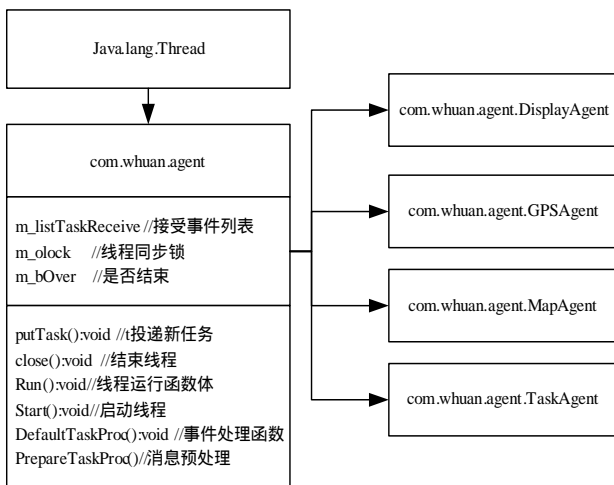


图 3 AgentThread 结构

作为 Agent 动轮源的 AgentThread 线程类主要提供了以下 3 个控制接口：

- (1)用于启动 Agent 的 start()函数。
- (2)停止 Agent 运行的 close()函数。
- (3)用于外部环境向 Agent 投递任务包的 putTask(Object tPack)函数。

Agent 排队处理任务包，并根据当前自身状态决定是否丢弃部分不重要的任务。例如，当车辆信息更新频率较高时，可以丢弃一部分数据，不会对数据显示效果造成较大影响。

Agent 在运行时，相互直接依靠消息机制通信，因此，系统特别定义了一组消息包，可分为地图区域改变消息 MapBoundsEvent、地图区域数据调度消息 MapLoadEvent(包括矢量数据加载消息和影响数据加载 2 个子消息)、车辆实时位置消息 GPSDataEvent、车辆轨迹数据消息 TraceDataEvent (该数据包放在 com.wuhan.event 下)。

4 实验及分析

本实验系统在普通 PC 机，Windows 操作系统下与利用 ESRI ArcObject 地图引擎实现的动态地图数据调度和车辆监控系统进行了模拟对比实验分析。使用 3 幅地图进行测试，

地图 1、地图 2、地图 3 的数据量分别为 30 MB, 30 MB, 100 MB，包含的动态车辆数分别为 100, 200, 300。实验结果如表 2 所示。

表 2 2 类数据模型的处理速度对比

地图编号	数据模型	帧率/(f.s ⁻¹)
1	ArcObject 版本	24
	Agent 版本	30
2	ArcObject 版本	20
	Agent 版本	28
3	ArcObject 版本	12
	Agent 版本	24

实验结果表明，在相同数据量前提下，Agent 版本模型的地图和车辆实时调度处理速度比 ArcObject 版本模型快，而且随着地图数据和车辆数据的增加，ArcObject 版本模型的处理速度下降较明显，Agent 版本模型下降平缓。

根据上述模型，建立 4 个逻辑上独立主动运行的 Agent 实体，分别负责地图数据加载、车辆数据加载、图形显示和任务管理的工作。各个部分是独立工作的，相互直接通过异步消息通信，在单机上如果不考虑机器性能对各 Agent 运行效率的影响，用户将看到一个很平滑的数据加载和车辆信息动态更新并行显示的效果，且任何平移、放大缩小操作都不会存在难以忍受的停滞现象。任务管理器所提供的数据加载和删除策略，保证了系统在同一时刻内存中的数据量不会超过硬件的负荷，从而实现海量地图数据的实时加载和显示。地图数据和车辆数据 Agent 各自独立，保证了车辆监控信息的实时显示。

实验结果表明，基于 Agent 的车辆监控 GIS 数据调度模型，具有较高的自治、主动和反应能力，在海量地图数据和车辆数据的同步加载的速度方面具有显著优势。

5 结束语

基于 Agent 思想的软件系统仍处于初级阶段，在软件层次上，现阶段的 Agent 只是 Object 对象的延伸和扩展，其本质还是对象 + 线程，但随着人工智能和软件工程的发展，Agent 最终将成为主流系统设计思想和手段，与计算机紧密连接的 GIS 领域也会受到深远影响。本文提出的基于 Agent 的车辆监控数据调度模型从软件层次上解决了海量静态地图数据和动态车辆信息调度同步调度显示的问题，为 GIS 在监控领域的应用提供了新方法，能在站点之间迁移的移动 Agent^[4]技术，使构建分布式车辆监控数据调度模型、充分利用网络的计算能力成为可能。本文在如何将 Agent 思想应用于 GIS 系统开发方面也做了有益的尝试。

参考文献

- [1] Wooldridge M, Jennings N R. Intelligent Agents: Theory and Practice[J]. Knowledge Engineering Review, 1995, 10(2): 1-67.
- [2] 毛新军. 面向 Agent 的软件开发[M]. 北京: 清华大学出版社, 2005.
- [3] 鲍远律, 夏冰. GPS 车辆监控系统开发的关键技术[J]. 中国公路: 交通信息产业, 2001, 9(3): 42-44.
- [4] 张云勇. 移动 Agent 及其应用[M]. 北京: 清华大学出版社, 2002.