

基于多 Agent 的大型地下建筑智能控制系统

杨启亮, 邢建春, 王 平

(解放军理工大学工程兵工程学院, 南京 210007)

摘要: 以大型地下建筑工程分区监控结构为基础, 提出了基于 Multi-Agent 的大型地下建筑工程智能控制系统的控制模型。该模型主要由操作管理 Agent、中间接口 Agent、区域监控单元 Agent 和工艺监控单元 Agent 等组成。讨论了操作管理 Agent 和区域监控 Agent 等重要 Agent 基于 UML 的行为模型和具体实现。

关键词: 多 Agent; 地下建筑工程; 智能控制系统; 通用建模语言

Intelligent Control System of Large-scale Underground Buildings Based on Multi-Agent Technology

YANG Qiliang, XING Jianchun, WANG Ping

(Engineering Institute of Engineering Corps, PLA University of Science and Technology, Nanjing 210007)

【Abstract】 On the basis of the multi-section structure of the monitoring-control system for large-scale underground buildings, one multi-agent based control model for the buildings are brought forward. The model consists of operation and management Agent, middle-interface Agent, domain-monitoring-control agent and work-flow-monitoring-control agent. The behavior models of the two important agents, operation and management agent and domain-monitoring-control agent are probed and described in depth with UML, and the detailed implementations of the two agents are discussed.

【Key words】 Multi-Agent; Underground buildings; Intelligent control system; Universal modeling language (UML)

地下建筑作为一类特殊的建筑工程, 在民用及国防领域发挥着重要的作用。随着智能楼宇等的研究逐步深入, 地下建筑工程的智能化已开始引起研究人员的关注。智能控制系统(即设备监控系统)是实现地下建筑智能化的基础。由于地下建筑特别是大型地下建筑的特殊性和复杂性、非线性、时变性、不确定性等特征, 使基于精确数据模型的传统控制理论面临严峻的挑战^[1]。目前的智能控制系统多采用递阶控制系统的刚性体系结构, 建筑工程内部通过网络将各个子系统连接起来, 具有最初级的智能, 但基本上还不具备人类智能的特点, 不具备推理、自学习、自适应等能力, 当前的智能建筑还属于“智而不能”的状况^[2]。将人工智能中的相关理论和技术有效地融合到此类系统中去, 使集成后的系统具有更高的智能, 是智能控制系统的发展趋势。人工智能领域中 Agent 及多 Agent 系统理论和技术产生和发展为解决当前智能控制系统面临的问题提供了新的途径^[1]。

本文设计了一种地下建筑工程智能控制系统的分区监控结构, 采用多 Agent 的方法对地下建筑工程智能控制系统进行了分析和实现。

1 多 Agent 技术概述

Agent 是用以说明一个具有自主能力、社交能力、反应能力和预动能力的软硬件系统^[3]。1986 年 Minsky 在《心智社会》(Society of Mind)中首次提出了 Agent, 认为社会中的某些个体经过协商可求得问题的解, 这个个体即是 Agent。Agent 的基本思想是使软件能模拟人类的社会行为和认知, 即人类社会的组织形式、协作关系、进化机制, 以及认知、思维和解决问题的方式。和传统的对象概念相比, Agent 概念具备更多

的知识、主动性和协作性, 具有更强的问题求解能力和自治能力。

多 Agent 系统是指若干 Agent 通过协作完成某些任务或达到某些目标的计算系统, 其更为一般性的含义是指由多个自主或半自主的构件所构成的各种大型的系统, 这种系统具有传统分布、并发问题求解的特点和能力。多 Agent 系统区别于传统智能系统的显著特征在于它具备与其所处环境、其他 Agent 和人进行交互、协调和协作的能力^[4]。

多 Agent 系统中的各个 Agent 成员与子程序、函数或过程有着本质的区别, 其目标与行为是相对自主与独立的。多 Agent 系统的协作求解问题的能力超过单个 Agent, 这是多 Agent 系统产生的最直接的原因。

2 基于多 Agent 的大型地下建筑智能控制系统

2.1 大型地下建筑工程的分区监控

大型地下工程一般担负着重要的作战保障任务, 地理区域庞大, 工艺设备繁多, 分布着大量的设备(发电机组、除湿机组、风机、电动阀门等)和工艺子系统。将工程内设备按地理位置或功能划分为若干区域, 每个区域配置一台监控设备, 称为区域控制器, 负责本区域内的设备监视和控制。

图 1 为基于区域控制器的分区监控实现方案。系统由操作站、区域控制器、其他工艺系统监控站(消防报警系统、安全保卫系统等)等设备组成。

作者简介: 杨启亮(1975 -), 男, 硕士、讲师, 主研方向: 计算机分布式控制系统, 防护工程智能化系统, 智能控制; 邢建春, 博士、教授; 王 平, 硕士、副教授

收稿日期: 2006-07-28 E-mail: yql@893.com.cn

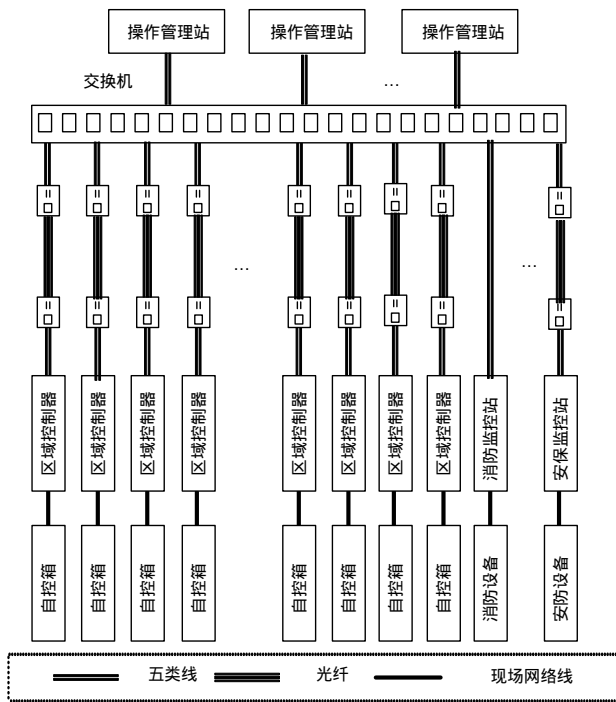


图1 智能控制系统分区监控设计方案

按照功能划分和区域划分相结合的原则决定所需配置的监控系统。

(1) 区域监控子系统

系统中分布着大量的智能监控模块(本系统主要指 LonWorks 网络 I/O 模块)及带智能口的智能设备,用于监控工程中大量设备(如水泵、空调除湿机等)。将整个工程划分为若干区域监控子系统,各区域配置一个区域控制器,实现对本区域内设备的监测和控制。

(2) 工艺设备监控子系统

工程中有些子系统,负责专门特殊的工艺操作,功能要求比较复杂、系统相对独立,可以设置专门的监控计算机,构成各自的监控子系统如工程中的备用电站、变电所、安全防范等监控子系统等。

(3) 操作管理站

区域控制器和工艺子系统主机直接面向现场,属于分散控制,分散控制对全局问题可能产生不可行解。要解决这一问题,需要一个协调机制,操作管理站用于实现协调功能,每一个操作管理站和区域控制器或子系统主机都可成为一个自治的控制单元,并用一种协调机制来协调控制决策。

本文提出了基于多 Agent 的分布式、协调式的智能控制系统实现方案。从图 1 可以看出,在相应层次的各个部分引入 Agent 是完全可行的,每一功能单元都可由单个 Agent 或组织良好的 Agent 群来实现。在递阶控制结构中引入 Agent,克服了递阶控制固有的刚性问题,使系统具备了自主性、自治性、协调性、协作性等良好品质。

2.2 基于多 Agent 的监控层控制结构

基于多 Agent 的控制系统结构有两种:(1)纯分布式结构,即系统中的所有 Agent 相互共享信息和知识,这种结构适用于规模较小的系统,因为当系统规模很大时,系统的通信量将会非常大,同时这种结构的局部自治性强,不易达到全局任务的目标。(2)联邦式结构,这种结构引入了中介子(Mediator)的协调机制,中介子将一组 Agent 聚集成为 Agent 集合,集合内部的 Agent 通过中介子进行通信和协调,基于

这种结构的多 Agent 系统简化了纯分布式系统的通信与控制的复杂性,能够解决复杂的实际问题。

本文设计的智能化系统监控层采用了如图 2 示的混合式结构,一方面保证了系统的柔性性与自适应性,另一方面简化了系统的结构,降低了系统的通信与控制的复杂性。

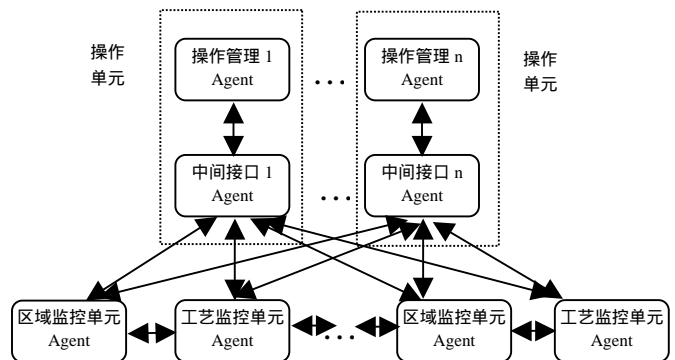


图2 基于多 Agent 的系统结构

各 Agent 的主要功能如下:

(1)区域监控单元 Agent。区域监控单元 Agent 主要负责对本区域的设备对象进行局部的过程监控,其通过 LonWorks 现场网络监视和控制现场设备。实时采集运行数据,并向其它 Agent 转发采集来的数据,同时能及时响应其它 Agent 发来的控制命令,并将命令发送到 LonWorks 网络中进而控制设备。具有友好的人机交互界面,能同系统维护管理人员在现场进行交互。

(2)工艺监控单元 Agent。该 Agent 主要负责对某一工艺系统进行操作、管理和监控(也可实现对其他遗留系统的封装),如空压机站工艺监控单元 Agent 能完成对高压空压机的启动、手自动排污、卸荷、故障报警、停机等一系列复杂的工艺操作。提供有同其它 Agent 进行交互的通信接口(如 OPC 等)和人机交互界面。

(3)操作管理 Agent。操作管理 Agent 是整个监控层的运行枢纽,几乎所有的监控功能都需要在该 Agent 上体现出来,如按地下工程的技战术要求实现协调控制、流程图监视、远程控制、数据报表、事件记录等。操作管理 Agent 又可分为风系统操作管理 Agent、水系统操作管理 Agent、电系统操作管理 Agent、复杂工艺操作管理 Agent 等多种形式。

(4)中间接口 Agent。该 Agent 是操作管理 Agent 与各区域监控单元 Agent 或工艺监控单元 Agent 间交互的接口,负责收集、整合数据和转发控制命令。

3 Agent 的行为模型

Agent 的行为模型是指对各个 Agent 的内部结构、外在功能、运作方式的刻画,是各个 Agent 编程的基础。采用通用建模语言(UML)来刻画 Agent 的行为模型,可使用标准的表示法(如用例关系图、协作关系图等)来定义各种用户和应用程序活动。

用例图是在开发的初期对创建过程的叙述性说明。用例是主角通过系统完成一个过程时出现的一组事件,通常用例是较大的过程,而不是步骤或事务。协作图用以显示对象角色之间的关系,如为实现某个操作或达到某种结果而在对象间交换的一组消息。协作图忽略了时间的作用。

(1) 区域监控单元 Agent

区域监控单元 Agent,是整个监控层系统的基础单元。各 Agent 自主管理一个区域,负责相应设备的操作管理、控

制命令传递和数据采集。

该 Agent 的用例图如图 3 所示，其描述了区域监控单元 Agent 的基本功能和与外部角色间的交互情况。

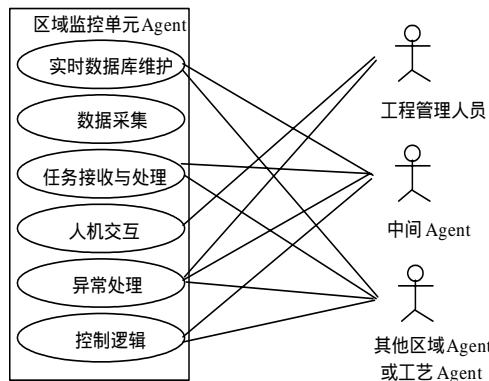


图 3 区域监控单元 Agent 用例图

1)实时数据库维护。实时数据库是以内存的形式动态存储着 LonWorks 网络采集来的本区域设备的运行参数。实时数据库的维护主要包括实时库的创建、实时库规模的动态增加或减小、实时数据记录的实时更新与访问等内容。

2)数据采集。对现场设备的运行数据的采集是该 Agent 的重要任务，数据采集由专门的数据采集线程实现，采集来的数据定时写入到实时数据库中。

3)任务接收与处理。接收其他 Agent 来的任务指令，为提高对指令的执行效率，设计了一个指令队列，实现对批量指令的缓存和管理。

4)人机交互。提供图形用户界面(graphic user interface, GUI)为工程管理人员提供直观的、生动形象的设备运行工艺流程图、故障报警信息、控制按钮等画面，实现工程管理人员的就地监控。

5)异常处理。按照设定的数据限范围，根据采集来的设备运行参数，实时判断设备运行故障与否，并以不同的报警级别为其他 Agent 或工程管理人员显示异常信息，在严重情况下能发出控制指令，强制停止故障设备的运行或自动采取其他保护措施。

6)控制逻辑。应用于特殊区域的区域监控 Agent，需要执行复杂的逻辑运算实现设备联动或调节功能，如批量控制、设备定时巡检、PID 调节等。对于这种复杂的控制逻辑的算法执行，Agent 提供了专门的控制逻辑运算线程。

图 4 为区域监控单元 Agent 的协作图，示意了构成区域监控单元 Agent 的示例对象，并详细列出了程序内部各个程序模块之间为完成用例图所描述的用例而产生的消息传递关系及消息内容。图 4 中的消息编号并不代表严格的时间上的先后顺序。图 4 中，1 为设备控制逻辑的触发命令，控制逻辑模块接收后执行相应的逻辑(如火警的联动逻辑)；2 为控制逻辑执行后产生的控制命令，缓存到指令队列；3 为其他 Agent 来的任务指令，如操作管理 Agent 发来的开排风系统指令；4 为控制逻辑运算时从实时库取出的设备运行参数；5 为向其 Agent 发送设备运行数据；6 为向操作管理 Agent 发送的异常报警信息；7 为异常处理所需的设备实时运行数据；8 为向人机界面发送的报警信息；9 为向人机界面发送的设备运行参数；10 为就地控制指令；11 为采集的现场设备运行数据(包含设备的故障信息)；12 为发往 LonWorks 网络的控制命令；13 为异常情况下的紧急停机指令。

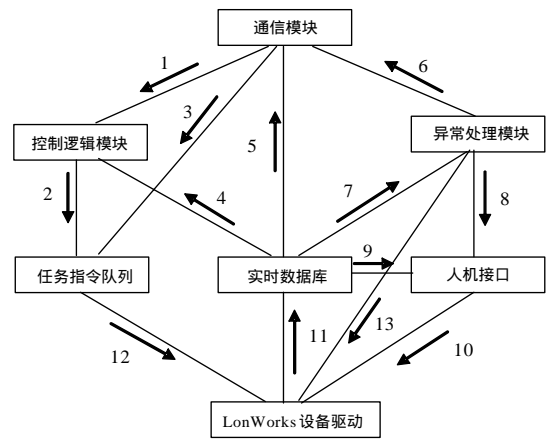


图 4 区域控制单元 Agent 协作图

(2)工艺监控单元 Agent

工艺监控单元 Agent 主要负责对单一的工艺子系统(如造水子系统、供油子系统等)管理、监控和组织生产，其行为模型基本同区域监控单元 Agent 相同，主要差别是该 Agent 所连接的 I/O 设备不一定是 LonWorks 网络，可能是 RS-232、RS-485 或其他类型的协议网络。在此不再讨论其用例图和协作图。

(3)操作管理 Agent

操作管理 Agent 负责对整个工程设备运行过程、生产过程的监控、调整、调度，负责系统内各个单元 Agent 的运行协调，负责处理监控或生产活动中异常事件。该 Agent 的用例图，如图 5 所示。

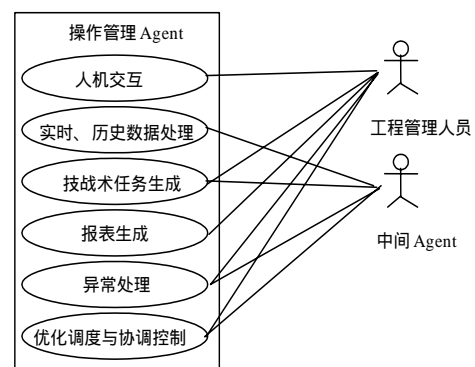


图 5 操作管理 Agent 用例图

由图 5 可知，该 Agent 主要有交互、实时历史数据处理、技战术任务生成、数据报表、异常处理、优化调度和协调控制等功能组件构成，实现同工程管理人员、中间 Agent 等的交互。

1)人机交互提供图形化的操作界面，便于工程管理人员进行操作和及时发现监控过程出现问题。

2)实时、历史数据处理主要处理来自中间 Agent 的现场实时数据，并将实时数据转储为历史数据，以便于长期保存和进行数据统计分析。

3)技战术任务生成是指工程管理人员根据工程当前所处的战术环境条件利用操作管理 Agent 进行任务指令的下达，该任务指令通过中间 Agent 转发给相关区域监控单元 Agent 和工艺监控单元 Agent，进而控制现场设备，完成任务目标。

4)报表生成主要利用该 Agent 中的设备运行的实时和历史数据，对设备的运行状况进行统计分析，并可进行深层次的数据发掘、故障诊断等。

5)优化调度与协调控制是操作管理 Agent 的非常重要的功能, 实现在工程范围内对工程中的所有子系统进行调度和协调。如消磁系统的用电负荷较大, 则在进行消磁工艺作业时, 需要使发电机组满负荷运行, 这就是消磁子系统与发电子系统间的协调控制; 又如, 在进行消磁作业时, 所有发电机组都投入运行仍不能满足需要, 可停止一些区域的除湿机组和空调机组来减轻工程用电负荷的压力以保障消磁的需要, 这属于优化调度。

操作管理 Agent 的异常处理功能基本与区域监控单元 Agent 相同。图 6 为该 Agent 的协作图, 1 为管理层接口 Agent 生成的技战术任务指令; 2 为监控层生成的技战术任务指令; 3 为到人机界面的实时数据; 4 为到人机界面的历史数据; 5 为到管理层接口 Agent 的实时数据; 6 为到管理层接口 Agent 的报警信息; 7 为异常处理所需的设备实时运行数据; 8 为数据分析综合用的实时数据; 9 为实时数据流; 10 为数据分析综合用的历史数据; 11 为异常情况下的紧急停机指令; 12 为从中间 Agent 来的现场设备运行数据(包含设备的故障信息); 13 为发往中间 Agent 的设备控制命令; 14 为发往中间 Agent 的技战术任务指令。

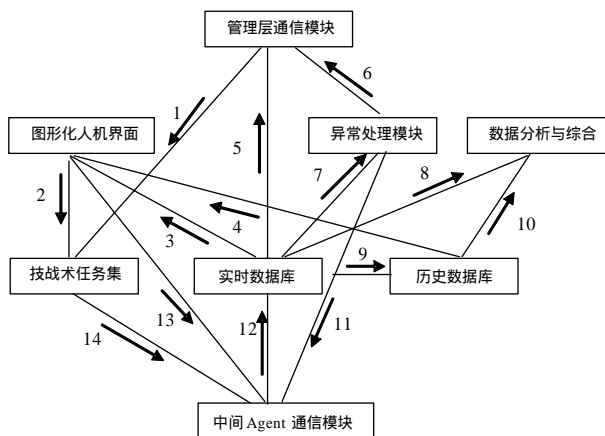


图 6 操作管理 Agent 协作图

(4)中间接口 Agent

中间接口 Agent 是操作管理 Agent 与各单元 Agent 的(区域监控单元、工艺监控单元)通信接口。负责向各单元 Agent 转发监控层来的各种控制指令, 同时向监控层上传各单元 Agent 发来的现场设备运行数据。这种接口 Agent 结构较简单, 在此不再讨论其用例图和协作图。

4 Agent 的实现策略与原理

多数多 Agent 方法应用者都倾向于使用一些知名的 Agent 开发环境来生成多 Agent 系统; 如: IBM 的 ABE (Agent building environment) 等。采用现成的 Agent 开发环境, 可以集中精力于应用领域问题的解决, 省去构造 Agent 程序生成工具、通信协作机制等麻烦。但也存在如下问题^[4]: (1) 这些开发环境所提供的推理引擎难于胜任比较复杂的工业应用, 特别是大型建筑工程智能化控制领域中的应用。(2) 开发环境一般不提供二次开发工具, 而采用这些 Agent 开发工具来生成 Agent, 很难根据应用需要对 Agent 结构、功能、运行环境等进行扩展。(3) 开发系统本身在操作系统、异质环境应用等方面的限制, 不便于利用 Agent 封装遗留系统, 难以保证多 Agent 系统的异质性和开放性。(4) 复杂的系统程序通常要涉及数据库访问、文件存取等多方面的问题, 一般的多 Agent 生成工具很难做到像成熟的商品化编程工具那样提供很多这方面的便利。

因此, 本研究采取了自行开发 Agent 程序的方案。在此重点讨论具有代表性的区域监控单元 Agent 的实现如下:

区域监控单元 Agent 采用 Visual Basic 作为开发工具实现, 该开发工具比 VC++ 有着更强的界面窗体生成能力。下面主要讨论该 Agent 若干重要部分的具体实现。

(1)实时数据库的构建

实时数据库用来存放和管理 LonWorks 网络扫描线程扫描来的实时数据, 为 Agent 中其它模块提供实时数据。实时数据库采用动态结构数组实现, 其结构定义代码如下:

```
Type DataPoint
    PointOrder As String    '测点编号
    BoardNum As Long       '模块地址
    ChannelNum As Long     '通道号
    BoardType As String    '模块类型
    IOType As String       'I/O 类型
    AlarmDownLimit As Single '报警下限
    AlarmUpLimit As Single '报警上限
    CurValue As Single     '当前值
    AlarmTimeLength As Long '存储连续的报警持续时间, 在实现保护程序时便于判断时间
    AlarmCategory As String '报警类别
    PointName As String    '测点名称
End Type
```

实时数据库规模大小(数组长度)由 Agent 初始化时确定。Agent 启动时首先扫描 LonWorks 网络, 识别出网络中测控模块类型和数量, 然后根据类型和数量生成实时数据库。

(2)命令队列的实现

命令队列用来按照先进先出的队列运算算法缓存从中间接口 Agent 发来的控制命令。本 Agent 的队列采用结构数组实现, 设计的队列的最大能存储 256 控制命令。同时也实现了对队列进行操作的出队和入队算法, 在此不再列出。队列实现代码如下:

```
Type CTRLMSG
    lAddr As Long          '队列元素-消息的结构
    sCmd As String        '模块地址
    cmdNum As Long        '控制命令
    a(3) As Single        '参数个数
    a(3) As Single        '参数数组
End Type
Public CtrlMsgQueue(1 To 256) As CTRLMSG '命令队列
```

(3)通信接口设计

该 Agent 提供了 3 种接口同其它 Agent 通信: 1) OPC 接口, OPC 接口是工业控制领域通用的软件接口, 这里控制器端软件作为 OPC Server, 同中间 Agent 的 OPC Client 通信; 2) WinSock 方式, 这种通信方式直接面向 TCP/IP 协议, 具有很高的通信速率, 但需要复杂的编程技术; 3) Remote API 方式, 其实质是 COM/DCOM 接口提供的一组方法, 通过调用这组接口方法实现对 LonWorks 网络的读写, 这种方式专为 LonWorks 网络远程管理 Agent 所使用。

(4)数据采集与命令输出

读写线程用于实现对 I/O 设备的读写。该线程在用户设备配置文档加载时创建, 线程根据设备配置信息按照设定周期扫描相应 I/O 设备通道, 并将扫描结果写到实时数据库中。在扫描的过程中, 线程同时检查写标志 WriteCmdIsComing, 如果 WriteCmdIsComing 为 True, 说明在写命令队列中存在写命令, 则线程按照先进先出的算法读取写消息队列中的写

(下转第 246 页)