

基于移动代理和本体的实时证券监管系统的设计

Design of Real - Time Stock Monitoring System Based on Mobile Agent and Ontology

余文利 (衢州学院 信息与电子工程系 浙江 衢州 324000)

摘要: 针对目前实时证券监管系统数据传输量大导致实时性差的不足,本文提出了一种基于 Mobile agent 和 Ontology 的实时证券监管系统模型,在该模型中引入了 Mobile Agent、Ontology 及数据挖掘技术,使得系统在实时性、冗错性及智能性方面得到了提高。

关键字: 客户机/服务器 实时证券监管系统 Mobile agent Ontology 数据挖掘

传统的实时证券监管系统是基于客户机/服务器 (C/S) 模式的,通过远程过程调用或消息传递等方式^[1,2]进行远程通信,由于数据传输量大而目前证券网络带宽较小,导致系统实时性差,影响了证券业务的正常进行。移动 Agent 技术集智能 Agent、分布式计算、通信技术于一体,提供了一个强大的、统一的和开放的计算模式,更适用于提供复杂的网络服务。移动 Agent 最有效的应用之一就是分布式信息检索,尤其是在移动信息处理情况下的应用。通过利用程序代码到数据端的移动,移动 Agent 技术能够降低在每个单个步骤的潜伏时间,避免媒介数据在网络之间的传输^[3]。

本文将移动 Agent 技术、Ontology 技术和数据挖掘技术引入到实时证券监管系统中,对原系统进行了重新的设计,提出了一种基于移动 Agent 和 Ontology 的实时证券监管系统模型,该模型很好地解决了原系统中存在的问题。

1 实时证券监管系统模型

系统模型包含三个子系统,分别为客户端子系统、监管中心子系统和营业部子系统(如图 1)。

在该模型中,每个子系统均包含一个移动 Agent 中间件平台,该平台为系统内的 Agent 提供创建、运行和销毁等 Agent 生命周期活动的环境以及其它有关的服务。Agent 之间以 FIPA ACL 作为通信协议,所有消息和数据用 XML 表达,并将这些消息和数据存放到系统智能目录系统 IDS (Intelligent Directory System) 中。

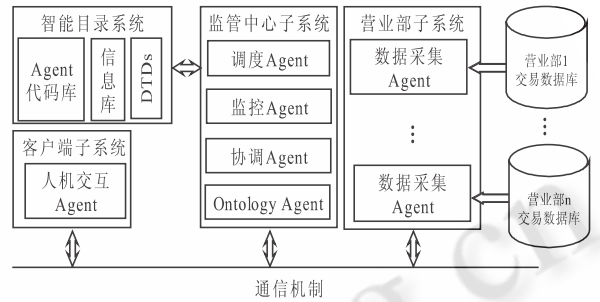


图 1 基于移动 Agent 的实时证券监管系统模型

智能目录系统不仅包含这些消息,它还存放移动 Agent 所需要的代码库,Agent 的特征信息,以及 XML 的文档类型定义 (DTDs)。

1.1 监管中心子系统

该子系统是整个实时监管系统的核心,实现基于 FIPA 标准进行相互通信的基础架构,并具有创建和部署 Agent 的功能。在该子系统中,存在四个 Agent,分别是调度 Agent,监控 Agent、协调 Agent 及 Ontology Agent。

(1) 调度 Agent 管理 Agent 平台的活动,包括创建、删除 Agent,负责营业部移动 Agent 平台的注册以及监视 Agent 从一个平台向另一个平台的移动,同时还维护着系统中当前所有 Agent 的索引。

(2) 监控 Agent 周期性地和智能控制 Agent、数据采集 Agent 和协调 Agent 通信,从而收集这些正在运行的 Agent 信息,并按照一定的组织方式将这些信息写到信息库中。当数据采集 Agent 发生意外事件(比

如异常死亡)需要恢复时,监控 Agent 可以通过查询信息库,将数据采集 Agent 恢复到最近的状态。

(3) 协调 Agent 为系统内的 Agent 提供黄页服务,是 Agent 目录管理者。其他 Agent 可以向该 Agent 注册服务,也可以通过该 Agent 查询某种服务的提供者 Agent。

(4) Ontology Agent 专门为其他 Agent 提供 Ontolog 服务,保证所有 Agent 对通信语言有一致的理解,实现语义上的一致性。为了向其他 Agent 提供服务,它必须在协调 Agent 上注册。

1.2 营业部子系统

该子系统只有数据采集 Agent,该 Agent 是一个移动 Agent,由调度 Agent 创建并移动到营业部子系统中,负责实时过滤证券交易信息,并将收集的违规信息实时上传给监管中心。

1.3 客户端子系统

该子系统负责和用户的交互,只有人机交互 Agent。该 Agent 是一个静态 Agent,在客户端启动后自动运行,提供用户登录验证、设置监控阈值及查询违规数据等服务。

2 系统中 Agent 的结构

系统中六种类型 Agent 分布在三个子系统中,共同协作来完成实时监管的任务。而每个 Agent 可以用如图 2 所示的通用结构来构建。

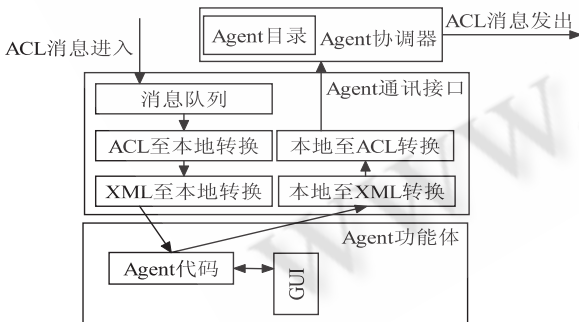


图 2 Agent 的通用结构

图 2 中的每个 Agent 都包括 Agent 协调器、Agent 通信接口和 Agent 功能体。Agent 协调器是 Agent 相互通信的控制中心;Agent 通信接口为 Agent 与人或其他 Agent 的通信提供了一个转换接口;Agent 功能体则包含了完成具体任务的 Agent 代码,只有人机交互 A-

gent 才有 GUI 图形界面。

3 系统结构实现的关键技术

3.1 Agent 之间的通信

在基于移动 Agent 的实时证券监管系统中,Agent 之间的通信采用的是 ACL (Agent Communication Language) 通信语言。ACL 是 FIPA^[4] 提出的,包括传递的消息格式、消息内容的表示及通信的动作以及交互协议的描述。如:

```
( request
: sender ( agent - identifier: name dataGathering
@ stockDep1: 1099/JADE )
: receiver ( set ( agent - identifier: name monitoringManager@ host: 1099/JADE )
: content " ( ( action ( agent - identifier: name monitoringManager@ host: 1099/JADE ) Warn turnover ( stock ShangHai ) )
: language fipa - sl
: ontology turnoverWarning
: protocol fipa - request
: conversation - id " 1"
```

在上面的消息中,营业部数据采集 Agent 的名称为 dataGathering,调度 Agent 名称为 monitoringManager,数据采集 Agent 位于机器名为 stockDep1 的计算机上,端口号为 1099,调度 Agent 位于机器名为 host 的计算机上,端口号为 1099。dataGathering 向 monitoringManager 发送了一条通信动作为 Request 的消息,该消息遵守 FIPA - REQUESTU 交互协议,要求 monitoringManager 收到消息后,针对单笔股票成交量超标执行一个名为 Warn 的行为。Warn 的行为在 Ontology 的 " turnoverWarning " 中定义,该行为要求 trunoverWarning 对上证所股票交易单笔超标进行报警。

3.2 Agent 的恢复机制

为了保证数据采集 Agent 可靠运行,在每一数据采集 Agent 运行的主机上,均有一后台挽救进程与之对应。挽救进程负责监视数据采集 Agent 的生命周期,并以一定周期通过预定的端口定时检查数据采集 Agent 的状态。当数据采集 Agent 在规定的时间内没有响应,挽救进程加快发送信号的频率。如果数据采

集 Agent 仍然没有响应,在发送一组信号后,就假定该 Agent 失效。此时挽救进程在操作系统进程表中查找有无该 Agent 对应的进程,如无则挽救进程通知调度 Agent 重新创建数据采集 Agent,并用 IDS 中保存的该数据采集 Agent 最近的信息进行初始化,再分派到证券营业部中。

上述机制要能正常进行,前提是确保挽救进程始终活着。因此,与挽救进程周期性地检查数据采集 Agent 相似,数据采集 Agent 反过来也以一定的周期通过信号检查挽救进程的状态。当然重新实例化挽救进程的工作由数据采集 Agent 来完成。

3.3 数据采集 Agent 的设计

设计思想:该 Agent 能对营业部的客户进行自动聚类,将客户分为特户、大户、中户和散户,给分类后的客户打上相应的标记。数据处理时,先判断采集数据的客户类别,再将采集数据(如单笔成交量)与该客户类别相对应的阈值进行比较,判断是否是违规数据,如果是则上传至监管中心。为了实现客户自动聚类,在该 Agent 中引入了改进后的层次聚类算法。

以下是该算法进行客户自动聚类的过程:

1) 数据的预处理

使用数据库工具对营业部交易数据库进行数据清洗、转换生成了客户聚类分析所需要的 Customer Clustering 事实表(如表 1 所示)。

表 1 Customer Clustering 事实表

字段名	字段说明	类型	长度
ZJZH	资金账号	CHAR	12
XM	姓名	CHAR	8
KHLB	客户类别	CHAR	2
ZJGM	资金规模	MONEY	NOT NULL
XCSR	息差收入	MONEY	NOT NULL
YJ	佣金	MONEY	NOT NULL
CZPL	操作频率	INT	NOT NULL

将所得数据进行标准化,再利用相关性分析将客户聚类相似度公式构造为:

$$d(i, j) = \sqrt{\mu(m_i - m_j)^2 + \eta(c_i - c_j)^2 + \varphi(a_i - a_j)^2 + \omega(f_i - f_j)^2} \quad (1)$$

在式(1)中, m_i 表示资金规模(资金余额+证券市值)指标, μ 表示该指标的影响因子,值为 0.2325; c_i 为佣金收入指标, η 表示该指标的影响因子,值为 0.4945; a_i 表示息差收入指标, φ 表示该指标的影响因

子,值为 0.0051; f_i 表示操作频率指标, ω 表示该指标的影响因子,值为 0.2673。下面根据上述内容对客户数据进行聚类。

2) 执行客户聚类分析,具体算法如下:

设定:客户集合 $G = \{g_1, g_2, \dots, g_i, \dots, g_n\}$,其中, g_i 为单个客户,以资金账号字段(ZJZH)标示。

(1) 将 G 中的每个客户看作是一个具有单个成员的类 $c_i = \{g_i\}$,这些类构成 G 的一个聚类 $C = \{c_1, \dots, c_i, \dots, c_n\}$;

(2) 根据加权的聚类相似度公式(式(1))计算 C 中每对类(c_i, c_j)之间的相似度 $d(c_i, c_j)$;

(3) 选取具有最大相似度的类对 $\arg \max \text{sim}(c_i, c_j)$,并将 c_i 和 c_j 合并为一个新类 $c_k = c_i \cap c_j$,从而构成 G 的一个新聚类 $C = \{c_1, \dots, c_n - 1\}$;

(4) 重复上述步骤,直至 C 中剩下最后一个类为止。

通过上述计算,建立一个表示客户聚集关系的生成树。经过对聚类过程的进一步分析,取第 $n-3$ 层的四个类别作为营业部的客户分类。

本文使用上述算法对营业部中具体数据进行了测试,测试结果如表 2 所示。通过将表 2 的聚类结果与人工划分的训练集比较,共有 529 个账号发生了类别迁移,迁移率为 2%,即客户聚类的准确率为 98%,由此可以证明该算法在客户自动聚类方面是有效的。

表 2 层次凝聚方法客户聚类结果

	账户数	平均 资金规模	平均 佣金收入	平均 息差收入	平均 操作频率
特户	116	5962415.83	95896.30	2035.69	613.49
大户	293	163843.20	23082.41	79.56	225.63
中户	572	96253.29	10567.46	71.89	168.21
散户	20808	18023.82	653.25	6.09	13.37

3.4 Ontology

为了保证 Agent 在交换消息时能相互理解,Agent 之间必须在领域的模型上或至少在彼此交换信息所涉及领域的模型上取得一致。在解决特定问题的多 Agent 系统中,需设定一个专门的 Ontology Agent 为其他 Agent 提供以下服务:发现公共的 Ontology^[5]以便用来访问;向协调 Agent 注册、更改一组公共的 Ontology;促进两个通信 Agent 之间共享本体论的一致性。

应用 Agent(如数据采集 Agent、调度 Agent 等)通

过 Ontology Agent 可以方便地访问到自己所需要的 Ontology, 其服务模型如图 3 所示。

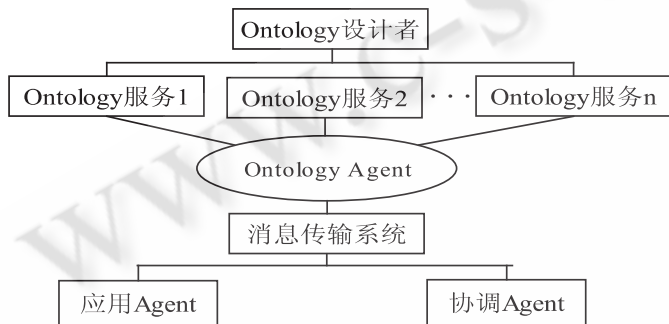


图 3 Ontology 服务模型

4 结束语

实时证券监管系统具有实时性、分布性和并行性等特点, 本文将 Mobile Agent 技术、Ontology 技术和数据挖掘技术引入到系统的设计中, 使系统在满足以上要求的同时还增加了智能性的特点。该系统具有很强的可扩展性、适应性和实用性。

参考文献

- 1 Z P Lazar, P Holfelder. Web Database Connectivity with Scripting Languages. World Wide Web J, Spring, 1997.
- 2 S P Hadjiefthymiades, D I Martakos. A Generic Framework for the Development of Structured Databases on the World Wide Web. In: Proc Fifth Int'l World Wide Web Conf, 1996 - 05.
- 3 Yang J, Honavar V, Miller L, et al. Intelligence Mobile Agents for Information Retrieval and Knowledge Discovery from Distributed Data and Knowledge Source. IEEE Information Technology Conference, Syracuse, NY, 1998.
- 4 Yannis Labrou, Tim Finin. Semantics for Agent Communication Language. Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, 1996.
- 5 Stojanovic L. Methods and tools for ontology evolution. Germany: University of Karlsruhe, 2004.