

自动售货机金钱管理系统中 AOP 框架设计与应用^①

鲍 陈, 汪千松

(安徽工程大学 现代教育技术中心, 芜湖 241000)

摘 要: 针对传统自动售货机软件体系结构中软件各组件间关系复杂、组件职责不清晰等问题基础上, 转向 AOP 技术, 提出了一种面向自动售货机实时系统软件开发领域的可复用、分层的面向方面软件实现框架 R-AoSAS, 该架构使得功能组件模块和非功能组件模块分离, 为了实现功能组件和非功能组件的绑定或组合, 该框架通过调用方面的 Join Point 作为它们之间的接口。在应用该框架进行实时系统软件开发时, 对于定时服务、任务创建和任务调度等非功能约束的实现可以交给 R-AoSAS 框架核心控制层提供的时间管理模型和任务调度模型去完成, 从而在软件产品层面上提高系统软件开发的的质量和软件的复用性。为了验证 R-AoSAS 的性能, 将该框架应用于金钱管理系统的开发设计, 从该系统的应用中充分体现了该框架的应用前景。

关键词: 面向方面; 自动售货机; AOP 框架; 编织技术

Design and Application of AOP Framework for Vending Machine Money Management System

BAO Chen, WANG Qian-Song

(Modern Education Technology Center, Anhui Polytechnic University, Wuhu 241000, China)

Abstract: Considering the complex relationship between the various components of software components, not clear responsibilities, this paper introduces aspect-oriented programming technology and proposes a kind of reusable and hierarchical aspect-oriented software implementation framework—R-AoSAS (Real-time Aspect-oriented Software Architecture Style) for the facing of vending machine real-time system in software development. This framework can effectively separate functional components and non-functional components. In order to bind or combine functional components and non-functional components, this framework can call aspectual Join Point as their interface. When applying R-AoSAS to software development, we may use time-management model and task-scheduling model supplied by the core control layer of R-AoSAS, to meet non-functional requirements such as timing, task creation and task scheduling, thus improving quality and reusability of software. In order to validate the functions of R-AoSAS, this paper introduces this framework to the design of the money management system. This application of the system has fully reflected the prospect of application.

Key words: aspect-oriented; vending machine; AOP framework; weaving technology

自动售货机是一种用于销售盒装、灌装等有形、防震防压性能好的商品的机器^[1]。本文介绍的自动售货机金钱管理系统是自动售货机软件系统应用层的重要组成部分, 它作为自动售货机实时系统的子系统, 其主要功能是将顾客从入金口投入的纸币(日元)搬送到收纳库, 在这个过程中同时对纸币进行真伪检测

以及对纸币进行分类。

目前, 随着自动售货机广泛应用, 自动售货机实时系统软件的复杂性也在不断增加, 传统的面向对象框架技术, 在自动售货机软件的开发中存在诸如软件模块化程度低、体系结构层次不清晰, 可维护性、可复用性差等许多问题^[2]。

^① 收稿时间:2011-09-05;收到修改稿时间:2011-10-08

为了使我们在自动售货机实时系统软件可持续提高,本文提出了一种基于 AOP 的自动售货机实时系统软件开发框架 R-AoSAS (Real-time Aspect-oriented Software Architecture Style),该架构使得功能组件模块和非功能组件模块分离,为了实现功能组件和非功能组件的绑定或组合,通过调用方面的 Join Point 作为它们之间的接口,与传统的面向对象框架相比,降低了关注间的关联度,对于定时服务、任务创建和任务调度等非功能约束的实现可以交给 R-AoSAS 框架核心控制层提供的时间管理模型和任务调度模型去完成,从而在软件产品层面上提高系统软件开发的质量和软件的复用性。

1 传统框架设计存在的问题

自动售货机实时系统软件具有硬件控制和实时控制两大特征,因此实时系统软件通常由对硬件的控制、对时间的控制,及管理上述两类控制的应用逻辑三部分构成,在传统的实时系统软件开发中,通常采用如图 1 所示的软件体系结构^[3]。

分析这种软件体系结构可以发现有许多问题,主要表现为:①软件各关注模块间的关系复杂。不仅模型层与控制层之间的关系复杂,而且根据所实现功能的需要,模型层间或者控制层间的组件间可能都会发生复杂的关联关系;②当某一模块功能发生变更时,与之相关的其它模块也要发生相应的变更。由于关注模块间关系紧密,当某一组件的功能发生变更时,势必导致与之相关的组件实现发生变更,这尤其表现为当某一硬件发生变更或者增加硬件时,与该硬件相关的控制层就会发生非常巨大的变化,使得软件维护性变得非常困难;③由于各个组件的耦合度高,组件职责不清晰,软件重用性变得非常差。

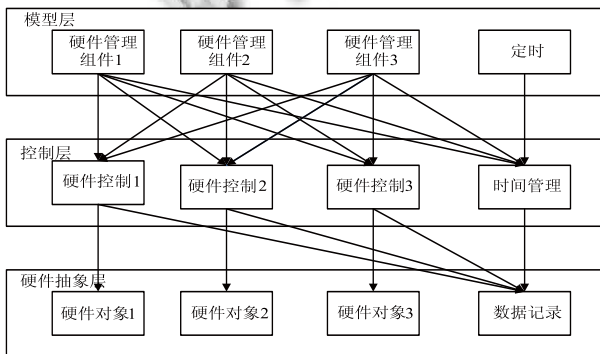


图 1 传统的软件框架

2 R-AoSAS框架设计

2.1 总体设计

针对原有框架的三个突出问题,根据多元关注分离(Multi-Dimensional Separation of Concern, MDSOC)的思想^[4-5],我们在原有开发框架中做出了比较大的改动,从而构建一种适合自动售货机实时系统软件开发领域的可复用的分层面面向方面软件实现框架 R-AoSAS(Real-time Aspect-oriented Software Architecture Style)。如图 2 所示,本框架分为 3 层,自顶向下依次为,具体业务层、公共业务层、核心控制层。①具体业务层。该层根据功能与非功能分类原则,构建的应用程序,核心功能需求只负责完成系统具体的业务性需求,非功能需求完成系统非业务性需求,将非功能需求从功能组件模块中分离出来,为它们单独设计组件模块。正面核心功能组件和侧面非功能组件间的接口由原来的直接的函数调用变成隐式调用方面的 Join Point,同时对并发性、实时性、任务调度等非功能性需求的实现交给公共业务层和核心控制层去完成。②公共业务层。该层是我们改进原有开发框架中,新添加的一层。它完成业务域中绝大多数应用任务需要的公共服务功能,主要有两个关键部件:一个是主动对象类 CActiveObject,它是具有相同性质和行为的主动对象的集合;另一个是 AOP Weaver 模块,它是仿真平台框架实现面向方面的关键部件。这两个部件都为具体业务层提供了丰富的公共业务层 API 接口。③核心控制层。该层对于定时服务,任务创建和调度等非功能需求的实现可以交给开发平台核心控制层提供的时间管理模型和任务调度模型去完成,从而缩短实时系统软件开发的周期。该层也提供了丰富的核心控制层 API 接口,供上两次调用。

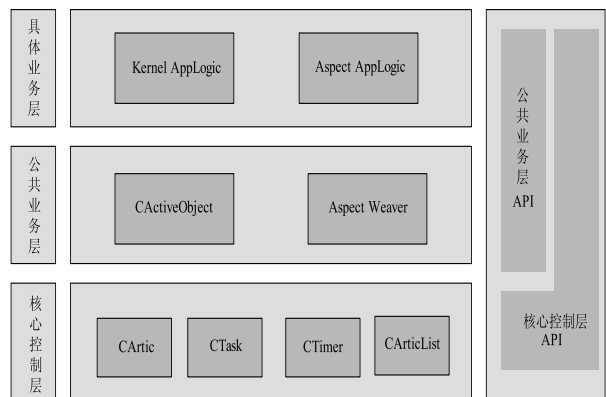


图 2 R-AoSAS 总体架构图

2.2 CActiveObject 类

R-AoSAS 一般由多个并行过程或任务组成，这些任务是顺序的，有时任务需要实时相互通讯与同步。公共业务层提供了主动对象类基类，它封装了执行线程、事件管理队列和状态机。①CActiveObject 中的 mState 变量对应当前任务状态；②支持不同任务之间进行相互通信和同步；③对于系统状态转移采用状态转移表实现。CActiveObject 具体类图，如图 3 所示。类中虚拟函数 Task_Run()方法表示主动对象的执行线程。具体的主动对象是从 CActiveObject 类中派生，同时还需要与定时器绑定实现定时或超时控制。

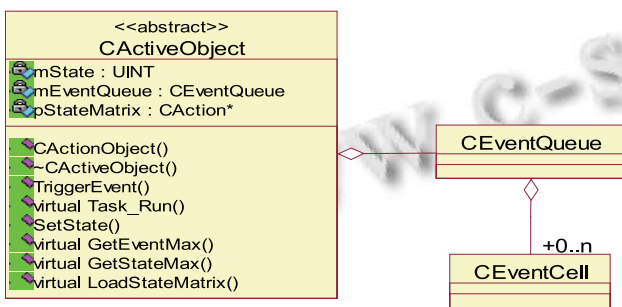


图 3 CActiveObject 类图

2.3 Aspect Weaver 编织器

在 R-AoSAS 框架应用环境下，为了实现功能组件和非功能组件绑定后组合。本文采用核心功能组件模块的连接点 (Join Point) 调用非功能组件模块的行为，类似于委托模式或装饰模式，通过方面代码的切入点 (Pointcut) 可以捕获输入/输出消息，我们只需要在 Advice 体中加入相应的非功能性组件代码即可。图 4 中显示了单点和多点传送结构连接模型，来实现不同功能组件间的绑定和组合。

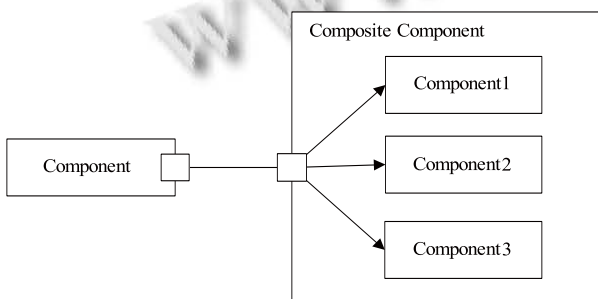


图 4 单点和多点传送结构连接模型

本文通过方面间描述文件来描述面向方面，方面

间描述文件是一种类似于 AspectJ 的方面编程语言 ACV，采用基于 ANSI C 语言 AOP 扩展的编译器^[6]进行源代码级的程序转换，转换成 C 语言编译器能编译的 C 代码，然后和其它非方面模块一起编译得到系统的可执行文件。通过基于 ANSI C 语言 AOP 扩展的编译器对以上已经模块化的组件模块进行编织，从而构建一个完整的系统。

2.4 任务调度

任务调度 CTask 类所含内容是对任务特性的精确描述，在控制处理任务状态中至关重要。其中主要有 4 个函数，如表 1 所示。一旦任务建立，CTask 就被赋值。每个主动对象都有一个对外提供的线程接口即 CTask_Run()方法。任务调度 CTask 类是一个数据结构，R-AoSAS 框架核心层用 CTask 来保存该任务的状态、任务的创建和任务调度。

表 1 CTask.h 中 4 个主要函数

函数	实现功能
CTask_Initialize()	任务队列的初始化
CTask_New()	新任务创建
CTask_Run()	任务调度的执行
CTask_Free()	注销主动对象任务

2.5 定时器管理

定时器 CTimer 是 R-AoSAS 核心控制层功能和非功能分离设计的产物，它的组合可以帮助主动对象实现实时系统软件的实时性要求。其中有 6 个函数，如表 2 所示。主动对象可以根据需要被分配多个定时器对象，被分配的定时器不应该被共享。

表 2 CTimer.h 中 6 个主要函数

函数	实现功能
CTimer_Initialize()	定时器的初始化
CTimer_Start()	定时器启动
CTimer_Stop()	定时器停止
CTimer_Pause()	定时器暂停
CTimer_Clock()	定时器的周期运行
CTimer_Kill()	定时器的解除

3 R-AoSAS 框架的应用扩展

3.1 系统设计

3.1.1 系统需求分析

自动售货机金钱管理系统是自动售货机应用层的

一部分。本系统是在 Visual C++6.0 开发环境下，运用 C 语言实现面向对象的设计，应用前文提出 R-AoSAS 框架，在软件设计阶段运用 AOP 思想进行系统分析和设计，从而实现了一个基于 AOP 技术的松散耦合的实时控制系统。该系统实现了自动售货机金钱管理的核心功能，以及实时处理、日志记录等非功能约束。

自动售货机金钱管理系统，其主要功能是将顾客从入金口投入的纸币（日元）搬运到收纳库，在此过程中还要负责对纸币的真伪检测以及纸币的分类。以下是对各子控制部的具体组成和功能的描述：

(1)投入部：顾客通过此口向贩卖机投入硬币。在此配置了充当自动售货机眼睛的投入口传感器和用于硬币搬运的马达。

(2)鉴别部：对从投入口搬运的纸币进行真伪检测，若纸币为假货则被送往出金口，反之，真货被送到下一个流程判别部。在该部配置有（四）个传感器、一个马达、一个用于门控的电磁继电器。根据纸币的真伪，这个门控电磁继电器有两个选择方向，它们分别是出金口和判别部。

(3)判别部：负责判别从鉴别部过来的纸币的面额大小，并根据纸币面额的大小送往不同的收纳库。因此在各种纸币的收纳库前都配有检测通过的传感器，用于纸币的计数和投入纸币的总额计算，此外还有一个负责把纸币搬运到各个收纳库前的一个马达。

其中千元库（1000Stock）主要用来存放千元纸币，混合库（CompStock）主要用来存放两千元和五千元，而精查库（InspectStock）主要用来做一些辅助存放及其他处理的临时库。

3.1.2 系统总体架构

系统总体架构如图 5 所示，系统主要有三个状态管理模块组成，分别为金钱管理模块，驱动管理模块

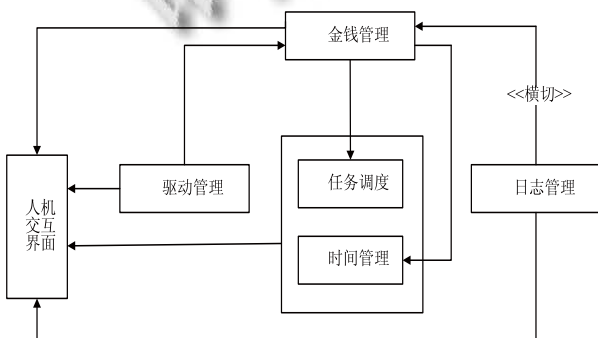


图 5 系统构架图

和日志管理模块，以及 R-AoSAS 框架核心控制层由任务创建、调度和时间管理模块组成。各模块应用事件驱动，即通过内部、外部的事件来驱动执行相应的模块，内部事件主要指时间队列中的任务以及执行时间的结束，外部事件主要指相应界面操作事件。系统通过人机交互界面来进金钱管理仿真监测和管理。

3.1.3 系统功能分析

根据自动售货机金钱管理控制需求，系统可分为入金、入金取消、出金、万元兑换等控制命令。

(1) 入金

入金部分的主要功能是根据入金口投入纸币的真假，把纸币送往判别部或者出金口，如果纸币为真，并且为千元纸币，则将其从判别部送往千元库（1000Stock）；若是两千元、五千元或万元纸币，则将其从判别部送往混合库。其纸币入金搬运定义如表 3 所示。

表 3 入金搬运定义

功能	搬运运行顺序	搬运转移条件	搬运
入金	投入口→鉴别部	真币	鉴别部→判别部
		假币	鉴别部→出金口
	鉴别部→判别部	千元	判别部→千元库
		其他	判别部→混合库

(2) 入金取消

入金取消部分的主要功能是将千元库和混合库的临时保留的纸币搬运到出金口。其实现过程是，先从千元库往出金口搬运，每次一张直到千元库临时保留的纸币数为零；再从混合库往出金口搬运，直到混合库临时保留纸币为零。其纸币入金取消搬运动作定义如表 4 所示。

表 4 入金取消搬运定义

功能	搬运运行顺序	搬运	结束条件
入金取消	千元库→出金口	千余库→判别部	千余库临时保存张数为零
		判别部→出金口	
	混合库→出金口	混合库→判别部	混合库临时保存张数为零

(3) 出金

出金的主要功能是将千元库的千元纸币搬运到出金口，其他种类的纸币搬运到精查库；把混合库纸币搬运到出金口，其他的搬运到精查库；直到指定的出

金张数搬送结束。其纸币出金搬送定义如表 5 所示。

表 5 出金搬送定义

功能	搬送运行顺序	搬送转移条件	搬送	结束条件
出金	千元库→判别部	千元	判别部→出金口	出金张数完了
		其他	判别部→精查库	
	混合库→判别部	混合纸币	判别部→出金口	出金张数完了
		其他	判别部→精查库	

(4) 万元兑换

万元兑换的主要功能是将精查库和混合库的万元纸币搬送到出金口。其实现过程是将精查库的万元纸币搬送到出金口，其他类型纸币搬送到混合库，等到把精查库的万元纸币搬完后，再把从精查库搬到混合库的纸币再搬回精查库；用类似的方法可以把混合库的万元纸币搬送到出金口，中间借助精查库。其纸币万元兑换搬送定义如表 6 所示。

表 6 万元兑换搬送定义

功能	搬送运行顺序	搬送转移条件	搬送	结束条件
万元兑换	精查库→判别部	万元	判别部→出金口	精查库万元纸币张数为零
		其他	判别部→混合库	
	混合库→精查库	/	/	已经搬送的张数
	混合库→判别部	万元	判别部→出金口	混合库万元纸币张数为零
		其他	判别部→精查库	
	精查库→混合库	/	/	已经搬送的张数

3.2 关键部件及技术

3.2.1 系统 Class 图

Class 图显示系统中类与类之间的交互关系，主要关注系统的层次和结构，包括对象、接口、属性和关系。图 6 显示了金钱管理系统的基于主动对象类图。其中，在具体应用 R-AoSAS 框架时，软件开发人员将实时系统的时间管理和任务调度等非功能约束的实现可以直接调用 R-AoSAS 框架核心提供的时间管理和

任务调度模型，从而缩短了系统开发周期，提高了系统开发质量和软件的复用性。

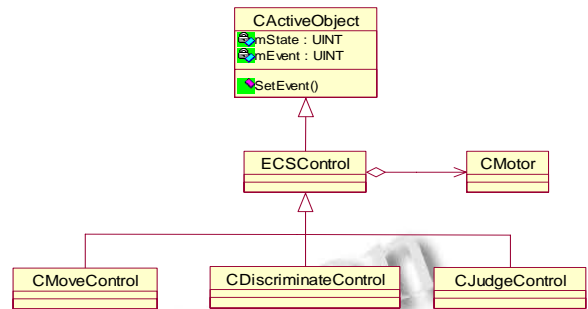


图 6 主动对象 Class 图

R-AoSAS 框架的一个最大特点是应用基于状态机的主动对象设计模式，对自动售货机实时软件系统构成多个主动对象的组合，实现了系统并发性关注。在本系统中设计主动对象区别于传统面向对象方法，它提供对事件队列、控制线程以及表示主动行为状态机封装，这种设计模式非常适合实时系统软件系统实时性处理和系统任务调度管理。为了使系统具有扩展性，R-AoSAS 框架提供继承主动对象类 CActiveObject 进行主动对象的派生操作。为了实现系统的功能设计了其它 4 个主动对象，它们分别是：

- ① ECSCControl 为金钱管理系统控制类基类
- ② CMoveControl 为金钱管理系统搬送控制类
- ③ CDiscriminatorControl 为金钱管理系统鉴别控制类
- ④ CJudgeControl 为金钱管理系统判别控制类

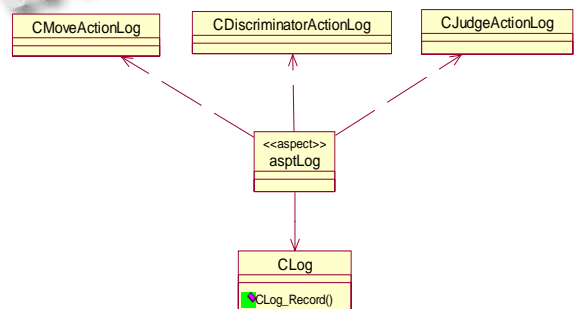


图 7 日志记录 Class 图

图 7 日志记录模块全部由 CLog 继承而来，单独行成横切模块。控制类的代码不用出现日志类的代码，两个模块之间将会通过编织器进行编织。

3.2.2 搬送控制类

首先给出控制类 ECSCControl 的构成, 其中主要有 5 个函数, 如表所示。这个类是搬送控制、鉴别控制和判别控制的父类。它从 CActiveObject 类继承而来, 由于控制的对象又是马达, 所以把马达类作为控制类的聚合。ECSCControl 类对 R-AoSAS 中 CTask、CTimer 类的直接引用。这也体现了直接调用非功能模块实现系统所需的非功能需求。

表 7 ECSCControl.h 中 5 个函数

函数	实现功能
ECSCControl_Initial()	控制类初始化
ECSCControl_Entry()	控制入金
ECSCControl_EntryCancel()	控制入金取消
ECSCControl_Exit()	控制出金
ECSCControl_Stop()	控制停止

CMoveControl 类该模块控制模块的核心类, 该类在继承 ECSCControl 类。搬送控制主要负责各个库之间的搬送工作, 但是搬送任务同时可能不止一个, 所以还要聚合一个搬送队列 CMoveQueue, 同时还关联 CMoveMotorRouterControl 搬送马达路由控制类, 其中还需要关联搬送路径的判断也需要判别部配合根据纸币面值来进行。其中主要有 9 个函数, 如表 8 所示。

表 8 CMoveControl.h 中 9 个函数

函数	实现功能
CMoveControl_1000StockToJudge()	千元库到判别出金
CMoveControl_CompStockToJudge()	混合库到判别部
CMoveControl_InspectStockToJudge()	精查库到判别部
CMoveControl_InspectStockToCompStock()	精查库到混合库
CMoveControl_1000StockToExitPort()	千元库到出金口
CMoveControl_CompStockToExitPort()	混合库到出金口
CMoveControl_RegisterEndJudge()	判别完了通知登录
CMoveControl_RemoveEndJudge()	判别完了通知取消
CMoveControl_Run()	搬送实现

3.2.3 鉴别控制类和判别控制类

鉴别控制类 CDiscriminatorControl, 主要负责纸币真伪的辨别, 通过鉴别部门控电磁继电器进行控制根据真伪的结果, 若为真币, 门控继电器开向判别部, 否则为假币, 则门控继电器开向出金口。其中要 4 个函数, 如表 9 所示。

表 9 CDiscriminator.h 中 9 个函数

函数	实现功能
CDiscriminatorControl_Start()	鉴别控制启动
CDiscriminatorControl_ToJudge()	鉴别部到判别部
CDiscriminatorControl_ToExitPort()	鉴别部到出金口
CDiscriminatorControl_RegisterCallback()	鉴别部注册回调

判别控制类 CJudgeControl, 主要负责纸币面额判别, 根据纸币面额, 若为千元, 则送往千元库; 若为两千元、五千元或万元纸币, 则送往混合库。其中要 7 个函数, 如表 10 所示。

表 10 CJudgeControl.h 中 9 个函数

函数	实现功能
CJudgeControl_To1000Stock()	判别部到千元库
CJudgeControl_ToInspectStock()	判别部到精查库
CJudgeControl_ToCompStock()	判别部到混合库
CJudgeControl_ToExitPort()	判别部到出金口
CBEndTo1000Stock()	千元库搬送完成
CBEndToInspectStock()	精查库搬送完成
CBEndToCompStock()	复合库搬送完成

3.2.4 日志记录实现

R-AoSAS 框架允许软件开发人员在系统设计时从核心功能需求中分离出不同的关注, 通过方面的 Join Point 成为功能组件模块和非功能组件模块的绑定或组合的接口。我们将金钱管理系统中的日志记录模块作为非功能组件模块, 通过前文的 Aspect Weaver 来实现静态编织。纸币入金流程可以通过人机交互界面显示出来, 如货币在入金控制命令发布下, 从判别部搬送到千元库中。在一次搬送完成后, 必须在程序界面上显示日志记录: 判别部→千元库, 开始时间→结束时间, 状态(完成或者错误)。如图 7 所示, 实现日志记录模块在判别部到千元库搬送面向方面文件描述。

这个方面代码描述功能是, 在核心组件模块执行判别部到千元库搬送时, 通知日志处理 CMoveActionLog, 采用面向方面技术, 切入点和通知绑定连接点, 同时, 把通知所实现的功能作为方面的普通服务, 在其接口处描述, 从而使得可以在连接点之前、之后插入日志记录代码, 不会影响到核心功能组件本身结构。如图 8 所示是使用方面代码编织器编织后 C 语言代码。

```

/*方面间描述 */
/* 自动生成 C 源代码后, 在原代码中加入注释
#include "MoveActionLog.h"
typedef GMoveActionLog;
typedef CMoveActionLog;
typedef CJudgeControl;
typedef OBJECT;
/*连接点(Join Point)定义. 判别部到千元库开始搬送方法声明 */
void CJudgeControl_To1000Stock(CJudgeControl* object_p) in JudgeControl;
/*连接点(Join Point)定义. 判别部到千元库完成搬送方法声明 */
static void CBEndTo1000Stock(OBJECT self_p) in JudgeControl;

/*方面 (Aspect) 定义 */
aspect asptMoveActionLog{
/*连接点(Pointcut)定义. 判别部到千元库开始搬送函数 */
pointcut pcCJudgeControl_To1000Stock(CJudgeControl*
object_p): CJudgeControl_To1000Stock(object_p);
/*连接点(Pointcut)定义. 判别部到千元库完成搬送函数 */
pointcut pcCBEndTo1000Stock(OBJECT self_p): CBEndTo1000Stock(object_p);

/*通知(Advice)定义. 判别部到千元库开始搬送, 启动搬送日志记录 */
before(CJudgeControl* object_p): pcCJudgeControl_To1000Stock(object_p)
{
short pattern;
pattern = JudgeTo1000Stock;

CMoveActionLog_Record(&GMoveActionLog.MOVE_ITEM_PATTERN,&pattern);/*记录开始搬送时间 */
CMoveActionLog_Record(&GMoveActionLog.MOVE_ITEM_STARTTIME,NULL);/*记录开始搬送状态 */
}
/*通知(Advice)定义. 判别部到千元库结束搬送, 启动日志记录 */
after(OBJECT self_p): pcCBEndTo1000Stock(self_p)
{
short _JudgeTo1000Stock_Result;
_JudgeTo1000Stock_Result = STATUS_NORMAL;

CMoveActionLog_Record(&GMoveActionLog.MOVEACTION_ITEM_ENDTIME,NULL);/*记录结束搬送时间 */
CMoveActionLog_Record(&GMoveActionLog.MOVEACTION_ITEM_ENDSTATE,&_JudgeTo1000Stock_Result);/*记录结束搬送状态 */
}
}

```

图 7 方面间描述文件

```

#include "JudgeControl.h"
#include "MoveActionLog.h"

extern CMoveActionLog GMoveActionLog;

#define _asptMoveActionLog_before(object_p){
short int pattern;
pattern = _JudgeTo1000Stock;
CMoveActionLog_Record(&GMoveActionLog.MOVEACTIONLOG_ITEM_PATTERN,&pattern);
CMoveActionLog_Record(&GMoveActionLog.MOVEACTIONLOG_ITEM_STARTTIME, NULL);
}

#define _asptMoveActionLog_after(self_p){
short int _JudgeTo1000Stock_Result;
_JudgeTo1000Stock_Result = STATUS_NORMAL;
CMoveActionLog_Record(&GMoveActionLog.MOVEACTION_ITEM_ENDTIME, NULL);
CMoveActionLog_Record(&GMoveActionLog.MOVEACTION_ITEM_ENDSTATE,&_JudgeTo1000Stock_Result);
}

void _original_CJudgeControl_To1000Stock(CJudgeControl *object_p);
void CJudgeControl_To1000Stock(CJudgeControl *object_p){
_asptMoveActionLog_before(object_p);
_original_CJudgeControl_To1000Stock(object_p);
}

static void _original_CBEndTo1000Stock(OBJECT self_p);
static void CBEndTo1000Stock(OBJECT self_p){
_original_CBEndTo1000Stock(self_p);
_asptMoveActionLog_after(self_p);
}

```

图 8 编织后的代码

把这个 Aspect Weaver 编织器编织后代码和核心代码通过编译器一块编译后, 形成最终系统。这样虽然在系统的核心模块代码中没有出现日志记录代码, 但是通过编织实现了日志记录功能。具体效果参见下文的系统测试。

3.3 系统实现效果讨论

本系统的开发利用 Visual C++6.0 IDE 集成开发环境。如图 9 所示的系统人机交互界面, 该界面应用 MFC 的 CDialog 类实现, 通过此界面来模拟展示系统的运行效果。界面上的入金、入金取消、出金、万元兑换 4 个按钮分别代表系统相应的 4 个不同的功能, 点击之后将启动不同的功能, 还有为了模拟系统中的硬件控制, 在界面上特意加入了对传感器的控制, 如: S3LR、S9、S4LR 等按钮都可以控制其开关状态。M1、M2、M3 等按钮代表不同的马达, 通过界面上的这些马达, 我们可以时刻了解它们的运转状态。每次搬送的动作、时间和状态都会在搬送日志中显示出来。如图 9 所示系统仿真测试运行界面。



图 9 系统仿真运行界面

图 9 中所显示的是一个货币从判别部到千元库的画面。从系统运行状态中可以看出, 由 Aspect Weaver 编织的日志记录功能运行正常。从系统运行状态中可以看出, 由方面代码编织器编织的效果很好。通过系统的测试可以证明了分层架构的面向方面软件开发方法可以满足系统运行的需要, 而且在解决模块级横切问题上也取得了令人满意的效果。

4 结语

本文在分析传统自动售货机实时系统软件体系结构的软件各组件间关系复杂、组件职责不清晰等问题基础上, 转向 AOP 技术, 提出了一种面向自动售货机

(下转第 25 页)

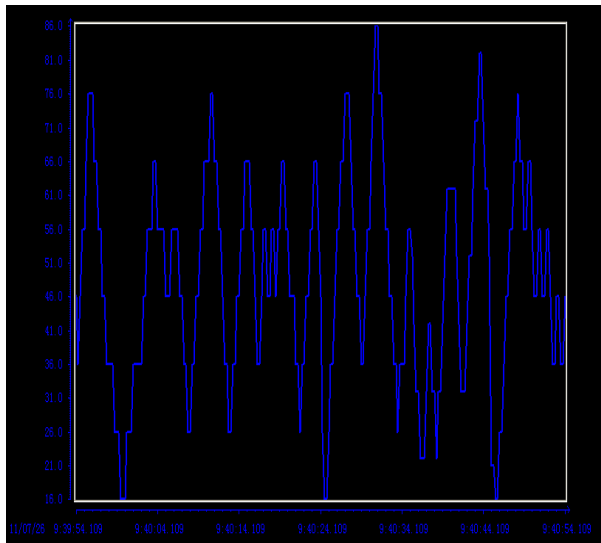


图 6 某段时刻储气罐 1 的压力趋势曲线

5 结论

经测试,该风电混合动力系统工作稳定、可靠,监测更加直观,成本较低,而且充分利用了风能,节约大量的不可再生能源。该系统优点突出,是风能利

用的新途径,在现代工业中,将具有广阔的发展前景。

参考文献

- 1 叶杭冶.风力发电系统的设计、运行与维护.北京:电子工业出版社,2010.
- 2 吴博.液压与气压传动原理及应用.北京:中国电力出版社,2010.
- 3 周晓平,姜建芳,苏少钰,陈迅.S7-200 系列 PLC 与监控计算机通信实现的研究.微计算机信息,2004,(1).
- 4 张延,周毅,万春林,赵全利,李会萍,贾磊.S7-200PLC 基础及应用.北京:机械工业出版社,2010.
- 5 Diego BC, Rodilla VM, Carames CF, Moran AC, Santos RA. Applying a software framework for supervisory control of a PLC-based flexible manufacturing systems. The International Journal of Advanced Manufacturing Technology, 2010,48 (5/8).
- 6 Da'na S, Sagahyoon A, Elrayes A, Al-Aydi AR. Development of a monitoring and control platform for PLC-based applications. Computer Standards & Interfaces, 2008,30(3).
- 7 西门子有限公司自动化与驱动集团.深入浅出西门子 Win-CCV6.北京:北京航空航天大学出版社,2004.

(上接第 7 页)

实时系统软件开发领域的可复用的分层面向方面软件实现框架 R-AoSAS,该框架在软件产品层面上提高实时系统软件开发的质量和软件的复用性。通过转向 AOP 技术,把金钱管理系统的实时性、并发性、任务调度和日志记录等非功能性关注从系统正面功能组件中分离出来,为它们单独的设计组件模块。在 R-AoSAS 框架应用环境下,对于定时器服务、任务创建和任务调度等非功能约束的实现交给 R-AoSAS 框架的核心层提供的时间管理模型和任务调度模型去完成。为了实现功能组件和非功能组件的绑定或组合,通过方面的 Join Point 作为它们之间的接口,各组件之间的关联关系变弱,使得组件职责更加清晰。最后结合一个具体应用,说明该框架在实际项目开发中的应用价值。

参考文献

- 1 宋晓峰,厉小军,邓阿群,俞蒙愧,胡上序.自动售货机故障管理子系统的设计与实现.计算机工程,2002,28(5):8-10.
- 2 熊光泽,古幼鹏,桑楠.嵌入式应用软件设计方法学研究综述.计算机应用,2004,24(4):1-4.
- 3 邓阿群.面向方面技术在大规模嵌入式软件中的应用[博士学位论文].杭州:浙江大学,2007.
- 4 何丽莉,金淳兆,冯铁,张家晨.关注分离问题研究综述.计算机科学,2005,32(2):129-132.
- 5 曹东刚,梅宏.面向 Aspect 的程序设计——一种新的编程范型.计算机科学,2003,30(9):5-10.
- 6 葛峰,张云华,赵国平.基于 C 语言的 AOP 编织器的设计与实现.计算机应用与软件,2008,25(12):161-163.