

# 基于 Spring 与 Hibernate 的网络故障管理系统的研究

郭文婷, 石 坚

(华中科技大学 电子与信息工程系, 湖北 武汉 430074)

**摘 要:**在网络技术飞速发展、网络地域迅速扩大的今天,建设具备跨平台性、高效性等特性的综合网络管理系统的需求越来越明显。文中提出了一种新的采用 Spring 框架与 Hibernate 技术的网络管理系统体系结构,以及一些新的故障管理实现方法,构建出一个更适应网络实际状况的网络故障管理系统,可以更有效地构建网络管理系统,维护网络。

**关键词:**网络管理系统; 故障管理; Spring 框架; Hibernate 技术

中图分类号: TP393.07

文献标识码: A

文章编号: 1673-629X(2007)06-0239-03

## Research and Implementation of Network Fault Management Based on Spring and Hibernate

GUO Wen-ting, SHI Jian

(Dept. of Electronics and Info. Eng., Huazhong Univ. of Sci. and Techn., Wuhan 430074, China)

**Abstract:** Nowadays, the network area has expanded quickly, and it becomes more and more required to construct a network management system which has the quality of high efficiency and multi-platform. A new system framework of network management system and several methods that implement fault management are presented in this paper. It can more efficiently construct the network management system to maintain the network.

**Key words:** network management system; fault management; Spring framework; Hibernate technology

### 0 引言

随着网络技术的发展,网络日益变得复杂化、分布化,网络地域的扩大也使得网络管理正朝着分布化、WEB化发展,对网络管理系统的分布特性、跨平台性和可扩充性的要求也更加突出。现有网络管理系统中的故障管理面临以下困难:其扩展性无法适应电信运营商丰富多变的需求;故障分析和分离功能不能适应现有网络的复杂结构;无法对网络中出现的故障信息的真实性进行有效的判断等。

Spring 是一个基于 IoC 和 AOP 的构架多层 J2EE 系统的框架,它实现了很优雅的 MVC,对不同的数据访问技术提供了统一的接口<sup>[1]</sup>。Spring 框架提供了对 Hibernate, JDO 和 iBATIS SQL Maps 的集成支持。框架对 Hibernate 的支持符合 Spring 通用的事务和数据访问对象(DAO)异常层次结构。而 Hibernate 是一个面向 Java 环境的对象/关系数据库映射工具,用来把

对象模型表示的对象映射到基于 SQL 的关系模型结构中去。Hibernate 不仅仅管理 Java 类到数据库表的映射,还提供数据查询和获取数据的方法,可以大幅度减少开发时人工使用 SQL 和 JDBC 处理数据的时间<sup>[2]</sup>。

文中提出的网络故障管理系统是利用 Spring 及 Hibernate 这两种开源框架来搭建三层架构实现的,包含视图的表现层(Client)、业务的逻辑层(Server)、数据的持久层。

### 1 系统总体概述

#### 1.1 系统描述

本网络故障管理系统是构建于北向接口层之上的上层网管系统,实现网管系统故障管理功能,系统采用 C/S 模式,利用 Spring 及 Hibernate 技术搭建三层架构,与北向接口的通信使用 Socket 通信,客户端和服务端的数据通信采用 RMI 通信,服务端与系统后台数据库的通信在本地进行。

本网络故障管理系统的开发目标是:

1) 实现网管系统核心管理服务(包括用户管理、授权管理、验证管理和网络传输数据加密功能等);

收稿日期: 2006-08-30

作者简介: 郭文婷(1986-),女,江西吉水人,硕士研究生,研究方向为计算机网络;石 坚,副教授,硕士生导师,研究方向为数据通信与计算机网络。



- 2) 实现网管系统故障管理功能;
- 3) 实现事件和日志管理;
- 4) 网管后台服务,提供一个可扩展的后台服务的管理框架;

5) 网管平台客户端向用户提供一个美观、易操作的可视化管理界面。

主要包括以下几方面的功能:

(1) 实时监视北向接口上报的告警信息,完成对告警进行显示、告警过滤、告警查询与统计等基本管理功能;

(2) 通过告警配置平台,实现告警自动通知功能,包括声音、图形、邮件等方式;

(3) 通过告警配置平台,实现告警屏蔽功能;

(4) 通过故障处理平台,实现故障相关性分析与定位功能、故障诊断功能。

## 1.2 系统体系结构

本系统是利用 Spring 及 Hibernate 技术搭建三层结构实现的,包含视图的表现层(Client)、业务的逻辑层(Server)、数据的持久层。在功能模块上相互分离,互不干涉,具有强大的 XML 模型定制功能;从总体实现上相互协作、共同完成,在开发效率、安全可靠性和灵活性等方面都有极大的提高。图 1 为系统体系结构图。

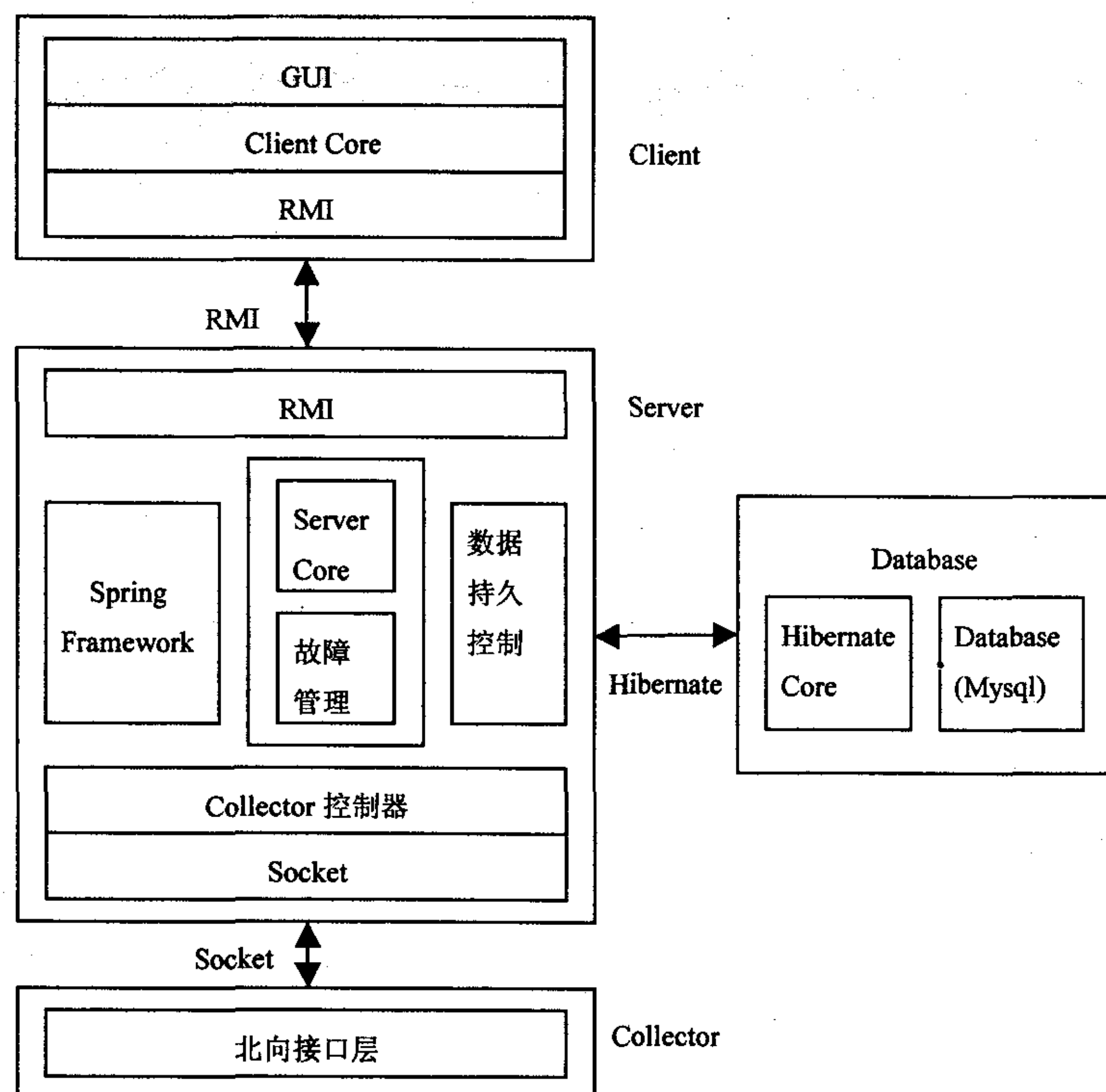


图 1 系统体系结构图

## 1.3 故障管理实现步骤

### 1.3.1 故障采集

利用 Socket 通信技术从北向接口获取故障数据,

实现告警同步,包括主动下发同步请求与屏蔽请求以及被动接收上报数据。

### 1.3.2 故障处理

系统将根据内部的不同事件判定条件(告警事件的唯一标识符)对获取到的故障数据进行重复告警的合并压缩,即将重复告警自动合并成一条告警,只标识告警发生的次数,并在告警的详细信息中记录每次发生的时间,防止同一故障持续告警造成的告警数据量剧增,从而减少告警信息数量,防止告警风暴的产生。

执行相关性分析,有效地屏蔽各种衍生事件、干扰事件和误告警,减少告警信息的冗余度,尽可能缩小故障根本原因的范围,用于在网络层对故障进行准确定位。

本系统在故障后台处理中主要执行两项相关性分析<sup>[3]</sup>:

#### (1) 根源告警相关性分析。

网络环境中来自不同信息源的告警和事件信息(无论是来自网络节点、主机系统、还是应用软件)必然是相互依赖不可分割的。经常发生由于网络端口不通,导致服务器失去联系、业务无法访问等连锁反映,这种情况下如果将 IT 分割为不同层次分别管理,管理员将会看到多个管理工具同时出现大量告警,很难在短时间内定位真正的故障根源。系统实现了 IT 整体

架构的集成管理,使得管理员能够合理建立从 IT 元素到业务的、跨层次的任何事件依赖关系,当有依赖关系的多个告警在指定的时间窗口内出现的时候,管理员只在告警浏览器中看到一个根源告警,如果点击此条根源告警的“相关告警”则可以查看到其它所有相关告警的信息。

#### (2) 成对事件相关性分析。

IT 环境中有些事件总是成对出现的,比如系统重启、端口重启、服务重启时总是产生“down”和“up”两个事件,而这种事件是没有必要通知管理员做任何干预处理的。系统允许管理员配置“成对事件”的各种告警规则:比如当“up”事件发生时,清除前一个“down”事件;或者当在指定时间窗口中连续出现“成对事件”时,忽略并屏蔽这两个事件。

### 1.3.3 告警存储

将按一定规则处理好的告警信息与数据库中已有告警信息进行比对,对于同一告警源产生的相同类型的告警采用更新数据库的处理方法,避免数据库中出现冗余信息。而新告警信息将存入数据库中。



### 1.3.4 告警显示与通知

告警显示功能涉及网络中发生的相应事件和情况的检测和报告。在网络中,设备内和输入信号中检测到的以及那些在此设备之外的事件/情况都是可报告的。根据 NG 北向接口上报的告警信息,在中央控制台上通过列表的方式将当前告警信息与历史告警信息显示出来,并以闪亮和声音来通知告警事件的发生提示用户对告警进行确认。当新发生的事件满足预先定制的规则条件时,通过调用内部或外部命令的方式实现对告警事件的自动响应,如自动发送 E-mail、执行声音等方式,及时将告警信息通知到相关的管理员。图 2 为故障管理处理流程图。

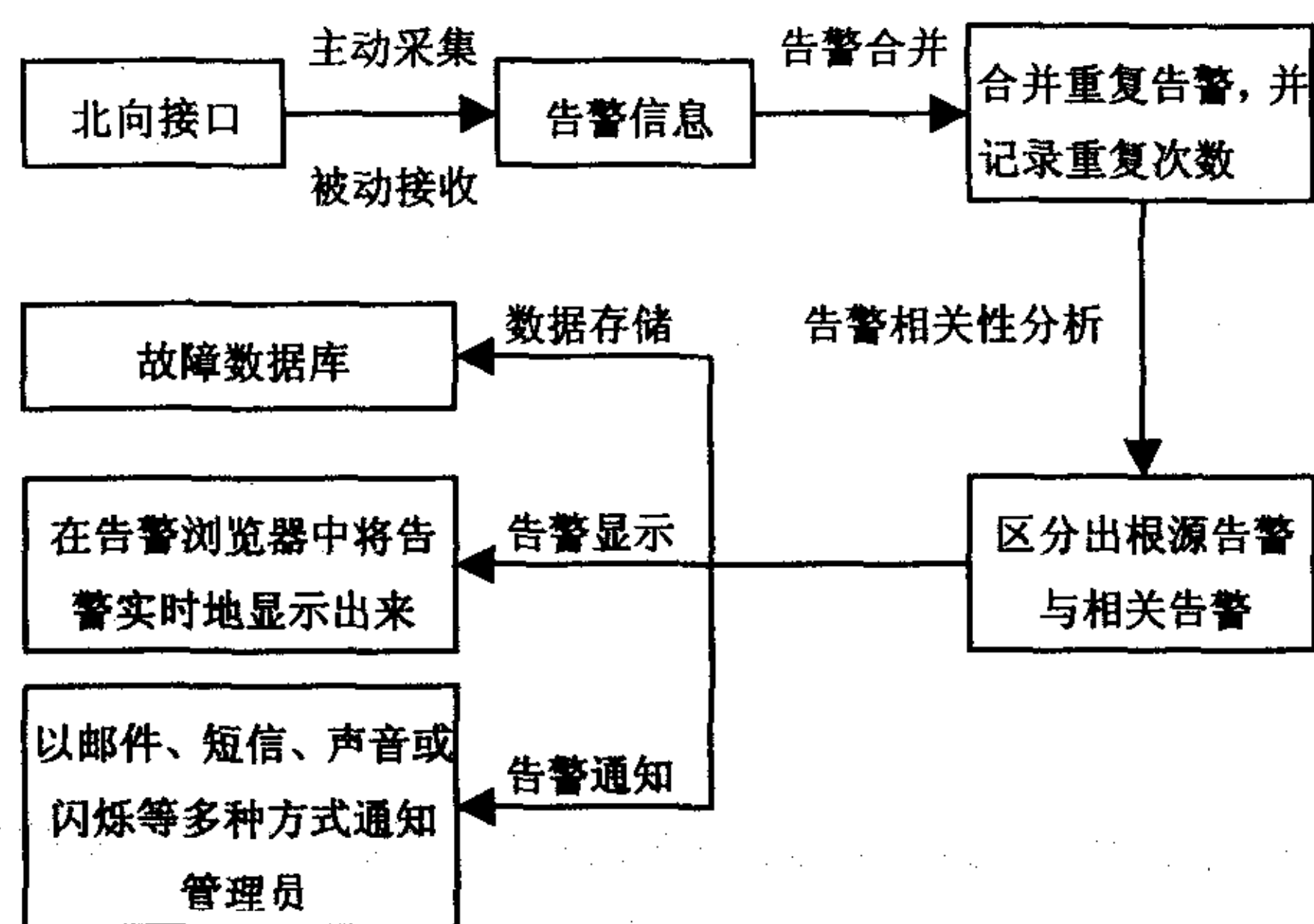


图 2 故障管理处理流程图

## 2 系统的实现

### 2.1 客户端

用户交互界面,系统设计支持 C/S 架构的胖客户端、B/S 架构的瘦客户端以及可以扩展其他语言的界面或者集成显示环境等。目前实现 C/S 架构的胖客户端,为了将来扩展性,在实现客户端时尽可能地减少客户端的数据处理,复杂数据处理都要放于服务器端,客户端只是对交互界面进行控制管理。

客户端界面显示采用 MVC 模式,分离业务数据、显示和控制,减小业务数据改变或显示方式改变而造成的相互影响。Core 封装与服务器连接方式,给业务开发提供基于接口的编程模式,屏蔽中间传输过程<sup>[4]</sup>。

### 2.2 服务器端

服务器端主要是数据计算处理以及系统控制中心,负责协调、控制其它层次。控制中心为核心商务逻辑与管理功能如故障、安全等服务提供统一连接接口并与连接协议分开,为业务处理人员的数据处理和传输提供统一透明的接口,不管数据持久层使用哪种数据库类型或者与客户端、数据采集端使用哪种连接协议,业务开发人员都是使用系统提供的统一接口。即

对数据库及远程连接进行封装提供统一接口,以中间件的模式提供数据库和远程连接。此模式分离技术细节与业务的耦合,便于将来技术变更给业务处理带来连动改动。本系统在服务器端实现上,采用 Spring 框架实现。

### 2.3 Hibernate 数据库连接的实现

在本系统中,数据持久层实现数据持久保存,支持基于 JDBC 驱动的 RDBMS 和基于 Hibernate 的数据服务。目前使用 Hibernate 提供数据服务。基于 Hibernate 可以快速而集中处理业务逻辑,加速模块设计,提供统一的事务控制模式。屏蔽后台数据库,根据业务需求可以快速切换后台数据库平台。

Hibernate 控制的是持久层,这里通常是程序最容易失控的地方。Hibernate 为 Java 提供了 OR 持久化机制和查询服务, Hibernate 将我们在 Java 类里使用的 HQL 语句转换为 SQL 语句,利用 JDBC 驱动进而操作数据源,完成数据的增加、删除、修改等最低层关系数据库。Hibernate 使用 XML 文件把 Java 类映射到表,把 JavaBean 属性映射到数据库表。Hibernate 的持久对象基于 POJO 和 Java collections<sup>[5]</sup>。

### 2.4 北向接口采集数据的实现

本网络故障管理系统的数据采集功能主要是通过北向接口来实现。系统与北向接口的通信使用 Socket 通信来实现,服务端与系统后台数据库的通信在本地进行。在服务器端建立一个数据采集控制器,主要是采用多线程机制建立与北向接口的 Socket 通信,支持数据报的上报和下行。控制器对通信端口进行实时监听,以获取从北向接口上报的数据,并将所获取的数据进行格式化处理后,之后对获得的告警信息进行重复告警过滤及相关性分析,然后存入数据库中;同时也可以将客户端传送过来的数据格式化成下行数据报下发给北向接口。

## 3 结 论

在该网络故障管理系统的开发实践中,采用了 Spring 与 Hibernate 这两种开源框架来搭建整体架构。在技术上,有免费、开源、丰富的文档和稳定的开发背景,保证了该系统在开发上的可行性;另外,这种架构的分层思想,使得各层之间相对独立,缩短了开发周期,增加了重用性,可扩展性高;在功能上实现了网络管理系统中的故障管理,对网络中的运行状况可以进行实时监视;在系统性能上,事务处理迅速,统计处理响应时间短,用户操作方便简单,具有很高的实用性。

当然,基于这两种框架的架构也并非没有缺点,

(下转第 245 页)



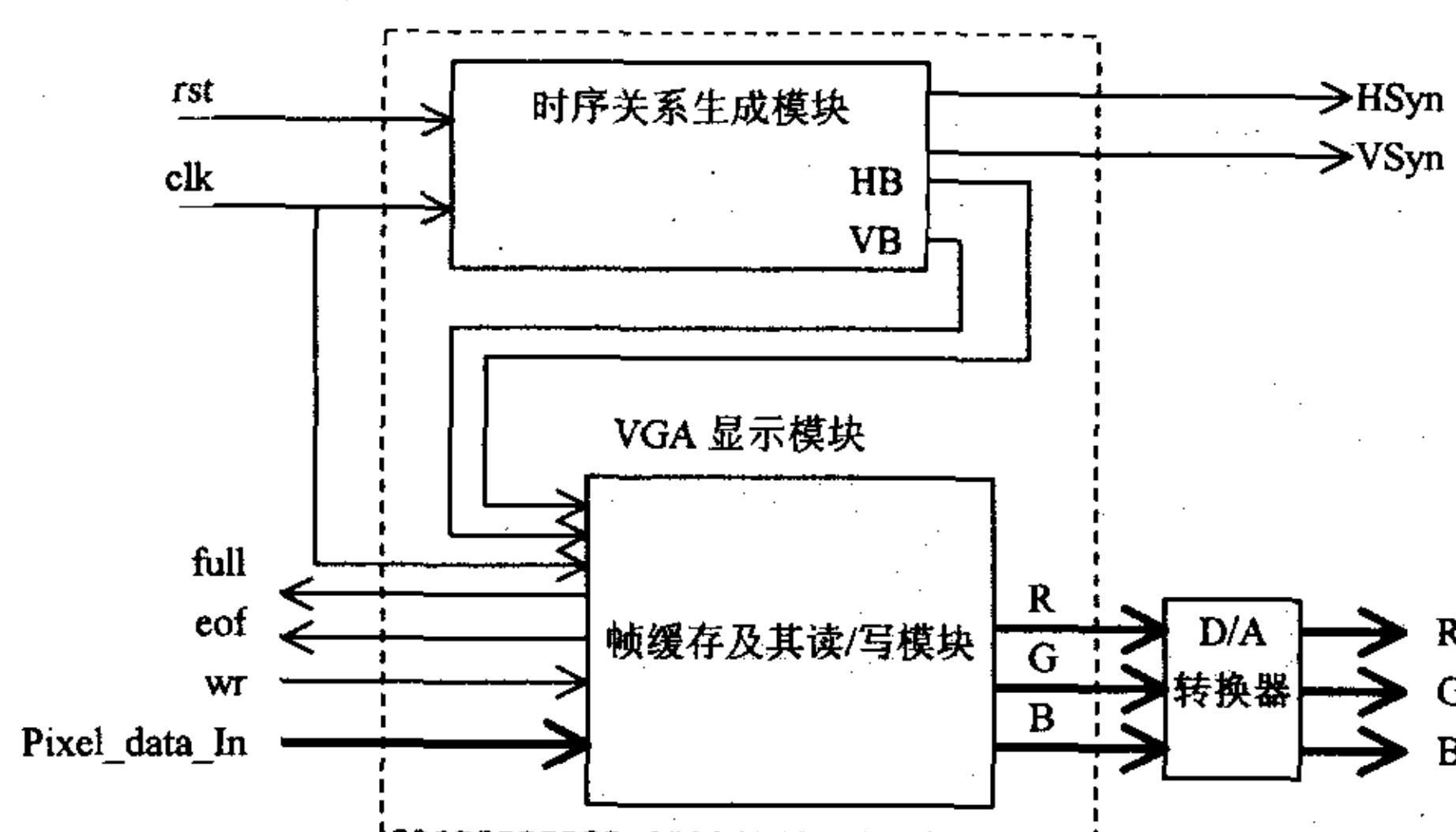


图5 VGA模块顶层视图

其中Clk是VGA显示模块的工作时钟信号,Rst为复位信号。Full是VGA显示模块中的帧缓存满标志信号,当帧缓存中的数据满时,VGA显示模块将此信号发送给SDRAM(视频存储器)控制器,Wr为VGA显示模块中的帧缓存写操作信号,Pixel\_data\_in为读入到VGA显示模块的帧缓存中的像素数据,该数据来自SDRAM(视频存储器),而Eof则表明VGA显示模块的帧缓存已空,可以重新读入一帧新的像素数据以供显示。信号HSyn是由VGA显示模块产生的水平同步信号,发往显示器的VGA接口,信号VSyn是由VGA显示模块产生的垂直同步信号,发往显示器的VGA接口,信号R是经D/A转换器分离的像素数据中的红色基色信号,发往显示器的VGA接口,信号G是经D/A转换器分离的像素数据中的绿色基色信号,发往显示器的VGA接口,信号B是经D/A转换器分离的像素数据中的蓝色基色信号,发往显示器的VGA接口。因为VGA显示接口处理的RGB信号都是数字信号,而传统的CRT显示器所采用的都是模拟接口,D/A转换器完成数字信号到模拟信号的转换工作。

其中VGA显示时序驱动是完成设计的关键。在VGA显示控制过程中,完成一帧(整屏)扫描所需要的时间称为垂直扫描时间,其倒数称为垂直扫描频率,又称刷新频率,即刷新一屏的频率。前文提到过,工业标准的VGA显示模式为:640×480×16色×60Hz。也

就是说,标准VGA显示的场频为60Hz,通过计算可知其行频(水平同步信号的频率)约为31.5kHz。图6显示了VGA显示的时序关系。

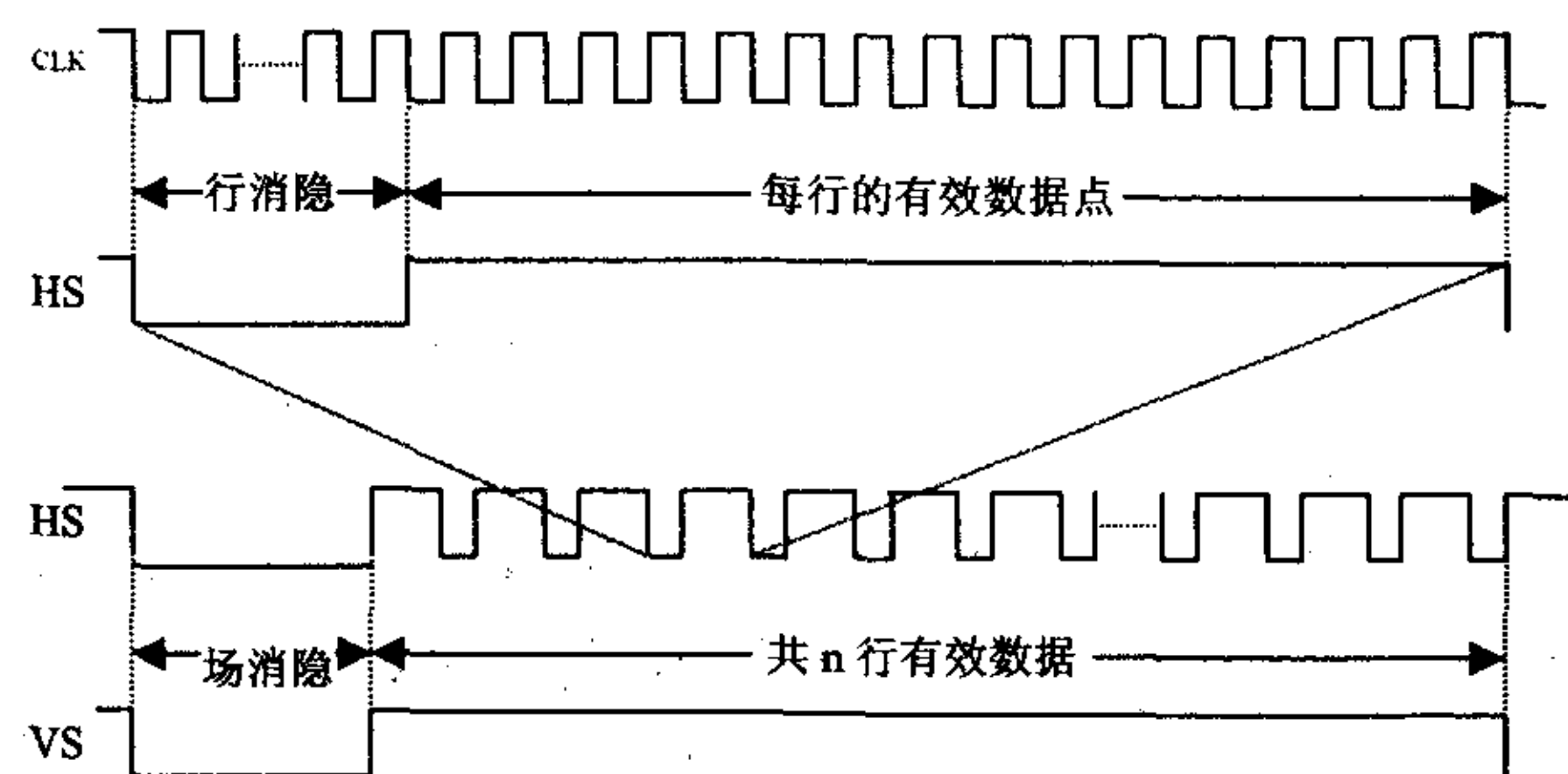


图6 VGA显示时序关系

#### 4 结束语

利用可编程逻辑器件(FPGA/CPLD)可以很方便地实现数字系统设计,而在Altera的QuartusII软件平台下,FPGA设计的各个阶段都得到了很好的支持,两者的有效结合使得数字系统的设计更加方便快捷。基于FPGA器件的VGA显示控制模块可以实现VGA显示的各项功能。其性能取决于VHDL代码的编写和QuartusII软件工具的综合优化的结果<sup>[6]</sup>。

#### 参考文献:

- [1] 王 诚,吴继华,范丽珍,等. Altera FPGA/CPLD设计(基础篇)[M]. 北京:人民邮电出版社,2005.
- [2] PLD/FPGA结构与原理初步[EB/OL]. 2006. <http://www.fpga.com.cn/advance/structures/lut.htm>.
- [3] 求是科技. FPGA数字电子系统设计与开发实例导航[M]. 北京:人民邮电出版社,2005.
- [4] 罗健军. VGA实用编程技术[M]. 北京:清华大学出版社,1995.
- [5] XESS Corporation. VGA Generator for the XSA Boards[EB/OL]. 2004-10-12. <http://www.xess.com/projects/an-101204-vgagen.zip>.
- [6] Pedroni V A. Circuit Design with VHDL[M]. USA: MIT Press,2004.

(上接第241页)

如:Hibernate的对象、关系之间的映射工作量很大;Spring和Hibernate的配置文件都很复杂等等。其实,这正是其灵活之处。因为把各层都完全分离了所以才易于重用和扩展,因而其灵活映射的配置文件才会很复杂。

#### 参考文献:

- [1] Johnson R. Spring Reference Documentation[EB/OL]. 2004.

<http://www.springframework.org/>.

- [2] Johnson R. Introducing the Spring Framework[EB/OL]. 2004. <http://www.theserverside.com/>.
- [3] 孟庆利. 故障管理系统中事件相关性分析的运用[J]. 世界电信,2004(10):43-44.
- [4] 黄烟波,张红宇,李建华. 基于Struts与Hibernate的J2EE架构[J]. 计算机时代,2004(10):29-30.
- [5] Alur D, Crupi J, Malsk D. J2EE核心模式[M]. 北京:机械工业出版社,2002.