

# 基于 UML 的设备磨损诊断系统分析与设计

程树林<sup>1</sup>, 毛 宁<sup>2</sup>

(1. 安庆师范学院 计算机与信息学院, 安徽 安庆 246011;

2. 广东工业大学 机电学院, 广东 广州 510090)

**摘 要:**应用 UML 建模工具对设备磨损诊断系统进行了分析和设计。讨论了 UML 在软件系统建模中的三个主要方面: 需求建模、静态建模和动态建模, 根据该建模方法详细分析和设计了设备磨损诊断即磨损判断和部位识别系统, 给出了系统主要模型, 包括需求模型、静态模型和动态模型及其详细模型视图。结合系统实现结构, 完成了系统分析与设计。

**关键词:**UML; 面向对象; 设备磨损诊断; 部位识别

中图分类号: TP277

文献标识码: A

文章编号: 1673-629X(2007)06-0217-04

## Analysis and Design of System about Equipment Wearing Diagnosis Based on UML

CHENG Shu-lin<sup>1</sup>, MAO Ning<sup>2</sup>

(1. Computer Dept. of Anqing Teachers College, Anqing 246011, China;

2. Institute of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510090, China)

**Abstract:** Analyzes and designs the diagnosis system of equipment wearing by using the modeling tool of UML. It firstly discusses the main three aspects of UML in modeling software system, which are demand modeling, static modeling and dynamic modeling. According to the modeling method, the analysis and design of equipment diagnosis system, including wearing judgment and place identification, are given. Also, the main system models, which are demand model, static model and dynamic model and detail model views, are demonstrated. At last, the analysis and design of system are finished with the system realization structure added.

**Key words:** UML; O-O; equipment wearing diagnosis; place identification

## 0 引 言

在工业生产中, 设备状态监测的一个重要方面就是设备磨损的诊断, 包括磨损判断和发生剧烈磨损时预测其可能的磨损部位。利用历史数据来判断设备磨损状况, 企业工程师无需深入生产一线就可以掌握设备运行情况, 以便快速响应、做出相关决策, 并能够提供丰富的知识库案例。笔者应用 UML 分析和设计了基于 B/S 环境下的分布式设备磨损判断与部位识别系统。UML 是第三代用来为面向对象开发系统的产品进行说明、可视化和编制文档的方法<sup>[1]</sup>, 从系统不同的角度分析和刻画系统视图, 清晰而明朗地将系统概貌和细节展现在设计者面前, 有利于系统的开发, 同时能够提高系统分析模块的维护性和开发尺寸合适的组

件模块, 并加以复用<sup>[2]</sup>。UML 的可视化好处在于使用户、开发人员、分析人员、测试人员、管理人员和其他项目所涉及的人员之间可以进行更好的沟通和交流。

## 1 UML 分析与设计方法

UML 已经广泛应用于软件系统的建模, 同时还可以用于非软件领域的建模。其分析与设计的主要方法是采用多种可视化的图形对系统进行描述和说明, 同时还可以设计 UML 中没有的特殊的泛型对系统模型进行补充说明<sup>[3]</sup>, 灵活自由。应用 UML 对系统进行分析和设计主要从三个方面考虑: 需求建模、静态建模和动态建模。

(1) 需求建模。即得到系统的需求分析模型。主要是应用 UML 对系统进行领域角色分析、业务分析和建立用例模型。其中用例模型分析非常重要, 根据用例的粒度大小可以得到系统的功能概貌和详细功能。

(2) 静态建模, 从对象角度来分析和设计系统的静

收稿日期: 2006-08-03

基金项目: 安徽省教育厅资助项目(2005KJ363ZC)

作者简介: 程树林(1979-), 男, 安徽安庆人, 讲师, 硕士, 研究方向为软件系统结构及知识工程研究; 毛 宁, 教授, 硕导, 主要从事制造信息化研究。



态模型。主要是系统类图、对象图、包图的分析。

(3)动态建模,从系统的工作流程和对象之间的相互作用建立系统模型。主要有协作图、序列图、活动图等。

以上三点是 UML 分析建模的主要方面,当然还可以根据系统建模的需要,进一步补充分析其他视图,如对象图、状态图等。

## 2 系统分析与设计过程

### 2.1 需求建模

需求建模,即对系统进行需求分析,找出系统所有的需求并加以描述,主要建立系统领域角色模型和用例模型(系统的数据流和数据字典在此不作为主要内容分析)。

(1)领域角色模型。角色是与所建系统交互的人或物,是描述系统范围外的一切<sup>[3]</sup>,既包括系统用户又包括与系统相关的其他系统、物或时间,即三大类:系统用户、与系统交互的其他系统和时间。在本系统中,系统用户角色有 Admin 系统管理员角色、Manager 监测项目负责人角色、Customer 用户角色、Person 业内人士角色和分析员基类 *Analyst*,它们均继承于用户基类 *UserBase*。Expert 专家角色继承于 *Person*,另有 *AnalystIron* 铁谱分析员、*AnalystPhysicChem* 理化分析员、*AnalystSpectral* 光谱分析员继承于 *Analyst*;与系统交互的外部系统角色有 *CreditSys* 银行信用系统角色和 *PRMSys* 设备运行系统角色;*Time* 角色在本系统中主要用于刷新和系统数据监控。

(2)用例模型。用例用来描述一个参与者(一个外部的主动者)使用系统完成某个过程时的事件顺序<sup>[4]</sup>。用例是系统的使用过程,表示了系统的功能块,换句话说,演示了人们如何使用系统<sup>[3,4]</sup>。每个用例应表示用户与系统间的完整事务,为用户提供一定的价值。用例之间也有关系,包括包含关系、扩展关系和泛化关系。图1和图2分别从粗粒度和细粒度对系统的主要用例进行分析。

图1是系统高级用例图,从系统全局的角度抽象系统的过程和功能。主要有四大块:管理客户资料及油样、管理检测数据、监测管理和协同会诊,其中还包含一个发送会诊邀请用例。用例间主要有包含、关联和泛化关系。涉及到用户角色、管理员角色、分析员角色以及会诊人员角色。用户可以对自已的资料和油样信息进行远程在线管理,分析检测员(即铁谱、光谱、理化分析员)对数据进行检测,其结果交给管理员进行进一步分析和预测,作出结论,同时可以将数据移交到客户的设备运行管理系统。当出现疑难问题,难以做出

结论时,可以进行远程协同会诊,参加会诊的人员有业内专家和专业人士等。

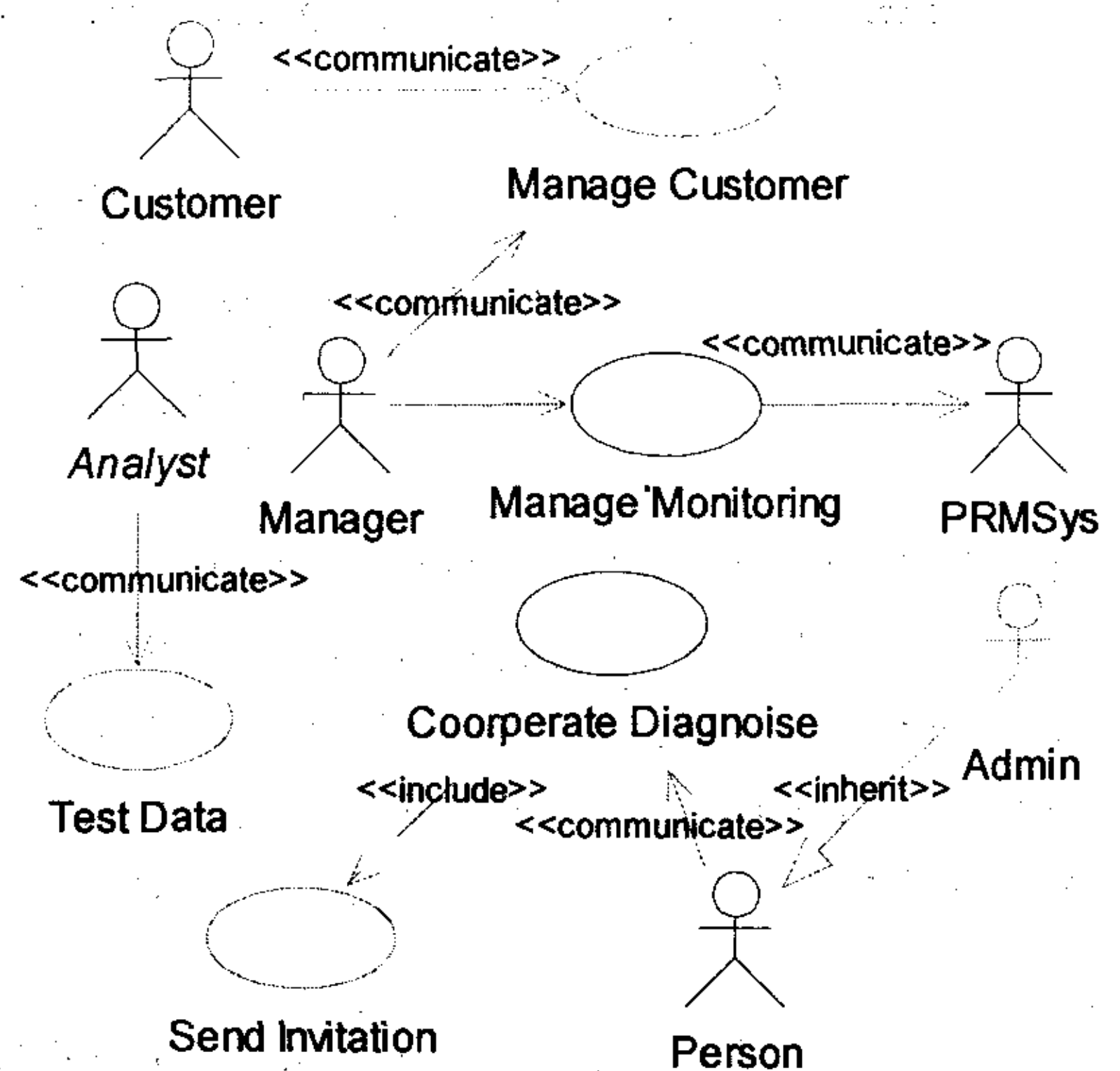


图1 系统高级用例图

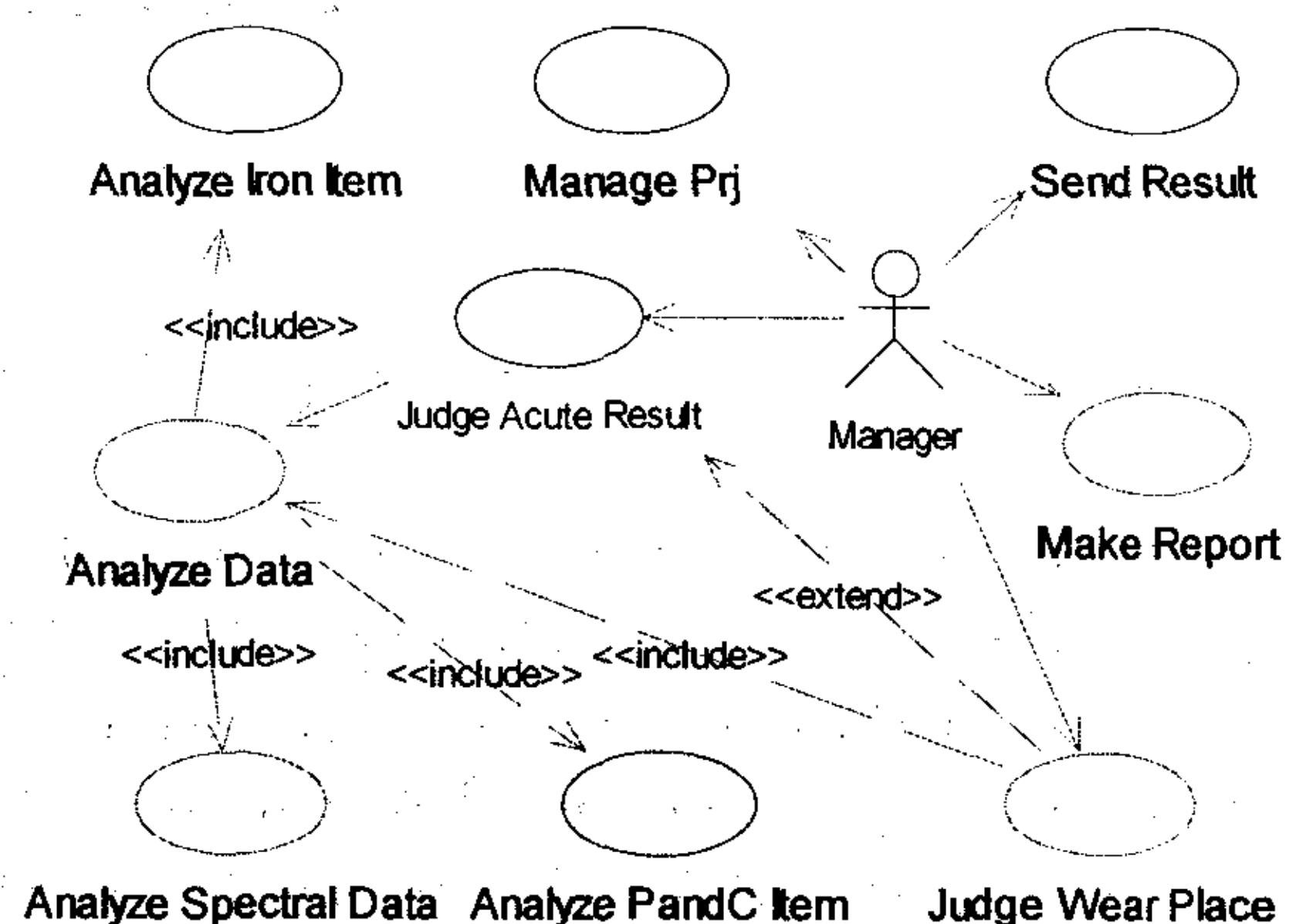


图2 监测管理扩展用例图

图1仅从高层描述系统过程,不够具体,表达的数据信息也很有限。在实际开发过程是不够的,这时就需要对系统高级用例进行扩展,对过程和需求进行深层次挖掘。图2是系统监测管理 *Manage Monitoring* 的高级用例的扩展用例图:从局部角度更详细地表示了系统的用例。如图中 *Analyze Data* 包含了 *Analyze Spectral Data*, *Analyze PandC Item* 和 *Analyze Iron Item* 三个用例,分别是分析光谱数据、理化数据和铁谱数据。*Judge Wear Place* 用例扩展了 *Judge Acute Wear* 用例,因为在执行 *Judge Acute Wear* 用例时,如果有剧烈磨损存在时,则需要进一步执行 *Judge Wear Place* 用例,否则无需执行它,因而它们之间是扩展的关系。

### 2.2 静态建模

静态建模主要是通过类图、对象图和包图对系统进行描述和说明。这里只给出类图模型。类图描述的是系统所包含的类以及各类之间的静态关系。类包装了信息和行为,按照面向对象的方法,将对象的属性和方法封装在一起<sup>[5]</sup>。常见的类有三种:边界类、实体类和控制类。类之间普遍使用四种关系:关联、依赖、累



积和一般化。

图3是系统类图的一部分,即剧烈磨损判断和磨损部位识别类图。图中表示了各类及类之间的关系,有继承、关联、引用或依赖以及链接的关系。其中 *DBAccess* 类用斜体表示,为 Abstract 类,不能直接实例化,作为控制类等的基类,其内部方法访问性为 protected。图中涉及到两个页面接口类,分别为 *AnalySpectral* 和 *JudgePlacePage*,它们之间为《link》关系,同时它们又各自依赖于与自己相关的控制类,控制类又继承于数据访问类,获得与后台数据库通信的方法,从而完成数据的操作。类图对于系统设计开发,提供了很好的参考作用,它列出了系统中应涉及到的类和各类之间的关系。

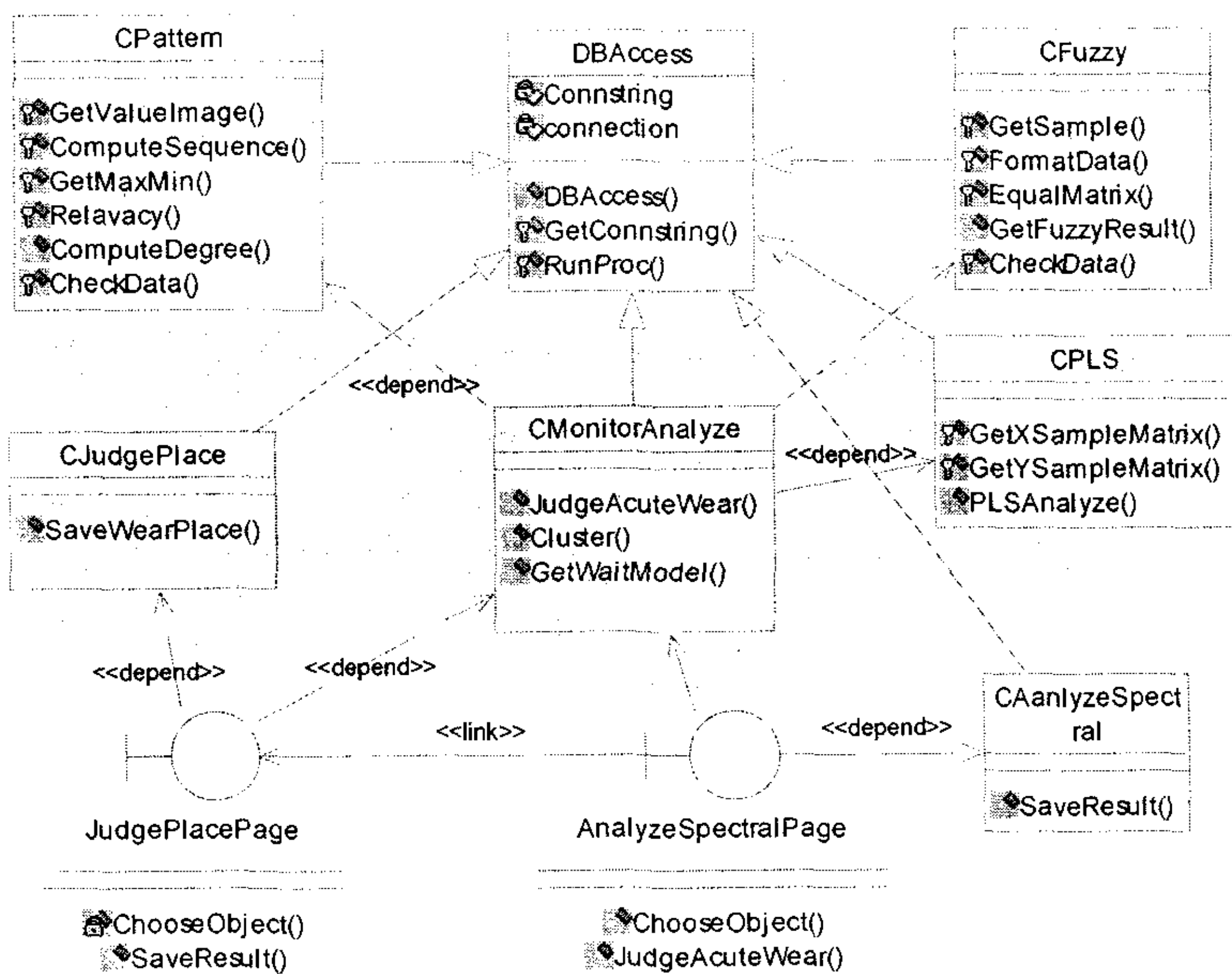


图3 系统的部分类图—剧烈磨损判断和部位识别

### 2.3 动态建模

对系统的动态建模这里给出序列图模型。序列图描述了系统中用例的实现过程,从时间顺序的角度出发,说明其操作过程<sup>[3]</sup>。每个序列图都是针对一个系统用例的特定事件流(如果一个用例有多个事件流则需要多个序列图表示),表示与系统直接发生交互的外部参与者、系统以及参与者所发起的系统事件等<sup>[4]</sup>。图4是磨损部位识别序列图。

图4由角色、对象行为(操作)和消息组成,其中还有垂直方向的虚直线为对象的生命线,其最末端的叉号表示对象的生命期的终止。它反映了角色 Manager 的一个实例在执行系统的操作功能“识别剧烈磨损部位”的一个事件过程,详细给出了整个事件过程中所涉

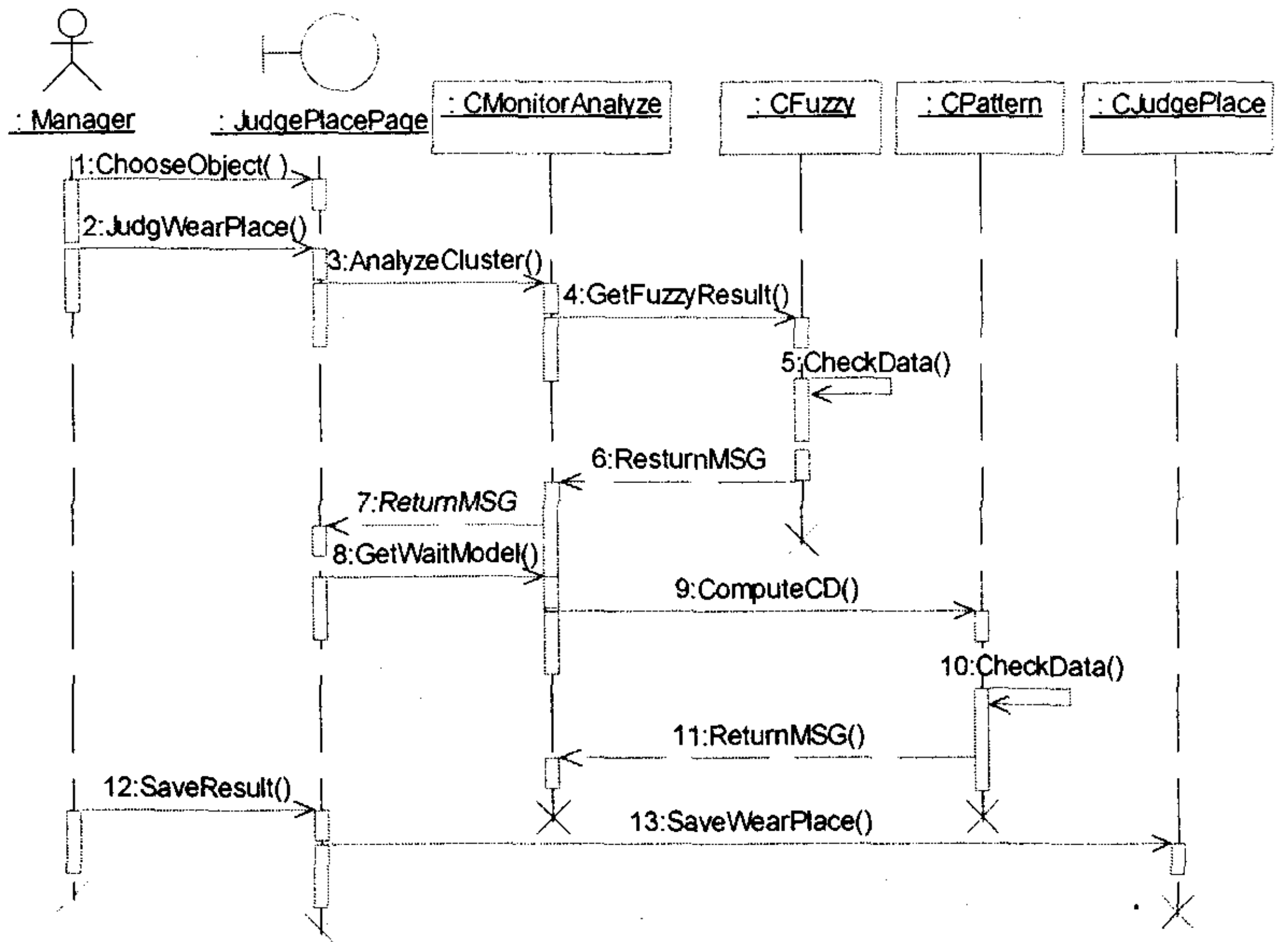


图4 磨损部位识别序列图

及到的对象、行为和消息。事件的执行过程为: Manager 实例触发 JudgePlacePage 实例的 ChooseObject() 方法确定剧烈磨损的判断对象, JudgePlacePage 实例调用监测分析控制类 CMonitorAnalyze 的 AnalyzeCluster() 和 GetWaitModel() 方法分别进行模糊聚类分析和获取待监模式,最后调用 CJudgePlace 的保存方法 SaveWearPlace() 保存分析结果。需要说明的是图中各对象的操作行为需要进一步契约化,这里没有标出。

## 3 系统结构

### 3.1 B/S 下系统三层结构

B/S 下系统的三层结构分为用户界面层、业务逻辑层和数据层<sup>[6]</sup>。

#### (1) 用户界面层。

用户界面层也叫表现层,主要是提供与用户交互的接口。由客户使用浏览器打开网页向 Web 服务器发出 HTTP 请求, Web 服务器得到请求后作出 HTTP 应答,将服务器端接口程序用 HTML 的形式发送给客户浏览器,同时客户浏览器要支持一些脚本语言。

#### (2) 业务逻辑层。

业务逻辑层也叫中间件层或应用层,它的主要组成是应用服务器。在应用服务器上提供了应用程序代码和组件代码。这些代码是由 .NET 编写的,封装了系统的业务逻辑思想和业务逻辑算法,其中采用 Web-Service 编写,以服务的形式提供给调用者,这些都是系统的主要控制部分,由 Web 服务器的接口页面调用。

#### (3) 数据层。

数据层也叫实体层,主要是数据库服务器,由主流



数据库支持如 SQL Server2000、Oracle8i、DB2 等。主要是存放系统数据和部分后台数据库逻辑程序。它与用户界面层和业务逻辑层分开,易于管理和维护,并且执行效率高、数据安全性好。

### 3.2 软件架构设计

基于上述的 B/S 下的系统结构,在设计软件系统架构时考虑以下几个方面来架构本系统,分别从纵向和横向两个方面分析,如图 5 和图 6 所示。

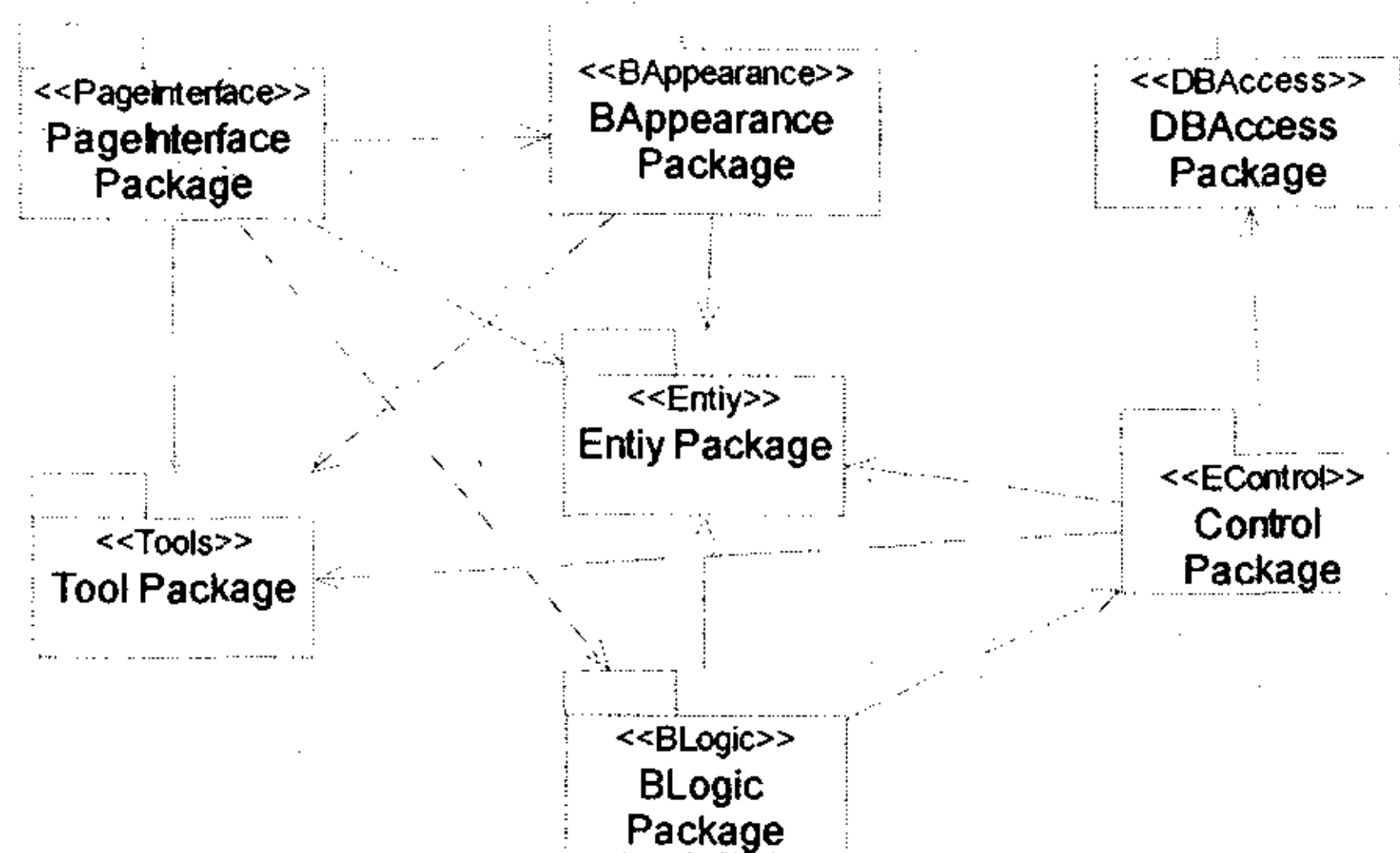


图 5 系统软件纵向架构图

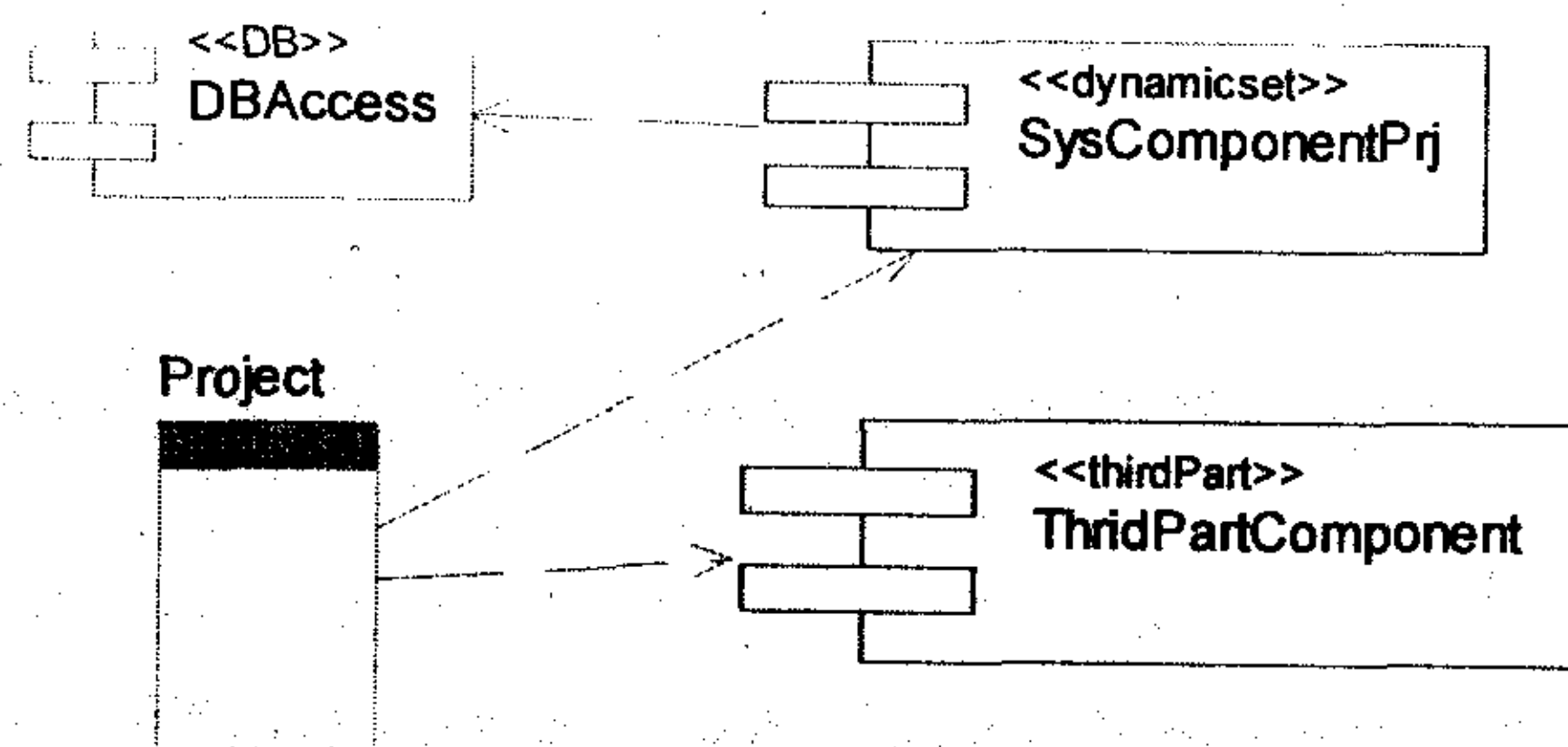


图 6 系统软件横向架构图

图 5 是从系统的纵向角度描述系统的架构,共有 7 个包,分别是页面接口包、实用工具包、业务外观包、业务逻辑包、实体包、实体控制包和数据访问包。package 和 package 之间具有依赖或引用关系。Tool

Package 和 Entity Package 引用较多,因为其他几个包都要使用它们。

图 6 是从系统的横向角度描述系统的架构,Project 表示系统运行项目,DBAccess 为数据库访问组件,SysComponentPrj 为系统组件群,ThirdPartComponent 为系统中所引用的第三方组件,各组件都是与 Project 级别相同,与 Project 同为一个解决方案下的不同项目,最终是以动态连接库的形式发布。

## 4 结束语

使用 UML 对 B/S 环境下的设备磨损判断和部位识别系统进行了分析和设计,给出了主要模型图和系统实现的结构模型,为后续的系统程式实现的过程起到较强的指导性作用,提高了系统的开发效率,缩短了开发周期。

### 参考文献:

- [1] Schmuller J. UML 基础、案例与应用[M]. 李 虎,王美英,万里威译.北京:人民邮电出版社,2000:5-10.
- [2] Gamma E, Ralph R H, Vlissides J J. Design Patterns Elements of Reusable Object-Oriented Software[M]. 李英军,马晓星译.北京:机械工业出版社,2000.
- [3] 蒋 慧,吴礼发,陈卫卫. UML Programming Guide 设计核心技术[M]. 北京:北京希望电子出版社,2001.
- [4] Larman C. UML 和模式应用-面向对象分析与设计导论[M]. 姚淑珍,李 虎等译.北京:机械工业出版社,2002.
- [5] Boggs W, Boggs M. UML 与 Rational Rose 2002 从入门到精通[M]. 邱仲潘 等译.北京:电子工业出版社,2002.
- [6] 程树林,毛 宁,蔡德辉,等. 基于 UML 的 .NET 系统建模方法[J]. 微电子学与计算机,2003(12):47-50.

(上接第 216 页)

中直接调用。如果父类对象中功能不完善,便可以直接修改父类对象的功能,以减少修改量。

## 4 结 语

软构件技术的出现,极大地满足了多个应用领域的要求,使得用各种技术形成的软构件可以最大程度地进行重用。这就引出了大规模软件开发所面临的另一个问题:如何建造面向对象的软构件库结构,并有效地组织和管理?在大型软构件库的支持下,利用现有的、质量好、可靠性强的软构件,按照大规模软件开发的工程规范进行开发,是满足这些大型系统要求的良好途径。现在人们越来越多地寄希望于具有分散和联合处理能力的开放性分布式软构件技术。分布式软构

件技术的目标是实现开放的软构件产品,使应用程序能相互操作,降低开发与管理费用。到目前为止,虽然还没有出现一个完整的开放式软构件系统的实施方案,但软构件化成为软件工业化将是必然的。

### 参考文献:

- [1] 杨美清. 软件复用技术[J]. 计算机科学,1999(5):21-26.
- [2] 梅 宏,常继传. 软件构件和软件构架技术[J]. 计算机世界,2000(8):30-34.
- [3] Pressman R S. 软件工程——实践者的研究方法[M]. 第 5 版. 梅宏译.北京:机械工业出版社,2002.
- [4] 何旭洪,余建英. PowerBuilder 8.0 数据库系统开发实例导航[M]. 北京:人民邮电出版社,2002.
- [5] Norman R J. Object-Oriented Systems Analysis and Design [M]. [s.l.]:Prentice-Hall International, Inc, 2000.