
软件工程课程作业

应用软件工程方法分析设计一个病人观察系统



(计算机应用技术)

作者: 赵卓宁

学号: 02S140010

电邮: zhaozhn@eyou.com.cn

目录

前言	1
一、作业背景	1
二、选题说明	1
三、课程题目	1
第一章 系统分析	2
第一节 现有系统介绍	2
一、现有版本及开发环境	2
二、基本功能	3
三、基本用例	3
四、现有系统的主要不足	4
第二节 新版本系统的项目计划	4
一、项目范围和目标	4
二、项目可行性分析	4
三、项目进度计划	5
第二章 需求分析	5
第一节 基本需求规格说明	5
一、系统概貌	6
二、功能要求(Functional requirements)	6
三、性能要求(Performance requirements)	6
四、运行要求(Operational requirements)	6
五、扩充要求(Possible evolution)	6
第二节 用例分析	7
一、系统基本用例	7
二、系统业务逻辑描述	8
三、数据流图分析	8
第三节 分析数据	10
一、系统的概念模型	10
二、系统的逻辑模型	11
第四节 项目计划的修订	11
一、成本估计	11
二、资源使用计划	11
三、改进的进度计划	11
第五节 系统原型	12
第三章 概要设计	12
第一节 系统结构设计	12
一、用户接口包	13
二、商业对象包	13
三、数据库包	15
四、实用包	15
第二节 子系统划分	15
第三节 数据库设计	15
第四章 详细设计	18
第一节 系统功能子模块的实现	18
一、系统管理子系统	18
二、费用管理子系统	19
三、在线就诊子系统	20
四、挂号咨询子系统	24
第二节 系统的部署	25
第五章 结束语	26
致谢	26
后记	26
主要参考文献	26
附：UML 课程作业要求	27

前言

一、作业背景

本文根据杨中海教授《软件工程概论》课程要求而做。



课程作业要求如下：

任意选择一个现有软件或者自己构思一个软件，采用软件工程方法对其进行设计（包括系统分析、需求分析、概要设计、详细设计）。

二、选题说明

由于我个人能力和时间有限，不可能性对一个复杂的系统进行系统分析、需求分析、概要设计、详细设计。因此，我选择《统一建模语言 UML》（杨国伟教授）的课程作业——《病人观察系统 Patient Observation》，按照软件工程的方法对其进行分析和设计。并借此进一步地加深自己对软件工程的认识和理解。

根据老师的要求，我认为自己的课程设计应完成以下目标（不知符合作业要求与否）。

- 1、以软件生命周期为线索（实际只能完成软件定义和部分软件开发工作），（由于个人时间和精力所限）选择瀑布模型开发系统，并能够初步应用课程中介绍软件开发技术和部分工具；
 - 2、能够编写较完整和标准的软件文档（实际只能将标准文档的主要部分写入本作业之中），以记录软件开发活动和阶段性成果、以及有助于自己 and 他人理解软件所必需的阐述性资料。
 - 3、以课件内容为准绳，将讲课内容应用于此作业。课程本身作为概论来讲，重要的是学生应将软件工程的一般过程能够反映在文档作业中。实现一个具体的系统是不现实的，因此，文档中有些部分略写。
-

三、课程题目

- 1、课程题目：《病人观察系统 Patient Observation 新版本的分析与设计》
- 2、编写目的：完成《软件工程概论》课程设计内容，供老师进行课程考核。

第一章 系统分析



作业任务

- 1、完成对旧系统的分析
- 2、初步设定新系统的目标
- 3、进行可行性分析
- 4、确立项目计划

第一节 现有系统介绍

一、现有版本及开发环境

Patient ObservationV1.00 是我在 2002 年 11 月 26 日，完成的一个课程设计作业。起初以为时间充足可以设计的更周详一些，但毕竟个人软件开发过程的管理能力差，不得不将系统设计成为一个定制系统。但作业还是完全满足课程设计要求，得到老师的肯定。



图 1-1 Patient ObservationV1.00 主界面

Patient ObservationV1.00 采用面向对象的分析和设计技术开发，并使用 Visual UML2.8.2 进行系统分析和设计，应用 Jbuilder4.0 编码实现。Patient ObservationV1.00 主界面如图 1-1 所示。

二、基本功能

Patient ObservationV1.00 具有如下功能：

- 使用动态数组，建立和保存多个病人的档案数据；
- 可以对病人进行心率、血压、XX 反应、YY 反应的测试；
- 对病人的观测数据，根据相应的标准，产生诊断结果；
- 可以对病人各个不同的档案数据进行查询。



图 1-2 Patient ObservationV1.00 的观测实例

三、基本用例

Patient ObservationV1.00 如图 1-3 所示。

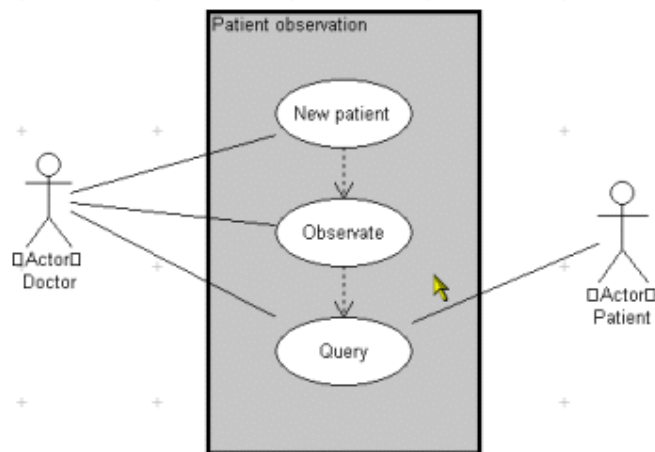


图 1-2 Patient ObservationV1.00 的基本用例

四、现有系统的主要不足

Patient Observation V1.00 的不足表现在：

- 1) 没有对使用软件的用户进行身份区别。
- 2) 系统没有对观测系统的数据输入进行细致的考虑，仅以手工输入数据，并进行界面的输入校验，以确保数据输入的有效性。
- 3) 系统没有对诊断系统进行较为规范的定义，并建立相应的诊断系统的维护和管理功能。
- 4) 全部数据保存在内存中，没有输出到相应的文件（如数据库）中。
- 5) 系统对病人查询的处理只能处理单项指标的查询工作。如果一个病人一次观察多个观测项目，那么需查询多次方可将一次观察内容查询清楚。
- 6) 系统部署的软件、硬件环境没有考虑。

第二节 新版本系统的项目计划

一、项目范围和目标

新系统将以 V2.00 为最终发布版本，在 V2.00 版将克服以上 V1.00 版中六点不足，Patient Observation V2.00 的所要实现的功能目标是：

1. 架 B/S 结构的 Patient Observation 系统，实现医院门诊信息管理工作的分布式计算机处理；
2. 所有有效数据存放到信息中心，通过大型数据库管理系统进行管理；
3. 对使用本系统的用户设计合理的使用权限，促进业务工作的顺利进行；
4. 系统将接收终端观测设备的数据，并根据专家系统返回实际诊断结果，以供主治医师使用。
5. 提供安全的、使用方便的查询功能；即能够保护病人隐私、又能方便查询。

二、项目可行性分析

本系统已运行了一阶段，不仅了解用户的需求，而且也树立了一定的产品形象，并且新系统是可行。

1. 经济可行性

开发本系统的投资约在 5 万左右，随着我国广大医院的信息化工作的不断深入，医院势必要将其信息优势扩大为业务优势；针对特定病人（老年、腿脚不便、需要不断跟踪观测者）提供网络在线医疗服务。随着人中老龄化，本系统有一定的应用市场，投资可以完全收回，并产生良好的经济和社会效益。

2. 技术可行性

本系统采用先进的数字信息采集终端，收集病人的相关数据，通过互联网传输信息给医生，所用技术都是较成熟并广泛应用的技术，在技术上是可行的。（还需评价系统方案，此处略。）

3. 社会可行性

本系统的主要目标市场定位是社区医护中心；但也可以适用于普通的医院。

能够将社区医护中心的管理工作信息化，可以通过互联网进行网上就诊，给病人带来极大的方便；这样的技术已在日本等发达国家应用。

综上所述，本系统的开发立项是可行的。

三、项目进度计划

本项目按照作业内容，安排进度如表 1-1：

表 1-1 项目进度计划表

序号	项目	天数	主要工作
1	系统分析	1	完成对旧系统的分析，初步设定新系统的目标，并进行可行性分析，确立项目计划。
2	需求分析	2	确定系统的要求：功能、性能、运行、扩充； 分析数据：系统的概念模型、逻辑模型； 修正项目计划； 给出系统原型。
3	概要设计	2	确定最佳方案； 结构设计； 数据库设计； 测试计划； 文档审核。
4	详细设计	3	用 UML 的行为视图和结构视图描述系统外观
5	系统测试	0.5	提供测试方案
6	系统移交	0.5	向老师交作业

第二章 需求分析



作业任务

- 1、确定系统的要求：功能、性能、运行、扩充
- 2、分析数据：系统的概念模型、逻辑模型
- 3、修正项目计划

第一节 基本需求规格说明

一、系统概貌

- 本系统是一个医务信息支撑系统。作为网络在线医务服务平台，系统基于 B/S 运行方式的提供网络在线医疗服务。

二、功能要求(Functional requirements)

- 病人可以通过医院挂号窗口，或者通过互联网网上挂号。
- 病人可以使用网络终端设备观测相关项目，并可将观测数据传送到医护信息中心。
- 病人也可将自己观测的数据发送到医护信息中心。
- 医护信息中心向病人提供医生信息，以及和医生网络沟通方式；医护信息中心向病人提供个人病历查询。
- 医护信息中心向医生提供病人信息，以及诊断报告，供医生对病人进行相应的用药和护理指导。
- 医生可以管理和跟踪自己的病人。

三、性能要求(Performance requirements)

- 安全性：提供用户身份验证、病人隐私保护，病员数据安全策略。
- 友好性：好的图形用户界面。
- 系统容量：系统可以同时为 500 个病人提供在线服务。
- 系统响应：系统设置病人观测等级，对重要病人能够提供优先诊断和紧急护理呼叫。
- 稳定性：系统 7 天×24 小时工作，并提供数据安全备份。

四、运行要求(Operational requirements)

1. 运行环境：系统能运行于通用的技术环境(如 Unix、Windows 等)中，并能够通过互联网和局域网支持医务工作。

2. 软件配置：

服务器端：Microsoft WindowsNT2000 Server；

Microsoft SQLServer2000；

Internet Information Server 5.0。

客户端：MicroSoft Windows95/98/2000/XP

Internet Explorer5.0/6.0

The observer for Patient Observation V2.00 (Client)

3. 硬件配置

(略)

五、扩充要求(Possible evolution)

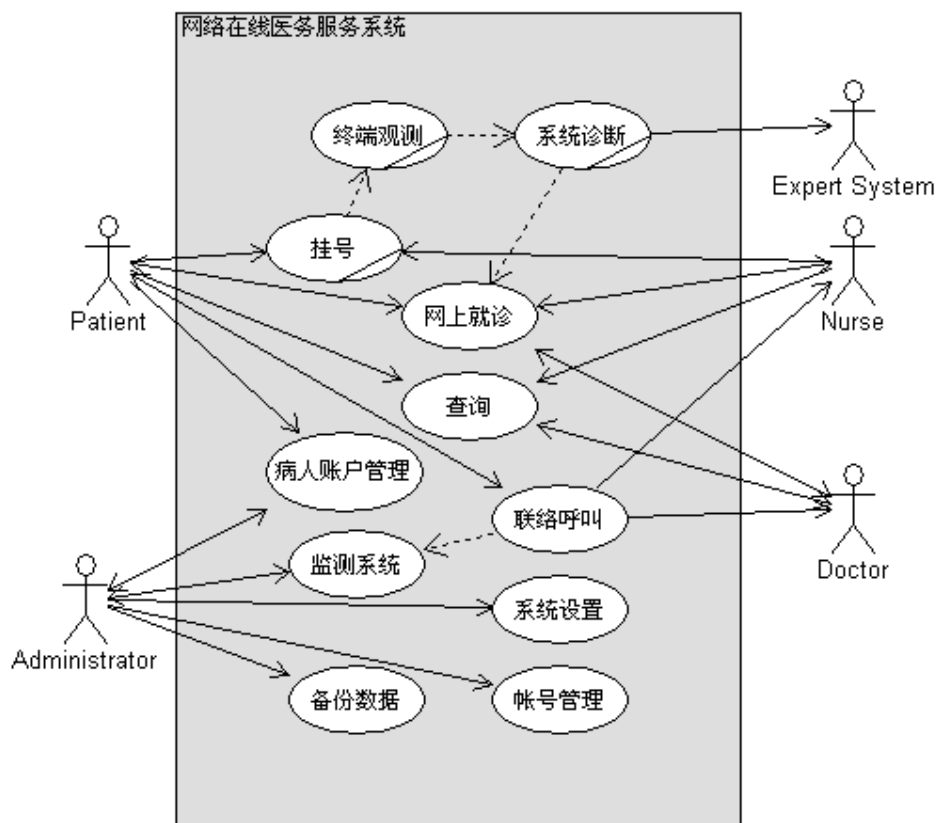
- 系统容易维护，便于功能扩充。

第二节 用例分析

一、系统基本用例

本系统的系统边界是：网络在线医务服务平台。

系统的参与者有：病人，医生，护士，医疗专家系统，系统管理员。用例图如图 2-1。



注：为了图示清晰，本用例图省去了“登录系统”、“退出系统”两个用例。

图 2-1 Patient Observation V2.00 用例图

1、病人通过系统可实现以下使用需求。

- 挂号
- 终端观测
- 紧急呼叫
- 网上就诊
- 查询病历
- 病人账户管理

2、医生通过系统可实现以下使用需求。

- 查询病人资料
- 就诊服务
- 获得紧急呼叫

3、护士通过系统可实现以下使用需求。

- 挂号
 - 就诊服务
 - 联络病员
- 4、医疗专家系统通过系统可实现以下使用需求。
- 诊断观测
 - 医疗系统服务接口
- 5、系统管理员通过系统可实现以下使用需求。
- 病人账户管理
 - 用户帐号管理
 - 备份数据
 - 监测系统
 - 系统设置

二、系统业务逻辑描述

本系统的业务逻辑是这样的：病人首先使用挂号用例，取得相应帐号和权限后，使用“终端观测”发送观测数据给“网络在线医疗服务系统”，“网络在线医疗服务系统”的系统诊断用例将访问“专家系统”，“专家系统”将相应的医治专家系统，对病人的观测数据进行诊断；产生和数据存放在数据库中。

医生、病人、护士使用“在线联络”用例，可以进行信息交流。

病人可通过“紧急呼叫”用例，与医生、护士、系统管理员（值班人员）取得联系。

专家系统实现对观测数据的分析，并将结果保存在病人病历中。

三、数据流图分析

1、数据流图

将系统业务逻辑用数据流图描述，可表示为：

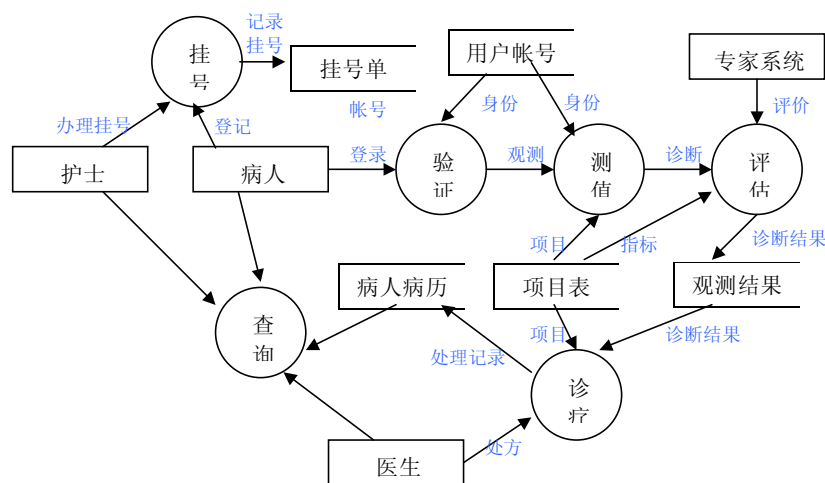


图 2-2 Patient ObservationV2.00 数据流图

2、数据流分析

编 号	01	数 据 流 名 称	办理挂号
别 名	无	来 源	护士
数 据 流 量	500 条/天	去 向	挂号
组 成	挂号号码		

编 号	02	数 据 流 名 称	登记
别 名	无	来 源	病人
数 据 流 量	500 条/天	去 向	挂号
组 成	挂号号码 + 姓名 + 性别 + 年龄 + 通信地址 + 电话号码		

编 号	03	数 据 流 名 称	记录挂号
别 名	无	来 源	挂号
数 据 流 量	500 条/天	去 向	挂号单
组 成	挂号号码 + 1		

编 号	04	数 据 流 名 称	登录
别 名	无	来 源	病人
数 据 流 量	500 条/天	去 向	验证
组 成	挂号号码 + 用户帐号 + 口令		

编 号	05	数 据 流 名 称	身份
别 名	无	来 源	用户帐号
数 据 流 量	500 条/天	去 向	验证、测值
组 成	用户帐号		

编 号	06	数 据 流 名 称	观测
别 名	无	来 源	验证
数 据 流 量	500 条/天	去 向	测值
组 成	挂号号码 + 用户帐号		

编 号	07	数 据 流 名 称	诊断
别 名	无	来 源	测值
数 据 流 量	N*500 条/天	去 向	评估
组 成	挂号号码 + 用户帐号 + 项目 + 观测数据		

编 号	08	数 据 流 名 称	评价
别 名	无	来 源	专家系统
数 据 流 量	N*500 条/天	去 向	评估
组 成	挂号号码 + 用户帐号 + 项目 + 诊断结果		

编 号	09	数 据 流 名 称	诊断结果
别 名	无	来 源	评估
数 据 流 量	N*500 条/天	去 向	观测结果
组 成	挂号号码 + 用户帐号 + 项目 + 诊断结果		

编 号	10	数 据 流 名 称	项目
别 名	无	来 源	项目表
数 据 流 量	N*500 条/天	去 向	测值、诊疗
组 成	项目编号 + 项目名称		

编 号	11	数据流名称	指标
别 名	无	来 源	项目表
数据流量	N*500条/天	去 向	评估
组 成	项目编号 + 值上界 + 值下界 + 诊断结果		

编 号	12	数据流名称	诊断结果
别 名	无	来 源	观测结果
数据流量	M*N*500条/天	去 向	诊疗
组 成	挂号号码 + 用户帐号 + 项目 + 诊断结果		

编 号	13	数据流名称	处方
别 名	无	来 源	医生
数据流量	M*N*500条/天	去 向	诊疗
组 成	挂号号码 + 用户帐号 + 处理方案		

编 号	14	数据流名称	处方
别 名	无	来 源	医生
数据流量	N*500条/天	去 向	诊疗
组 成	挂号号码 + 用户帐号 + 处理方案		

编 号	15	数据流名称	处理记录
别 名	无	来 源	诊疗
数据流量	N*500条/天	去 向	病人病历
组 成	挂号号码 + 用户帐号 + 处理方案		

第三节 分析数据

一、系统的概念模型

根据业务逻辑，建立系统的 E-R 模型。

- 1、目前，初步涉及到的系统实体有：病人、医生、护士、挂号、观测、观测结果、观测项目、评价指标。
- 2、病人和医生是多对多的关系，实体挂号作为二者的关系存在。

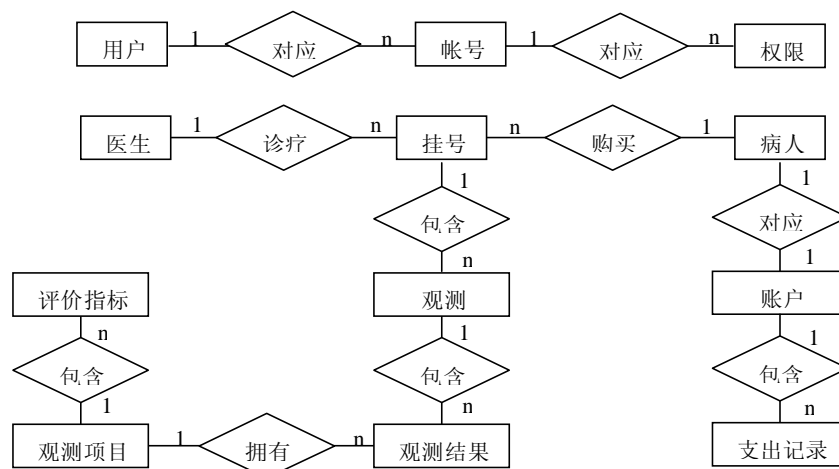


图 2-3 2 Patient ObservationV2.00 E-R 图

3、每一个挂号实体对应着一次就诊，而每次就诊可能有一个或多个观测项目，每个观测项目对应一个观测结果。

基本的 E-R 模型图，设计如图 2-3 所示。

二、系统的逻辑模型

将 E-R 图转换成 Microsoft SQL server 数据库的逻辑设计。

病人（病人编号，姓名，性别，出生日期，住址，电话，建档日期，消档日期）

医生（医生编号，医生姓名，性别，出生日期，住址，电话，建档日期，消档日期）

挂号（挂号流水号，挂号号码，*病人编号*，观测日期）

观测（观测流水号，*挂号号码*，*病人编号*，观测者，观测日期，医生处方）

观测结果（流水号，*观测流水号*，*观测项目编号*，*病人编号*，值类型，观测值）

观测项目（观测项目编号，项目名称，项目说明）

评价指标（指标编号，指标名称，*观测项目编号*，编号类型，上界值，下界值）

病人账户（账户流水号，病人编号，现有余额，建账日期）

病人支出记录（流水号，挂号流水号，支出额，支出日期）

病人存款记录（流水号，存款日期，存款金额）

用户帐号（用户编号，口令，*组类型ID*，帐号过期日期，启用标志）

组（组类型ID，组名称，组说明）

权限表（权限编号，*组类型ID*，功能编号，授权标志）

第四节 项目计划的修订

一、成本估计

（略）

二、资源使用计划

（略）

三、改进的进度计划

表 2-1 新进度计划

序号	项目	天数	主要工作
1	系统分析	1	略
2	需求分析	2	略
3	概要设计	3	确最佳方案；结构设计；数据库设计；测试计划；文档审核。
4	详细设计	4	用 UML 的行为视图和结构视图描述系统外观
5	系统测试	0.5	提供测试方案
6	系统移交	0.5	向老师交作业

根据目前的完成情况，本作业的时间估计如表 2-1 所示。

第五节 系统原型

由于时间关系，在此不能设计相应的原型，暂且将 V1.00 版作为附件上交。

第三章 概要设计



作业任务

- 1、系统结构设计
- 2、系统数据库设计

第一节 系统结构设计

本系统主要包括以下四个包：用户接口包、商业对象包、数据库包、实用包。

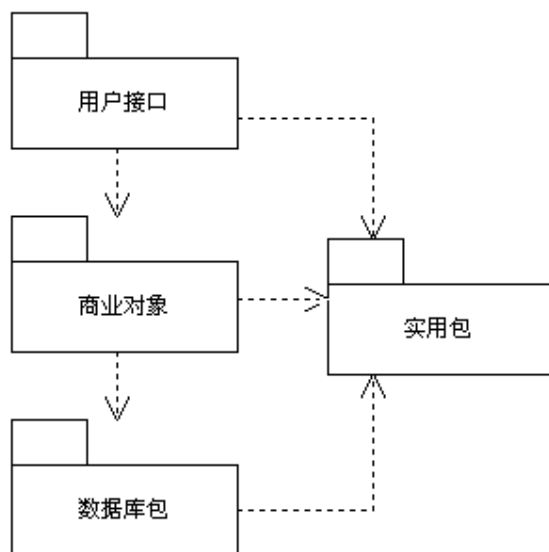


图 3-1 系统的子系统组成

1. 用户接口包：用户接口类允许用户访问系统数据和加入新数据。在商业对象中,用户接口包跟商业对象包合作,调用商业对象的操作,实施数据的检索和插入。
2. 商业对象包：包括来自分析阶段的特定领域类。在设计阶段,详细设计这些类,以完

整定义他们的操作,支持对数据库的存取。所以,所有商业对象类必须继承数据库包中的类。

3. 数据库包: 为商业对象包中的类提供服务,便于永久存储。
4. 实用包: 包含系统其他包要使用的服务。

一、用户接口包

在用户接口包中,主要有以下类:

- 1、病人接口类: 提供病人信息注册, 和病人个性化服务的类;
- 2、挂号台类: 用于向病人提供挂号服务功能的类;
- 3、在线医务系统类: 用于向病人提供查询和在线联络功能的类;
- 4、终端观测类: 用于病人提供观测数据输入功能的类;
- 5、专家系统接口类: 此类向系统提供其他医务信息专业处理的接口, 以提高系统的扩展性。

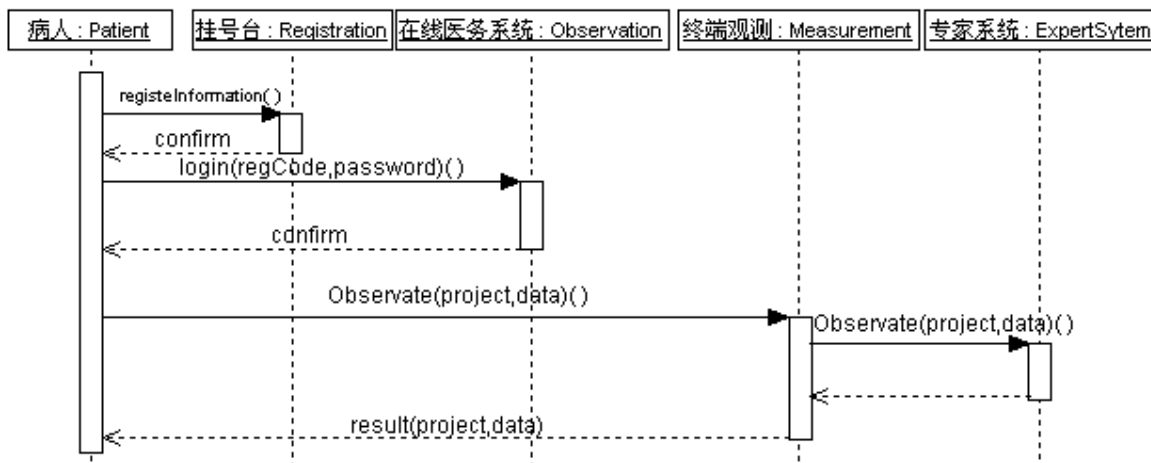


图 3-2 表示用户接口包中基本的类之间的交互

二、商业对象包

在商业对象包中有以下主要类, 见图 3-3:

- 1、病人类: 主要存放有关病人的数据, 以及对观察和测量的检索功能。
- 2、观察类: 主要存放病人每次的观察情况。
- 3、测量类: 对观测的数据进行诊断的类。
- 3、现象类型类: 描述某个分类的观测项目的类。
- 4、现象类: 描述某类观测项目的诊断指标的类。
- 5、处方类: 描述医生对病人所做的治疗方案及相关操作。
- 6、一些描述量度单位的类: 如, 范围、数量、量纲等。

除此之外, 还有:

- 账户类: 对病人的账目进行收支管理。
- 计费类: 对观测和诊断按时进行计费的类。

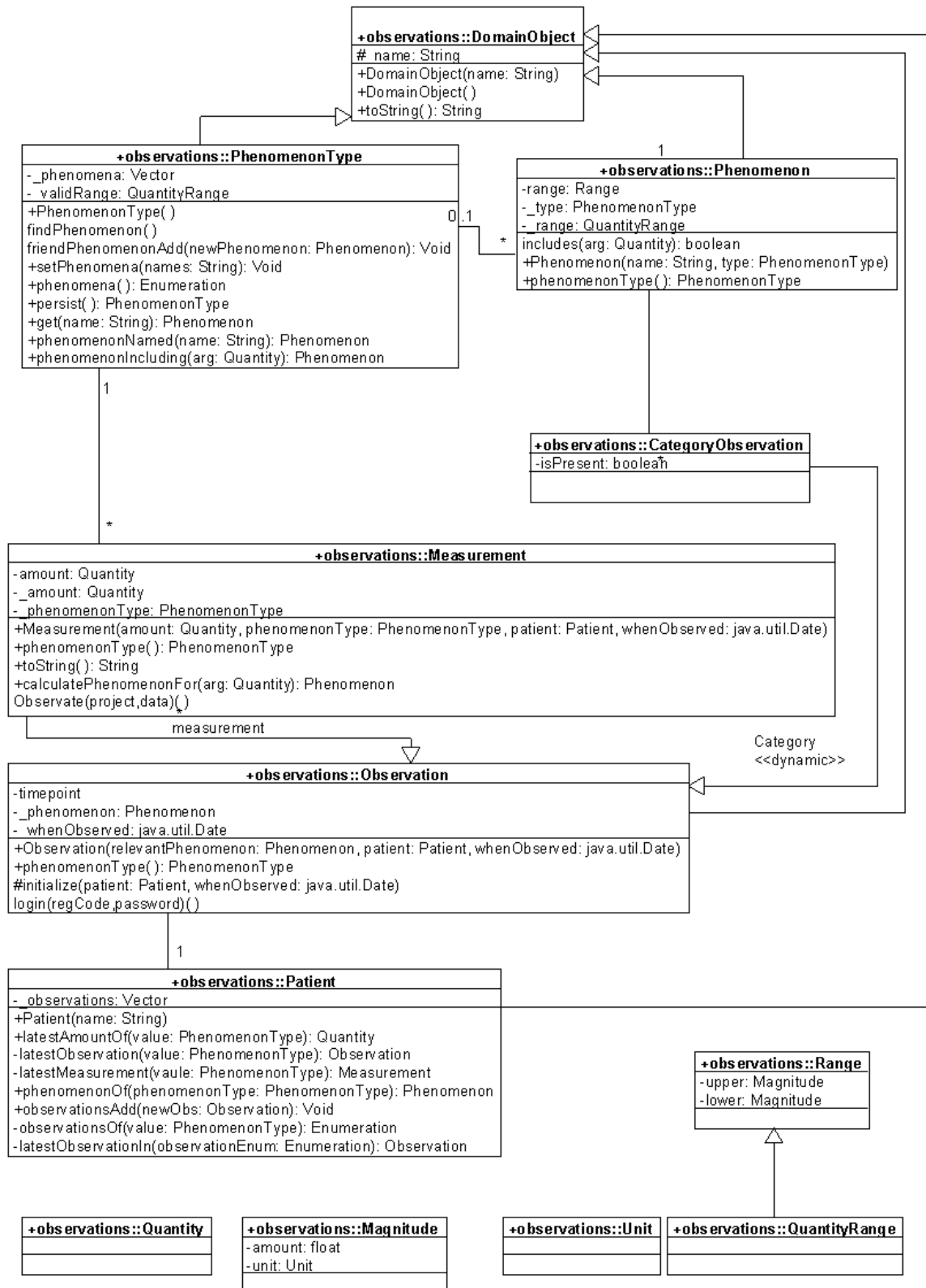


图 3-3 商业对象包中有关观测的类

三、数据库包

提供了访问后台数据库的相关类，针对业务逻辑的需要设计。具体有：

- 查询类：主要提供查询相关业务数据的类。如查询观测项目指标体系的类，查询病人病历、就诊情况的类、查询病员账目情况的类；
- 存储类：主要有完成数据存储和转换的类。如完成业务数据从内存转换到数据表的类；将数据表的类导入内存中的数据结构的类。
- 更新类：实现更新、备份数据表的类。

四、实用包

包含系统其他包要使用的服务。如系统状态信息类，系统事务管理类等。

第二节 子系统划分

应用 U/C 矩阵对 Patient Observation V2.00 进行功能子系统的划分，如表 3-1 所示。本系统主要划分为四个功能子系统。

- 系统管理：进行用户帐号管理和相关系统参数的设置；
- 费用管理：主要进行病人账户的维护管理工作；
- 在线就诊：对病人进行身体状态观测，并提供用药指导；
- 挂号咨询：提供挂号、就诊排队、查询病历的功能。

表 3-1 用 U/C 矩阵对 Patient Observation V2.00 进行功能子系统的划分

功能 \ 数据		病人	医生	护士	管理员	项目	指标	病人 账户	服务 费用	账户 余额	观测 数据	观测 项目	观测 结论	医生 处方	挂号 单	病历
系统 管理	用户帐号管理	C	C	C	C											
	设置项目				U	C										
	设置指标				U		C									
费用 管理	服务计费管理	U						C	U	U		U				
	病人缴费管理	U						U	C	U						
	病人账户管理	U								C						
在线 就诊	采集病人数据	U									C					
	拟定观察方案	U	U			U						C				
	诊断观察结果						U				U	U	C			
	编写处理方案	U	U										U	C		U
挂号 咨询	挂号单管理			U											C	
	就诊管理	U	U												U	C
	查询	U	U	U		U	U	U	U	U	U	U	U	U	U	U

第三节 数据库设计

根据需求分析阶段的 E-R 模型，建立实际的数据表，将有关 SQL 语句呈现如下。

1、病人

```
create table Patient (patientCode char(10) ,
                    patientName char(10) NOT NULL,
                    patientSex boolean,
                    patientBirthday datetime,
                    patientAddress varchar(40),
                    patientPhone char(18),
                    patientCreateDate datetime,
                    patientCancleDate datetime)
```

2、医生

```
create table Doctor (doctorCode char(10),
                    doctorName char(10) NOT NULL,
                    doctorSex boolean,
                    doctorBirthday datetime,
                    doctorAddress varchar(40),
                    doctorPhone char(18),
                    doctorCreateDate datetime,
                    doctorCancleDate datetime)
```

3、挂号

```
create table Doctor ( registrationID char(10) indentify(1,1) NOT NULL,
                    registratCode char(4),
                    patientCode char(10),
                    doctorCode char(10),
                    Observateday datetime)
```

4、观测

```
create table Observation ( observationID char(10) indentify(1,1) NOT NULL,
                    registrationID char(10),
                    patientCode char(10),
                    Observer char(10),
                    observeDay datetime,
                    DoctorDealWith varchar(200))
```

5、观测结果

```
create table ObservationResult
( observationResultID char(10) indentify(1,1) NOT NULL,
  observationID char(10),
  projectCode char(4),
  patientCode char(10),
  vaulesType char(2),
  vaules varchar(10))
```

6、观测项目

```
create table ObservationProject
( observationProjectID char(10) NOT NULL,
  observationProjectName char(10),
  observationProjectDetailText varchar(50)
)
```

7、评价指标

```
create table item (  itemCode char(8)  NOT NULL,
                    itemName char(20) NOT NULL,
                    observationProjectID char(10),
                    itemType char(3),
                    itemUpper float(6,2),
                    itemlower float(6,2))
```

8、病人账户

```
create table PatientAccount
( patientAccountID char(10)  NOT NULL,
  patientCode char(10),
  accountTotal money,
  createAccountDate datetime
)
```

9、病人支出记录

```
create table AccountSpent
( AccountChangID char(10) indenty(1,1) NOT NULL,
  registrationID char(10),
  spent money,
  spentDate datetime)
```

10、病人存款记录

```
create table AccountSave( AccountChangID char(10) indenty(1,1) NOT NULL,
                          saveDate datetime,
                          save money)
```

11、用户帐号

```
create table User(  userID char(10) indenty(1,1) NOT NULL,
                  password char(10),
                  groupID char(10),
                  endDayOfLimit datetime,
                  avaiable boolean)
```

12、权限表

```
create table userPower( PowerID char(10) indenty(1,1) NOT NULL,
                      groupID char(10),
                      FunctionCode char(4),
                      avaiable boolean)
```

13、组

```
create table Group( groupID char(10) indenty(1,1) NOT NULL,
                  groupName char(10),
                  groupDeatils varchar(100))
```

第四章 详细设计



作业任务

- 1、功能模块的实现
- 2、系统的部署

第一节 系统功能子模块的实现

在概要设计阶段，我们应用 U/C 矩阵将 Patient Observation V2.00 划分为四个功能子系统。现对各个系统的实现进行详细设计

一、系统管理子系统

系统管理主要由“用户帐号管理”，“设置项目”，“设置指标”组成。另外，还有“数据备份”，“系统性能监测”。

1. 用户帐号管理

1) 用户帐号管理的形式

- 用户帐号管理分为：组管理，用户帐号管理。（类似于 WindowsNT）
- 用户帐号的安全属性有：口令，所属组，过期期限，启用帐号。
- 组的安全属性依赖于对权限表的设置。权限表可以通过字节编码或数据表来实现，本系统采用数据表实现。
- 任何一个用户至少隶属于一个组。其权限取各组权限的并集。

2) 常设组的权限

系统设置四个常设组：PatientGroup、NurseGroup、DoctorGroup、AdimistratorGroup。

Function	PatientGroup	NurseGroup	DoctorGroup	AdimistratorGroup
用户帐号管理	×	×	×	✓
设置项目	×	×	×	✓
设置指标	×	×	×	✓
系统性能监测	×	×	×	✓
数据备份	×	×	×	✓
服务计费管理	×	×	×	✓
病人缴费管理	✓	×	×	✓
病人账户管理	×	✓	×	×
采集病人数据	✓	×	×	×
拟定观察方案	×	×	✓	×
诊断观察结果	✓	✓	✓	×
编写处理方案	×	×	✓	×
挂号单管理	×	✓	×	×
就诊管理	×	✓	✓	×
查询	✓	✓	✓	✓

2. 设置项目和指标

以心率为例说明，项目和指标设置关系。

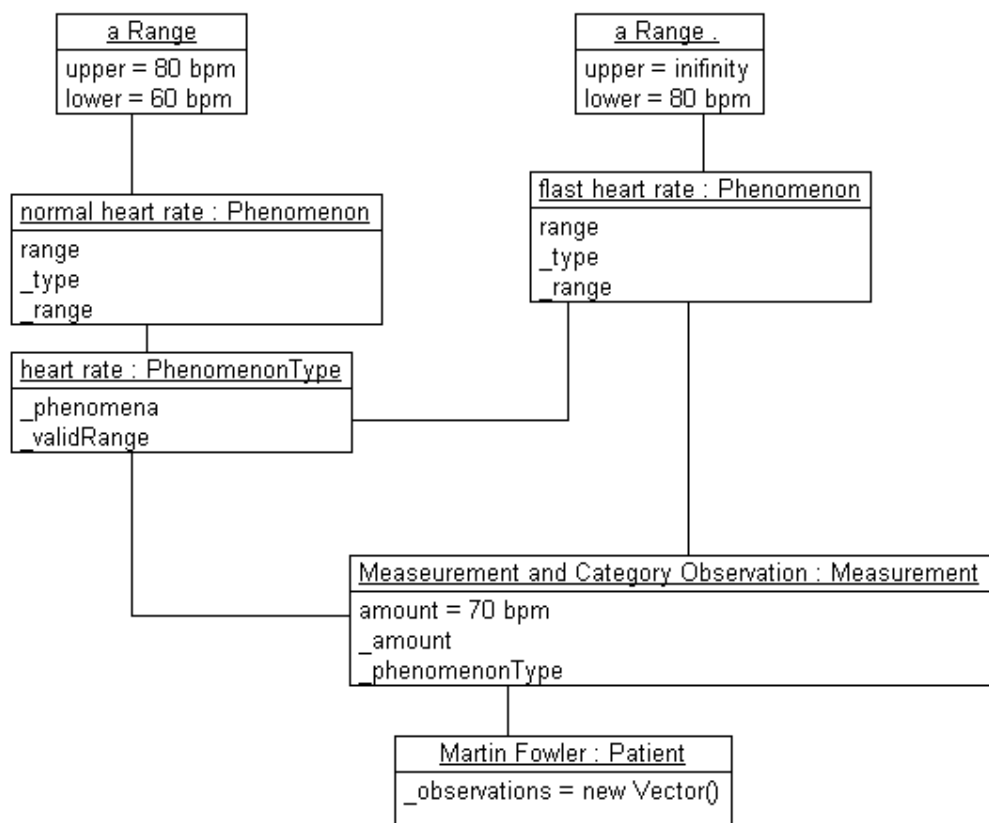


图 3-4 设置项目和指标的规约

二、费用管理子系统

费用管理子系统主要包括：服务计费管理、病人缴费管理、病人账户管理等。另外还包括科目设置，与其他会计系统的数据接口等。

1. 服务计费管理

以挂号为单位，对每次就诊进行计费。

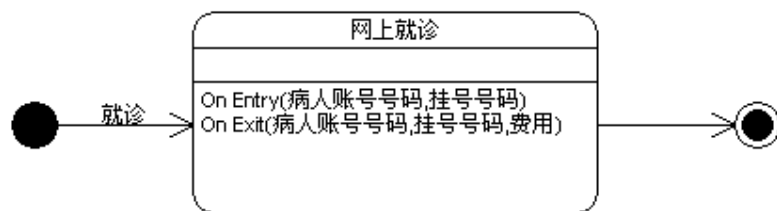


图 3-5 服务计费管理

2. 病人缴费管理与病人账户管理

1) 病人缴费管理

病人可以通过网络电子支付，或到服务中心交费。

功能有：存款、退款、催款。

2) 病人账户管理

一个病人只能有一个账户。

功能有：建立账户、消除账户、设置密码。

三、在线就诊子系统

本子系统包括如下功能：采集病人数据、拟定观察方案、诊断观察结果、编写处理方案、联系呼叫。

1. 采集病人数据

病人数据的采集方式可通过终端联网医疗设备或通过人工观测手工输入的方式。

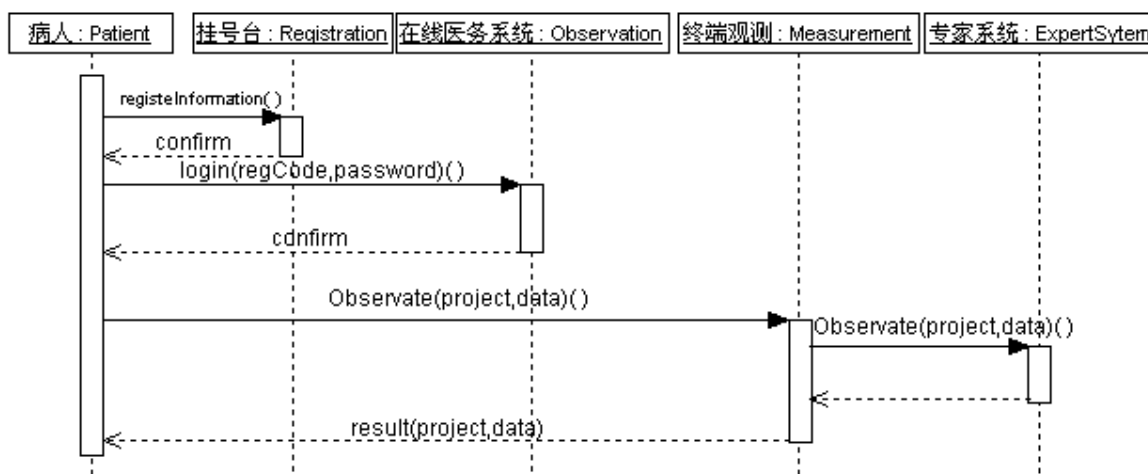


图 3-6 采集病人数据

2. 拟定观察方案

观察方案的拟定是“网上就诊”用例中的一部分。是由病人和医生或医生完成。

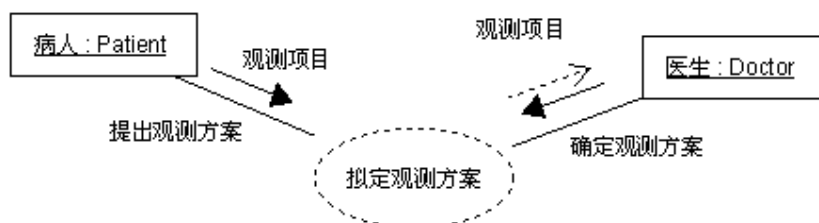


图 3-7 拟定观察方案

3. 诊断观察结果

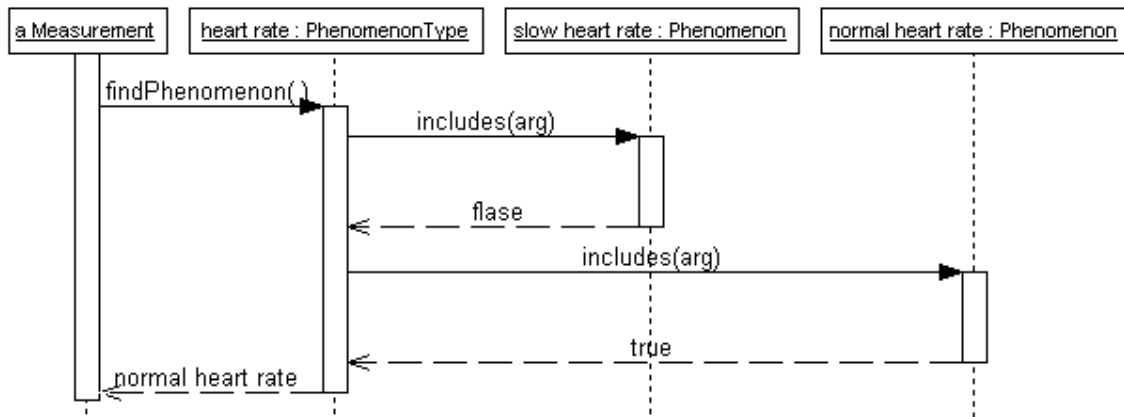


图 3-8 诊断观察结果

```

// =====
// File Name: Measurement.java
// Model Name: PatientObserver
// Generated by Visual UML 2.8.2 at 17:31:15 on 2002 年 11 月 16 日
// =====

package observations;

// Class Name: Measurement
// Package Name: observations
// Generated by Visual UML 2.8.2 at 17:31:15 on 2002 年 11 月 16 日

/** 度量 */
public class Measurement extends Observation {
// Attributes:

    private Quantity _amount;
    private PhenomenonType _phenomenonType;

// Operations:

    public Measurement(Quantity amount, PhenomenonType phenomenonType, Patient
patient, java.util.Date whenObserved) {
        initialize(patient,whenObserved);
        _amount = amount;
        _phenomenonType = phenomenonType;
        _phenomenon = calculatePhenomenonFor(_amount);
    }

    public Quantity amount(){
        return _amount;
    }
    public PhenomenonType phenomenonType() {
        return _phenomenonType;
    }

    public String toString() {
        return _phenomenonType + ":" + _amount;
    }

    public Phenomenon calculatePhenomenonFor(Quantity arg) {
        return _phenomenonType.phenomenonIncluding(arg);
    }
}

```

```
// =====
// File Name: PhenomenonType.java
//
// Model Name: PatientObserver
//
// Generated by Visual UML 2.8.2 at 17:31:14 on 2002 年 11 月 16 日
// =====

package observations;
import java.util.Vector;
import java.util.Enumeration;

// Class Name: PhenomenonType
// Package Name: observations
// Generated by Visual UML 2.8.2 at 17:31:15 on 2002 年 11 月 16 日

/** 现象类型 */
public class PhenomenonType extends DomainObject {
// Attributes:

    private Vector _phenomena = new Vector();
    private QuantityRange _validRange;

// Operations:

    public PhenomenonType(String name) {
        super(name);
    }

    void findPhenomenon() {}

    void friendPhenomenonAdd(Phenomenon newPhenomenon) {
        //RESTRICTED: only used by phenomenon
        _phenomena.addElement(newPhenomenon);
    }

    public void setPhenomena(String[] names) {
        for (int i = 0; i < names.length; i++) new
Phenomenon(names[i],this);
    }

    public Enumeration phenomena() {
        return _phenomena.elements();
    }

    /** this operation used to save phenomeon to login object. */
    public PhenomenonType persist() {
        /**所有的 persist()操作把现象类型储存到登录对象中
        *以便以后可用静态的 get()方法再次取得它
        */
        return this;
    }

    public PhenomenonType get(String name) {
        return this;
    }

    public Phenomenon phenomenonNamed(String name) {
        Enumeration e = phenomena();
        while(e.hasMoreElements())
        {
            Phenomenon each = (Phenomenon) e.nextElement();
            if (each.name().equals(name)) return each;
        }
        return null;
    }
}
```



```

        public Phenomenon phenomenonIncluding(Quantity arg) {
            Enumeration e = phenomena();
            while(e.hasMoreElements())
            {
                Phenomenon each = (Phenomenon)e.nextElement();
                if (each.includes(arg)) return each;
            }
            return null;
        }
    }

// =====
// File Name: Phenomenon.java
//
// Model Name: PatientObserver
//
//
// Generated by Visual UML 2.8.2 at 17:31:15 on 2002 年 11 月 16 日
//
// =====

package observations;

// Class Name: Phenomenon
// Package Name: observations
// Generated by Visual UML 2.8.2 at 17:31:15 on 2002 年 11 月 16 日

/** 现象 */
public class Phenomenon extends DomainObject {
// Attributes:

    private PhenomenonType _type;
    private QuantityRange _range;

// Operations:

    public Phenomenon(){}

    public Phenomenon(String name, PhenomenonType type) {
        super(name);
        _type = type;
        _type.friendPhenomenonAdd(this);
    }

    public boolean includes(Quantity arg) {
        return(_range == null ? false : _range.includes(arg));
    }

    public PhenomenonType phenomenonType() {
        return _type;
    }
}

```

4. 编写处理方案

编写处理方案的实现过程，如图 3-9。

医生利用图 3-6 的观测数据，以及医生和病人的网上（文字或语音）诊询，医生向病人提供用药、养护、医疗的处理方案（处方）。

本部分功能的实现依赖于处方类 `Reciper` 的实现。其中 `getScript()`显示处方内容，`writeScript()`编写处方内容，`saveScript()`保存处方内容。

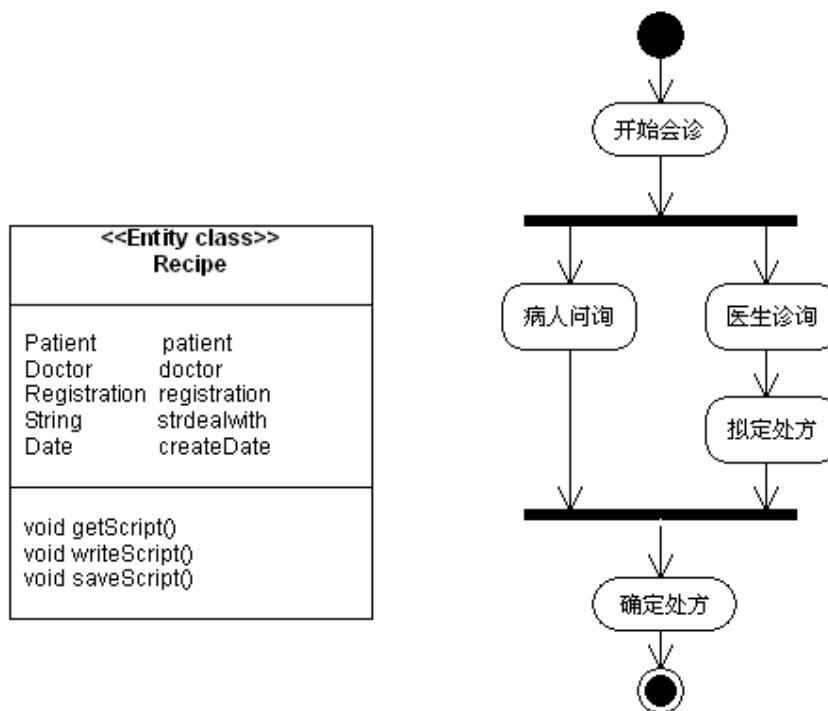


图 3-9 诊断观察结果

5. 联系呼叫

当然对于紧急情况联系呼叫的最好方法，就是打电话，本系统的联系呼叫，除有此功能外，主要提供预约、互联网发送短消息、以及与呼叫中心的数据接口等功能。

四、挂号咨询子系统

本子系统包括：挂号单管理、就诊管理、查询三个功能。

1. 挂号单管理

主要包括：当日挂号初始化、出售挂号、预约挂号。

- 当日挂号初始化，系统根据日期的变化，次日零之前更新挂号相关数据。包括：当班医生信息、挂号总数等信息，并于零时发布。
- 出售挂号：根据网上请求，经过合法性检查后，出售挂号。其中合法性检查是指对本系统的用户帐号。
- 预约挂号：用户不拥有本系统用户帐号的情况下，或想购买以后某时间的挂号，可启用此功能。

2. 就诊管理

主要包括：查询排队、加入排队、挂号通知、取挂号。

- 查询排队：可查询目前等待就诊的队列长度和评估所需等待的时间。

模块名称	就诊管理	编号	
输入	处理	输出	
医生, 日期	1. 查询排队数据表 2. 计算当前排名, 评估等待时间	排名序号, 等待时间。	

- 加入排队: 病人将自己的挂号加入队伍。

模块名称	加入排队	编号	
输入	处理	输出	
挂号, 日期, 医生	1. 添加数据到排队数据表中 2. 显示操作成败信息	成功或失败信息	

- 挂号通知: 一个挂号处理完毕, 系统自动启动挂号通知功能, 告诉医生从等待队列中取一个挂号。

模块名称	挂号通知	编号	
输入	处理	输出	
挂号处理完毕的信息	1. 还有要处理的挂号吗? 2. 有: 发送消息给医生 无: 发送消息给前台挂号系统, 挂号空闲。	响铃 “新挂号已就绪” “挂号空闲”	

- 取挂号: 医生取出挂号后, 此功能将使系统自动转入在线就诊子系统。

模块名称	就诊管理	编号	
输入	处理	输出	
就绪挂号	1. 取就绪挂号 2. 调用“在线就诊”子系统	启动“在线就诊”子系统	

第二节 系统的部署

系统的部署图如下

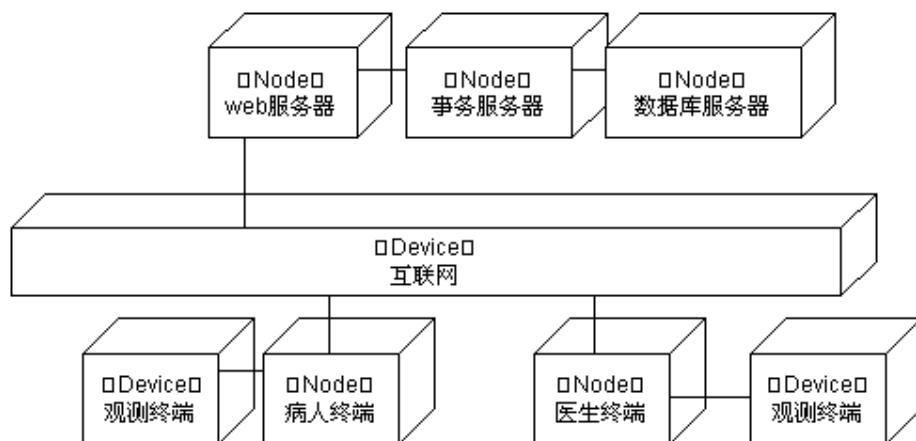


图 3-10 系统部署图

观测终端是一个观测设备, 用户观测后数据通过终端发送到服务器上。当然, 用户也可以不使用观测终端, 通过浏览器发送数据。这种系统较适用于社区医疗服务。

第五章 结束语

致谢

本文的完成，要感谢杨中海教授和李斌博士生导师授课，并通过课程设计提高了我对软件工程进一步的认知。

后记

虽然手头有两个较全面的系统分析与设计文档可应付作业，但我还是按照老师的要求从头做。由于时间和精力有限，我对作业内容做了一些限制，但仍然不能做到细致入微，尽善尽美，请老师予以谅解。

本作业累积用时 52 小时。

主要参考文献

1. 《UML 精粹（第 2 版）标准对象建模语言简明指南》
2. 《实用软件工程》（第二版）；郑人杰 等；清华大学出版社
3. 《UML 用户指南》机械工业出版社 2001-6-1
4. 《Java 面向对象编程指南》电子工业出版社 2001-11-1



附：UML 课程作业要求

以下作业要求说明，UML 课程作业目标以应用 UML 实现系统的部分功能为主，并不考虑整个软件系统的结构和用户的诸多需求。请老师予以审查比较。

Course Project

1, understand the "Patient Observation" programming listed on chapter 11 in the book, Use JBuilder or/and Rational Rose to write a application program which will include UI interface, perform their basic function and yield an executable Jar file.

2, The application should include at least four phenomenon types and definite available phenomenons as follows:

Heart rate : slow, normal, fast

XX response: -, +, ++, +++

yy response: +, -, --, ---

Blood pressure: low, lower, normal, higher, high

3, From the UI interface, the patients can show their own information, and the doctor may create new phenomenon type and collect phenomenons which belong to different phenomenon types and patients.

4, (Not necessary!) Adding four patient names and another two phenomenon type in your applicaion. Keeping these data fora long time.

5, For submitting, put your programs into a Jar file and e-mail to the teacher's mailbox. (gwyang@uestc.edu.cn) Make sure that you have mark your name and your study number clearly on your projects. By the way, you may also take any other way to submit your projects.

Due date: 2002/11/10---2003/1/10

Gwyang 2002/10/26

