

# 项目3

## 需求分析和编写

### ● 拟实现的能力目标

N1.1 能够运用需求分析方法,对软件需求进行分析;保证需求的七个特性(完整性、正确性、必要性、可行性、划分优先级、无二义性、可验证性)都得以体现;

N1.2 能够运用需求编写方法,规范并完整地编写软件需求。

### ● 须掌握的知识内容

Z1.1 需求的分类、层次以及特性;

Z1.2 需求分析的主要方法(ADMEMS矩阵分析法);

Z1.3 需求编写涉及的内容(功能需求、性能需求、质量属性需求等);

Z1.4 功能需求编写的两种方式(use case方式和IPO方式)。

本项目包含了2个学习任务,具体为:

任务1:需求分析方法;

任务2:需求编写方法。

通过该项目任务的学习,能够掌握需求分析过程的设计与编写。

## 任务1 需求分析方法

### 3.1.1 任务引导训练

引导任务:

什么是需求?需求有哪些层次?应该如何分析需求?

训练任务分析:

需求(requirements):指被描述对象“做什么”(功能需求)及“做什么”时的水平(非功能需求,如性能需求、质量属性需求、外部接口,其他需求如设计约束、复用、国际化等)。当被描述对象特指系统、软件、硬件、结构或者组件时,需求就是系统需求(system requirements)、软件需求(software requirements)、硬件需求(hardware requirements)、结

构需求(construct requirements)或者组件需求(component requirements)。

需求层次(requirement hierarchy):需求层次一般划分为市场需求、产品需求、子系统需求、模块需求。其中,后三者是面向开发(解决方案域)的。

**注意:**原始需求不是一个单独的需求层次,它可以是四个需求层次的任意一个层次,如图 3-1 所示。

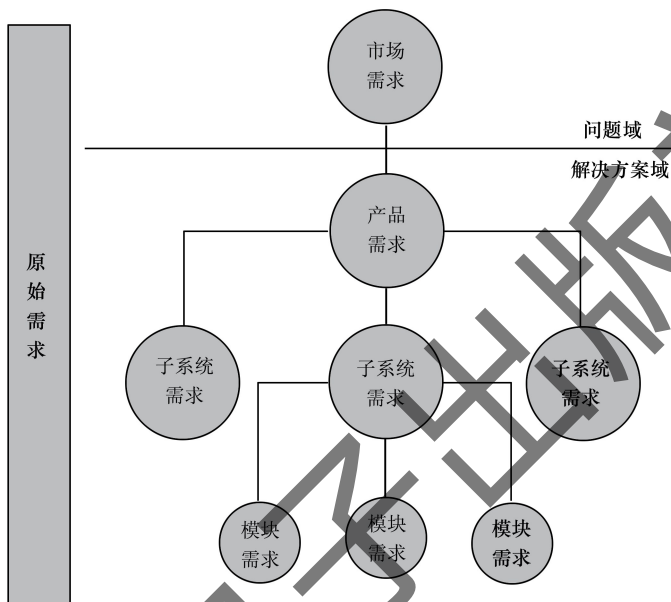


图 3-1 需求层次划分图

需求分析:为“能够高质量地描述需求”而进行的活动。所谓“高质量地描述”是指:单条需求具备 7 大特性(完整性、正确性、必要性、可行性、划分优先级、无二义性、可验证性)、需求说明书具备 4 大特性(完整性、一致性、可修改性、可追踪性)。在分析时,一般需要对上层需求进行细化、转换、显式化。

## 3.1.2 支撑知识

### 3.1.2.1 需求背景分析

#### 1. 需求背景分析需要包含的内容

(1)软件的背景和起源。对于在老版本之上升级的软件,则还应说明:

- ①老版本出现的主要问题。
- ②新版本需要增加或改进的主要内容。

(2)本软件的上下文。应使用上下文图(context diagram)说明本软件与外界(可能包括整个产品外的实体)之间的联系。图形应能清晰地表达软件与外部环境的边界、软件与外部环境的关系,以帮助读者更好、更快地理解被描述的软件。实践中经常出现的错误是:

①不画图。

②画了软件内部组成图、协议栈或软件结构图。关于最后一点,正确的做法是:应将软件结构图以“被描述对象”为中心,转化为上下文图,即去掉各个外部系统之间的联系,当然,各个外部系统与“被描述对象”的联系还得保留。

③在上下文图中,还描述了外部系统之间关系。

## 2. 需求背景分析示例

某网络设备管理软件需要管理两种设备:一种是 A 设备,一种是 B 设备。两种设备不同之处在于前者用的是 A 型机平台,后者用的是 B 型机平台。两者的操作方式差异很大:设备 A 的操作以字符人机命令为主;设备 B 的操作命令基本上已实现图形化。对于用户来说,需要在—个设备管理软件上实现对这两种设备的统一处理。本文即描述了统一管理设备 A 和设备 B 的设备管理软件的需求。

图 3-2 描述了本网络设备管理软件的上下文图。

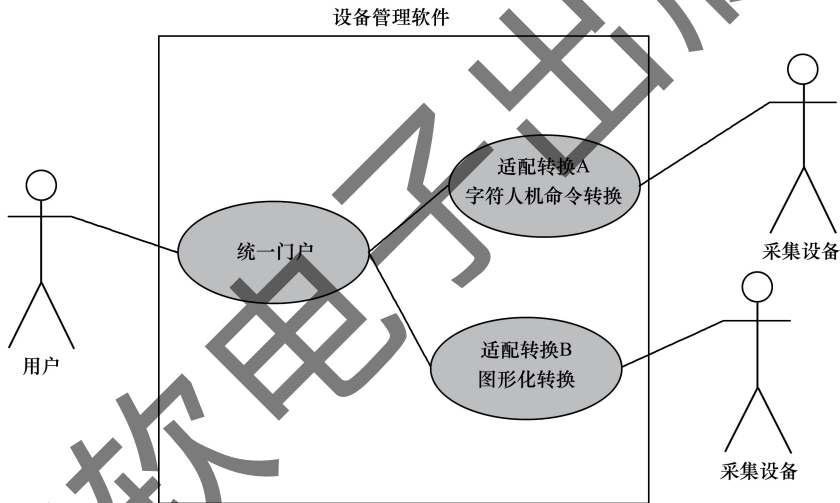


图 3-2 网络设备管理软件上下文图

本软件与外部环境及其接口的说明见表 3-1。

表 3-1

说明表

| 外部环境   | 外部环境描述 | 相关接口说明            |
|--------|--------|-------------------|
| 采集设备 A | 数据采集设备 | 通过字符串人机命令方式控制采集设备 |
| 采集设备 B | 数据采集设备 | 通过界面交换方式控制采集设备    |
| 用户     | 操作人员   | 操作人员通过统一门户与采集设备交互 |

### 3.1.2.2 软件概述分析

#### 1. 软件概述分析需要包含的内容

软件概述是概述软件所具有的主要功能、性能指标、质量属性、外部接口、领域建模、上下文和用例等。由于详细内容将在后续功能需求中描述,因此此处可以通过软件主要

的使用场景(Scenario)来组织,使读者在比较高的层面上了解本软件是如何被使用的。直接罗列后续的各篇中的所有需求(如用一个表格)并不合适,因为这会引起内容冗余,以致引起维护问题,还会增大文档篇幅。

## 2. 软件概述示例

本软件具备以下主要功能:

- 工程及设计管理;
- 用于教育领域的考试管理系统;
- 数据管理(如设计试卷、考试参数等);
- 试卷规划与分析(如试卷题型规划、分值规划、考试结果分析、易错试题分析等);
- 考试模拟(如模拟会考、模拟中等考);
- 试题优化(如题型统计分析、书库优化等);
- 视图管理(提供各种相关数据、分析结果的显示等功能);
- 实用及辅助工具(提供报告生成、考试信息查看等功能)。

### 3.1.2.3 运行环境分析

#### 1. 运行环境分析需要包含的内容

描述软件的运行环境,包括硬件环境、操作系统及其版本,还有其他的软件组件或与其共存的应用程序。如果有版本配合要求、补丁要求,则应明确列出(如使用 Windows 98,则 Office 可以使用 97 及其以上版本,如使用 Windows 2000,则应打上 SP2,且 Office 只能使用 2000 版本且应打上 SP1)。

#### 2. 运行环境示例

运行环境见表 3-2。

表 3-2 运行环境

| 名称         | 硬件(CPU/RAM/HD 等) | 操作系统及其版本            | 其他软件环境    |
|------------|------------------|---------------------|-----------|
| COLL       | PIII700/1G/18G   | VxWorks X           | 无         |
| DEAL       | PIII700/1G/18G   | VxWorks X           | 无         |
| PROC       | PIII700/1G/18G   | Windows NT 4 Server | Oracle 8i |
| DB Server  | SPARC/1G/18G     | SUN Solaris X       | Oracle 8i |
| DIR Server | SPARC/1G/18G     | SUN Solaris X       | Oracle 8i |

### 3.1.2.4 用户类分析

#### 1. 用户类分析需要包含的内容

应根据用户使用软件的频度、应用领域和计算机系统知识、需要完成的任务、地理上的布局以及访问优先级等对用户进行分类。不同的用户类,其需求是不同的,某些用户类的需求可能更加重要,需要优先考虑。

每一个用户类都有自己的一系列功能和非功能需求。如一个没有经验或偶尔使用电脑的用户关心软件是否简单易用,因此,菜单、提示符和向导是很重要的。然而,对于那些

一天使用几小时软件的用户,他们更关心软件的易用性和高效性,所以他们喜欢使用快捷键、宏。

有一些受软件影响的人并不一定是软件的直接使用者,而是通过相关的软件访问本软件的数据和服务。这些非直接的或次级(secondary)的用户也有自己的需求,可以将这些人加入“附加的用户类”。

用户类不一定都指人,其他与本软件存在接口的软件(及其开发人员)也是比较重要的用户类。

实践中,应尽早为软件确定出用户类及其代表,以广泛、完整地收集需求。

## 2. 用户类分析示例

本软件涉及到的用户类见表 3-3。

表 3-3 用户类

| 序号 | 用户类      | 说明           |
|----|----------|--------------|
| 1  | 设备维护人员   | 关注软件的功能和易用性  |
| 2  | 内部维护人员   | 关注软件的可维护性    |
| 3  | 内部售后人员   | 关注软件的价值所在    |
| 4  | 外部开发者    | 关注外部与本软件的接口  |
| 5  | 相关设备的开发者 | 关注软件的功能和实现限制 |

### 3.1.2.5 需求综合分析

软件需求综合分析主要对整个软件系统做全景式的需求分析,用结构化的需求分析思想在模块级层次上使需求更全面,没有遗漏。

#### 1. 需求结构化

需求结构化主要是利用 ADMEMS 矩阵分析工具,以用户需求说明书为输入,对软件需求进行查漏补缺式的分析,主要关注需求的全面性,避免遗漏。需求的粒度到模块即可。

ADMEMS 需求矩阵见表 3-4。

表 3-4 ADMEMS 需求矩阵表

| 分类  |      | 功能                 | 质量                | 约束  |
|-----|------|--------------------|-------------------|---|
| 组织级 | 产品规划 | 产品线 RoadMap 中的功能规划 | 产品规划中应该将“快好省”原则细化 | 产品规划中应该考虑的约束:<br>产品研发人力资金的投入;<br>标准的约束;<br>竞争对手的情况;<br>相关产品的市场占有率情况;<br>相关技术的发展趋势;<br>计算机技术的发展趋势;<br>产品市场前景(售前售后反馈);<br>产品推广策略(试点……)。 |

(续表)

| 分类  | 功能  | 质量   | 约束   |
|-----|---|--|--|
| 用户级 | 直接用户<br>用户需求  | 用户的需求往往不够详细,需要需求分析时进一步细化,必要时直接和用户交流:<br>UI<br>用户体验(反应速度)<br>可操作性<br>资源消耗<br>稳定性<br>安全性 | 国内、国外用户;<br>用户官方语言;<br>具体使用该需求部门。  |
|     | 竞争对手<br>竞争对手资料  | 比较详细和成型的需求,需要仔细体会,取其精华,去其糟粕:<br>UI<br>用户体验(反应速度)<br>可操作性<br>资源消耗<br>稳定性<br>安全性         | 分析竞争对手如何解决国际化问题,如何应对不同层次的用户群。  |
|     | 内部规划<br>内部自行规划的需求   | 内部的沟通:<br>UI<br>用户体验(反应速度)<br>可操作性<br>资源消耗<br>稳定性<br>安全性                               | 用户群特点;<br>用户水平;<br>多国语言。   |
|     | 工程<br>工程同事的需求   | 内部沟通:<br>UI<br>用户体验(反应速度)<br>可操作性<br>资源消耗<br>稳定性<br>安全性                                | 内部工程人员使用习惯;<br>考虑服务外包;<br>考虑运维托管。  |
| 开发级 | 开发部<br>总结细化上述各类功能性需求,形成需求说明书的功能性部分,case by case的描述,加上 UI。 | 代码可扩展性<br>代码可重用性<br>维护效率高  | 开发人员技术水平;<br>开发人员磨合程度;<br>开发人员分布情况;<br>开发人员业务知识;<br>管理:保密要求;<br>安装;<br>维护。 |
|     | 测试部<br>同上   | 同上   | 测试人员技术水平;<br>测试人员磨合程度;<br>测试人员分布情况;<br>测试人员业务知识;<br>管理:保密要求;<br>安装;<br>维护。 |

## 2. 关键性需求

关键性需求主要是利用质量关系属性矩阵和目标-场景决策图等分析工具,以上一节中 ADMENS 矩阵为输入罗列出能够决定软件架构的关键性质量和功能需求。

确定关键质量的五大原则:

- ①分类合适+必要补充;
- ②考虑多方涉众;
- ③检查性思维;
- ④识别矛盾+划定优先级;
- ⑤严格程度符合领域与规模特点。

确定关键功能的四条规则:

- ①核心功能;
- ②必做功能;
- ③高风险功能;
- ④独特功能。

### 3.1.3 任务同步训练

任务描述:

重庆市化工园区需要开发一个应急平台监测系统,以此为基础分析重庆市化工园区应急平台监测系统的需求。

实施步骤:

#### 1. 需求背景

党的十六届六中全会全面分析了当前的形势和任务,通过了关于构建社会主义和谐社会若干重大问题的决定,把加强应急管理作为社会管理的重要任务,纳入社会主义和谐社会建设的总体布局,第一次在中央全会的文件中明确提出了应急管理工作的目标和任务。2005年4月17日,《国务院关于实施国家突发公共事件总体应急预案的决定》(国发[2005]11号)是我国历史上首次从国家层面上制定的统筹应对突发公共事件的指导方针和纲领,是全国应急预案体系的总纲。《国家国民经济和社会发展第十一个五年规划纲要》提出要建立健全应急管理体系,加强指挥信息系统、应急物资保障、专业救灾抢险队伍、应急标准体系以及运输、现场通讯保障等重点领域和重点项目的建设,提高处置突发公共事件能力。《国务院关于印发国家中长期科学和技术发展规划纲要(2006—2020年)的通知》(国发[2005]44号)把公共安全列为我国科技发展的重点领域,将应急平台列为公共安全领域的首要优先主题。《“十一五”期间国家突发公共事件应急体系建设规划》要求,2007年底前,基本完成国务院、示范省级和部门应急指挥场所和主要基础支撑系统,实现互联互通;2008年底前,完成综合应用系统开发,初步完成应急信息资源整合,实现各地区、各有关部门互联互通和数据交换与信息共享等;2010年底前全面完成国务院、省级和部门以及地市指挥中心建设,并在有条件的县(市)建设适用的应急平台。

随着我国经济的发展,化工园区已经成为国家经济发展的重要组成部分,国家与市场的共同需求给化工园区带来了高速发展的契机。重庆市政府为了为发展重庆化工产业,

创建了重庆(长寿)化工园区(以下简称化工园区),随着重庆(长寿)化工园区进驻企业的增加,安全问题的形式日趋多样化、复杂化,新的安全问题不断涌现,灾害造成的危害程度正在逐步加大。化工园区是高能量聚集和有毒有害、易燃易爆物质密集的高危险性场所,是火灾、爆炸高发的行业。一旦发生灾害事故,极易造成人员伤亡和巨大的财产损失,通过连锁反应还可能给企业造成毁灭性破坏,同时由于重庆处于长江上游,污染物可能流入三峡大坝,威胁三峡大坝下流的众多省份,由此造成的环境破坏,甚至对整个国家和社会稳定产生负面影响。

近年来我国化工企业发生了多起安全事件,例如,2003年重庆市开县境内的西南油气田分公司川东北气矿发生天然气井喷事故,大量含有高浓度硫化氢的天然气喷出并扩散,导致243人因硫化氢中毒死亡,事发地方圆5公里内的4.1万多名群众被迫疏散,给国家和人民造成了巨大损失,社会影响严重。

2005年11月13日13时40分,中石油吉化双苯厂苯胺装置硝化单元发生爆炸事故,由于缺乏完善的应急预案,对泄漏物处理不当,部分硝基苯等污染物沿吉化公司东10号排水管线排入松花江,造成江水中苯系物超标,并向松花江下游扩散,导致哈尔滨市停水四天,污染物继续沿江扩散进入俄罗斯境内,引发国际外交事件,造成严重的国际影响,损害了中国作为一个负责任大国的国家形象。

为了更好地应对各类突发事件,国务院于2006年1月正式颁布的《国家突发公共事件总体应急预案》提出必须建立公共安全应急体系,指出企事业单位应根据有关法律法规制定相关应急预案。

为了不断提高安全管理水平,《总体应急预案》中指出“要积极开展公共安全领域的科学研究;加大公共安全监测、预测、预警、预防和应急处置技术研发的投入,不断改进技术装备,建立健全公共安全应急技术平台,提高我国公共安全科技水平;注意发挥企业在公共安全领域的研发作用”。

重庆(长寿)化工园区为了更好地应对园区内突发安全事件,也为了更好地服务园区内的化工企业,决定建设园区在线监测系统与应急平台。

## 2. 软件概述

为将重庆(长寿)化工园区(以下简称化工园区)在线监测项目建设成国内一流水平的安全环保管理设施,提高园区政府和园区内企业的工作效率,降低运作成本,提高政府与企业、企业和企业、区内与区外的联系紧密程度,最终提高化工园区整体的竞争优势,总体实现“实体可信、资源可管、行为可控、事件可查”的目标。到2008年底,建成统一指挥、功能齐全、反应灵敏、协同有序、运转高效、保障有力的长寿化工园区应急平台,并实现与上级应急平台的互联互通。

具体建设目标为:

建设化工园区应急平台,各企业环保在线监测、煤矿瓦斯监测监控系统、公共场所视频智能监控、园区监控中心等。

(1)根据国家应急平台体系的定位和分工,参照国家对地市级应急平台的要求设计长寿化工园区应急平台,实现长寿化工园区应急管理功能,并满足上级应急平台互联互通、调用各种数据以及综合应用系统在体系中的各种关联要求。



(2)平时满足长寿化工园区值守应急需要,根据上级应急平台统一的通信和计算机网络系统的设计保障与上级应急平台保持每天 24 小时不间断的联络畅通。

(3)设计并建设符合长寿化工园区管委会信息接报工作需求和流程的信息接报与发布系统,获取多种现场信息。

(4)建设长寿区突发公共事件的预测预警、模拟仿真和综合研判的功能,实现海量信息的智能检索和多部门、多领域数据的整合与叠加综合,为领导同志提供辅助决策支持。

(5)在长寿化工园区应急指挥厅实现应急指挥的指令下达、召开会议、接入现场图像、视频会商及提供上级应急平台有关数据和资料。

重庆市化工园区应急平台监测系统需要提供如下系统功能:

(1)基础支撑系统:基础支撑系统为长寿化工园区应急平台的正常运行提供硬件保障,主要包括应急通信、计算机网络系统、环境在线监测、图像监控系统、主机存储和安全保障等。

(2)综合应用系统:综合应用系统是应急平台的核心和灵魂,提供强大的应急业务管理和应急智能决策能力,主要包括综合统计查询系统、环境在线监测系统、图像监控系统、应急联动系统、模型分析系统、模拟演练系统、应急指挥调度系统、数字预案系统、地理信息系统、煤矿瓦斯监控系统等。

(3)数据库系统:长寿化工园区应急平台数据库系统采用集中式存储方式。数据库主要包括基础信息数据库、空间信息数据库、监测数据库、预案库、模型库、案例库、法规库、知识库等。

(4)应急指挥场所:应急指挥场所主要包括应急指挥厅、值班室、会商室等场所,这些场所又包括显示、灯光、会议、供配电、消防、防雷、防静电、装修等。

(5)移动应急平台:移动应急平台满足现场应急通信、现场会商、指挥调度、移动办公、现场图像采集等功能,主要通过卫星等通信手段与上级应急平台和其他应急平台以及有关方面进行联通。移动应急平台主要包括装载平台系统、通信传输系统、主机与存储系统、局域网、移动软件系统、移动数据库系统以及业务承载系统等。

### 3. 运行环境

运行环境信息表见表 3-5。

表 3-5

运行环境信息表

| 名称  | 硬件(CPU/RAM/HD 等)               | 操作系统及其版本                               | 其他软件环境                        |
|-----|--------------------------------|--|-------------------------------|
| 服务器 | 推荐使用主流服务器,最低配置 P4 2.8G,2G 内存   | Linux                                  | 数据库:Oracle;WEB 应用服务器:Weblogic |
| 客户机 | 推荐使用主流 PC 机,最低配置 CPU 双核,1G 内存。 | Windows 98/Windows 2000/<br>Windows XP | 浏览器:IE8.0                     |

### 4. 用户类 略。

### 5. 需求综合分析

以环境在线监测系统为例进行需求综合分析。

环境在线监测系统作为应急指挥系统的重要的子系统之一的综合环境信息系统,主要涉及的是环保方面,所以我们整理的也主要是重庆市环保局、长寿区环保局以及化工园区安环部等职能部门和环保监控相关的要求。根据近两周的调研工作,以及重庆市环保局对重庆市化工园区关于系统建设的若干问题的函的回答内容为基础,整理出重庆市化工园区综合环境信息系统建设的需求。

为了达到我国的“十一五”中相关环境保护的目标,总体控制环境污染,加快改善园区环境质量,初步遏制生态破坏的趋势,因此需要通过化工园区的大气、水源、噪声进行实时监控,针对污染超标的企业要及时进行预警,以最大化地控制环境污染环境事件的发生,保证化工园区的环境质量。整个工程为统一设计,分步实施。第I期实施化工园区已入住的38家企业,并建设整个环保监控系统的框架平台,包括空气大气监测、污染源水质监测、企业烟气排放监测、应急污水监测以及部分企业的危险源监测信号的接入等。第II期实施后期入住的其他企业的环保监测子站,并根据第I期实施中的经验,对整个环保监控系统的框架平台进行补充和完善。

综合环境信息系统的监控平台通过自动监控设备实时采集的大气、污染源、企业排放烟气、噪声相关的监测数据,经数据采集传输设备写入化工园区环境在线监测预警系统数据库。如果发生企业污染物排放浓度超标,主要生产设备、环保设备不运行的情况,预警系统则会自动地在园区配套的GIS系统上显示出发生事件的位置,智能系统根据前端采集设备采集回来的数据与应急响应和决策系统内的数据进行比对,智能判断事故等级,通过声光的形式进行报警,并以短信的形式通知到相关的责任人且告知相关处理方法;此外环境监测人员还可以通过对采集数据进行统计与加工,从而对化工园区的环境质量预测、环境信息的发布、环境的管理和执法提供可靠的依据。

(1)环保监测系统必须建立以重庆(长寿)化工园区基础地形图为支撑的空间数据库,包括地形图、在线监测点位、园区企业点位、排污口位置等;建立满足长寿化工园区实际情况的海量监测数据历史数据库,包括噪声、污染源水质监测的DO、氰化物、BOD5、PH、COD、TP、污染源烟气监测的 $\text{SO}_2$ 、 $\text{NO}_x$ 、 $\text{O}_2$ 、PM10、温度、压力、流量等,且所参照的标准必须统一。

(2)要求能够对企业污染源污染物排放情况实施24小时监控;同时,企业端监测子站也必须能够响应监控中心对数据主动采集的要求,包括实时数据采集和历史数据采集的要求。

(3)实时监测企业的污染物排放情况,监控主要生产设备和环保设施运行情况,现场数据采集器至少能保存1年的采集数据。对于重点监控企业,还需要对企业的排污口排放状况和主要污染治理设施的运行状况进行视频监控,例如企业的曝气池情况等。

(4)数据采集与存储功能:采集企业环保监测仪器的测量数据和企业污染治理设施实时的运转数据,在监测子站的数据采集器上进行存储。并利用有线或者无线等的通信手段将监测数据传送到监控中心。在监控中心的软件上进行基础运算,并存储原始数据。监控中心应该保存各监控点历史相关数据,供随时调用。

(5)数据自动传输,通信协议必须满足国家环保总局要求的HJ/T212-2005通信标准。现场的数据可根据管理中心的需要提供以下各种上传数据方式:

①在监控中心任何需要现场数据时,可以从监控中心发起数据传输,要求采集各个监测子站的监控数据。

②实时数据传输:监测子站在采集监测数据后自动实时地将监测数据上传监控中心。当通信链路发生故障,传输失败的情况下,记录未传输数据。待通信链路恢复后自动传送上次传送失败的数据。

(6)报警功能:要求根据实时监测数据,在满足报警情况时,及时进行报警。如果出现企业污染物排放浓度超标,或主要生产设备运行、环保设备不运行,直接在系统中产生声音或在系统地图上闪烁进行图文报警;报警信息能在中心控制室显示,并能随时跟踪查询具体信息;

(7)数据、报表自动生成:可以根据需求开发各种报表,提供定时(不定时)打印功能。

(8)数据分析、统计:中心控制室以监测数据为基础进行的数据加工和分析,如污染物指标查询、环境信息定时发布、环境评价、环境质量预测等,可以结合 GIS 在地图上直接进行专题发布,如:在地图上展示某季度园区内企业排污超标情况,绿色表示正常,红色表示超标;在地图上展示企业监测设备运转情况,绿色表示正常,红色表示出现问题等,一目了然,为环境管理和执法提供可靠、直观的依据。

(9)数据交换功能:要求预留接口,满足系统后期与重庆市环保局、长寿区环保局以及其他的环保职能部门进行数据交换。数据交换必须满足国家环保总局要求的 HJ/T352—2007 数据交换技术规范。其中向重庆市环保局和长寿区环保局传送的监测数据包括各个企业的各种污染因子的监测数据,各个企业的污染治理设施的运行状况数据和视频监控数据。由于重庆市环保局和长寿区环保局目前的监测系统正在升级中,所以视频监控数据在一期项目中暂时不考虑上传。但是应该提供通过外部网络和网页播放的方式提供远程监视视频图像数据的技术手段。

(10)在市/区环保局要求在监测子站现场直接采集监控数据,并通过自己的环保专用网络传输数据时,监测子站的数据采集器需要提供现场的数据通信接口,以备市/区环保局在现场的直接的数据采集。监测子站的数据通信接口可以采用 RS232/485 或者 RJ45 的网络接口,通信协议采用 Modbus 或者 Modbus/TCP。

(11)可扩展功能:要求预留接口,满足系统后期与其他渠道来源的数据资源进行交换整合,满足系统后期业务应用扩展所需的数据接口支撑,进而为建立数据仓库、进行环境综合评价等提供基础支撑。

通过以上用户需求分析我们可以得到如下 ADMEMS 需求矩阵表(示例),见表 3-6。

表 3-6 ADMEMS 需求矩阵表

| 分类  |      | 功能   | 质量         | 约束  |
|-----|------|--|------------|---|
| 业务级 | 产品规划 | 1. 污染源快速发现<br>2. 污染源精确定位<br>3. 污染源趋势分析<br>4. 污染源实时监控 | 1. 多/快/好/省 | 1. 接入各种采集设备<br>2. 兼容各采集设备的不同版本<br>3. 尽快推向市场,后续不断优化,分批实现 |

(续表)

| 分类  | 功能   | 质量   | 约束  |
|-----|------|--|---|
| 用户级 | 直接用户 | 1. 数据压缩比能达到 10:1<br>2. 支持导入或者导出 10 万条记录<br>3. 采集数据分析在 5 秒内需要分析完成<br>4. 实时监控最小粒度 10 秒,时延不超过 15 分钟<br>5. 门限可自定义<br>6. 用户可自定义报表格式,可以查询 1 年前的数据,查询等待时间小于 10 秒<br>7. 查询一周以内的记录,查询等待时间小于 10 秒<br>8. 用户可自定义污染源处理建议<br>9. 可以查询 1 年内的数据进行分析,分析结果展现时延不能超过 30 秒<br>10. 可 24 小时不间断监控 | 1. 广域网不超过 10M<br>2. 主要的用户是网络维护人员<br>3. 数据导入导出的文件格式无法确定<br>4. 数据能够接入专用系统<br>5. 当地地图<br>6. 需要留有接口以满足后续的数据交换 |
|     | 内部规划 | 1. 权限管理<br>2. 日志管理<br>3. 多采集设备接入<br>4. 多层灵活组网  | 1. 操作日志至少保存 3 个月<br>2. 可以接入所有采集设备的不同版本<br>3. 接入采集设备数不少于 100 个<br>4. 客户端启动不超过 15 秒<br>5. 客户端连接不少于 50 个     |
| 开发级 | 开发部  | 1. 接口采用 RS232/485 或者 RJ45 的网络接口,通信协议采用 Modbus 或 Modbus/TCP<br>2. 中英文版本合一   | 1. 同一套接口接入不同采集设备<br>2. 抽象出公共框架和公共流程<br>3. 尽量复用已有代码<br>4. 减少模块间的耦合   |
|     | 测试部  |  | 1. 对 Java 熟悉的人多,对 C/C++ 熟悉的人少<br>2. 对业务知识了解少<br>3. 外包部分功能<br>4. 与采集设备解耦合                                  |

### 3.1.4 自主学习任务

#### 1. 练习题

(1) 补充任务同步训练中用户类内容。

(2) 利用质量关系属性矩阵和目标-场景决策图等分析工具分析出此软件的关键功能需求和关键属性需求。

#### 2. 拓展学习任务

考虑如何编写需求,尽量保证需求的完整性、正确性、必要性、可行性、划分优先级、无二义性、可验证性。

## 任务 2 需求编写方法

### 3.2.1 任务引导训练

#### 引导任务：

编写一个查询交易需求,用户设置查询条件,软件显示满足条件的存取款交易数据,并在需要的时候用户可以将查询结果导出为 Excel 文件。

#### 训练任务分析：

用户查询交易数据,包括输入合法查询条件能正确查询到交易数据的正常过程,也包括输入不合法查询条件导致查询出错的异常过程,甚至还包括快捷键定义/排序等易用性特殊需求。

#### 实施步骤：

需求描述:用户设置查询条件,软件显示满足条件的存取款交易数据,并在需要的时候用户可以将查询结果导出为 Excel 文件。

Actor:用户

优先级:高

使用频度:经常

前置条件:

V01 用户已登录

后置条件:

B01 用户查询到交易数据

正常过程:N1

N01 用户选择“交易查询”功能,用例开始。

N02 软件显示查询界面,供用户设置查询条件。可供设置的信息包括:卡号、币种、开始查询日期、结束查询日期。其中,日期格式取 YYYYMMDD。

N03 软件查询用户名下的所有卡号,作为卡号的取值范围。

N04 用户选择:卡号(单选,不能为空)、币种(单选,不能为空)、开始查询日期、结束查询日期。

N05 软件确认用户选择了卡号、币种、开始查询日期、结束查询日期,且开始查询日期与结束查询日期是同一年度。

N06 软件查询满足条件的交易数据。

N07 软件确认存在满足条件的交易数据,并以表格形式显示出来。

N07-1 表头为:卡号、日期、支出、存入、余额、说明。

N07-2 数据以“日期”从小到大排序。

N07-3 软件显示“支出和”、“存入和”。

N08 等待用户退出。

N09 用户退出,用例结束。

可选过程:O1 将查询结果保存为 Excel 格式的文件(在 N1. N08 处)

O01 用户选择“导出到 Excel”。

O02 软件要求用户选择存放路径,并输入文件名。

O03 软件确认文件名及路径是有效的。

O04 软件将查询结果按照当前选择的排序方式,保存到 Excel 文件中。

O04-1 Excel 文件中,也应该有表头(同屏幕显示)。

O04-2 Excel 文件中,需要包括查询条件。

O04-3 Excel 文件中,需要对存入、支出进行合计。

O04-4 支持 Excel 2000/2002/2003/2007 版本。

O05 软件确认保存 Excel 文件成功,并提示“查询结果已成功保存为 xx”,其中,xx 为路径及文件名。

O06 回到 N1. N08。

可选过程:O2 改变排序方式(在 N1. N08 处)

O01 用户按下表头中的存入(或支出或日期)。

O02 软件按照存入(或支出或日期),对数据重新排序并显示。如果与当前排序字段相同,则反序显示;否则,则按从小到大显示。

O03 回到 N1. N08。

异常过程:E1 无效的查询条件(在 N1. N05 处)

E01 软件提示“未选择卡号、币种、开始查询日期、结束查询日期,或开始结束日期大于结束查询日期!请重新设置查询条件。”,并等待用户确认。

E02 回到 N1. N04。

异常过程:E2 没有满足条件的交易数据(在 N1. N07 处)

E01 软件提示“没有满足条件的交易数据!请重新输入。”,并等待用户确认。

E02 回到 N1. N04。

异常过程:E3 未安装 Excel(在 O1. O04 处)

E01 软件提示“没有安装 Excel,无法生成 Excel 文件!”,并等待用户确认。

E02 回到 N1. N08。

异常过程:E4 未成功保存文件(在 O1. O04 处)

E01 软件提示“未成功保存文件!可能是路径或文件名错误。”,并等待用户确认。

返回 N1. N08。

特殊需求:

S01 查询过程中,用户可随时按下 Esc 键,放弃查询。

S02 查询界面上,应该有提示:按下表头中的“日期”、“存入”、“支出”,可以按其排序。

S03 必须对每个具有活动焦点的控件定义快捷键。

## 3.2.2 支撑知识

### 3.2.2.1 功能需求

功能需求表示了产品的行为或期望产品能够完成的工作。这些需求通常是面向动作的,需要重点描述产品与外界的交互。

凡是功能需求(包括外部接口完成的功能),都是可以通过“激励、响应”模型(对应于 use case 描述方式)或“输入、处理、输出”模型(对应于 IPO 描述方式)来描述的。

**注意:**理解 IPO 中的 P 时,应与 use case 中的过程对照起来:P 就是正常过程、可选过程、异常过程的综合。否则会被误解为内部处理。

#### 1. USE CASE 需求描述方式

需求名称应当使用确定性的语句,建议采用动宾结构(如“建立任务”,而不是“小任务建立”),应尽可能简短,使得标题能够写在一行。

**需求描述:**对需求的正常过程、可选过程进行概述,以方便那些不需要知道需求细节的读者快速了解功能需求。

**Actor:**“参与者”、“角色”。Actor 是指同系统交互的所有事物,包括系统的用户、与系统拥有物理或逻辑连接的其他系统、外部事件(如满足特定日期或时间、满足特殊环境条件如涨潮/落潮、峰谷/峰顶等)。识别 Actor 的目的是为了更好地保证功能需求的完整性及与确定的对象(即 Actor)确认需求。

描述 Actor 时,应注意:

- ① Actor 一定存在于系统的外部,不可能是系统本身,但可能是系统的一个实例。
- ② Actor 一定要在后续的过程描述中出现,否则,要么是此处多写了,要么是过程描述少写了。实践中,后者占绝大部分。活动描述中经常出现“系统”,而没有任何 Actor。没有 Actor,怎么能体现“交互”呢?

优先级:5、4、3、2、1(5 最高、1 最低)。

**使用频度:**根据项目的实际情况取值,如“经常”、“偶尔”、“不关心”或“高”、“中”、“低”、“不关心”等。识别“使用频度”的目的是为后续设计提供约束:如果使用频度高,则在设计时,不仅应注重功能的完成,还应特别关注其性能及易用性。

**前置条件:**只有具备该条件(即前置条件应为真)才可执行该功能,即必要条件。

描述前置条件时,应注意以下问题:

- ① 一般由人、其他系统或其他需求负责设置前置条件。
- ② 不能将触发条件当成前置条件。如“收到 XX 消息”,正确的前置条件可能是“用户已登录并具有 XX 权限”。
- ③ 不能将应在异常过程中捕获的条件作为前置条件。如“通信正常”,由于前置条件是必须为真的,因此在实现时无需验证,而“通信正常”则是谁也不能保证的,在实现时需要采用某种机制保证。

后置条件:描述正常过程、可选过程执行后系统所处的状态。又叫“请求结果”、“最小保证及成功保证”。

后置条件是用例必须承担的义务,由正常过程、可选过程共享。一般应考虑三种情况:数据生成与删除;数据修改;数据之间关联的建立与删除。

通过前置、后置条件可以将功能需求有机地联系起来:某个功能需求的前置条件是由另一个功能需求通过其后置条件设置的。事实上,我们在开发用例时,经常可以从某些基本的、常见的功能需求的前置条件中发现更多的用例。

描述后置条件时,应注意以下问题:

- ①后置条件应表述成“条件”的形式。如“文件被保存”、“系统收到XX消息”。
- ②不要画蛇添足。如,“XX与YY间建立了ZZ链路”而不要写成“建立了正确的ZZ链路”,因为这样一来,就需要定义什么叫“正确”。
- ③后置条件一般不为空,因为系统总得处于某个状态。但如果用例执行前后,系统状态没有变化,则可以没有后置条件。

正常过程:当一切运转正常时就得到了正常过程。没有故障,没有错误。每一个用例都必须有一个正常过程。

识别正常过程的目的是:用尽可能少的语句让读者毫无障碍地了解“需求是做什么的”。

描述正常过程时,应注意以下问题:

- ①正常过程步骤的第一步,必须是描述用例由于“触发条件”被触发而执行,如“XX(指Actor)收到XX消息,用例开始”。
- ②正常过程的最后一步,通常为“用例结束”。
- ③存在几个不同的触发条件,就存在几个正常过程。
- ④一般将“管理/维护XX”作为一条需求进行描述:C(reate)作为正常过程、R(etrieve)/U(pdate)/D(elete)作为可选过程。这样冗余最少。但是如果分成几个用例描述更加方便且易于阅读,则也可分开描述。
- ⑤过程描述中应出现“Actor”中所列的所有的Actor,否则,要么是本处写错了,要么是“Actor”处多写了。

⑥过程描述中,通常应显式指明交互的发起者、交互的内容及交互的对象。

⑦不要将一个以上的活动放在一个步骤中描述。如“系统收到XX消息,并检查其是否合法。”应拆成两个步骤:

- 系统收到XX消息。
- 系统确认XX消息是合法的。

⑧不应该使用“如果…那么…否则”一类的句式,否则就没有必要区分过程类型了。如:用户输入数据后,一般应检查其合法性。关于合法性,正确的描述为:“N03系统确认数据是合法的”,这样N04就很自然地沿着正确的路径往下走了。

当然,如果只有一两个分支,放在一起描述反而更易于阅读,也是可以在正常过程中



出现分支的。取舍的基本原则是：易于阅读，易于描述，易于维护。

可选过程：可选过程也可促进功能的成功完成，但它们代表了功能的细节或用于完成功能的途径的变化部分。一个用例有 0 到多个可选过程。

描述可选过程步骤时，应注意以下问题：

①可选过程的名称应该描述对应的分支条件。如：A1 将查询结果保存为 Excel 格式的文件（在 N1、N08 处）。

②可选过程结束后，一般应回到正常过程的下一步继续执行，但这也不是绝对的，可视具体情况回到正常过程的合适的地方继续执行。应显式说明这一点。如：O08 回到 N1、N08 处继续执行。

③为了避免阅读困难，在可选过程中可以直接处理条件判断，而无需再编写独立的可选过程。

④可选过程中的异常，应放在用例的异常过程中统一描述。

异常过程：描述引起功能不能顺利完成的情况。正常过程、可选过程都会引起异常。一个用例有 0 到多个异常过程。

描述异常过程步骤时，应注意以下问题：

①异常过程的名称，应该描述对应的分支条件。如：E1 无效的查询条件（在 N1、N05 处）

②异常过程一般会结束整个用例的执行，但这也不是绝对的。对于输入错误等异常，一般应回到正常过程中重新输入，以提高易用性。

③为了避免阅读困难，在异常过程中可以直接处理可选和异常，而无需再编写独立的可选或异常过程。

特殊需求：补充描述某些特别的需求，如与本功能密切相关的性能、质量属性、业务规则等。如果本需求是分阶段实现的，则在上面的“需求描述”中描述当前需要交付的目标，而在此处描述后续需要交付的目标。

说明：这里列出的仅是最小要求，使用者可以根据实际需要进行扩充。另外，尽管没有强调，use case 图与包括正常、可选、异常过程的流程图也是十分鼓励使用的。

对一些复杂的软件，为了描述的方便，往往会引入抽象状态（机）的概念，这样就可以用一个抽象状态来代替一大堆的文字说明（最常见的是用“状态”代替 use case 中的“前置条件”和“后置条件”）。因此，可以对这些抽象状态进行描述。描述的方法建议采用 UML 中的 State Chart 图或其他等价的状态图。因为只是为了帮助说明，因此抽象状态可以不考虑具体实现，使用嵌套状态、并发状态、同步等高级描述方式。

一般来说，为了描述一个抽象状态机，需要提供下面两部分的信息：

一张完整的状态图。需要包含所有的状态（包括初始状态和终止状态）、所有的状态转换（包括内部转换）。但状态转换只需要说明事件名（event）和监护条件（condition）。

对每个状态进行简单的文字说明（建议使用表格），主要说明该状态的含义。

需要注意的是，在对抽象状态的描述中（包括状态图）中出现的具体的名词（如状态名、

事件名(接口)等),必须保持和各功能需求中的一致。

【示例1】 WarmUp 制热构件总的抽象状态图如图 3-3 所示。图 3-3 中各状态的说明见表 3-7。

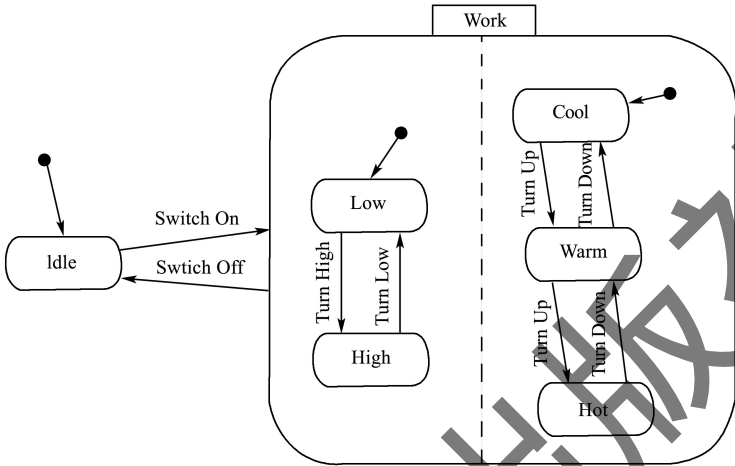


图 3-3 抽象状态图

表 3-7

图 3-3 中各状态的说明

| 状态名称 | 所属状态 | 状态说明          |
|------|------|---------------|
| Idle | —    | WarmUp 构件空闲中  |
| Work | —    | WarmUp 构件工作中  |
| Low  | Work | WarmUp 构件处于低档 |
| High | Work | WarmUp 构件处于高档 |
| Cool | Work | WarmUp 构件在制冷  |
| Warm | Work | WarmUp 构件在制暖  |
| Hot  | Work | WarmUp 构件在制热  |

【示例2】 恶意信息监控用例图如图 3-4 所示。

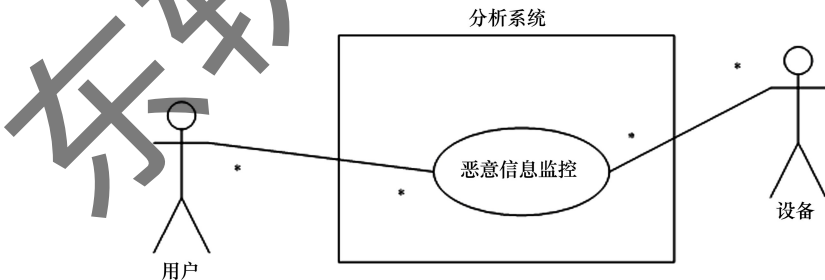


图 3-4 恶意信息监控用例图

需求描述:设备上发生业务,设备把相关业务信息上报给分析系统,分析系统对采集到的信息根据策略进行分析过滤,把恶意信息提取出来展现给用户,另外恶意信息还需要持久化保存,以便用户后续查看。

Actor:设备,用户

优先级:5

使用频度:经常

前置条件:

V01 设备已接入分析系统;

后置条件:

R01 恶意信息被监控到并展现给用户。

正常过程:N1

N01 设备上发生业务,用例开始。

N02 设备把相关业务数据上报给分析系统。

N03 分析系统根据策略对业务数据进行分析过滤。

N04 分析系统把恶意信息提取出来展现给用户并持久化保存。

N05 用户实时查看到恶意信息。

N06 用例结束。

可选过程:O1 设备上发生分析系统不关心的业务(在 N1、N02 处)

O01 设备上发生分析系统不关心的业务,用例开始;

O02 设备不把业务数据上报到分析系统。

O03 用例结束。

可选过程:O2 分析系统未配置过滤策略(在 N1、N03 处)

O01 分析系统未配置过滤策略,用例开始。

O02 分析系统直接把业务数据丢弃。

O03 用例结束。

可选过程:O3 上报的业务数据不包括恶意信息(在 N1、N04 处)

O01 上报的业务数据不包括恶意信息,用例开始。

O02 分析系统把此业务数据丢弃。

O03 用例结束。

可选过程:O4 用户设置了过滤条件不监视该用户(在 N1、N05 处)

O01 用户设置了过滤条件不监视该用户,用例开始。

O02 用户在实时监控界面上看不到恶意信息。

O03 用例结束。

异常过程:E1 设备与分析系统链路断(在 N1、N02 处)

E01 设备与分析系统链路断,用例开始。

E02 分析系统接收不到设备上报的业务数据。

E03 用例结束。

异常过程:E2 分析系统过滤策略异常(在 N1、N03 处)

E01 分析系统过滤策略异常,用例开始。

E02 分析系统直接丢弃该业务数据并记录日志备查。

E03 用例结束。

异常过程:E3 恶意信息持久化失败(在 N1、N04 处)

E01 恶意信息持久化失败,用例开始。

E02 分析系统丢弃该业务数据并记录日志备查。

E03 已经打开的实时监控界面上用户还是能查看到恶意信息。

E04 用例结束。

特殊需求:

S01 设备与分析系统链路断的情况下要求设备能够缓存 24 小时或者 50M 的业务数据,在链路恢复后能够补报上来。

S02 恶意信息要求能够持久化保存至少 3 个月。

## 2. IPO 需求描述方式

需求描述:对需求的输入、处理、输出进行概述,以方便那些不需要知道需求细节的读者。

优先级:见 use case 中对于“优先级”的描述。

使用频度:见 use case 中对于“使用频度”的描述。

触发条件:描述在何种情况下开始执行本功能。可能的触发条件包括:用户执行、其他系统调用、其他需求调用等。

输入:描述需求涉及的全部输入数据,包括:输入源、数量、度量单位、时间关系、有效输入的范围;用户动作,其中包括操作员活动的描述、控制台或操作员的位置(如在打印表格时,要求操作员调整打印纸位置的需求)等。

输出:描述需求完成后的输出数据,包括:输出的目的地,有效输出的范围,非法值的处理结果,错误信息等。

处理:为获得期望的输出而进行的主要处理步骤,对应于 use case 中的正常过程、可选过程、异常过程。

特殊需求:见 use case 中对于“特殊需求”的描述。

### 【示例 1】登录服务器

需求描述:提供一个公共界面,完成登录服务器的功能。

优先级:5

使用频度:经常

触发条件:

T0010 启动客户端或登录到另外一台服务器上。

输入:

I0010 用户名(不超过 20byte 长度的字符串,允许字符为:[a-z],[A-Z],[0-9]、一)。〈最好定义在“外部接口说明书”中〉

I0020 密码(不超过 10byte 长度的字符串),允许字符的 ASCII 为 33~26。〈最好定义在“外部接口说明书”中〉

输出:

O0010 登录结果(成功、失败、超时)。

处理:

P0010 用户在客户端输入用户名及密码。

P0020 客户端发送登录消息到服务器端。消息格式为……(最好定义在“外部接口说明书”中)

P0030 客户端等待服务器返回验证结果。

P0040 客户端根据验证结果显示登录是否成功,或超时。(超时值可以配置)  
特殊需求:无。

### 3.2.2 性能需求

#### 1. 性能需求描述方式

性能需求描述了软件实现预期功能的程度,如软件运行应有多快、精度应有多高等。本节与上一节功能需求中“特殊需求”部分含有的“性能”各有侧重点,前者描述的是“与某功能紧密相关的性能需求”。而在本节,主要从整体上针对一组或全部功能描述其性能需求。

性能需求包括但不限于下列方面:

(1)精度要求:数据传送的误码率要求;数据的精度要求;数学计算的精度要求。

(2)时间特性:平均响应时间;在峰值负载期内,与所规定的响应时间的允许偏离范围、更新处理时间、数据的转换和传送时间等的要求;时序的安排,错误时序出现时软件应在什么时间内作出处理等。

(3)容量需求:最大的并发用户数或事务数。

(4)数据业务流量、包交换速率等。

需求描述:性能的详细描述。应包括以下要素:

(1)性能对应的条件。

(2)具体的指标。必要时,需要给出指标的定义。

(3)可能的容差。

优先级:见 use case 中对于“优先级”的描述。

特殊需求:描述其他需要描述的内容。

特殊需求至少有两种应用场合:

场合 1:需求是分阶段实现的,则在上面的“需求描述”中描述当前需要交付的目标,而在此处描述后续需要交付的目标。

场合 2:需求是由更上层的设计“勉强分解”得到,且很难对需求进行单独测试验证,则可以在描述该性能具体值的同时,在“特殊需求”中描述由本层及其兄弟姐妹组成的更上层黑盒必须具备的总的性能,以给后续设计及测试提供更多的信息。

在描述性能需求时,应注意:

(1)明确列出性能所对应的条件。在以往的实践中,多数性能需求只是列出了具体指标,而未指明该指标是在什么条件下取得的,如“所有的查询应在 20 秒之内(含 20 秒)完成”就没有指明条件。

(2)严格区分系统性能需求与软件性能需求。由于被描述对象的层次不同,两者是不

同的,无法直接复用。

另外,应注意避免以往的实践中经常出现的问题:

(1)不描述或很少描述性能需求;

(2)未描述具体指标;

(3)未描述需求对应的条件。

(4)只描述一个单一值,而没有注明容差。一般情况下,应使用数值范围来描述某种性能。

## 2. 性能需求示例

(1)客户端首次启动效率

需求描述:客户端首次启动时延需要控制在10秒以内。

优先级:5

(2)客户端二次启动效率

需求描述:客户端二次启动时延需要控制在3秒以内。

优先级:5

(3)精度需求

需求描述:传感器的数据应具有14位的精度,在两年之内可扩展到18位。

优先级:5

(4)设备与恶意信息分析系统间性能要求

需求描述:设备与恶意信息分析系统间消息流量不能超过100条/秒,数据流量不能超过1M/秒;恶意信息分析过滤时长不能超过5秒;从业务发生到恶意信息展现在实时监控界面上时延不能超过10秒;恶意信息保存时长不能小于3个月。

优先级:5

### 3.2.2.3 质量属性需求

质量属性需求规定了软件被执行时的好坏程度,如能连续工作多长时间而不出错,出错后能在多长时间内修复,安全性有多高,软件的易用性如何等。

实践中,有些软件可以完成规定的功能,但软件不稳定(经常无任何响应,出现异常而退出程序甚至死机),非常难以使用,数据经常丢失,因此也就无法有效完成用户的工作。比起功能需求,是否满足质量属性需求往往是软件成功的最终决定因素。同时,质量属性需求一般是隐含的,很少被正式讨论,即使被讨论,往往也是定性的,如“界面要友好,容易使用”,“软件要稳定”,“程序要健壮,容错性好”等,这样的描述很难对设计方案有直接的约束,而且也无法测试、确认。

实际上,业界已有一些标准来规定质量属性需求的定量描述。为了更好地(即定量、可验证)描述质量属性需求,以下列出了IEEE的相关标准IEEE 1061-1998作为参考,见表3-8、表3-9。

表 3-8

| 质量属性           | 属性描述  | 子属性                  | 子属性描述   | 谁最感兴趣 |
|----------------|---|----------------------|---|-------|
| 可移植性           | 与软件可从某一环境转移到另一环境的能力有关的一组属性                                      | 硬件独立性                | 软件不依赖于特定硬件环境的程度   | 开发者   |
|                |   | 软件独立性                | 软件不依赖于特定软件环境的程度   | 开发者   |
|                |   | 易安装性                 | 将软件安装到某个新环境所需努力的程度  | 开发者   |
|                |   | 可重用性                 | 软件可重新应用到一个新应用(与原应用不同)的程度  | 开发者   |
| 可靠性            | 与在规定的的一段时间和条件下,软件维持其性能水平的能力有关的一组属性                              | 无缺陷性                 | 软件不包含未被检测出的错误的程度  | 用户    |
|                |   | 容错性/健壮性              | 软件在出现了错误的情况下能继续工作的程度,其中包括软件降级操作和恢复功能的程度(所谓恢复,是指软件发生错误后重新运行时,恢复到没有发生错误前的状态的能力) | 用户    |
|                |   | 可用性(availability)    | 一段时间内软件真正可用的程度  | 用户    |
| 功能性            | 与满足明确或隐含的用户需求及其指定的性质有关的一组属性(之所以叫“功能性”,可能是命名者认为这类需求最适合描述为“功能需求”) | 完备性                  | 软件具有的能满足用户需要的必需的和足够的功能的程度   | 用户    |
|                |   | 正确性                  | 所有软件功能与用户的需要相符合的程度(似乎无需特别要求。总需如此)   | 用户    |
|                |   | 安全性(safety&security) | 软件能够检测或防止信息泄漏、信息丢失、非法使用和系统资源破坏的程度。另外,也可描述“保护用户人身安全的”“安全性(safety)”             | 用户    |
|                |   | 兼容性                  | 在不改变环境及无需事先准备某些条件的情况下,软件可直接安装(使用)的程度(侧重于静态,如接口文件、数据格式的兼容性)                    | 用户    |
|                |   | 互操作性                 | 软件能否与其他系统方便连接并能操作的程度(侧重于动态交换数据,如 Word 就提供了很好的互操作性,其他软件可以很方便地控制 Word 进行文档编辑工作) | 用户    |
| 易用性(usability) | 与使用(含使用前准备和结果评估)所需作的努力和用户对使用所作的个人评价有关的一组属性                      | 易学性                  | 学习软件所需的努力程度   | 用户    |
|                |   | 易操作性                 | 软件操作与目的、环境和用户的生理特性相匹配的程度  | 用户    |
|                |   | 沟通性                  | 软件符合用户生理特性的程度   | 用户    |

表 3-9

| 质量子属性 | 直接度量      | 描述                   |
|-------|-----------|----------------------|
| 硬件独立性 | 硬件依赖性     | 计算硬件的依赖性             |
| 软件独立性 | 软件依赖性     | 计算软件的依赖性             |
| 易安装性  | 安装时间      | 测试安装时间               |
| 可重用性  | 能够用于其他软件中 | 计算能够用于或已经用于其他软件的软件数量 |
| 无缺陷性  | 测试覆盖      | 测量测试覆盖               |
|       | 审查覆盖      | 计算已做过代码审查的模块         |

(续表)

| 质量子属性                     | 直接度量           | 描述                       |
|---------------------------|----------------|--------------------------|
| 容错性/健壮性                   | 数据完整性          | 统计用户数据被破坏的情况             |
|                           | 数据恢复           | 测量恢复被破坏的数据的能力            |
| 可用性<br>(availability)     | 软件可用的时间百分比     | 软件可用时间除以总的软件使用时间(以百分比表示) |
| 完备性                       | 测试覆盖           | 调用对的测量或分支覆盖的测量           |
| 正确性                       | 缺陷密度           | 计算在软件发布前每一个版本中发现的缺陷      |
| 安全性<br>(safety&-security) | 数据完整性          | 统计用户数据被破坏的情况             |
|                           | 用户安全性          | 没有被阻止使用软件的非法用户数量         |
| 兼容性                       | 环境变化           | 软件安装后必须修改的环境变量数量         |
| 互操作性                      | 混合应用环境下软件的可操作性 | 软件能正确运行的混合应用环境的数量        |
| 易学性                       | 学习所用的时间        | 新用户学会如何操作软件提供的基本功能所花费的时间 |
| 易操作性                      | 人的因素           | 新用户关于人类工程学、人的因素等消极的评论数量  |
| 沟通性                       | 人的因素           | 新用户关于人类工程学、人的因素等消极的评论数量  |

需要强调的是,对于某一个具体的软件来讲,没有必要也不可能完全考虑以上所有的质量属性。事实上,与质量属性有关的标准如 IEEE 1061、ISO 9126、IEEE 830 对质量属性究竟应包括哪些内容都没有统一,更遑论“全部”考虑了。在需求分析时,应仔细分析产品的特性,确定重要的即用户关注的某几个质量属性。

下面将以子标题的形式对表 3-8 及表 3-9 中列出的一些质量属性作进一步的说明。对于这里没有描述而实际项目又需要的质量属性,可以根据表 3-8、表 3-9 及相关资料以子标题的形式进行扩充。同样,在描述质量属性需求时也应注意其可验证性。

质量属性需求的描述格式为:

需求描述:质量属性的详细描述

优先级:见 use case 中对于“优先级”的描述。

特殊需求:见性能需求中对于“特殊需求”的描述。

在描述质量属性需求时,应注意避免以往的实践中经常出现的问题:

- ① 不描述或很少描述质量属性需求;
- ② 很多“需求”体现了设计要求或思路,而没有给出具体的、可验证的要求;
- ③ 即使描述了质量属性,多数也是定性的;
- ④ 未明确列出可能的条件。

⑤ 只描述一个单一值,而没有注明容差。一般情况下,应使用数值范围来描述某种取值为数值的质量属性。

## 1. 可靠性需求

所谓可靠性,是指软件在规定的条件下及规定的时间内,完成规定功能的能力。应该考虑以下因素:

(1) 无缺陷性。应确定测试覆盖、审查覆盖的指标。

(2) 健壮性。一般指的是当软件或其组成部分遇到非法数据、异常操作、相关软件或硬件出现缺陷、超出规定性能要求等情况时,能继续正确运行功能的程度。



(3)平均故障间隔时间 MTBF。

(4)故障平均维修时间 MTTR。

(5)可用性:与 MTBF 相关。可用性指一段时间内软件能够提供规定功能的能力,一般用  $(MTBF) / (MTBF + MTTR)$  表示。因此,定义 MTBF、MTTR 时,必须满足软件的可用性。

(6)分析关键输入变量的初始信号源、关键输入/输出变量的信号传输通道和通讯接口,判断可能的错误信号出现时软件应作出的响应。一般应和功能需求同时描述。

(7)对可靠性要求高的软件,一般应列出可能的不期望事件,分析导致这些不期望事件的可能原因,可能造成的影响和危害,确定软件的功能失效等级和关键的输入/输出变量。一般应和功能需求同时描述。

#### 【示例 1】 健壮性需求

需求描述:所有的参数都要指定一个缺省值,当输入数据丢失或无效时,自动使用缺省值。

优先级:5

#### 【示例 2】 平均故障间隔时间 MTBF

需求描述:平均故障间隔时间 MTBF 应不小于 400 小时。

优先级:5。

## 2. 安全性需求

安全性主要包括数据完整性和用户安全性,指软件能够检测或防止信息泄漏、信息丢失、非法使用和系统资源破坏的程度。可以考虑以下情况:

(1)为防止非法用户进入系统,提出软件用户管理及权限分配要求,如身份验证、用户特权级别、访问约束等;

(2)为防止敏感数据被非法使用,提出对输入、输出数据的加密要求;

(3)数据安全,如数据的备份和恢复等;

(4)数据一致性要求与校验,如对多表的更新操作的要求、突然掉电/程序异常时数据一致性的要求;

(5)对可能造成意外人身伤害的设备及情况,提出软件安全性设计要求。

#### 【示例 1】 数据一致性需求

需求描述:软件必须在正常工作及发生异常(如掉电、异常退出)的情况下都保持数据间的一致性。如果发生了不一致,应提供工具检测及修复。

优先级:5

#### 【示例 2】 反修改需求

需求描述:软件应具有判断自身是否被修改(如被病毒感染)的能力。进一步地,如果自身已被修改,则应提醒用户如何进行恢复。

优先级:5

## 3. 可服务性需求

软件的可服务性。可服务性是针对产品而言的,指获得并实施服务(如产品安装、升级、改进和修复)的难易程度(对设备的影响,耗时,花费),包括可维护性、可安装性等。

可服务性由“可维护性(maintainability)”一词变迁而来。可维护性表明纠正一个缺陷或做一次更改的难易程度。可维护性取决于理解、更改和测试软件(发现、定位缺陷)的难易程度。一般可用“故障平均维修时间(MTTR)”、“最大维修时间”来表征。注意:MTTR与可用性(availability)相关,定义MTBF、MTTR时,必须满足“可用性”。

可服务性是一个综合的属性,可以体现为某些功能(如错误侦测、告警、自动恢复等),也与可靠性、可安装性、可维护性等相关。可以考虑以下情况:

- (1)支持“软件热插拔”。
- (2)支持故障隔离。发生异常后,错误应不扩散,且应在最短的时间内上报维护系统。
- (3)软件模块的相关性应尽可能少。
- (4)对于公用函数尽可能放在公用模块中。
- (5)输入数据有一致性检查,并且有保护。
- (6)支持远程操作。
- (7)支持后台升级重启。
- (8)管理重要的工程维护信息。

**注意:**如果为了提高“可服务性”而需要增加某一个功能,则必须将该功能描述在“功能需求”中。为了避免淹没在其他功能需求中,可以单独设置一节,如“可服务性相关”。如果实在有必要,也可以在此处重复罗列这些功能相关的可服务性,但前提是在“功能需求”中已经对这些需求进行了详细的描述,而且此处应该给出对这些功能需求的引用,这样做的唯一好处是:可在一个地方看到可服务性相关的所有需求。

#### 【示例1】 报表维护速度需求

需求描述:当用户更改报表相关规定时,所有受影响的报表必须可以在5个工作日内修改完毕。

优先级:5

#### 【示例2】 平均修复时间 MTTR

需求描述:平均修复时间 MTTR 应小于 2 小时。

优先级:5

#### 【示例3】 异常记录

需求描述:当发生异常时,软件应能记录操作系统版本、可用硬盘空间、最大物理内存及可用内存、用户的输入、错误类型;软件应能连续记录异常数据。

优先级:5

## 4. 易用性需求

所谓易用性,是指在特定使用环境下,为特定用户用于特定用途时所具有的有效性、效率和用户主观满意度。

- (1)有效性:指用户完成特定任务和达到特定目标时所具有的正确和完整程度;
- (2)效率:指用户完成任务的正确和完整程度与使用资源之间的比率;
- (3)满意度:指用户在使用产品过程中具有的主观满意和接受程度。

可以从以下方面考察“易用性”:

- (1)易理解性:标识/提示、界面、手册等符合规范、习惯的程度。

- (2)易学性。
- (3)易操作性:与用户的使用场景的一致程度,对用户体力、技巧等的要求程度。
- (4)防止误操作。
- (5)易记忆性。
- (6)易见:仅凭观察,用户就可知道产品的状态。
- (7)反馈:对用户的操作及时做出反馈。

“易用性”与“可服务性”有一定的交叉。实践中,重要的不是如何区分这两个属性,而是将必须满足的需求都描述出来。

另外,易用性一般都与某具体功能相关,因此在功能需求的描述中,应同时描述易用性方面的要求。

**【示例 1】** 数据编辑完成应有提示

需求描述:对于修改、删除、增加数据的操作,保存完成后应有提示。

优先级:5

**【示例 2】** 任何功能不允许只存在“向导”模式

需求描述:支持“向导”模式的同时,必须支持“专家”模式,以提高操作效率。

优先级:5

## 5. 可移植性需求

可移植性是度量把一个软件从一种运行环境转移到另一种运行环境所花的工作量的大小,或必须修改的代码比例。包括硬件独立性、软件独立性、易安装性、可重用性。要达到高的可移植性,在设计时应将“设备相关程序”与“设备无关程序”分开,将“处理逻辑”与“用户界面”分开。

需要说明的是,实践中的可移植性是指:当前已经决定软件要在若干个环境中运行了,但先实现环境 A,再移植到环境 B。在这过程中验证可移植性需求。

可移植性与“扩展性”不同,后者是指将来可能要做的需求。如果把将来要做的需求误放在可移植性中了,则该需求的验证就会出问题。我们不可能为了验证的需要,而做出一个目前不需要的东西。

**【示例 1】** 硬件独立性

需求描述:软件应能在 15 个工作日之内移植到 Alpha 环境,所花的工作量应不大于 30 人天(含)。

优先级:5

**【示例 2】** 软件独立性

需求描述:软件应能在 5 个工作日内支持 Netscape 浏览器(4.0 及以上版本),所花的工作量应不大于 10 人天。

**【示例 3】** 可移植性

需求描述:软件从〈环境 A〉移植到〈环境 B〉,代码的修改率应不大于 15%。

优先级:5

## 6. 扩展性需求

扩展性表明了软件适应“变化”的能力,对设计有重大影响。与“可维护性”在概念上

有一定的交叉,但实际描述时区别又很明显:扩展性重点描述“需求可能的变化”,而“可维护性”更多的是描述“应对的变化可能要采取的措施”。

原则上,应尽可能将要增加的需求、需求的可能变化,与明确要做的需求一起描述出来(可以在各类需求的“特殊需求”中描述)。可以考虑以下情况:

(1)已被提出,但由于某些原因必须在后续版本中实现的需求。如果该需求对设计有重要影响,则应在本节中进行描述。

(2)对需求变化的适应能力。需求可能的变化包括:

- ①操作方式上的变化;
- ②运行环境的变化;
- ③同其他系统的接口的变化;
- ④精度和有效时限的变化。

**【示例1】** 应为“支持 TEMS、Wanhe 手机路测数据分析”留下接口。

需求描述:当前版本要求支持 Safco 路测数据的分析。由于无法取得 TEMS、Wanhe 数据的格式,因此只能放到下一版本实现,但设计时应留有接口,不要每新增一种数据就对程序修改一次。

优先级:3

## 7. 可测试性需求

可测试性指发现缺陷的难易程度。可测试性不同于需求说明的可验证性,后者主要是指需求是否可以作为测试结果本身是否可验证,而前者是对设计与实现的要求,即设计与实现时应采取何种方式保证可测试性。当然,如果没有好的可测试性,即使需求本身是可验证的,但由于无法测试,因此也无法真正对需求进行验证。

如果软件中包含复杂的算法和逻辑,或具有复杂的功能性的相互关系,那么对于可测试性的设计就很重要。可以考虑以下情况:

(1)可观察性。

①对关键程序的运行状况和关键数据(或全局数据)变化情况应能够实时或准实时显示,或在运行中可查询。(可监视)

②过去的关键程序的运行状况和关键数据(或全局数据)可见,或在运行中可查询(例如:事务日志)。

③所有影响输出的因素都可见。

④容易识别错误输出。

⑤通过自测机制自动侦测内部错误。(系统有自测功能)

⑥自动报告内部错误。系统的各类程序在发生运行故障时应能告警并留下历史记录,通过该记录可以准确地定位故障。

(2)可控制性。

①所有可能的输出都产生于某种输入组合。

②通过某种输入组合,所有的代码都可能被执行。

③测试工程师可直接控制软件和硬件的状态及变量。

(3)可分解性。

①软件系统由独立模块构成。

②能够独立测试各软件模块。

(4)软件应具有接口、流程等的跟踪能力。

#### 【示例 1】 日志输出控制

需求描述:可以通过配置文件来控制内部关键信息的输出,输出目标也可通过配置文件配置。关键信息、输出目标如下:……。

优先级:5

#### 【示例 2】 消息转向

需求描述:可以通过配置文件将发往预定目标的某些消息转发至特定目标。消息、预定目标、特定目标及其可能关系如下:……。

优先级:5

### 3.2.2.4 其他需求

本节描述其他需求,如执行标准、设计和实现上的限制、国际化等。其他需求并非是不重要、可以不描述的需求,相反,它对设计与实现有比较大的影响。

其他需求的描述格式为:

需求描述:需求详细描述。

优先级:见 use case 中对于“优先级”的描述。

特殊需求:见性能需求中对于“特殊需求”的描述。

### 1. 通用化、系列化、模块化

通用化:在相互独立的系统中,最大限度地扩大具有功能互换和尺寸互换的功能单元使用范围的一种标准化形式。换句话说,通用化是选定或研制具有互换性特征的通用单元,并将其用于新研制的系统,以满足其需求的一种标准化方法。以前的通用化仅限于零部件、通用电路和结构要素,并以优选为主要手段。现代的通用化已扩展到整机、系统,甚至软件,实施手段也从优选发展到开发。

系列化:是根据同一类产品的发展规律和使用要求,将其性能参数作合理的安排和规划,或对其型式和结构进行规定和统一,从而有目的地指导同类产品发展的一种标准化形式。系列化的对象是产品的品种、规格。开展系列化,一方面需要对现有品种、规格进行压缩简化,另一方面需要开发新的品种、规格,使这类产品构成品种齐全、数量适宜、结构和功能优化的产品体系,以满足使用要求。系列化中每一规格的产品都应具有一定范围的通用性。系列化的产品给人以专业、独特的印象。

模块化:在对一定范围内的不同产品进行功能分析和分解的基础上,划分并设计、生产出一系列通用模块或标准模块,然后,从这些模块中选取相应的模块并补充新设计的专用模块和零部件,以构成满足各种不同需要的产品的一种标准化形式。模块化需求描述的重点是:

①将要继承和借用的模块信息;

②准备提供给其他产品或模块使用的模块信息。

与“三化”概念对应的是“复用”。所谓复用,是指已有的软件构件(组件/部件)可用于

构造新的软件或系统。换句话说,可复用的软件构件可以用于不同的软件或系统。一个“高复用性”的软件或系统由很多可现货供应或标准构件组成,一个“低复用性”的软件或系统则包含很多需要单独开发的、只用于自身的部件。对于软件来说,一般应考虑以下构件的复用:

- ①平台;
- ②模块。

### 【示例 1】 必须使用的平台

需求描述:本软件需要使用的平台见表 3-10。

表 3-10

| 平台名称 | 版本  | 平台概述 | 开发单位 |
|------|-----|------|------|
| ×××× | ××× | ×××× | ×××× |
| ×××× | ××× | ×××× | ×××× |
| ×××× | ××× | ×××× | ×××× |

平台名称:描述平台的正式名称。

版本:将要使用的平台的版本(一般为形如 Vx.yy 的系统版本,如有必要,也可采用更细的版本号)。

平台概述:概要描述平台的组成、主要功能及关键指标、应用情况等。

开发单位:开发单位名称。

### 【示例 2】 必须使用及提供的模块中

需求描述:在设计时应复用表 3-11 所列的模块,并提供表 3-12 中所列的模块。

表 3-11

| # | 复用的模块名称 | 复用的模块代码 |
|---|---------|---------|
| 1 | ×××模块   | ×××代码   |
| 2 | ×××模块   | ×××代码   |

表 3-12

| # | 待提供的模块名称 |
|---|----------|
|   | ××××     |
|   | ××××     |
|   | ××××     |

优先级:4

## 2. 设计和实现上的限制

设计与实现时必须考虑的约束条件,并尽可能说明施加这些约束条件的原因。它不影响系统的外部行为,但必须被满足以达到技术、商业或合同的要求。至少应考虑两类限制:一般限制、用户界面限制。表 3-13 列出了常见的一般限制。

表 3-13

常见的一般限制

| 序号 | 限制    | 详细描述  | 原因                             |
|----|-------|---|--------------------------------|
| 1  | 特定技术  | 必须使用××技术  | ××××                           |
| 2  | 专利    | 可以采用 Q 公司以下专利：<br>①〈专利号 1 专利名称 1〉<br>②〈专利号 2 专利名称 2〉<br>必须回避 N 公司以下专利：<br>①〈专利号 3 专利名称 3〉<br>②〈专利号 4 专利名称 4〉<br>计划开发的专利：<br>①×××<br>②×××  | 已取得 Q 公司专利许可；<br>与 N 公司专利谈判破裂。 |
| 3  | 工具    | 必须使用或避免×××工具或类库   | ××××                           |
| 4  | 编程语言  | 必须使用 C 语言   | ××××                           |
| 5  | 数据库   | ××××<br>说明：必须使用的表结构，必须保留的字段，甚至必须使用的数据库等。而使用频率、访问的能力，则应写在性能需求中   | ××××                           |
| 6  | 硬件限制  | 存储器限制：×××<br>其他限制：×××   | ××××                           |
| 7  | 数据格式  | ××××  | 客户需要据此进行二次开发                   |
| 8  | 软件许可  | ①必须使用加密技术以保证只有授权用户才能使用本软件。<br>②必须使用加密技术以保证只有特定数目(可配置)的用户能同时使用本软件。<br>说明：为防止盗版，提出软件加密技术和要求。是否需要加密？如是，采用硬件加密还是软件加密？是否限制同时使用的用户数？  | ××××                           |
| 9  | 升级方式  | ××××<br>说明：是否支持网上自动升级？必须以 Patch 方式或卸载后重新安装进行升级？这将影响方案设计，同时影响安装程序的实现   | ××××                           |
| 10 | 分模块发布 | ××××<br>说明：是否允许只发布众多功能中的若干功能？这需要仔细设计功能间的依赖关系，严重影响方案设计。  | ××××                           |
| 11 | 安装/卸载 | 本软件应使用 InstallShield 制作安装/卸载程序，并由安装程序自动完成数据库连接信息的设置。使用数据库管理系统的脚本执行程序建立数据库、表、存储过程。为了实现的方便，不对数据库进行细微的权限设置，应用程序都通过一个最大权限的用户访问数据库，“安全性”完全通过应用程序进行控制。<br>说明：安装与卸载方式(如采用商业安装软件制作标准安装或自行开发安装程序)、数据库连接信息设置方式(如由安装程序自动设置或由用户手工设置 ODBC、BDE 连接信息)、数据库/表/存储过程等的建立方式、数据库的权限设定方式等 | ××××                           |
| 12 | 兼容性   | ××××<br>说明：规定新旧版本数据之间的关系(如新版本必须能直接读取老版本的××数据，或必须提供格式转换程序等)、硬件如单板之间的关系、接口线缆之间的关系等  | 用户(×××)及市场部门(×××)提出。保护用户先期投资   |
| 13 | 其他    | ××××<br>说明：关键操作应尽可能支持回溯、支持软件版本的在线更换，软件版本更换失败应有回溯手段，系统要有流量控制功能，对于关键软件，应留有足够余量等   | ××××                           |

**【示例 1】 必须使用的技术**

需求描述:设计时必须采用××技术。该技术详细描述见……。

优先级:4

**【示例 2】 必须采用的专利**

需求描述:设计时必须使用的专利为:

①〈专利号 1 专利名称 1〉

②〈专利号 2 专利名称 2〉

优先级:4

**【示例 3】 用户界面限制**

需求描述:在设计与实现时,必须满足以下要求:

①应执行〈××××〉编程规范。

②应使用多文档界面。

③凡是超过 5 秒的内部操作,均应显示进程条;进程条应该显示在主窗口的状态栏上,并且进程条前面应显示当前正在进行的操作。

用户界面要求是指用户对界面的图形化方式、屏幕布局、标准按钮、告警显示方式、指示灯等外观视觉特征的要求,并不需要罗列真实的用户界面。可以考虑以下(但不限于)情况:

①将要采用的图形用户界面(GUI)标准、产品系列的风格。

②屏幕布局或解决方案的限制。

③快捷键。

④错误信息显示标准。

由于一般的编程规范已基本体现了以上 4 点,因此,可以直接引用某个编程规范(其中有用户界面标准)。当然,如有额外的用户界面需求时,也应明确列出。

另外,如确实需要描述用户界面的细节(如特定对话框的布局),为了便于维护,应编制独立的用户界面需求说明书,格式不限。一般应包括如下内容:

①具体界面罗列。一般应给出每个界面的示例,并同时说明实现时必须实现的属性,如“颜色、图标、文字、相对位置、控件大小应保持一致,而控件的绝对位置则无需关心”等。

②未尽事项的规定。如“出现了没有考虑到的界面问题,应与客户协商而不是由项目组成员自行决定”。

优先级:4

**【示例 4】 包装盒必须环保及可回收**

需求描述:软件包装盒必须满足如下企标要求:

①《禁止和限制使用的环境物质要求》。

②《元器件认定工艺要求》。

优先级:5

### 3. 执行标准

执行标准罗列上游及同级相关工作产品、软件必须执行的标准(也可以是协议、规范)或标准中的具体部分(不仅指国际、国家、行业和企业的相关标准,还包括相关法律、法规



方面的内容)。罗列顺序为:国家标准、行业标准、企业标准、文件、国际标准、国外标准,同类标准应按字典顺序排列。

“执行标准”与“参考文献”的区别是:前者是本文的一部分(通过引用或直接把标准中的内容写到本文件中),其具体条款必须被满足;而后者不是本文的一部分,它的作用主要是佐证本文的某些观点、引导读者进一步阅读等。

#### 注意:

①关于“执行标准”的版本问题。为了维护方便,引用处及本处都不列出版本信息,默认总是执行最新版本。如果某标准存在几个现行有效版本,且本文未执行(或不打算执行)最新版本,则应列出具体版本信息。

②本模板引用的标准,只能保证在编写模板时其版本是最新的。在使用本模板编制文档时,应对照最新标准进行引用。

③如果国际化需求涉及特定国家的特定协议,则在此处详细说明。

④尽管编程规范是对设计与实现的限制,在本模板中的“设计和实现上的限制”中已有描述,但还是应在此处罗列这些标准。

## 4. 国际化

产品正越来越多地走向国际市场,因此,国际化是产品开发必须考虑的一个问题。在本节中应列出有关国际化方面的需求。

国际化需求主要包括四大类:第一类与语种相关,后台软件一般都特别关注,这也是一提起国际化马上就想起的内容;第二类与协议相关,前台软件一般都特别关注,开发初期最易被忽略;第三类是与国际认证相关(如 EMC 的杂散要求);第四类与法律、宗教等相关。

### (1) 语种相关。

①首先应明确有哪些文档(一般为用户类文档)需要考虑支持多语种?具体有哪些语种?

②然后应明确代码的国际化要求。应区分前后台软件:日志、调试信息(一般为英语)、界面、输入(如各种配置文件)、输出(如报告)等。

③最后可能要规定国际化的实现方式。不同的实现方式其效果也不一样,特别是对于后台(有界面)的软件。方法一是动态设置,立即更新。方法二是 Borland 提供的国际化默认方法,根据操作系统的区域设置自动选择语种,用户不能主动设置,除非是更改区域设置或给资源 dll 文件改名。另外,还需要规定不同语种的界面是否需要调整,以适应不同语种的文字长度(好多商品化软件为了节省开发费用,完全不考虑该要求,导致界面很不美观,甚至内容无法显示完全,影响使用)。

(2)协议相关。如果与协议相关,应明确需要支持哪些国家的哪些特定的协议。

(3)国际认证相关。一般体现在“执行标准”中或单列。

(4)法律、宗教等相关。

①法律。如:欧盟关于环保的法令《EU 有害物质限制》,在法国销售的软件及文档必须使用法语,在比利时销售的软件必须支持当地开发的硬件,德国的竞争法严格限制生产商声称其产品比其他产品好,等等。

②宗教。如中东的某些国家往往反感与美国、英国有关的标志,因此应考虑某些部件是否不能标注其产地。

国际化是一个相当复杂的课题,以上考虑的仅是问题的很小的几个方面,其他还需考虑的有键盘布局(日文、德文、英文键盘布局不一样)、文字方向(从左至右、从右至左、从上到下等)、日期时间格式、排序方式等。在可能的情况下,亦应对这些作出具体的规定。详情请见《Windows 95、Windows NT 国际软件开发指南》。

**注意:**此处所谓“支持的语种”,是指“肯定需要实现的”,对于以后“可能需要支持的语种”,应在需要时通过需求变更的方式来体现。

### 【示例 1】 语种相关的国际化需求

**需求描述:**本软件有表 3-14 所列的语种相关的国际化需求。“/”表示应支持对应的语种,而“-”则无需支持。

表 3-14

| 类别 | 工作产品名称    | 语种                      |    |  | 备注 |
|----|-----------|-------------------------|----|--|----|
|    |           | 中文                      | 英语 |  |    |
| 文档 | 技术手册      | ✓                       | ✓  |  | 无  |
|    | 安装手册      | ✓                       | ✓  |  | 无  |
|    | 操作维护手册    | ✓                       | ✓  |  | 无  |
|    | 各类宣传资料    | ✓                       | ✓  |  | 无  |
|    | 各类培训资料    | ✓                       | ✓  |  | 无  |
| 实现 | 联机帮助      | ✓                       | ✓  |  | 无  |
|    | 调试信息      | -                       | ✓  |  | 无  |
|    | 日志        | ✓                       | ✓  |  | 无  |
|    | 输入(如配置文件) | -                       | ✓  |  | 无  |
|    | 输出(如报告)   | ✓                       | ✓  |  | 无  |
|    | 界面        | ✓                       | ✓  |  | 无  |
| 其他 | 实现方式      | 动态设置,立即刷新。界面应根据具体语种进行调整 |    |  |    |
|    | 键盘布局      | 采用操作系统设置,无需特别考虑         |    |  |    |
|    | 文字方向      | 采用操作系统设置,无需特别考虑         |    |  |    |
|    | 排序        | 采用操作系统设置,无需特别考虑         |    |  |    |
|    | 时间日期格式    | 采用操作系统设置,无需特别考虑         |    |  |    |

优先级:5

## 5. 参考资料

列出编写本文时的参考资料或其他资源。如参考了多篇、多种类型的文献,则应按国家标准、行业标准、企业标准、文件/书籍、国际标准、国外标准的顺序罗列。同类的文献应按字典顺序罗列。

“参考文献”与“执行标准”的区别是:后者通过引用自动成为本文的一部分,其中被引用的具体条款应被满足或执行;而前者不是本文的一部分,它的作用主要是佐证本文的某些观点、引导读者进一步阅读等。

原著类型的参考文献的编写格式如下：

[序号] 作者 1<, 作者 2, …>< 等>< 著/编/编著>, 资料名称<, 版本>.< 地点:>出版社/刊物名称, 时间<, 总期数(本年期号); 开始页号~结束页号)

**【示例 1】** 参考文献(原著类型)

[1] 林锐 等编著. 高质量程序设计指南——C++ / C 语言. 北京: 电子工业出版社, 2002. 6

[2] 范玉顺, 吴澄, 俞盘祥. 面向制造业的 CIMS 应用集成平台. 清华大学学报, 1998, 38(3): 104~107

翻译类型的参考文献的编写格式如下：

[序号] 作者 1<, 作者 2, …>< 等>< 著/编/编著>, 译者 1<, 译者 2, …>< 等>译. 资料名称<, 版本>.< 地点:>出版社/刊物名称, 时间<, 总期数(本年期号); 开始页号~结束页号)

**【示例 2】** 参考文献(翻译类型)

[1] Roger S. Pressman 著, 黄柏素, 梅宏等译. 软件工程: 实践者的研究方法, 第四版. 北京: 机械工业出版社出版, 1999. 10

### 3.2.3 任务同步训练

**任务描述：**

以任务 1 中描述的环境在线监测系统的实时监控需求为例来编写其功能需求和性能需求。

**实施步骤：**

(1) 环境在线监测系统实时监控功能需求如图 3-5 所示。

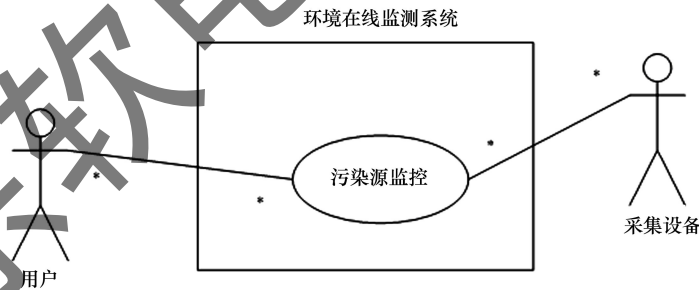


图 3-5 环境在线监测系统实时监控功能用例图

需求描述: 采集设备上发生业务, 采集设备把相关业务信息上报给环境在线监测系统, 环境在线监测系统对采集到的信息根据策略进行分析过滤, 把污染源信息提取出来展现给用户, 另外污染源信息还需要持久化保存, 以使用户后续查看。

Actor: 采集设备, 用户

优先级: 5

使用频度: 经常

前置条件:

C01 采集设备已接入环境在线监测系统；

后置条件：

R01 污染源信息被监控到并展现给用户。

正常过程：N1

N01 采集设备上发生业务，用例开始。

N02 采集设备把相关业务数据上报给环境在线监测系统。

N03 环境在线监测系统根据策略对业务数据进行分析过滤。

N04 环境在线监测系统把污染源信息提取出来展现给用户并持久化保存。

N05 用户实时查看到污染源信息。

N06 用例结束。

可选过程：O1 采集设备上发生分析系统不关心的业务(在 N1、N02 处)

O01 采集设备上发生分析系统不关心的业务，用例开始。

O02 采集设备不把业务数据上报到环境在线监测系统。

O03 用例结束。

可选过程：O2 环境在线监测系统未配置过滤策略(在 N1、N03 处)

O01 环境在线监测系统未配置过滤策略，用例开始。

O02 环境在线监测系统直接把业务数据丢弃。

O03 用例结束。

可选过程：O3 上报的业务数据不包括污染源信息(在 N1、N04 处)

O01 上报的业务数据不包括污染源信息，用例开始。

O02 环境在线监测系统把此业务数据丢弃。

O03 用例结束。

可选过程：O4 用户设置了过滤条件不监视该污染源(在 N1、N05 处)

O01 用户设置了过滤条件不监视该污染源，用例开始。

O02 用户在实时监控界面上看不到污染源信息。

O03 用例结束。

异常过程：E1 采集设备与环境在线监测系统链路断(在 N1、N02 处)

E01 采集设备与环境在线监测系统链路断，用例开始。

E02 环境在线监测系统接收不到采集设备上报的业务数据。

E03 用例结束。

异常过程：E2 环境在线监测系统过滤策略异常(在 N1、N03 处)

E01 环境在线监测系统过滤策略异常，用例开始。

E02 环境在线监测系统直接丢弃该业务数据并记录日志备查。

E03 用例结束。

异常过程：E3 污染源信息持久化失败(在 N1、N04 处)

E01 污染源信息持久化失败，用例开始。

E02 环境在线监测系统丢弃该业务数据并记录日志备查。

E03 已经打开的实时监控界面上用户还是能查看到污染源信息。

E04 用例结束。

特殊需求：

S01 采集设备与环境在线监测系统链路断的情况下,要求采集设备能够缓存 24 小时或者 50M 的业务数据,在链路恢复后能够补报上来。

S02 污染源信息要求能够持久化保存至少 3 个月。

(2)环境在线监测系统实时监控性能需求

需求描述:采集设备与环境在线监测系统间消息流量不能超过 100 条/秒,数据流量不能超过 1M/秒,污染源信息分析过滤时长不能超过 5 秒;从业务发生到污染源信息展现在实时监控界面上时延不能超过 10 秒;污染源信息保存时长不能小于 3 个月。

优先级:5

### 3.2.4 自主学习任务

#### 1. 练习题

(1)补充环境在线监测系统实时监控需要考虑的质量属性需求。

(2)编写一个环境在线监测系统的报表功能需求和性能需求。

#### 2. 拓展学习任务

熟悉 Oracle 数据库技术、SQL 语言。