

学位论文创新性声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：_____ 日期_____

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署名为西安电子科技大学。

（保密的论文在解密后遵守此规定）

本学位论文属于保密，在____年解密后适用本授权书。

本人签名：_____ 日期_____

导师签名：_____ 日期_____

摘 要

摘要按照三段论进行组织：（1）论文工作目的和本文主要解决的问题；（2）解决问题中所采用的技术方法、工作过程、实验和测试；（3）论文工作的结论、结果和应用效果等。

软件工程硕士学位论文是评价硕士生阶段工作成果的重要内容，通过撰写硕士学位论文，能够将自己所学到的知识与工程实践结合起来，为进一步的学习和工作奠定良好的基础。但通过相关研究发现，目前硕士生所撰写的软件工程硕士学位论文在内容组织、描述方式等方面存在诸多问题。

文本针对论文撰写过程中存在的问题，**研究了**软件系统设计类学位论文的撰写规范和模式的设计与实现。论文在**阐述了**学位论文的基本概念、原理和方法的基础上，**介绍了**学位论文撰写的相关技术；**分析了**软件系统设计类论文的撰写过程和应用需求，分别利用实体联系图和数据流图对系统的数据和过程进行建模；**给出了**系统的应用架构设计，**设计了**系统多个不同功能模块和硬件器件实现的撰写方法；**实现了**软件系统设计类硕士学位论文的撰写规范和模式。

基于西安电子科技大学软件学院软件工程硕士学位论文撰写的测试结果表明，论文撰写规范能够帮助硕士生组织论文内容，选择更合理的内容描述方式，有效提升软件工程硕士学位论文的质量，达到了预定设计目标。

关键词：学位论文 软件工程硕士 软件系统设计

Abstract

英文摘要撰写中，可直接将中文摘要按照准确的理解翻译为英语即可。切忌用翻译软件翻译后不经修改直接采用。

Keyword: Dissertation, Master of Software Engineering, Software System Design

目 录

第一章 绪论.....	1
1.1 选题背景及意义.....	1
1.2 国内外现状分析.....	1
1.3 论文工作内容.....	1
1.4 论文组织结构.....	1
第二章 相关技术概述.....	3
2.1 ×××理论.....	3
2.2 ×××技术.....	3
2.3 ×××技术.....	3
2.4 本章小结.....	3
第三章 ××业务需求分析.....	5
3.1 ×××业务陈述.....	5
3.2 ×××需求建模.....	5
3.3 ×××数据建模/×××数据分析.....	6
3.4 ×××过程建模.....	7
3.5 本章小结.....	7
第四章 ××系统设计与实现.....	9
4.1 ×××系统应用架构.....	9
4.2 ×××系统功能设计.....	10
4.2.1 ×××软件组件.....	10
4.2.2 ×××硬件选型.....	10
4.3 ×××系统数据库设计/×××数据结构设计.....	10
4.4 ×××系统功能实现.....	11
4.4.1 ×××组件实现.....	11
4.4.2 ×××系统集成.....	13
4.5 本章小结.....	13
第五章 ××系统测试及分析.....	15
5.1 系统运行环境.....	15

5.2 测试用例及过程.....	15
5.3 测试结果分析.....	16
5.4 本章小结.....	17
第六章 结束语.....	19
6.1 论文工作总结.....	19
6.2 后续工作展望.....	19
致 谢.....	21
参考文献.....	23
附录一：UML 图使用指南.....	25
附 1.1 UML2.0 图示类型.....	25
附 1.2 UML2.0 图使用方法.....	26
附 1.3 UML2.0 图绘制方法.....	27
附 1.3.1 活动图.....	27
附 1.3.2 用例图.....	28
附 1.3.3 类/对象图.....	30
附 1.3.4 顺序图.....	32
附 1.3.5 通信图.....	33
附 1.3.6 状态图.....	34
附 1.3.7 组件图.....	35
附 1.3.8 部署图.....	36

第一章 绪论

学位论文主要是向读者介绍工作的背景、现状、工作内容和过程。绪论部分通过介绍为什么做？别人做了什么？我要做什么？准备如何做？四个方面引出论文的主体内容。

1.1 选题背景及意义

论文工作的行业和领域背景，论文所依赖的项目背景和发展现状，论文工作预期的收益和社会价值等内容。（建议1页到2页）

1.2 国内外现状分析

主要介绍近5年内国内和国外与本论文相关的研究工作、工程项目和产品等。要注意突出特点和重点，主要介绍特色和优势。（建议1页到2页）

1.3 论文工作内容

本节按照软件系统开发的过程分不同阶段介绍论文工作的内容，通常包括应用需求分析、系统架构设计、质量属性设计、各模块功能设计、功能实现和测试等。可在描述过程中突出论文工作的特点和关键功能。（建议半页）

本文主要研究XX系统建设相关的项目技术管理工作，在整个过程中主要完成以下工作：

- 1、分析××的业务需求。在分析项目建设需求的基础上，分析了系统功能与性能需求。
- 2、设计××系统的架构和功能模块。在需求分析的基础上，设计了系统应用架构，并进行模块化分解。
- 3、实现××系统的软硬件功能组件。完成了×××××的模块开发与集成。
- 4、部署××系统并给出系统运行及测试结果分析。

1.4 论文组织结构

介绍论文工作过程，分章简要介绍每章的主要内容。（建议半页）

论文共分为六章，各章主要内容如下：

第一章：绪论。提出论文选题背景及意义、国内外研究现状分析、论文主要的

工作内容和组织结构。

第二章：相关技术概述。列举并简要描述了项目开发过程中涉及到的××理论和相关技术，其中包括××技术、××技术等。

第三章：××业务需求分析。在介绍系统业务数据和过程的基础上，完成系统应用需求建模、数据建模和过程建模。

第四章：××系统设计与实现。设计系统的应用架构，按照功能分解结构细化系统功能，分别设计软件组件和确定硬件选型，设计系统数据库。按照组件设计结构开发、采购相应的软件组件和硬件，完成系统集成。

第五章：××系统测试及分析。阐述系统运行和部署环境，采用测试用例介绍系统测试过程，针对测试结果进行分析给出结论。

第六章：结束语。总结了本文的主要工作，指出工作的不足及进一步的改进方向。

第二章 相关技术概述

相关技术概述部分。（建议不超过4000字，相关技术2~3个）在选定相关技术的基础上，将对该技术的理解采用书面语形式准确描述，切忌从网络上直接摘录比较随意的描述方式和内容。

2.1 ×××理论

阐述主要研究内容相关基本原理、理论和概念。

2.2 ×××技术

概述相关技术一。

2.3 ×××技术

概述相关技术二。

2.4 本章小结

第三章 ××业务需求分析

系统分析是研究系统能做什么，是软件系统设计的前提，在系统开发中决定着项目的成败。系统分析的过程是对系统所需要的功能、质量和约束进行准确描述的过程，即需求分析。需求分析通常包括业务过程陈述及问题分析、需求建模、数据建模和过程建模等。

3.1 ×××业务陈述

该目标系统开发前的业务过程及所存在的问题分析。通常可采用活动图（Activity Diagram）+文字形式进行描述。活动图构建过程：

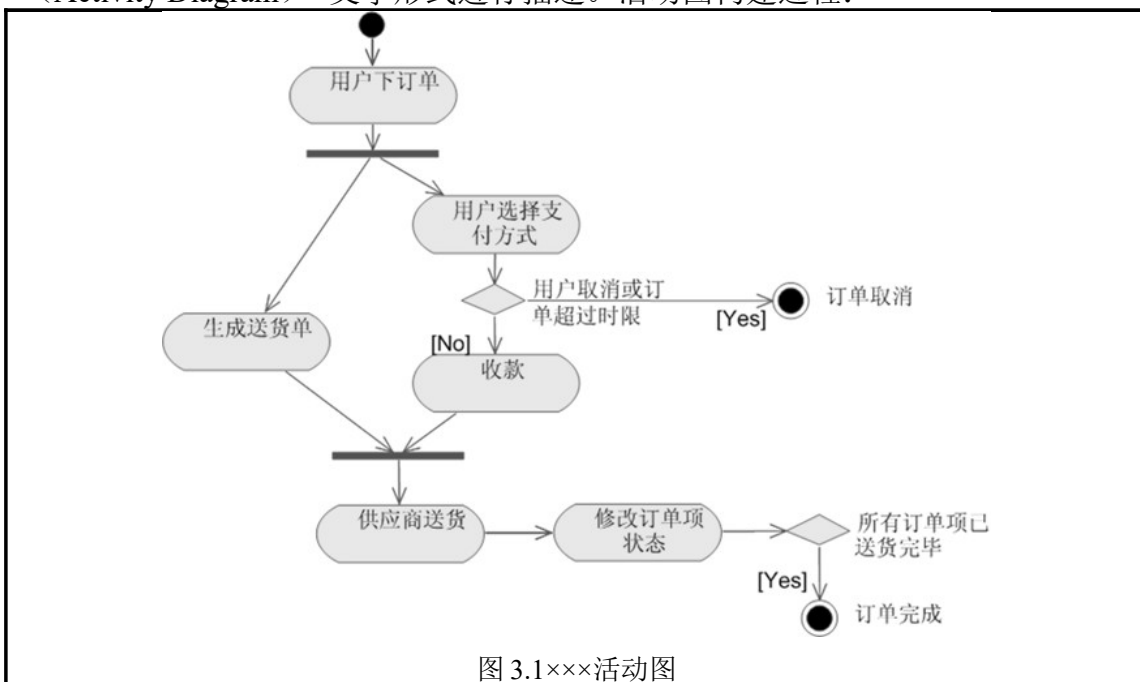
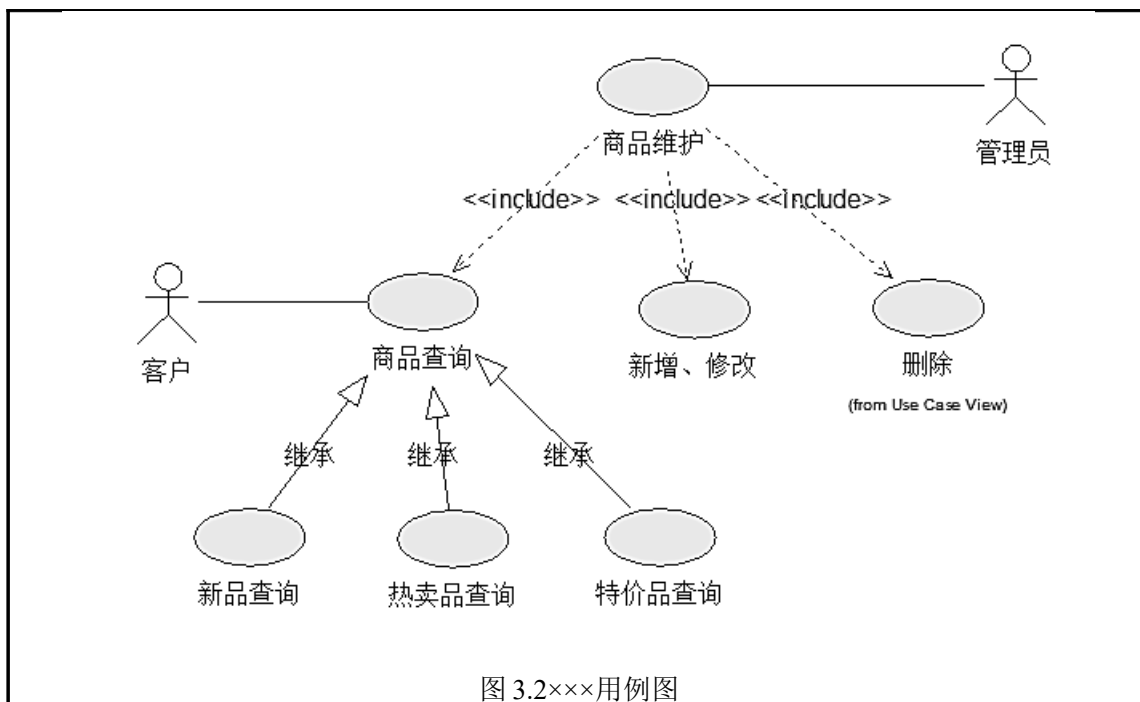


图 3.1×××活动图

3.2 ×××需求建模

对业务过程的内容进行分析，抽取出业务功能、质量属性和约束。业务功能可采用用例图（Use case Diagram）+文字形式进行描述，质量属性和约束可采用文字进行描述。用例图构建过程：

- （1）按照角色识别并定义系统参与者；
- （2）按照功能识别并定义系统用例；
- （3）分析参与者、用例的复用性；
- （4）将用例分配到参与者；
- （5）用文字描述用例图。

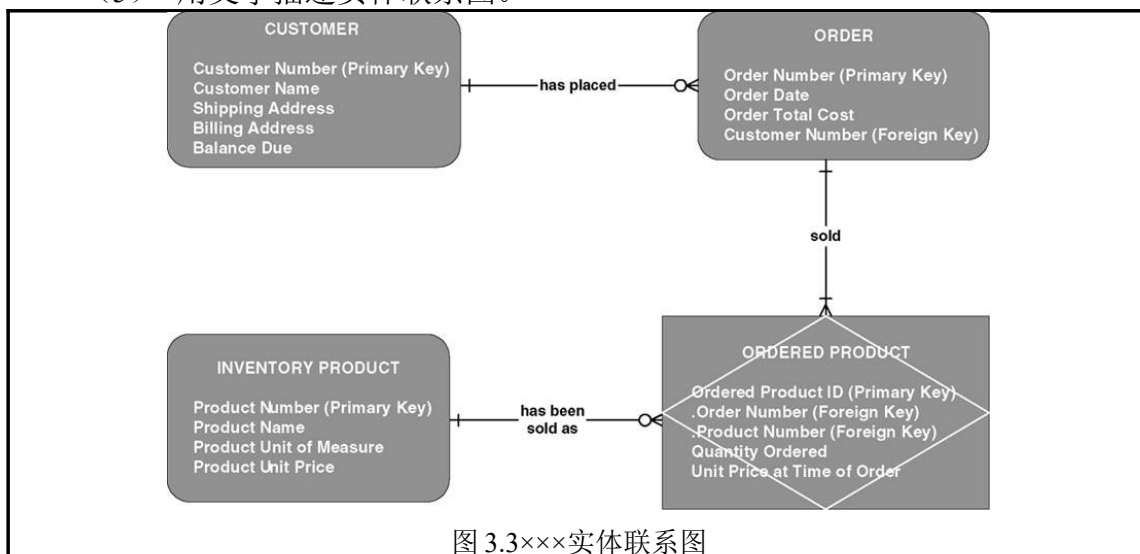


3.3 ××数据建模/××数据分析

如果项目规划中无数据库，本节标题为“××数据分析”，通过分析数据流，确定相关数据项和数据字典。

对业务过程进一步细化，按照数据建模过程分析数据实体及关系。通常可采用**实体联系图**（Entity Relation Diagram）+文字进行描述。实体联系图构建过程：

- （1）构造上下文数据模型：包括实体和联系，确定系统范围；
- （2）绘制基于键的数据模型：每个实体中加入主键；
- （3）构造完整属性的数据模型：实体中加入其他属性信息；
- （4）规范化：满足第三范式要求；
- （5）用文字描述实体联系图。



3.4 ×××过程建模

对业务过程进一步细化，按照过程建模过程逐层分析业务流程和活动。通常可采用**数据流图**（Data Flow Diagram）+文字进行描述。数据流图构建过程：

- （1）构造上下文数据流图：系统被看作一个过程，确定系统范围；
- （2）绘制功能分解图：将上下文数据流图中的过程按照功能划分子系统；
- （3）创建事件响应图：将子系统业务过程进一步细化，用数据流图描述；
- （4）创建系统图：将不同事件的数据流图合并形成系统数据流图；
- （5）用文字描述数据流图。

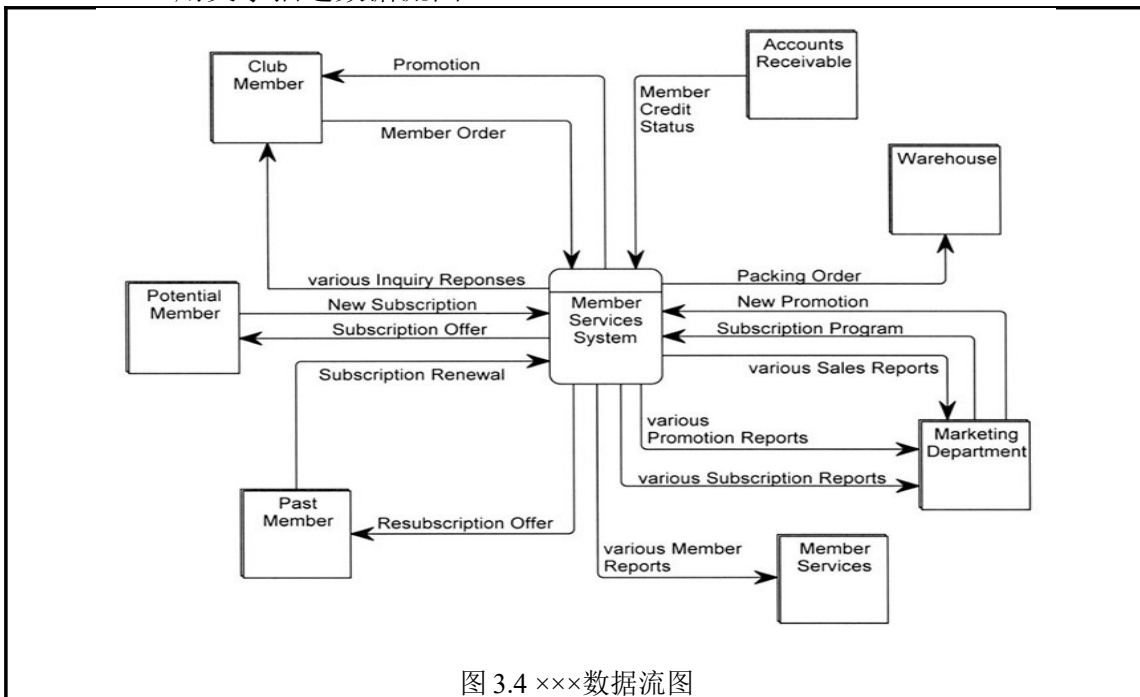


图 3.4 ×××数据流图

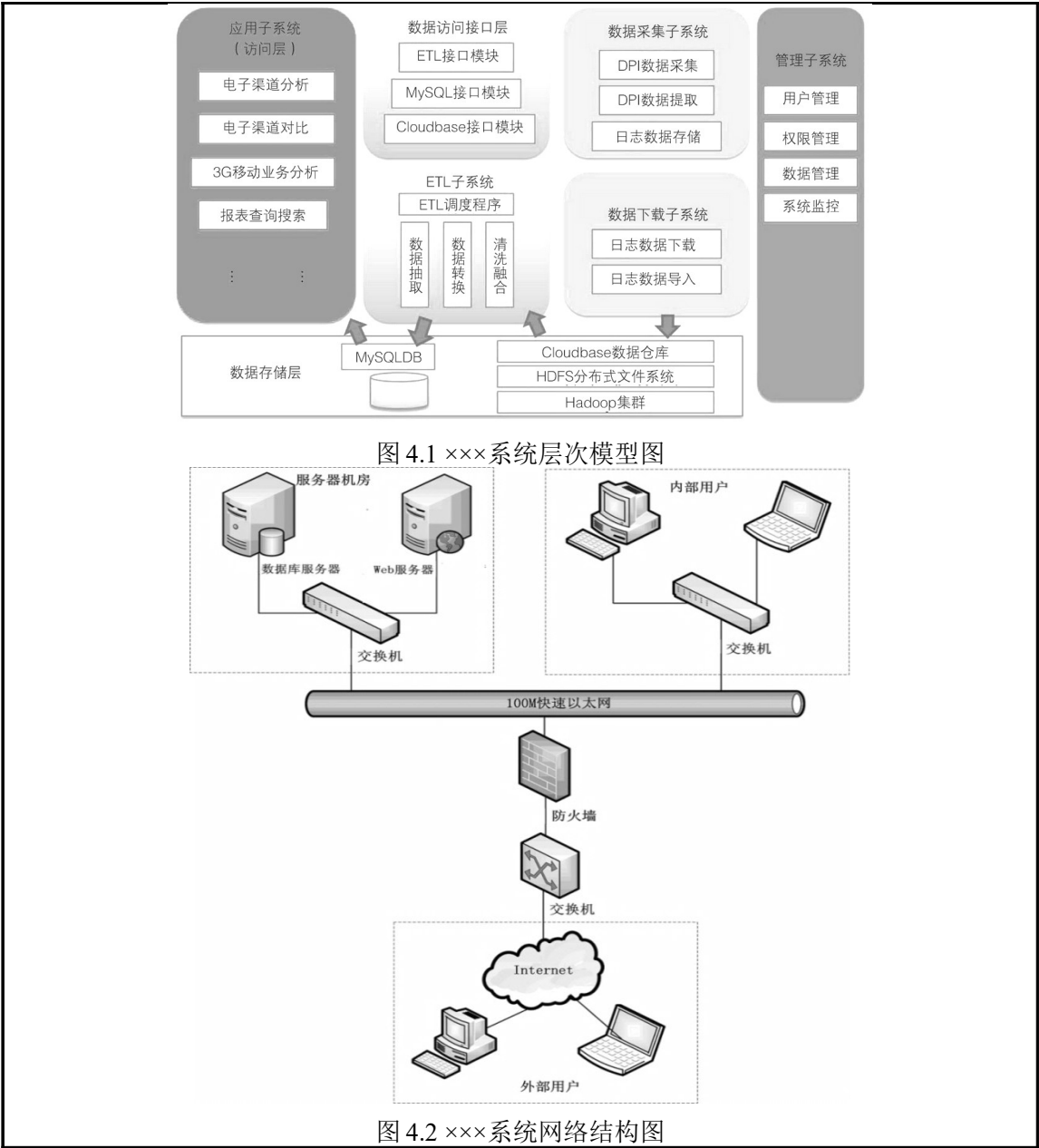
3.5 本章小结

第四章 ××系统设计与实现

系统设计是在系统分析的基础上研究系统如何实现需求中所描述功能。系统设计的过程包括系统应用架构设计、功能模块设计、数据库设计、界面和接口设计等。系统实现的过程是按照模块功能分别采用类图、序列图、接口、算法、程序流程图等方式描述实现方法和过程。

4.1 ×××系统应用架构

系统应用架构设计用来描述系统的骨架和轮廓。集中式系统可以采用层次模型图描述；分布式系统可采用网络结构图描述。



4.2 ×××系统功能设计

4.2.1 ×××软件组件

软件组件设计是在应用架构设计的基础上将系统内部进行细化，将系统按照功能划分为多个模块，并描述模块之间的关系和接口。通常可采用功能模块分解图+文字、组件图+文字进行描述，并给出各模块之间的接口描述。

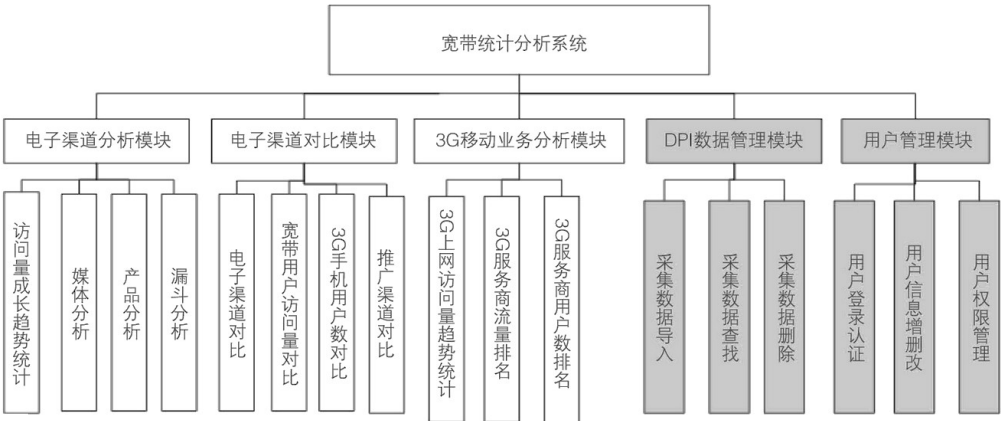


图 4.3 ×××系统功能模块分解图

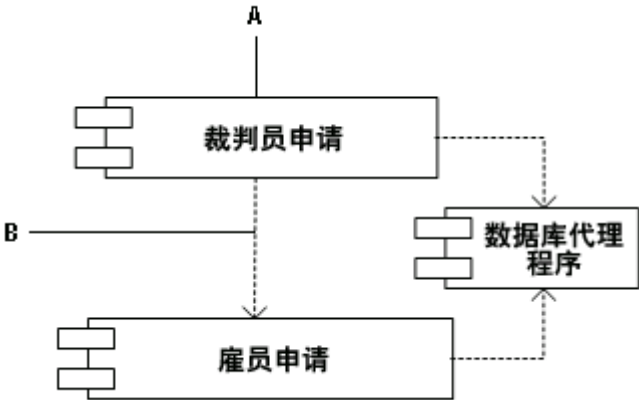


图 4.4 ×××系统组件图

4.2.2 ×××硬件选型

硬件选型设计是在应用架构设计的基础上制定系统内部的硬件平台和器件选型标准和指标体系。通常可采用选型条目+文字进行描述，并给出具体的选型指标体系。

4.3 ×××系统数据库设计/×××数据结构设计

如果项目规划中无数据库，本节标题为“×××数据结构设计”，通过数据结构给出数据项相关定义和构造过程。

数据库设计是在数据建模的基础上根据特定数据库系统构建系统物理数据

模型。通常可采用**物理数据模型**+文字进行描述。物理数据模型构建过程：

- (1) 每个实体创建一张表；
- (2) 每个属性创建一个字段；
- (3) 为主键创建索引；
- (4) 为关系指定外键；
- (5) 确定每个属性的数据类型、字段大小、空或非空、域及默认值；
- (6) 用文字描述物理数据流图。

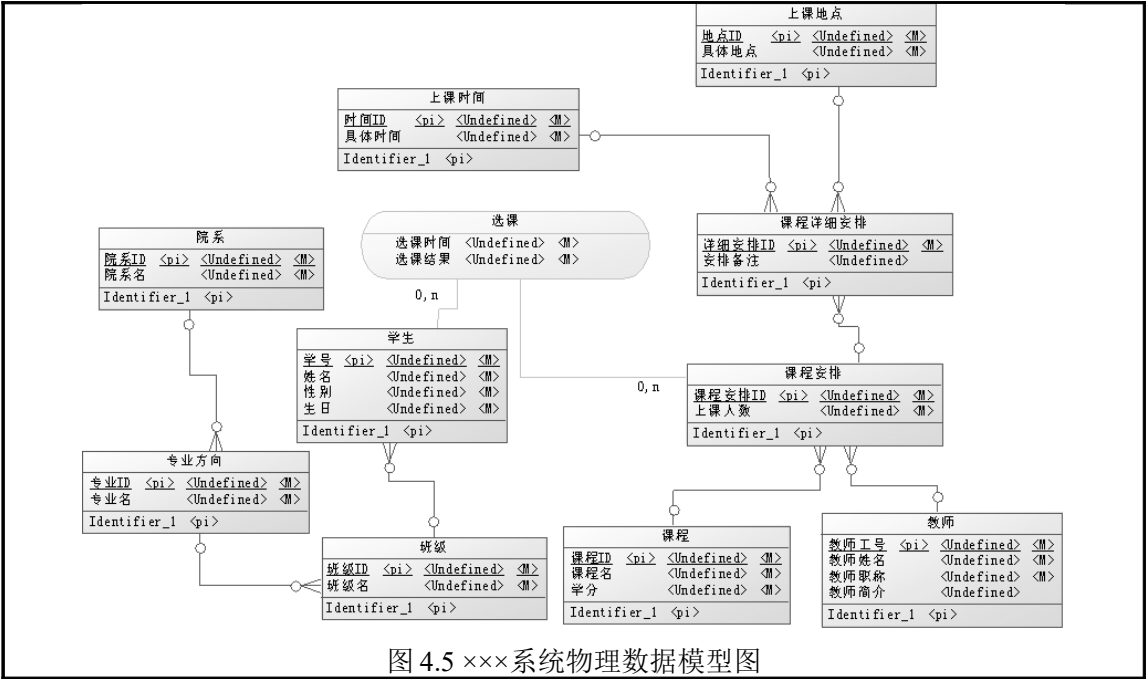


图 4.5 ×××系统物理数据模型图

4.4 ×××系统功能实现

4.4.1 ×××组件实现

系统功能实现是按照系统模块划分，基于第二章介绍的相关技术，实现各模块功能的过程。通常可采用**类图**+文字、**序列图**+文字、**算法**、**程序流程图**+文字等方式进行描述。（切忌出现太多系统界面截图）

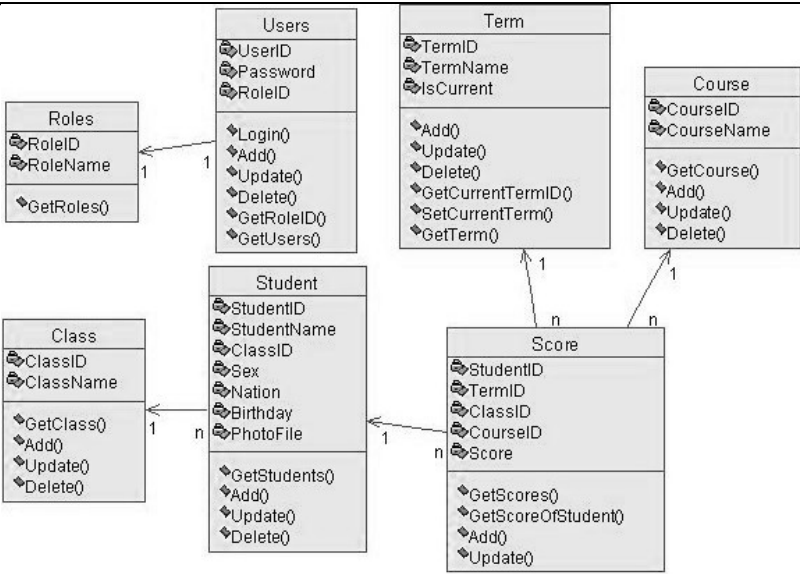


图 4.6 ×××系统类图

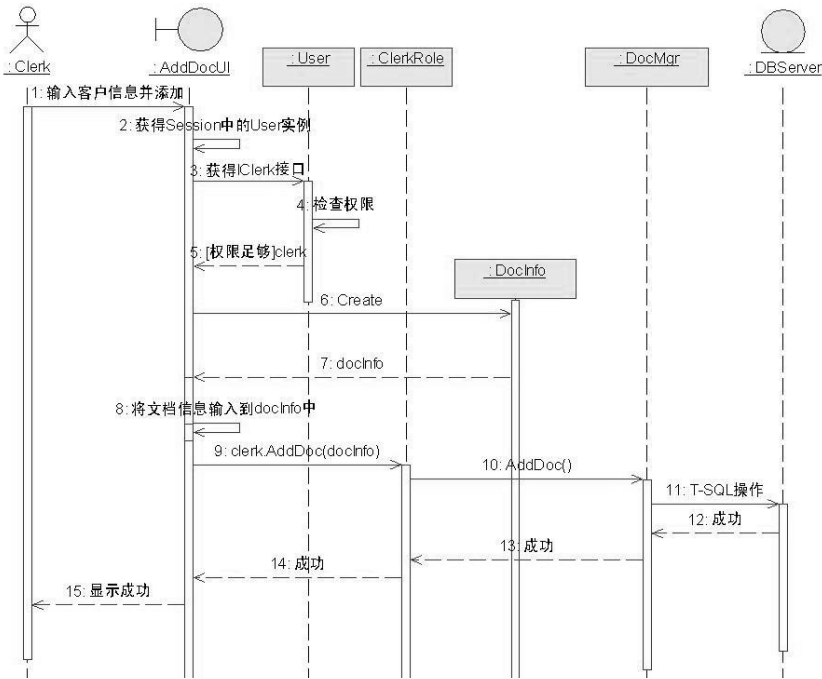
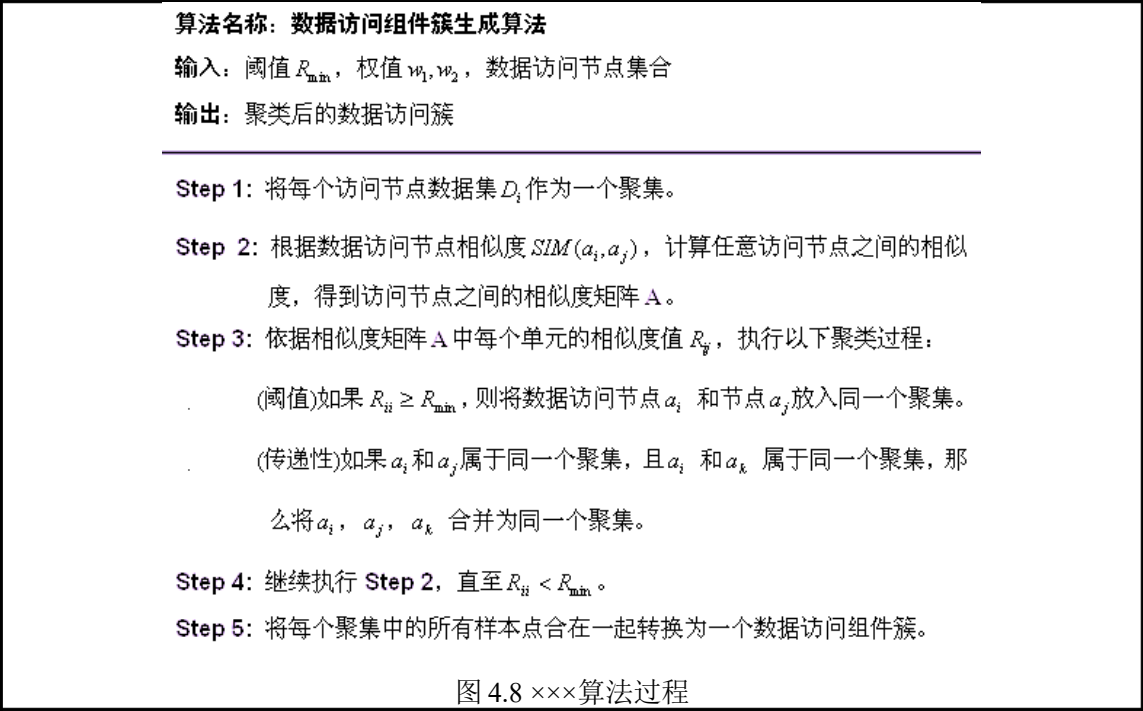


图 4.7 ×××系统顺序图



第五章 ××系统测试及分析

系统测试是在仿真或真实运行环境中，采用测试用例测试系统需求分析中所定义的功能和质量属性。系统测试的内容包括测试环境、测试用例、测试过程和测试结果分析等。

5.1 系统运行环境

系统测试平台需要真实或者模拟的运行环境支持。集中式系统通过软硬件平台描述；分布式系统在软硬件平台的基础上，还需要描述其物理分布和网络环境。通常可采用部署图+文字描述网络环境。

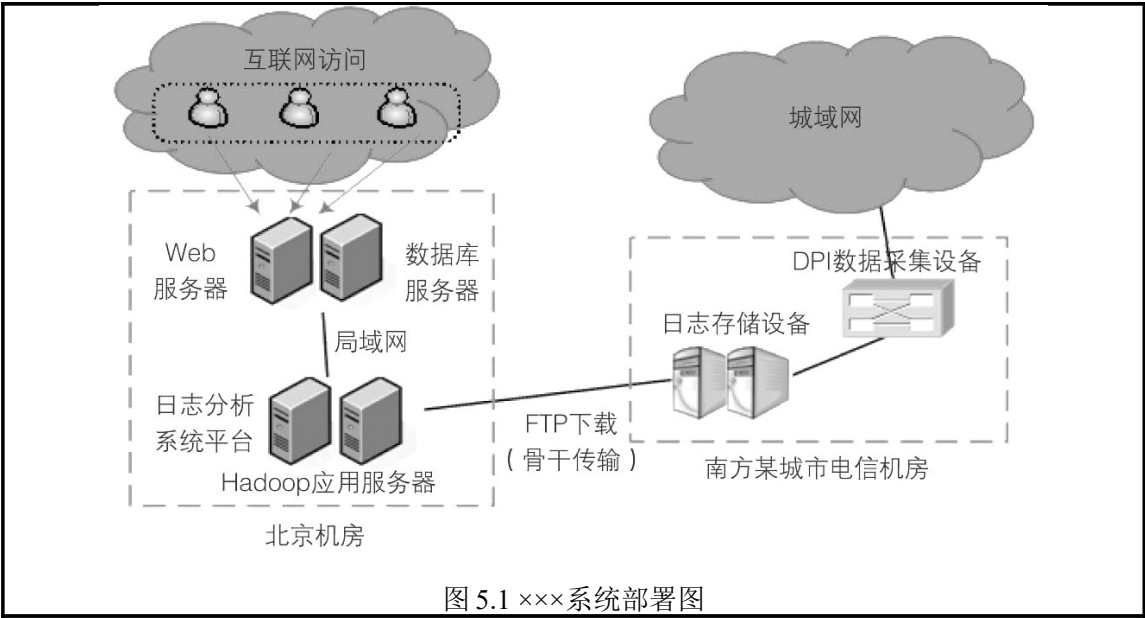


图 5.1 ×××系统部署图

5.2 测试用例及过程

测试用例是测试的大纲，通常可通过需求分析用例经过转化到得到。测试过程按照测试活动的先后顺序进行描述，通常分步骤采用文字进行描述。（切忌描述测试的概念和方法）

（1）小波包提取有效信号测试过程

这两部分在频域中距离较远，有利于使用小波包对信号进行分解和重构。本文采用 Daubechies 小波系列的 DB10 小波进行 3 层小波包变换仿真分析，用 $X(0,0)$ 代表原始信号， $X(1,0)$ 代表小波包分解的第 1 层低频系数， $X(1,1)$ 代表小波包分解的第 1 层高频系数，其他以此类推，由于保持了多分辨分析中的正交分解特性，每一个节点分解后的两个频带互不交叠，输出两个频带的带宽减半，假设在原始信号中，最低频率成分为 1，最高频率成分为 0，则第 3 层中每个节点代表的频带范围就是 $0.125^{[18]}$ 。需要注意的是，对于第 3 层按照自然数增

加顺序排列的 $X(3,i), i = 0,1,2,\cdots,7$ ，其小波包分解的频率分布不是严格按照自然数的递增排列的。为了保持主要频率随 i 的增加单调递增的性质，需要对小波包分解后的系数重新排序，以 $N_i(i = 1,2,\cdots,8)$ 来表示。表 3.3 是用 db10 小波 3 层分解后按频率重新排序的情况。

表 5.1 各频率成分所代表的频率范围

信号	频率范围	信号	频率范围
S_{30}	0~0.125	S_{34}	0.500~0.625
S_{31}	0.125~0.250	S_{35}	0.625~0.750
S_{32}	0.250~0.375	S_{36}	0.750~0.875
S_{33}	0.375~0.500	S_{37}	0.875~1.000

小波包分解结构树中第 3 层各结点所代表的频率范围为 2.5kHz。结点(3,1)，(3,2)，(3,3)，(3,4)所代表的频率范围为 2.5kHz~12.5kHz，与振动信号频率成分的频率范围相一致。因此，对结点(3,1)，(3,2)，(3,3)，(3,4)重构原信号可去除低频和低频干扰，更加真实的反映振动过程。如图 3.13 所示：

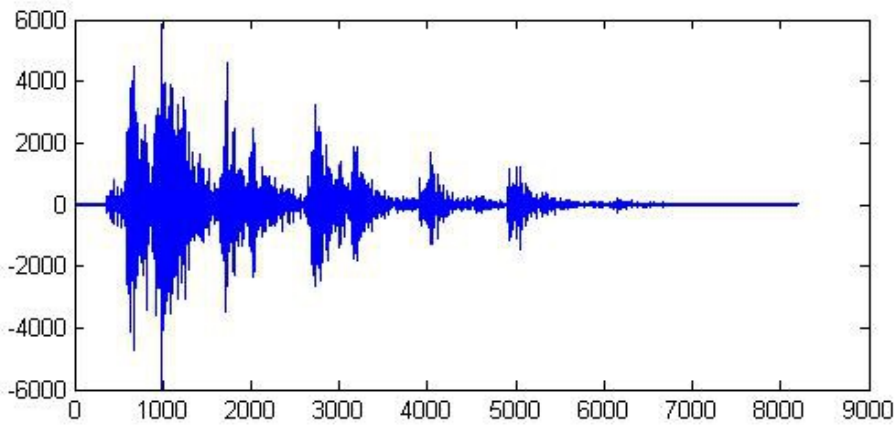


图 5.2 小波包分解滤波后的振动波形

5.3 测试结果分析

功能测试的结果能满足功能需求即可，通常可以用系统运行时截图进行描述；对于系统质量属性的测试结果需要进行分析，通常采用柱状图、折线图或数据表的形式描述测试结果。对于质量属性测试的结果进行分析、讨论，并给出能够满足系统质量要求的结论。（切忌只给出运行时截图，数据和图表必须进行分析并给出结论）

本文进行 100 组对比测试，得到 98 组有效数据，其中有 2 次测试出现明显错误，在剔除这两错误数据后得到如表 4.5 所示的分合闸起始运动时间实测数据与从动触头位移信号（行程信号）中得到分合闸起始运动时间之间平均值误差对比的测试结果：

表 4.5 大量实测数据与行程信号对比分析

98 次测试	实测数据	行程信号	平均误差
合闸起始运动时间/ms	23.54	24.14	2.49%
分闸起始运动时间/ms	19.63	20.08	2.24%

分析以上数据可以得出如下结论：在 100 次测试的有效数据中，所有测试数据的相对误差均不大于 3%，这在在线监测状态下是完全可以接受的。而在这 100 次测试期间发生 2 次明显的错误，检查监测系统并分析对应的振动信号发现，产生错误的原因是由于多次试验后振动传感器和永磁铁之间连接的松动造成的，实际使用时注意检查固定情况能有效减少这种错误的产生。

5.4 本章小结

第六章 结束语

结束语是对论文工作的总结。在给出论文工作结论的基础上，分条目介绍论文主要完成的工作，并对系统进行客观评价和分析，提出后续改进的建议。

6.1 论文工作总结

按照总分式介绍论文所完成的工作。先给出论文工作的结论，然后再按照工作内容分条给出所完成的具体工作。（切忌和摘要重复）

6.2 后续工作展望

针对系统中存在的问题，分析后续工作内容和改进方案。（切忌存在问题过多，一般2个为宜）

致 谢

对在项目实施、论文工作和学习过程中给予指导和帮助的人表示感谢。一般感谢的对象包括导师、同学朋友、家人等，对被感谢者不要直书其名，而应该冠以敬称，如某某教授、某某博士等学术头衔。（切忌天马行空，姓名和单位切勿写错，语言尽量认真、诚恳和严谨）

参考文献

【期刊论文】

[序号] 作者1, 作者2, 作者3, 等. 论文名称[J]. 期刊名称, 年, 卷(期): 页码-页码.

- [1] 宋胜利, 鲍亮, 陈平. 多层文本分类性能评价方法[J]. 系统工程与电子技术, 2010, 32(5): 216-222.

【会议论文】

[序号] 作者1, 作者2. 论文名称[C]. 会议名称. 出版地: 出版社. 年份. 页码-页码.

- [2] Shengli Song, Xiaofei Qiao, Ping Chen. Hierarchical Text Classification Incremental Learning[C]. Proceedings of the 16th International Conference on Neural Information Processing. Heidelberg: Springer-Verlag. 2009. 247-258.

【学位论文】

[序号] 作者. 论文名称[D]. 所在地: 学校, 年份.

- [3] 宋胜利. 文本语义表示及多层分类关键技术研究[D]. 西安: 西安电子科技大学, 2011.

【普通图书】

[序号] 作者. 书籍名称[M]. 版本. 出版地: 出版社, 年份. 页码-页码.

- [4] 陈平, 褚华. 软件设计师教程[M]. 第二版. 北京: 清华大学出版社, 2006. 1-24.

【电子文献】

[序号] 单位. 名称[OL]. 年份. 网址.

- [5] 中国互联网络信息中心. 2012年中国网民搜索行为研究报告[R/OL]. 2012.
http://www.cnnic.cn/gywm/xwzx/rdxw/2012nrd/201208/t20120806_32995.htm.

附录一：UML 图使用指南

UML是一种面向对象的建模语言，它是运用统一的、标准化的标记和定义实现对软件系统进行面向对象的描述和建模。UML是一种定义良好、易于表达、功能强大且普遍适用的建模语言。它融入了软件工程领域的新思想、新方法和新技术。它的作用域不限于支持面向对象的分析与设计，还支持从需求分析开始的软件开发的全过程。

附 1.1 UML2.0 图示类型

UML2.0所包含的图示：

- 类图 (Class Diagram)
- 组件图(Component diagram)
- 复合结构图(Composite structure diagram)
- 部署图(Deployment diagram)
- 对象图(Object diagram)
- 包图(Package diagram)
- 行为图(Behavior diagrams)
- 活动图(Activity diagram)
- 状态图(State diagram)
- 用例图(Use Case Diagram)
- 交互图(Interaction diagrams)
- 通信图(Communication diagram)
- 交互概述图(Interaction overview diagram) (UML 2.0)
- 顺序图(Sequence diagram)
- 时间图(UML Timing Diagram) (UML 2.0)

在学位论文写作中，常见的有9类图示，其作用包括：

- 用例图：用例图是从用户角度描述系统功能， 是用户所能观察到的系统功能的模型图，用例是系统中的一个功能单元。
- 类图：类图描述系统中类的静态结构。不仅定义系统中的类，表示类之间的联系如关联、依赖、聚合等，也包括类的内部结构(类的属性和操作)。

- 对象图：对象图是类图的实例，几乎使用与类图完全相同的标识。他们的不同点在于对象图显示类的多个对象实例，而不是实际的类。
- 顺序图：顺序图显示对象之间的动态合作关系，它强调对象之间消息发送的顺序，同时显示对象之间的交互，顺序图的一个用途是用来表示用例中的行为顺序。当执行一个用例行为时，顺序图中的每条消息对应了一个类操作或引起状态转换的触发事件。
- 通信图：通信图描述对象间的交互关系，显示对象间的动态合作关系。除显示信息交换外，通信图还显示对象以及它们之间的关系。
- 状态图：状态图是一个类对象所可能经历的所有历程的模型图。状态图由对象的各个状态和连接这些状态的转换组成。
- 活动图：活动图是状态图的一个变体，用来描述执行算法的工作流程中涉及的活动。活动图描述了一组顺序的或并发的活动。
- 组件图：组件图为系统的构件建模型—组件即构造应用的软件单元—还包括各构件之间的依赖关系，以便通过这些依赖关系来估计对系统组件的修改给系统可能带来的影响。
- 部署图：部署视图描述位于节点实例上的运行构件实例的安排。节点是一组运行资源，如计算机、设备或存储器。这个视图允许评估分配结果和资源分配。

附 1.2 UML2.0 图使用方法

UML是一种面向对象分析与设计的重要工具，常见的图示类型与面向对象分析与设计的对应关系如下表所示：

附表 1.1UML 图示与面向对象分析与设计对应关系

OOSAD 各阶段	图示类型
需求陈述	活动图、系统用例图、业务用例图
静态分析	类图、对象图
动态分析	顺序图、通信图、状态图
物理设计	组件图、部署图

附 1.3 UML2.0 图绘制方法

附 1.3.1 活动图

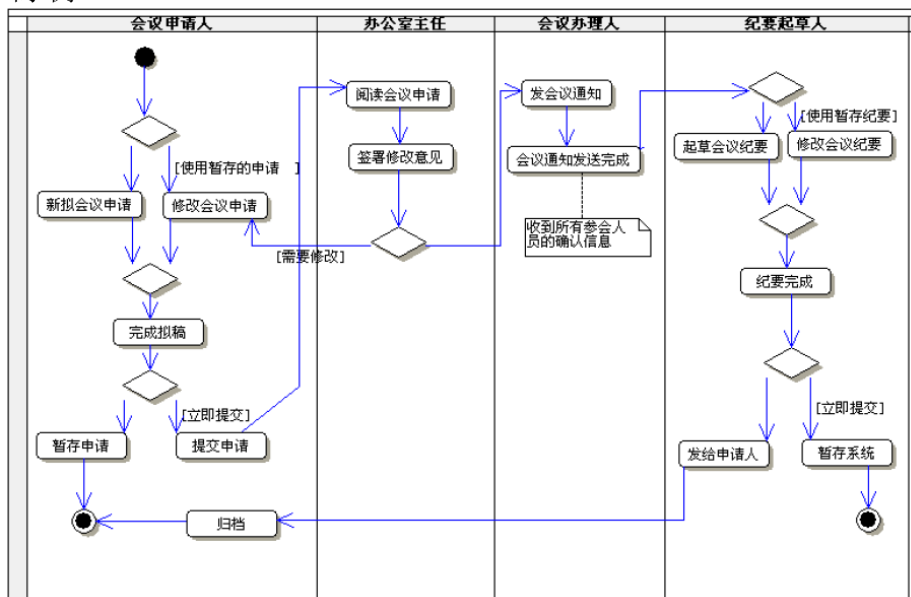
● 基本作用

活动图是用来描述一系列顺序动作、结果及其它们之间关系的图，主要用来表示系统控制流程和业务处理流程，它重点关注业务过程中的动作和结果。在软件开发中，活动图主要用来和用户交流，以辅助需求采集。有时候，可以把活动图看成一种特殊的状态图。活动可以绑定到任何建模元素，用来反映该元素的行为，该元素提供该活动的语境，活动绑定的元素有：用例、类、接口、组件、协作、操作，但在早期建模时不需要明确是哪个元素。活动图在使用时要注意目标用户，力求简单。活动图元素包括活动、转移、分支、分叉、合并、汇合、泳道、对象流。活动图的典型应用包括：用例中控制流和用例间的控制流，操作和算法的细节以及业务流程。

● 建模过程

- (1) 在采集的原始需求中选择重点流程。
- (2) 首先要确定要设计的活动图是针对业务流程还是用例。
- (3) 其次要设计活动过程的起点和终点。
- (4) 确定活动图所有执行对象。
- (5) 确定活动节点，并根据执行对象进行活动分组。
 - (a) 如果对用例建活动图，则把角色所发出的每一个动作变为活动节点。
 - (b) 如果对业务流程建活动图，则把每一个流程步骤(或片段)变为活动节点。
- (6) 确定活动节点之间转移。
- (7) 处理在活动节点之间的分支和合并。
- (8) 处理在活动节点之间的分叉和汇合
- (9) 用 UML 建模工具进行活动图建模。
- (10) 编写必要的补充文档。

● 绘图示例



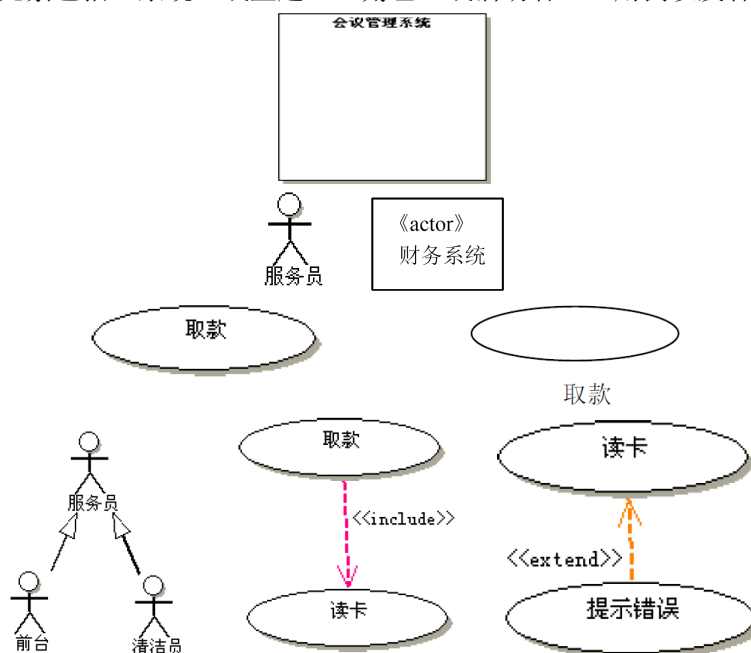
附 1.3.2 用例图

● 基本作用

用例模型主要用来描述系统和系统外部环境的关系，直接影响着其他模型。用例是一组系统使用场景的集合，每个场景又是由一些事件序列构成，发起这个事件的用户就是系统使用的参与者。用例图是系统的高层描述，角色和用例，在实现阶段则变成了对象和接口这样的底层描述。用例可以帮助每一用户知道他们未来怎样使用系统。对开发者来讲，在不关注细节的情况下，可以快速搜集系统需求，形成总体样式。

● 表示规范

用例模型元素包括：系统（或主题）、角色（或活动者）、用例以及各种关系。



● 用例描述

用例名称：起草会议申请。

参与者：会议申请人。

前置条件：会议申请人有条件通过网络访问系统并已成功地登录系统。

后置条件：系统保存一份新的会议申请。

基本事件流：

1. 用户通过网络登录后成功访问系统。
2. 用户选择会议管理后，再选择浏览会议信息。
3. 浏览结束后用户选择查看暂存会议申请。
4. 在确认无合适的会议申请后，用户选择起草会议申请。
5. 用户输入会议申请的相关信息。
6. 会议申请经过校验后提交办公室主任。

可选事件流：

1. 用户发现有可用的暂存申请可以修改，系统进入修改会议用例界面。
2. 新起草的会议申请被暂存。

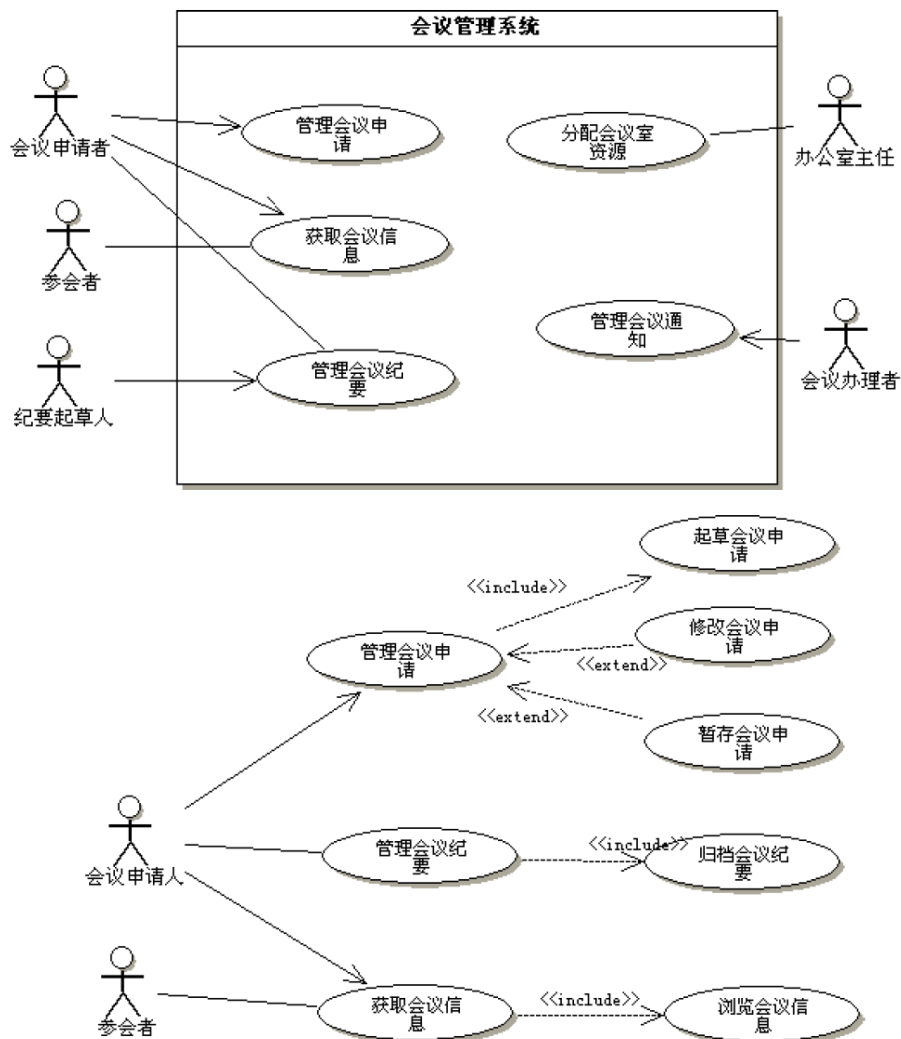
异常事件流：

1. 会议室已被预定，给出错误信息提示。
2. 会议信息校验失败，给出错误信息提示。

● 建模过程

- (1) 首先要确定将要设计的系统和它的边界。
- (2) 其次要确定系统外的活动者。
- (3) 从活动者（用户）和系统对话的角度继续寻找一下两方面的特征：
 - (a) 寻找活动者怎样使用系统。
 - (b) 系统向活动者提供什么样的功能。
- (4) 把离用户最近（接口）的用例作为顶级用例。
- (5) 对复杂的用例做进一步分解，并确定低级用例以及用例间的关系。
- (6) 对每一用例做进一步细化。
- (7) 寻找每一个用例发生的前提条件和发生后对系统产生的结果。
- (8) 寻找每一个用例在正常条件下的执行过程。
- (9) 寻找每一个用例在非正常条件下的执行过程。
- (10) 用 UML 建模工具画出用例模型图。
- (11) 编写用例模型图的补充说明文档。

● 绘图示例（系统用例和业务用例）



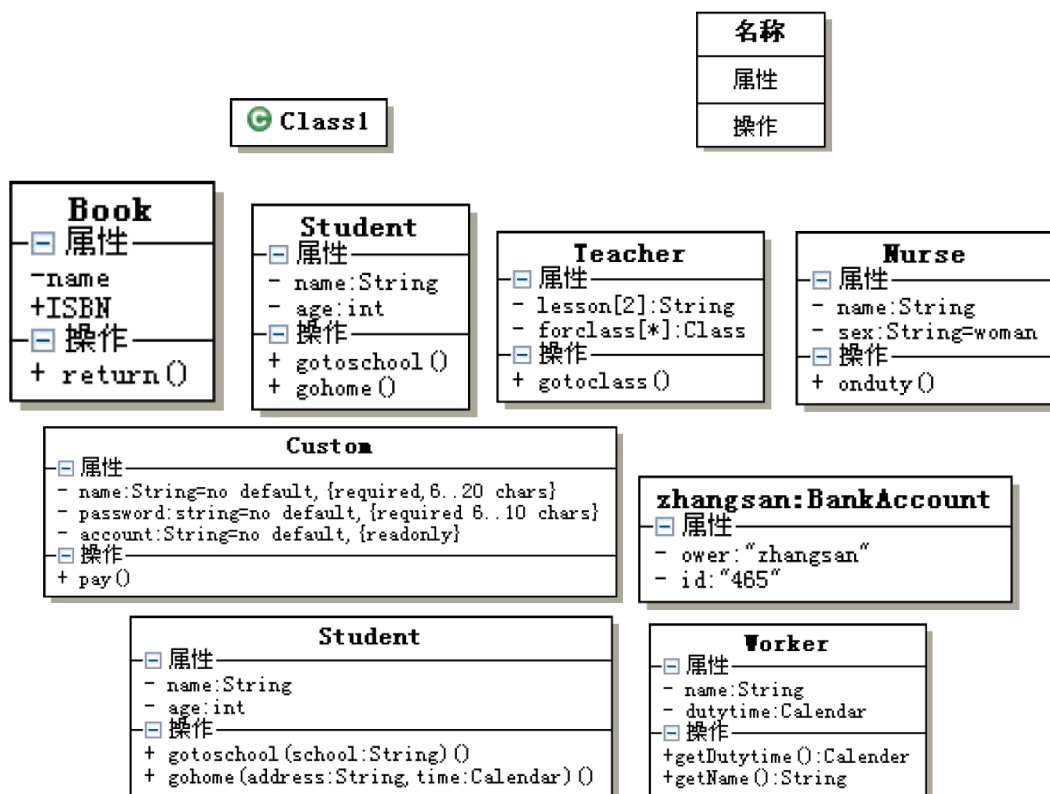
附 1.3.3 类/对象图

● 基本作用

面向对象的思想来自人们对客观世界的认识，希望把对客观世界的认识，映射到计算机世界。对象是真实世界中某一具体事务，可以是有形的，也可以是无形的。如果把真实世界的每一个对象都搬进计算机，其开发和管理的复杂程度及效率可想而知，因此，面向对象思想提出了一种类的概念。即把满足一组规则的所有对象的集合称为类。这组规则就是类属性和操作。类是一种模板，它的实例化就是对象。

● 表示规范

类图元素包括：类、可见性、操作名称和数据类型、多重性、默认值、特定字符串、对象图、参数列表、返回值、。

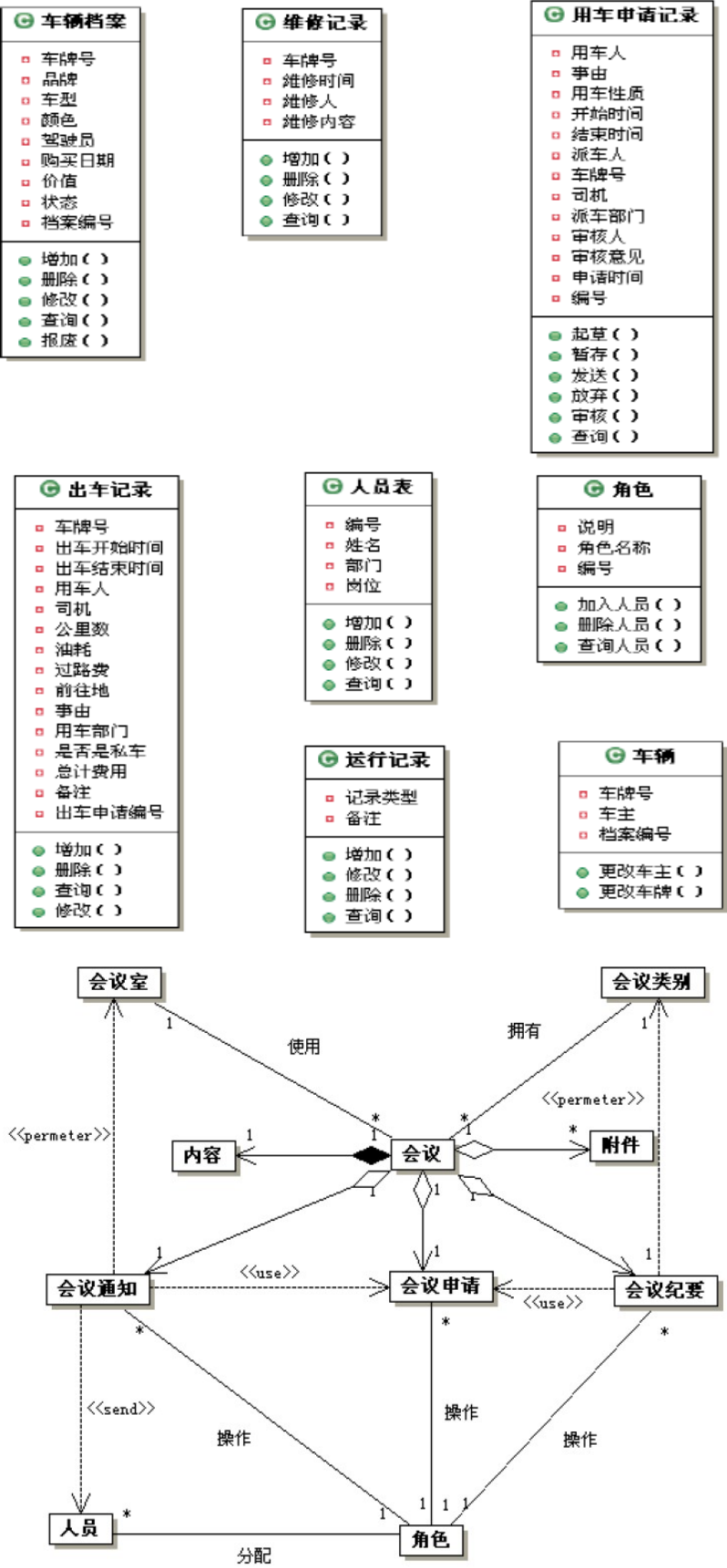


● 建模过程

- (1) 寻找出需求中的名词(候选对象)。
- (2) 合并含义相同的名词，排除范围以外的名词，并寻找隐含的名词。
- (3) 去掉只能作为类属性的名词。
- (4) 剩下的名词就是要找的分析类（候选类）。
- (5) 根据常识、问题域、系统责任确定该类有那些属性。
- (6) 补充该类动态属性，如状态、对象间联系（如聚合、关联）等属性
- (7) 从需求中的动词、功能或系统责任中寻找类的操作（候选操作）。
- (8) 从状态转换，流程跟踪、系统管理等方面补充类的操作。
- (9) 对所寻找的操作进行合并、筛选。
- (10) 对所寻找的操作在类间进行合理分配（职责分配）。形成每个类候选操作。

(11) 补充每个类的的分析文档，为类的进一步设计打下基础。

● 绘图示例



附 1.3.4 顺序图

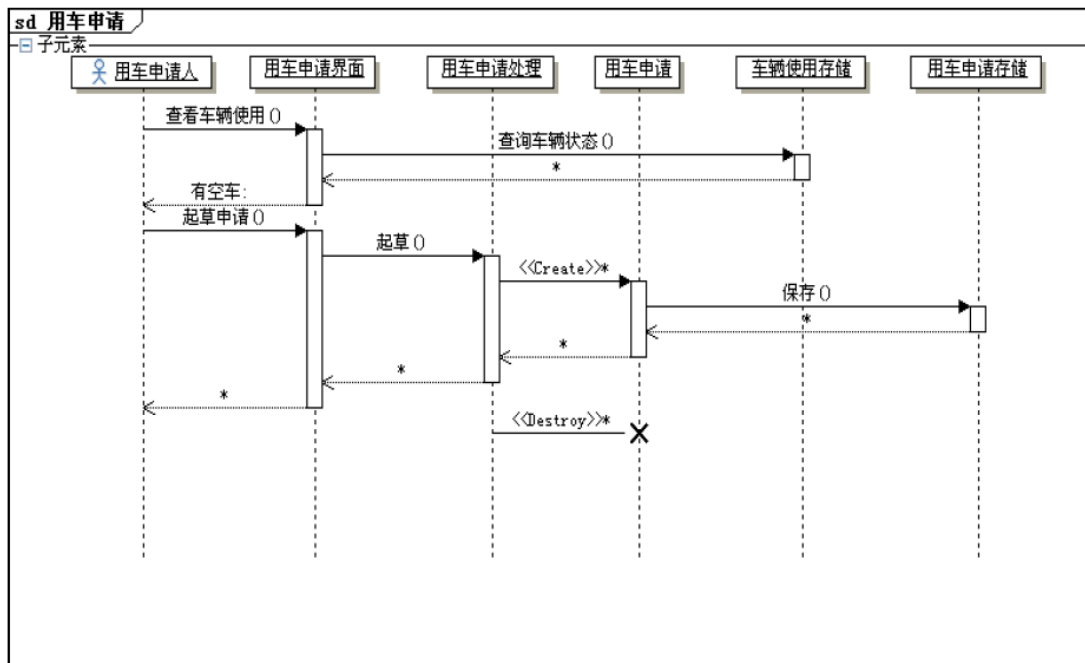
● 基本作用

显示一组对象为了实现某种功能，而彼此发送和接收的一串消息，这组对象可能是类、接口、构件和节点的具体的或原型化的实例。它是一种以时间为序的表示方法。消息的接收方是通过接口（操作）来接收消息的，消息中的数据是具体的，而不是象类中数据是定义的规则，通过顺序图，可以发现新的操作。顺序图强调消息的事件时间顺序，首先把参与交互的对象或角色放在图上方沿水平轴方向排列，对象的排列顺序从左到右依次为边界对象、控制对象、实体对象。对象的发送和接收消息沿垂直轴方向按时间顺序从上到下放置。

● 建模过程

- (1) 完成用例图的分析。
- (2) 对每个用例，识别出参与基本事件流的对象(包括接口、子系统、角色等)。
- (3) 识别出这些对象是主动对象还是被动对象。
- (4) 识别出这些对象发出的消息是同步消息还是异步消息。
- (5) 从主动对象开始向接收对象发消息。
- (6) 接收对象再调用自己的服务为主动对象返回结果。
- (7) 如果接收对象需要再调用其他对象的服务，需要向其他对象再发消息。
- (8) 如此反复，最后返回给主动对象有意义的结果。
- (9) 用 UML 建模工具绘出顺序图。
- (10) 给顺序图补充必要的说明文档。

● 绘图示例



附 1.3.5 通信图

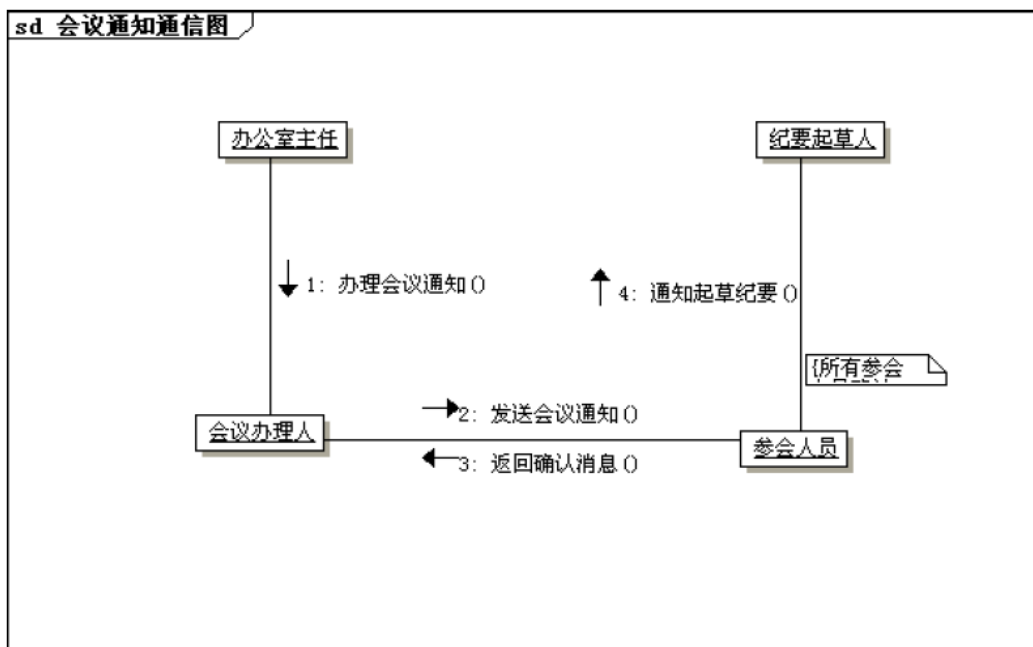
- 基本作用

在 UML2.0 中，通信图实际上是以前版本的协作图，使用通信图重点是把消息和对象之间的链直观的布局展示出来，它从空间角度反映对象之间的组织关系。通信图侧重对象之间的交互、对象的结构，有助于验证类间的关联。它同样可以表示消息的类型，如同步消息、异步消息、返回消息、丢失消息、发现消息以及对象的创建消息，但其表示方法和顺序图截然不同。

- 建模过程

- (1) 完成用例图的分析。
- (2) 对每个用例，识别出参与基本事件流的对象(包括接口、子系统、角色等)。
- (3) 识别出对象间的连接关系。
- (4) 识别出这些对象发出的消息顺序。
- (5) 从主动对象开始向接收对象发消息。
- (6) 接收对象再调用自己的服务为主动对象返回结果。
- (7) 如果接收对象需要再调用其他对象的服务，需要向其他对象再发消息。
- (8) 如此反复，最后返回给主动对象有意义的结果。
- (9) 用 UML 建模工具绘出通信图。
- (10) 给通信图补充必要的说明文档。

- 绘图示例。



附 1.3.6 状态图

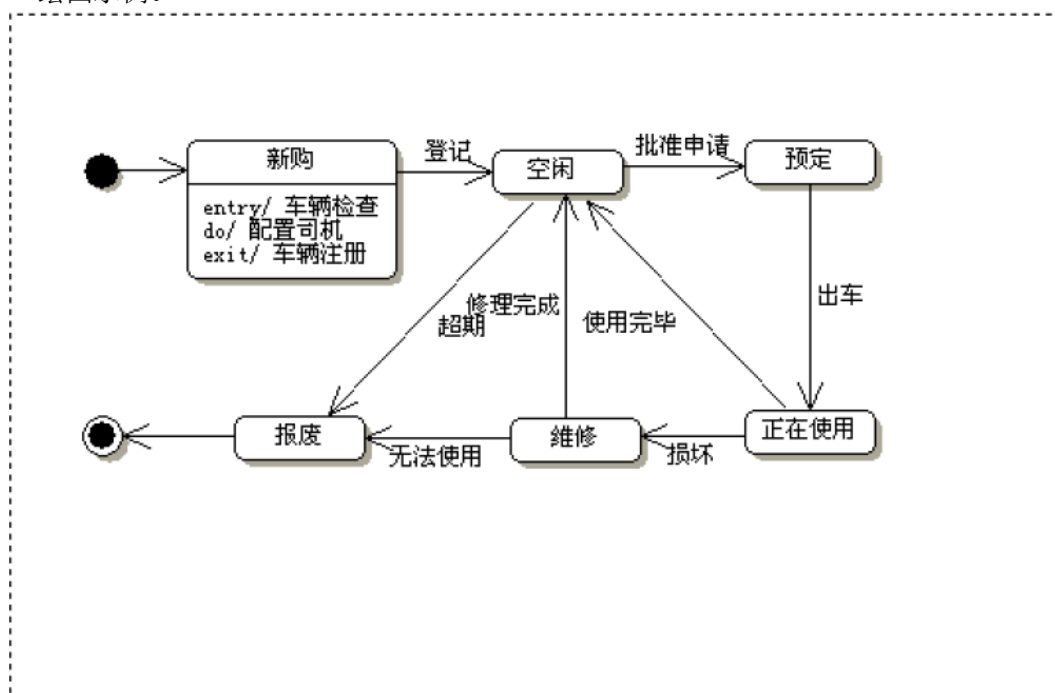
- 基本作用

状态图描述交互对对象内部的影响，交互图中的消息在这里变成外部事件对对象发出的命令，对象对这些命令的响应导致对象的状态发生变化。因此，从这个意义上说，状态图是顺序图的进一步细化，并且是对核心对象（选择核心对象的依据是看是否在多个交互图中有多个消息指向该对象）的细化。

- 建模过程

- （1）首先要确定进行系统控制的对象，可以从前面分析的顺序图中寻找。
- （2）确定对象的起始状态和结束状态。
- （3）在对象的整个生命周期寻找有意义的控制状态。
- （4）寻找状态之间的转换。
- （5）补充引起转换的事件。
- （6）UML 建模工具画状态图。
- （7）补充必要的文档。

- 绘图示例。

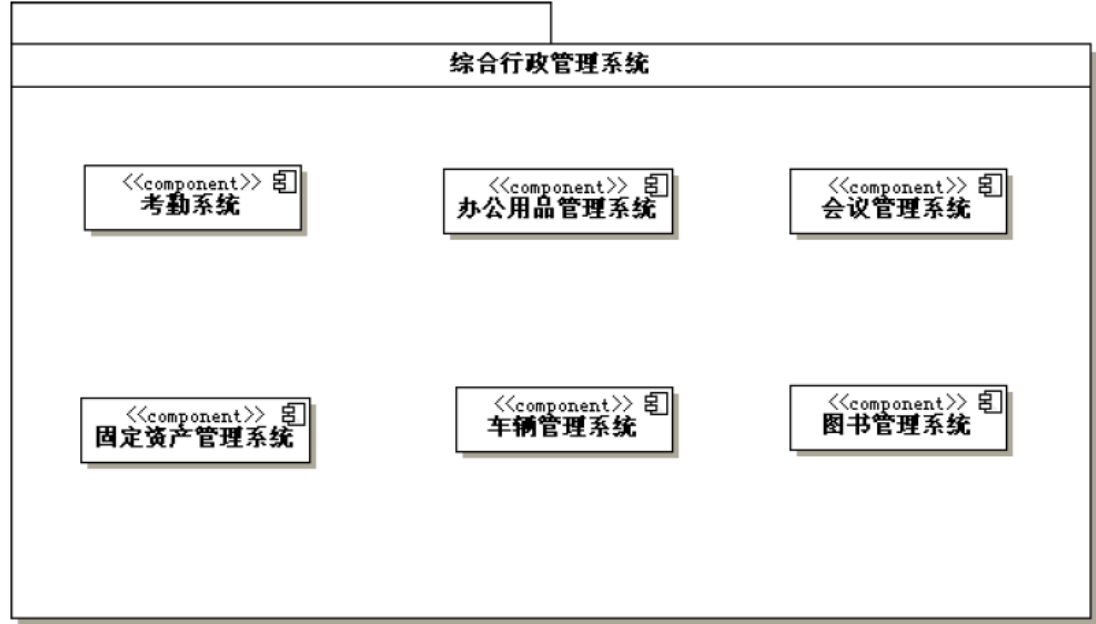


附 1.3.7 组件图

● 基本作用

组件又叫构件，代表从类到应用、子系统和系统的任何事物，组件是对物理实现类型的定义，是一个抽象的物理软件需求。每个组件定义了一个或多个和其他构件通信的接口，组件暴露了所包含类的一个或多个接口，而不是全部接口（其余用于内部通信）。一方面组件依靠自身的类指定其行为。另一方面，组件依靠多个实现它的制品。组件是一种容器，没有自己的特征，一般用组件表示一个业务过程。

● 绘图示例。



附 1.3.8 部署图

- 基本作用

部署图是展示系统实现环境的静态视图，对一个系统的管理可以从多个方面描述，每一个描述构成了一个部署视图，对整个系统的完整描述必须由多个部署视图来完成。部署图是由节点和节点之间的关联组成，节点代表处理资源的类型或执行工作的任何事物，其上的制品可以被部署和执行。节点具有属性和操作，也可参与关联，节点的属性和操作一般用注释来表示。

- 绘图示例。

