软件开发类论文撰写说明

**目 录**

[1 毕业论文的选题 1](#_Toc322463297)

[1.1 论文选题注意事项 1](#_Toc322463298)

[1.2 课题来源 2](#_Toc322463299)

[1.3 毕业设计（论文）任务书 3](#_Toc322463300)

[2 毕业论文撰写 5](#_Toc322463301)

[2.1 摘要 5](#_Toc322463302)

[2.2 绪论（引言） 6](#_Toc322463303)

[2.3 相关技术（关键技术） 8](#_Toc322463304)

[2.4 需求分析（系统分析） 8](#_Toc322463305)

[2.5 系统设计 9](#_Toc322463306)

[2.6 系统实现 9](#_Toc322463307)

[2.7 系统测试 10](#_Toc322463308)

[2.8 总结及展望 10](#_Toc322463309)

[3 附录 11](#_Toc322463310)

[3.1 用例图 11](#_Toc322463311)

[3.2 业务流程图 12](#_Toc322463312)

[3.3 活动图 14](#_Toc322463313)

[3.4 数据流图 17](#_Toc322463314)

[3.5 功能模块图 18](#_Toc322463315)

[3.6 程序流程图 20](#_Toc322463316)

毕业论文是学术论文的一种形式，为了进一步探讨和掌握毕业论文的写作规律和特点，需要对毕业论文进行分类。由于毕业论文本身的内容和性质不同，研究领域、对象、方法、表现方式不同，不同的院系，不同的专业，毕业论文通常有不同的类型。就软件学院本科而言，毕业论文通常有下面两种类型：

1. 算法设计类论文

这一类型毕业设计的特点是带有探索性，经过文献调研后，对软件工程某一领域的先进技术或成熟产品进行分析、比较，进而提出自己的评价和有针对性的创见，对某一理论问题有一定见解，产生出一个题目（课题），利用自己所学的专业知识和数学工具，得出一个（些）有用（或者有潜在的价值）的结论，并能将该创新性技术用于自己研制的系统中。这类毕业设计工作要注意把分析和实验相结合，不要只停留在消化上。消化是前提，吸收和转化才是工作的重点。

这一类毕业设计一般先根据具体条件建立一个数学模型，推导出相应的表达式，利用计算机计算出结果，然后对结果加以分析，再提出结论性意见。最好将研究成果应用到实际进行验证。

1. 软件开发类论文

这一类型的毕业设计主要依据所学的知识，完成一个相对完整的实际项目或在某一个较大的项目中设计并完成一个子系统，或者对已有的软件产品进行改进。最后以软件工程的思路，结合项目开发文档，进行归纳总结，综合扩展形成论文。

本文主要介绍软件开发型论文的撰写原则和内容指导，重点介绍如何进行毕业设计选题，论文应该包括哪些内容，每部分内容应占论文总体篇幅多大比重，以及应该如何用语言、图表来进行说明。

1 毕业论文的选题

毕业论文的选题要与学院的培养目标相联系，重在培养学生综合运用所学专业理论知识去解决实际问题的能力，使其受到科学研究的基本训练，所以选题一般不应超出专业内容的范围。毕业论文的题目应符合所学专业范围和知识结构的基本要求，毕业设计的内容要紧密结合当前国家经济发展或最新科技术情况做到既要有理论与方法的研究，又要有应用前景。

软件开发类论文选题时需要有具体的应用背景，围绕着某应用系统中的某个“复杂工程问题”设计解决方案并进行实现测试。其中，“复杂工程问题”必须具备下述特征（1），同时具备下述特征（2）-（7）的部分或全部：

（1）必须运用深入的工程原理，经过分析才可能得到解决；

（2）涉及多方面的技术、工程和其它因素，并可能相互有一定冲突；

（3）需要通过建立合适的抽象模型才能解决，在建模过程中需要体现出创造性；

（4）不是仅靠常用方法就可以完全解决的；

（5）问题中涉及的因素可能没有完全包含在专业工程实践的标准和规范中；

（6）问题相关各方利益不完全一致；

（7）具有较高的综合性，包含多个相互关联的子问题。

## 1.1 论文选题注意事项

毕业论文选题时，常见的问题有下列几种：

(1) 选题过大；

(2) 选题过难；

(3) 选题陈旧。

因此，在选题过程中，应注意以下几个问题：

(1) 选题时应对自己有正确的客观估计

客观评价自己掌握材料的深度和广度，驾驭材料的能力，对课题的理解程度等。根据自己的长处和兴趣爱好，扬长避短充分发挥主观优势。同时还要充分了解学术界的研究现状。如，本课题研究已有的成果，还存在哪些问题，尚待研究的问题，尚待解决的问题及迫切程度，社会需要和科学发展的趋势，另外，只有把主客两方面的条件结合起来，才能选出最适合自己的课题来。

(2) 课题难易要适度

选择的课题难易要适度。难度大的课题当然更有科学价值，但对刚刚涉足科学领域的大学生来说，往往力不胜任，难以完成。而难度小的课题，学生就会失去一次科学研究规范训练的机会，达不到写作毕业论文的目的。因此，课题既要有一定的难度，有一定的工作量，又要结合学生的知识水平和实际能力。

(3) 课题大小要得当

毕业论文主要是反映学生能否运用所学基础和专业知识来分析和解决本学科内某一基本问题的学术水平和运用能力。所以，毕业论题不可能囊括学习期间的全部知识，也不可能解决本学科的全部问题。一篇毕业论文只需论述某一基本问题的某一重要侧面，或是对某些基本的理论、原理有比较系统的整理等。因此，在选题时，要根据学生的专业基础和时间及其它相关因素，如资料条件、经费许可，指导力量等，综合考查以选择大小适当的课题。否则，课题过大，问题均难以研究深入，可能导致虎头蛇尾，草草收摊；题目过小，不能充分挖掘学生的潜力，发挥才能，论文达不到应有的水平和深度，也反映不出学生的实际功底和能力。

## 1.2 课题来源

软件学院鼓励学生毕业设计到企业去，承担企业真实项目。因此大部分同学应通过各种渠道（例如，导师推荐，就业实习，自己联系等方式）进入企业，融入企业真实项目中，根据前一节的选题注意事项，在导师的指导下，选定合适的论文题目。

但是由于我院毕业设计学生情况不同，论文题目不能全部来自企业。因此不同情况需不同处理。我院毕业设计学生分成以下几种情况：

(1) 本科毕业即就业

此类学生有相当比例可到就业单位实习，如若就业岗位为软件开发岗，可与就业单位沟通，根据我院选题注意事项，选定毕业论文题目。如果就业岗位非软件开发岗（如产品推广、测试或非软件行业），可根据企业实际生产或运营需求，选定软件开发项目。如若不能到就业单位实习，或实习单位不能提供符合软件学院要求的论文题目，请参考第3条。

(2) 保送学生

保送学生可参与到自己导师的科研项目中，导师通常愿意从项目中选取本科生能完成的模块，交给学生作，然后写成论文。教师熟悉项目，项目有实用背景，一般而言，多数学生经过努力都能完成。

(3) 无实习单位学生

由于学生考研、出国等原因没有找到合适的企业实习，此类学生需尽快与校内导师沟通。

① 可由校内导师推荐到企业实习，或者自己联系短期实习单位，此时课题来源于企业。

② 有些学生对一些问题有自己独到的看法和理解，知识面较宽，理论基础较深厚，学生可根据自己的科研兴趣及爱好拟定论文题目，与校内教师沟通，由校内导师认定。此时课题属学生自选题目。

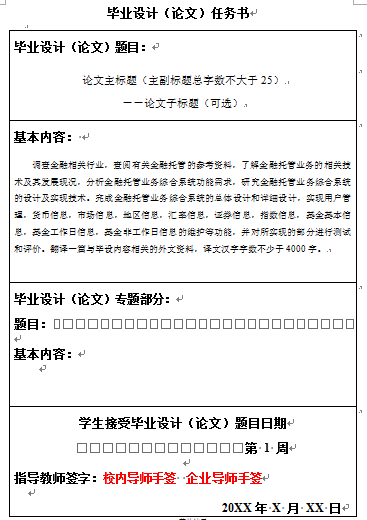
③ 校内教师根据学生能力情况，及自己科研方向拟定题目。

总之，毕业论文的题目大部分应来源于企业，其次是教师科研项目的子课题，最后才是学生自选题目。选题要尽量早些，以便有充分的时间积累材料，也有足够的时间和精力深入探讨。如果选题太晚，就会显得很仓促，无暇把问题考虑成熟并加以实现。

## 1.3 毕业设计（论文）任务书

毕业设计论文题目及任务书，应该在开学第一周内下达给毕业设计学生。对于来源于企业的论文选题，应由企业导师下达毕业设计任务书，最迟在开学第一周经由学生提交到校内导师审核；对于来源于教师科研项目的论文选题，应由校内导师下达毕业设计任务书；对于来源于学生自选的论文选题，学生应该尽快与校内导师沟通，由老师认定并下达毕业设计任务书。毕业设计（论文）任务书如图1.1所示，在《毕业设计（论文）手册》的首页，主要包括“毕业设计（论文）题目”和“基本内容”。

图1.1 毕业设计（论文）任务书



(1) 毕业设计（论文）题目

对论文题目的要求是：

① 简短精炼

**论文题目必须正确无误，且不得超过25个汉字。**

② 准确得体

论文题目应是以最恰当、最简明的词语反映论文中最重要的特定内容的逻辑组合，应避免使用不常见的缩略词、首字母缩写字、字符、代号和公式等。

③ 外延和内涵要恰如其分

毕业论文的题目应仔细推敲，尽可能从各个角度充分考虑，选择最合适的。原则上，题目要简单明了，能反应毕业论文的主要内容，使读者能一眼看出论文的中心内容要讲什么，切忌笼统、空泛。

④ 醒目

毕业论文的标题不能像小说、散文那样经过艺术加工而引起读者的好奇心。语言要补实，同时也能引起读者的注意。

(2) 基本内容

① 由于毕业设计（论文）任务书，是毕业设计实质性开始的标志，是导师下达给学生的任务说明。因此，**应以将来时的口吻书写**。

② 一般包括学生为了完成毕业设计需要学习哪些基本原理、使用哪些工具、要研究的课题以及预计实现的功能等内容。

③ 必须包括“翻译一篇与毕设内容相关的外文资料，译文汉字字数不少于4000字。”

2 毕业论文撰写

东北大学软件学院本科毕业设计（论文）的撰写有以下几点要求：

(1) 内容要正确充实；

(2) 格式要符合标准要求；

(3) 篇幅要满足要求

毕业论文的撰写不应占用全部毕业设计时间，也不应该在毕业设计实习结束，回到学校之后大概一周的时间一蹴而就。而是应该采用循序渐进，日积月累的方法。在企业实习期间，在做工程项目的间歇，分阶段、分篇章的搜集整理论文素材，待到实习结束，回到学校之后，总结整理这些论文素材，形成一篇毕业设计论文。

毕业论文的写作步骤应该首先确定论文提要材料，论文页数和字数的大致分配，拟定论文的写作提纲，进而对论文提纲进行细化和扩展，形成论文初稿。通过与指导教师沟通讨论，对初稿进行修订，包括观点的订正、深化，材料的增、删、改、换，结构及语言的修订，最后形成一篇合格的本科毕业设计论文。

**我院软件开发类的论文一般包含摘要、绪论、相关技术（关键技术）、需求（系统）分析、系统设计、系统实现、测试、总结和展望等几个章节，论文正文篇幅应在40页以上，论文总篇幅尽量满足50页以上**。下面就逐一介绍各章节应该论述的内容、撰写原则，及占总篇幅的比例。

## 2.1 摘要

(1) 摘要的含义

摘要是毕业论文的高度概括和总结，是一篇完整的短文，是毕业论文的内容不加注释和评论的简短陈述。摘要应具有独立性、完整性和自含性，让读者尽快了解论文的主要内容，以补充题名的不足，即不阅读全文就能获得必要的信息。另外，摘要（特别是关键词）也可为科技情报文献检索数据库的建设和维护提供方便。

(2) 摘要的内容

摘要的撰写一般分三段式来阐述：

① 第一段描述毕业设计从事的研究或开发课题的目的和重要性，简短的几句话足以。

② 第二段是摘要的主体部分，描述毕业论文所阐述的研究内容、研究方法和研究成果。

③ 第三段描述论文所阐述的研究（开发）成果，对现实社会造成的影响、意义。

(3) 关键词的选取

关键词是从论文的题名、摘要和正文中选取出来的，是对表述论文的中心内容有实质意义的词汇。关键词选用是否合适，关系到该论文被检索的概率和该成果的利用率。

(4) 摘要注意事项

摘要的篇幅不易过长，最好在500~700字左右，以英文译文一页之内为宜。撰写摘要时，应注意以下几点：

① 不要把应在引言（绪论）中出现的内容写入摘要。

② 一般不要对论文内容作诠释和评论（尤其是自我评论）。

③ 要用第三人称。

④ 除了实在无法变通以外，一般不用数学公式，不出现插图、表格。

⑤ 缩略词、略称、代号，除了相邻专业的读者也能清楚理解的之外，在首次出现时必须加以说明。

(5) 摘要的翻译

一般而言，英文摘要应是中文摘要的转译，所以只要简洁、准确地逐段将文章译出即可，时态常用一般现在时间、一般过去时，少用或不用现在完成时、过去完成时、进行时态和其他复合时态。尽量使用短句，但也要避免单调和重复。

## 2.2 绪论（引言）

**绪论的篇幅不应超过总论文的10%，即3~5页即可。**应言简意赅，不要与摘要雷同。一般教科书中有的知识，在绪论中不必出现。

在绪论中，首先要阐明选题的实际背景和解决该问题的现实意义和重要作用等，结合问题背景的阐述，使读者感到此选题确实具有实用价值和学术价值，有研发和开发的必要性。

其次，应简述本课题在国内外的研究和发展状况；本课题研究的指导思想、欲解决的主要问题以及解决此课题所需要的条件；也可适当简要地介绍一些与本课题有关的预备知识。

接着，应说明本选题的来源、目的、范围及应达到的技术要求。若属子课题，在引言中还应对主客体的全貌加以介绍，说明本人的工作内容以及在整个课题中所起的作用和关系。

最后，介绍一下论文的组织结构。

本科毕业设计论文**绪论的内容应包括以下小节**：

1.1课题研究背景

阐述选题的理由。

1.2课题研究意义

1.3国内外现状

对本课题现有的研究进展情况的简要介绍。

1.4论文研究内容

本文所要解决的问题，采用的手段、方法和步骤，所需要的条件，预期成果。

1.5论文的组织结构

**例如：**本论文结构安排如下：

第1章，绪论。介绍了课题的研究背景、意义、国内外研究现状、发展特点和趋势，论文的组织结构。

第2章，相关技术。简要介绍了高压发生器的系统构成、cpu单元结构及uclinux嵌入式操作系统简介及驱动程序开发概述。

第3章，需求分析。通过用例的方式对高压发生器的控制软件进行需求分析[10]，包括功能性需求分析和非功能性需求分析，进而得出高压发生器的用例模型。

第4章，系统设计。进行软件及架构设计[7]，对软件进行分层和模块划分。将软件分为硬件接口层、驱动程序层和应用程序层；将软件划分为硬件接口模块、控制模块、算法模块和数据模块。

第5章，系统实现。实现了高压控制软件，给出硬件接口层模块、驱动程序层各驱动程序、应用层各模块的具体实现。

第6章，系统测试。对高压基本功能编写测试用例[16]，进行测试，得到相关波形。

第7章，总结与展望。对工作做了简要的总结，并对后续工作提出了设想。

## 2.3 相关技术（关键技术）

相关技术或关键技术章节不是论文组织所必需的章节，如果论文中所涉及的相关理论或关键技术不是很常见，对后续论文的理解需要该知识，则有必要在相关理论或关键技术这一章节进行简要介绍。**其篇幅不应超过总论文的15%，即5~8页即可**。

相关技术或关键技术主要包括两部分内容：

(1) 与课题内容相关的理论知识。

(2) 课题项目开发用到的关键技术介绍，包括开发方法、开发工具和环境等方面的内容。

**最后写一段本章小结**，最好不要少于3行。

## 2.4 需求分析（系统分析）

需求分析（系统分析）是论文组织所必需的章节。**其篇幅应占总论文的15%左右。**

如果论文所研究的课题属于一个相对较独立的完整课题，则应该进行较完整的需求分析。如果论文所研究的课题属于某个较大项目的子课题，则应该首先进行系统分析，对主课题的全貌加以介绍分析，说明本人的工作内容以及在整个课题中所起的作用和关系。然后重点对子课题进行较全面的需求分析。

软件需求包括三个不同的层次：业务需求、用户需求和功能需求。其中，业务需求和用户需求中包括功能性和非功能性需求。

需求分析建模的方法有多种，包括结构化方法、面向对象模型等。建模工具主要有四种，用例图、活动图、业务流程图、数据流图。它们之间的关系如表2.1所示。

表2.1 需求分析建模方法

|  |  |
| --- | --- |
| 结构化方法 | 面向对象方法 |
| 业务流程图 | 用例图 |
| 数据流图 | 活动图 |

(1) 功能分析可从静态和动态两种角度来进行分析，静态分析过程中可使用“**用例图**”来进行描述，从业务的角度进行动态分析可使用“**业务流程图**”或“**活动图**”来描述，从数据的角度进行动态分析可使用“**数据流图**”来描述。附录中逐一介绍了这几种分析工具。

(2) 非功能性需求分析可包括可行性分析（技术可行性、经济可行性等）、安全性分析、性能效率分析等等。

**最后写一段本章小结**，最好不要少于3行。

## 2.5 系统设计

系统设计是论文组织所必需的章节。**其篇幅应占总论文的30%以上。**

系统设计也可采用结构化设计方法和面向对象的设计方法。

(1) 传统的结构化设计方法主要有3种：

① 功能模块划分设计；② 面向数据流的设计；③ 输入输出设计。

(2) 面向对象的设计方法主要有对象设计和动态模型设计。

① 对象设计：对象模型描述系统的静态结构，由**类图**和**对象图**表示，包括构成系统的类和对象、它们的属性和操作，以及它们之间的联系。

② 动态模型设计：动态模型确定对象的可能事件的顺序。动态模型由**状态图、时序图、协作图**等表示。

系统设计可分为概要（总体）设计和详细设计。

(1) 概要（总体）设计可包括体系结构设计和模块设计。体系结构设计一般需论述采用何种架构（B/S或C/S架构），何种设计模式和框架（MVC等），何种开发环境及工具等等。

模块设计一般应画出总体功能模块图。（参考附录功能模块图画法）

(2) 详细设计可分模块的详细介绍设计过程，包括算法设计、功能设计，数据库设计，接口设计、协议设计、界面设计等。可通过“**时序图**”、“**类图**”、“**E-R图**”及**数据库表**等设计工具进行更直观的阐述。

**最后写一段本章小结**，最好不要少于3行。

**需要注意的是**，论文中的图表只是对文字论述的一个补充说明，不能是论文的主体，论文还应是以论述为主，图表为辅，所以不能连续页面都是图表。

## 2.6 系统实现

系统实现是论文组织所必需的章节。**其篇幅应占总论文的20%以上。**实现这一章节的安排应与设计一章相呼应，设计一章出现了哪几个模块的设计，在实现一章应有相应的模块实现。每一模块的实现可通过“**程序流程图**”、“**代码**”和“**界面**”更加直观的论述。

程序流程图要规范，有“开始”，有“结束”，分支要使用“菱形框”，要有分支条件，跳转箭头要指向流程线，而不能指向“执行框”。具体见附录图3.15所示。

代码要选取关键代码或伪代码，篇幅不可太长，连续代码不应超过一页。

界面图不应太多，几个足以。

**最后要写一段本章小结**，最好不要少于3行。

## 2.7 系统测试

测试是论文组织所必需的章节。**其篇幅应占总论文的7%左右。**

测试一章的内容没有必要大段大段摘抄测试原则、测试方法等，应重点描述测试方案、测试用例和测试结果，**最后给出测试结论或评价**。

## 2.8 总结及展望

总结是对整个研究工作进行归纳和综合而得出的总结，对所得结果与已有结果的比较和课题尚存在的问题，以及进一步开展研究的见解与建议。总结要写得概括、正确、完整、明确、精炼。

总结不是个人总结，不是自己在毕业设计期间的流水帐，在总结中要以整个研究工作为主体，进行阐述相关的问题。

总结与摘要不同：总结主要是对课题研究内容进行总结，摘要是对论文本身进行概括。

3 附录

## 3.1 用例图

用例图主要用来描述客户的需求，即用户希望系统具备的完成一定功能的动作，通俗地理解用例就是软件的功能模块，所以是系统分析阶段的起点，设计人员根据客户的需求来创建和解释用例图，用来描述软件应具备哪些功能模块以及这些模块之间的调用关系，用例图包含了用例（Use Case）和参与者（Actor），用例之间用关联来连接以求把系统的整个结构和功能反映给非技术人员（通常是软件的用户），对应的是软件的结构和功能分解。

用例是从系统外部可见的行为，是系统为某一个或几个参与者（Actor）提供的一段完整的服务。从原则上来讲，用例之间都是独立、并列的，它们之间并不存在着包含从属关系。但是为了体现一些用例之间的业务关系，提高可维护性和一致性，用例之间可以抽象出包含(include)、扩展(extend)和泛化(generalization)几种关系。

(1) 包含(include)关系：虽然每个用例的实例都是独立的，但是一个用例可以用其它的更简单的用例来描述。这有点像通过继承父类并增加附加描述来定义一个类。一个用例可以简单地包含其它用例具有的行为，并把它所包含的用例行为作为自身行为的一部分，这被称作包含关系。如图3.1所示。

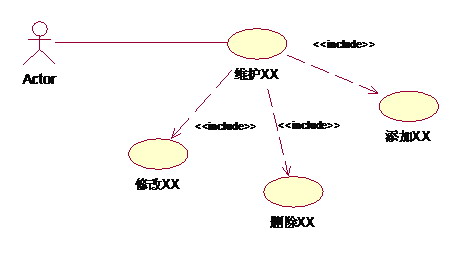
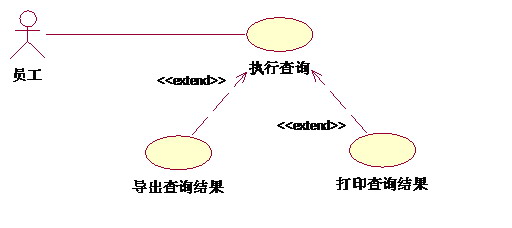


图3.1 用例包含关系示例图

(2) 扩展(extend)关系：一个用例也可以被定义为基础用例的增量扩展，这被称作扩展关系，扩展关系是把新的行为插入到已有的用例中的方法。同一个基础用例的几个扩展用例可以在一起应用。基础用例的扩展增加了原有的语义，此时基础用例而不是扩展用例被作为例子使用。在UML中，扩展关系表示为虚线箭头加<<extend>>字样，箭头指向被扩展展的用例。如图3.2所示。

图3.2 用例扩展关系示例图



(3) 泛化(generalization)关系：子用例和父用例相似，但表现出更特别的行为；子用例将继承父用例的所有结构、行为和关系。子用例可以使用父用例的一段行为，也可以重载它。父用例通常是抽象的。在实际应用中很少使用泛化关系，子用例中的特殊行为都可以作为父用例中的备选流存在。例如，业务中可能存在许多需要部门领导审批的事情，但是领导审批的流程是很相似的，这时可以做成泛化关系表示。如图3.3所示。

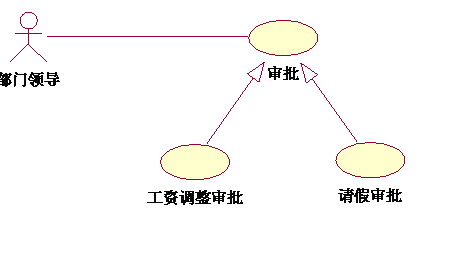


图3.3 用例泛化关系示例图

## 3.2 业务流程图

业务流程图是一种描述系统内各单位、人员之间业务关系、作业顺序和管理信息流向的图表，利用它可以帮助分析人员找出业务流程中的不合理流向，它是物理模型。业务流程图主要是描述业务走向，例如病人，首先要去挂号，然后在到医生那里看病开药，然后再到药房领药，然后回家。业务流程图描述的是完整的业务流程，以业务处理过程为中心，一般没有数据的概念。

业务流程图的绘制是按照业务的实际处理步骤和过程进行的。业务流程图是一种系统分析人员都懂的共同语言，用来描述系统组织结构、业务流程。

(1) 业务流程图的基本符号及含义如图3.4所示。

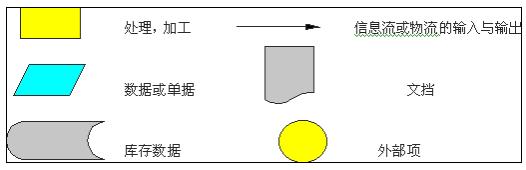


图3.4 业务流程图的基本符号和含义说明

(2) 画业务流程图的步骤与例子

① 现行系统业务流程总结

在画业务流程图之前，要对现行系统进行详细调查，并写出现行系统业务流程总结。例如，开发人员在系统调查阶段了解到某[企业](http://wiki.mbalib.com/wiki/企业" \o "企业)的[会计核算形式](http://wiki.mbalib.com/wiki/会计核算形式" \o "会计核算形式)是[科目汇总表](http://wiki.mbalib.com/wiki/科目汇总表" \o "科目汇总表)的核算形式，其帐务处理业务流程如下：

* 根据审核无误的[原始凭证汇总表](http://wiki.mbalib.com/wiki/原始凭证汇总表" \o "原始凭证汇总表)编制[记帐凭证](http://wiki.mbalib.com/wiki/记帐凭证" \o "记帐凭证)，包括[现金](http://wiki.mbalib.com/wiki/现金" \o "现金)收付、[银行](http://wiki.mbalib.com/wiki/银行" \o "银行)收付、转帐凭证。
* 根据现金收付款凭证登记现金[日记帐](http://wiki.mbalib.com/wiki/日记帐" \o "日记帐)。
* 根据银行收付款凭证登记[银行存款](http://wiki.mbalib.com/wiki/银行存款" \o "银行存款)日记帐。
* 根据银行送来的对帐单对银行存款日记帐核对。
* 根据记帐凭证及所付[原始凭证](http://wiki.mbalib.com/wiki/原始凭证" \o "原始凭证)登记有关[明细帐](http://wiki.mbalib.com/wiki/明细帐" \o "明细帐)。
* 根据记帐凭证，按相同的借贷方汇总出[科目汇总表](http://wiki.mbalib.com/wiki/科目汇总表" \o "科目汇总表)。
* 根据科目汇总表登记汇[总分类帐](http://wiki.mbalib.com/wiki/总分类帐" \o "总分类帐)。
* 将明细帐科目余额与财产物资实用数核对。
* 把总分类帐余额与有关明细帐余额核对。
* 根据总帐、明细帐余额编制各种[会计报表](http://wiki.mbalib.com/wiki/会计报表" \o "会计报表)。

② 业务流程图的绘制

根据上述业务流程可以绘制出该企业帐务处理业务流程图，如图3.5所示。

(3) 业务流程图的特点

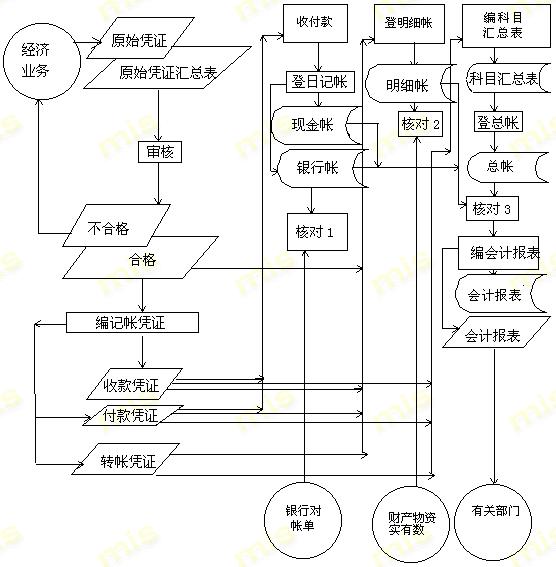
① 图的形式是按业务[部门划分](http://wiki.mbalib.com/wiki/部门划分" \o "部门划分)的横式图。

② 图描述的主体是[票据](http://wiki.mbalib.com/wiki/票据" \o "票据)、帐单的业务处理。

③ 票据、帐单流动路线与实际业务处理过程一一对应。

④ 图中票据、帐单是有“ 生”、“ 死”的，即用它的一次生命周期来表示出一笔业务的处理情况。

图3.5 帐务处理现行系统业务流程图



(4) 业务流程图的作用

① 制做[流程图](http://wiki.mbalib.com/wiki/流程图" \o "流程图)的过程是全面了解业务处理的过程，是进行系统分析的依据。

② 它是系统分析员、管理人员、业务操作人员相互交流思想的工具。

③ 系统分析员可直接在业务流程图上拟出可以实现计算机处理的部分。

④ 用它可分析出业务流程的合理性。

## 3.3 活动图

活动图本质上是一种流程图，每个“活动”可以是某个具体的“事务”，比如审核单据等。用不太严谨的表述来说，活动图是一种粒度比较粗的事件流程图，多在需求阶段使用。

**(1) 基本活动图**

一个活动图可能包括以下元素：

① 活动状态：表示在工作流程中执行某个活动或步骤。

② 转移：表示各种活动状态的先后顺序。

③ 决策与警戒条件：显示业务用例的工作流程中的备选线程。

④ 同步示意条：用于显示平行分支流，能够显示业务用例的工作流程中的并行线程。如图3.6 所示。

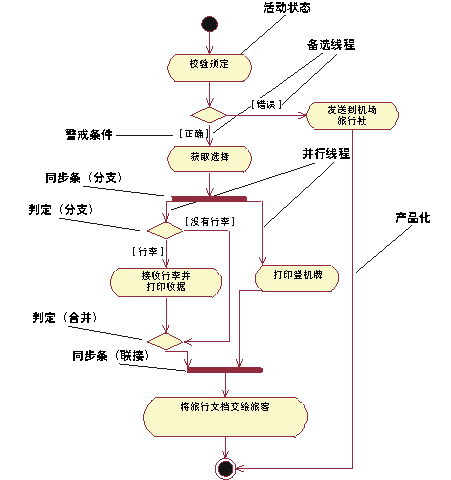


图3.6 “机场登记”业务用例模型中“个人登记”业务用例的活动图

**(2)** **条件线程**

警戒条件用于说明一组并行线程中的某个线程是有条件的。例如，在上面的“个人登记”示例中，进行登记的乘客可能是频繁乘机旅行的顾客。在此情况下，您需要给他奖励一些飞行里程数。如图3.7所示。

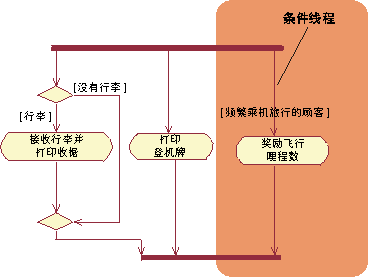


图3.7 “机场登记”业务用例模型中“个人登记”业务用例的活动图

(3) 嵌套活动图

一个活动状态可能要引用另一个活动图，因为后者显示了前者的内部结构，称为嵌套活动图。可以显示活动状态中的子图或是让活动状态引用另一个图。如图3.8所示。

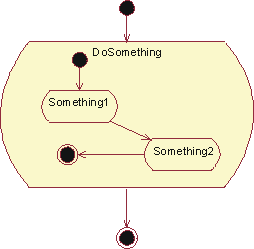


图3.8 活动状态中嵌套的活动图

(4) 使用泳道

可以使用垂直实线将活动图划分为**泳道**。每条泳道代表整个工作流程的某个部分的职责，该职责由组织的某个部门来执行。泳道最终可以由组织单元或者业务对象模型中的一组类来实施。泳道之间的排序并不会影响语义。每个活动状态都指派了一条泳道，而转移则可能跨越数条泳道。如图3.9所示。

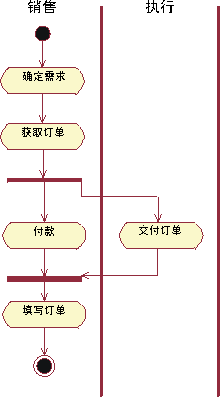


图3.9 泳道活动图

## 3.4 数据流图

系统分析阶段必须进行全面准确的收集、整理、分析收集的数据及其流程。

数据流图(Data Flow Diagram，简称DFD)是描述系统逻辑模型的工具之一——只反映信息在系统中流动和处理情况的图。它能精确地在逻辑上描述系统的功能、输入、输出和数据存贮等，而摆脱了其物理内容。

(1) 数据收集

从客户的现行组织机构、现行系统的业务流程、现行的决策方式、各种报表、报告、图示等各种渠道进行数据收集。通过查阅档案，面谈调查，发调查表，测定，采样等多种方法进行数据收集。

(2) 数据分析

收集上来的数据是“原材料”，其中有些数据不能用作系统设计的依据，要把这些原材料加工成系统设计可用的资料，就必须做数据的分析工作。

(3) 数据流图（DFD）

将收集、整理、分析的数据用数据流图的形式进行描述。

① 数据流图的基本符号

数据流图由四种基本符号组成，如图3.10所示。

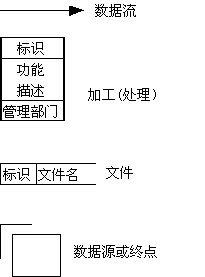


图3.10 数据流图的基本符号

② 简单数据流图举例

图3.11是一个简单的数据流图，它表示数据X从源S流出，经P加工转换成Y，接着经P加工转换为Z，在加工过程中从F中读取数据。

|  |
| --- |
|  |

图3.11 简单数据流图举例

③ DFD的画法

对于不同的问题，数据流图可以有不同的画法。一般情况下，应该遵守“由外向里”的原则。即先确定系统的边界或范围，再考虑系统的内部，先画加工的输入和输出，再画加工内部。具体实行时可按下述步骤进行：

* 识别系统的输入和输出，画出顶层图。
* 画系统内部的数据流、加工与文件，画出一级细化图。
* 加工的进一步分解，画出二级细化图。

其它注意事项：

* 数据流的命名，名字要确切，能反映整体。
* 从左至右画数据流图。各种符号布置要合理，分布均匀，尽量避免交叉线。
* 先考虑稳定态，后考虑瞬间态。如系统启动后在正常工作状态，稍后再考虑系统的启动和终止状态。

(4) 数据字典(Data Dictionary,DD)

在数据流图的基础上，还需对其中的每个数据流、文件和数据项加以定义， 我们把这些定义所组成的集合称为数据字典。数据流图是系统的大框架，而数据字典以及加工说明则是对数据流图中每个成分的精确描述。它们有着密切的联系，必须结合使用。

## 3.5 功能模块图

所谓功能模块图就是将系统的功能进行分解，按功能从属关系画成的图表。管理信息系统的各子系统可以看作是系统目标下层的功能，对其中每项功能还可以继续分解为第三层、第四层……甚至更多的功能。图中的每一个框都称为一个功能模块。功能模块可以根据具体情况分的大一点或小一点，分解得最小功能模块可以是一个程序中的每个处理过程，而较大的功能模块则可能是完成某一个任务的一组程序。

画出功能模块图是系统设计过程中的一个重要环节，它将显示出工程所要实现的各种功能，并分类，然后软件工程师根据功能模块图来具体实现这些功能（通过代码）

对于系统概要设计可以画出总体功能模块图如图3.12所示。



图3.12 总体功能模块图

**注意：**模块图上层与下一层只有包含关系没有流程关系，所以要使用无箭头的线条。

在详细设计中对其中的某一模块再进行细化，如图3.13所示。



图3.13 系统设置功能模块图

如果模块细化不多，也可以总体画出，如图3.14所示。

图3.14 总体功能模块图



## 3.6 程序流程图

在系统实现章节，可以通过程序流程图来说明某个function的实现，如图3.15所示。

图3.15 流程图示例图



**注意：**程序流程图要规范，有“开始”，有“结束”，分支要使用“菱形框”，要有分支条件，跳转箭头要指向流程线，而不能指向“执行框”。