概要设计

概要设计是一个设计师根据用户交互过程和用户需求来形成交互框架和视觉框架的过程，其结果往往以反映交互控件布置、界面元素分组以及界面整体板式的页面框架图的形式来呈现。这是一个在用户研究和设计之间架起桥梁，使用户研究和设计无缝结合，将对用户目标与需求转换成具体界面设计解决方案的重要阶段。[1]

概要设计的主要任务是把需求分析得到的系统扩展用例图转换为软件结构和数据结构。设计软件结构的具体任务是：将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面等。数据结构设计包括数据特征的描述、确定数据的结构特性、以及数据库的设计。显然，概要设计建立的是目标系统的逻辑模型，与计算机无关。

1引言

1.1编写目的

指出目的和预期读者

1.2背景

a. 待开发系统名称；

b. 列出项目任务提出者、开发者、用户及运营和运营站点（计算机中心、数据中心、网络服务器、云服务器）

1.3定义

定义文件中用到的专门术语和外文字母组成的原词组

1.4参考资料

包括但不限于：

1. 经核准的项目计划书或合同，及上级批文；
2. 属于本项目的其它发布文件（软著、证书、知识产品的援引和授权等）
3. 本文件中各处引用的文件、资料，包括所要用到的软件开发标准。列出这些文件的标题、文件编号、发表日期和出版单位，说明能够得到这些文件资料的来源。

2总体设计

2.1需求规定

说明对本系统的主要的输入输出项目、处理的功能性能要求。

2.2运行环境

简要地说明对本系统的运行环境（包括硬件环境和支持环境）的规定。

2.3基本设计概念和处理流程

a. 说明本系统的基本设计概念和处理流程，尽量使用图表的形式。

b. 程序设计的基本概念有程序、数据、子程序、子例程、协同例程、模块以及顺序性、并发性、并行性、和分布性等。

2.4结构

用一览表及框图的形式说明本系统的系统元素（各层模块、[子程序](https://baike.baidu.com/item/%E5%AD%90%E7%A8%8B%E5%BA%8F)、[公用程序](https://baike.baidu.com/item/%E5%85%AC%E7%94%A8%E7%A8%8B%E5%BA%8F)等）的划分，扼要说明每个系统元素的[标识符](https://baike.baidu.com/item/%E6%A0%87%E8%AF%86%E7%AC%A6)和功能，分层次地给出各元素之间的控制与被控制关系

2.5功能需求与程序的关系

2.6人工处理过程

说明在本软件系统的工作过程中不得不包含的人工处理过程（如果有的话）。

2.7尚未解决的问题

说明在概要设计过程中尚未解决而设计者认为在系统完成之前必须解决的各个问题。

3接口设计

3.1用户接口

说明将向用户提供的命令和它们的语法结构，以及软件的回答信息。

3.2外部接口

说明本系统同外界的所有接口的安排包括软件与硬件之间的接口、本系统与各支持软件之间的接口关系。

3.3内部接口

说明本系统之内的各个系统元素之间的接口的安排。

4运行设计

4.1运行模块组合

说明对系统施加不同的外界运行控制时所引起的各种不同的运行模块组合，说明每种运行所历经的内部模块和支持软件。

4.2运行控制

说明每一种外界的运行控制的方式方法和操作步骤。

4.3运行时间

说明每种运行模块组合将占用各种资源的时间。

5系统数据结构设计

5.1逻辑结构设计要点

给出本系统内所使用的每个数据结构的名称、[标识符](https://baike.baidu.com/item/%E6%A0%87%E8%AF%86%E7%AC%A6)以及它们之中每个[数据项](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%A1%B9)、记录、文卷和系的标识、定义、长度及它们之间的层次的或表格的相互关系。

5.2物理结构设计要点

给出本系统内所使用的每个数据结构中的每个[数据项](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%A1%B9)的存储要求，访问方法、存取单位、存取的物理关系（索引、设备、[存储区域](https://baike.baidu.com/item/%E5%AD%98%E5%82%A8%E5%8C%BA%E5%9F%9F)）、设计考虑和保密条件。

5.3数据结构与程序的关系

说明各个数据结构与访问这些数据结构的形式

6系统出错处理设计

6.1出错信息

用一览表的方式说明每种可能的出错或故障情况出现时，系统输出信息的形式、含义及处理方法。

6.2补救措施

说明故障出现后可能采取的变通措施，包括：

a. 后备技术说明准备采用的后备技术，当原始系统数据万一丢失时启用的副本的建立和启动的技术，例如周期性地把磁盘信息记录到磁带上去就是对于磁盘媒体的一种后备技术；

b. 降效技术说明准备采用的后备技术，使用另一个效率稍低的系统或方法来求得所需结果的某些部分，例如一个自动系统的降效技术可以是手工操作和数据的人工记录；

c. 恢复及再启动技术说明将使用的恢复再启动技术，使软件从故障点恢复执行或使软件从头开始重新运行的方法。

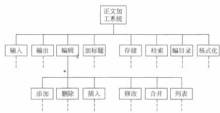
6.3系统维护设计

说明为了系统维护的方便而在程序内部设计中作出的安排，包括在程序中专门安排用于系统的检查与维护的检测点和专用模块。 各个程序之间的对应关系，可采用如下的矩阵图的形式

图形工具

（1）层次图。层次图用来描绘软件的层次结构。一个矩形框代表一个模块，方框间的连线表示调用关系。如图，最顶层的方框代表正文加工系统的主控模块，它调用下层模块，完成正文加工的全部功能。第二层的每个模块完成正文加工的一个主要功能。

（2）HIPO图。HIPO图是IBM公司发明的“层次图加输入/处理/输出图”。

[](https://baike.baidu.com/pic/%E6%A6%82%E8%A6%81%E8%AE%BE%E8%AE%A1/9827718/0/86d6277f9e2f0708ecdd4cdfe824b899a801f211?fr=lemma&ct=single)

层次图加上编号称为H图。在层次图的基础上，除最顶层的方框之外，其余每个方框都加了编号。层次图中每一个方框都有一个对应的IPO图（表示模块的处理过程）。每张IPO图应增加的编号与其表示的（对应的）层次图编号一致。IPO图是输入/加工/输出图的简称。

（3）结构图。结构图是Yordon提出的进行软件结构设计的工具，结构图和层次图类似，一个方框代表一个模块，框内注明模块的名字或主要功能。方框之间的直线（箭头）表示模块的调用关系。用带注释的箭头表示模块调用过程中来回传递的信息，尾部是空心的，表示传递的是数据，实心的表示传递的是控制。

概要设计的目的

1. 在前期需求分析“要解决什么问题”的基础上，重点讲清楚“总统的实现方案”：讲总体布局，讲各个组成之间的关系，讲与外部的联系，并有一定的研究基础和论证性。
2. 概要设计尚处于概念阶段，此时应着眼于总体，看联系看基础看相关，无需细枝末节。
3. 为详细设计提供总体和分模块的指引，为工程项目提供蓝图。

输出：

书面说明（设计文档）

软件结构图（插图、附图）