

Contents

第一章 引言	3
1.1 项目背景	3
1.2 总体要求	3
第二章 软件开发	5
2.1 需求分析	5
2.1.1 功能需求	5
2.1.2 系统功能图	5
2.2 概要设计	5
2.2.1 基本数据流程	5
2.2.2 系统功能模块划分、功能分配	5
2.2.3 数据结构设计	5
2.3 详细设计	6
2.3.1 导航控制器	6
2.3.2 制导控制器	6
2.3.3 姿态控制器	6
2.3.4 数据流	6
2.4 编码和单元测试	6
第三章 仿真	7
3.1 软件仿真	7
3.1.1 软件准备	7
3.1.2 组成部分	7
3.1.3 数据流走向	7
3.1.4 效果展示	7
3.2 硬件仿真	7
3.2.1 硬件准备	7
3.2.2 组成部分	7
3.2.3 数据流走向	7
3.2.4 效果展示	7
3.3 综合测试	8

第一章 引言

1.1 项目背景

1.2 总体要求

第二章 软件开发

2.1 需求分析

2.1.1 功能需求

2.1.2 系统功能图

这个阶段的任务仍然不是具体地解决问题，而是准确地确定“为了解决这个问题，目标系统必须做什么”，主要是确定目标系统必须具备哪些功能。

用户了解他们所面对的问题，知道必须做什么，但是通常不能完整准确地表达出他们的要求，更不知道怎样利用计算机解决他们的问题；软件开发人员知道怎样使用软件实现人们的要求，但是对特定用户的具体要求并不完全清楚。因此系统分析员在需求分析阶段必须和用户密切配合，充分交流信息，以得出经过用户确认的系统逻辑模型。通常用数据流图、数据字典和简要的算法描述表示系统的逻辑模型。在需求分析阶段确定的系统逻辑模型是以后设计和实现目标系统的基础，因此必须准确完整地体现用户的要求。系统分析员通常都是计算机软件专家，技术专家一般都喜欢很快着手进行具体设计，然而，一旦分析员开始谈论程序设计的细节，就会脱离用户，使他们不能继续提出他们的要求和建议。软件工程使用的结构分析设计的方法为每个阶段都规定了特定的结束标准，需求分析阶段必须提供完整准确的系统逻辑模型，经过用户确认之后才能进入下一个阶段，这就可以有效地防止和克服急于着手进行具体设计的倾向。

2.2 概要设计

2.2.1 基本数据流程

2.2.2 系统功能模块划分、功能分配

2.2.3 数据结构设计

实体对象关系设计

E-R图到关系模式

这个阶段必须回答的关键问题是：“概括地说，应该如何解决这个问题？”首先，应该考虑几种可能的解决方案。例如，目标系统的一些主要功能是用计算机自动完成还是用人工完成；如果使用计算机，那么是使用批处理方式还是人机交互方式；信息存储使用传统的文件系统还是数据库……。通常至少应该考虑下述几类可能的方案：低成本的解决方案。系统只能完成最必要的工作，不能多做一点额外的工作。中等成本的解决方案。这样的系统不仅能够很好地完成预定的任务，使用起来很方便，而且可能还具有用户没有具体指定的某些功能和特点。虽然用户没有提出这些具体要求，但是系统分析员根据自己的知识和经验断定，这些附加的能力在实践中将证明是很有价值的。高成本的“十全十美”的系统。这样的系统具有用户可能希望有的所有功能和特点。系统分析员应该使用系统流程图或其他工具描述每种

可能的系统,估计每种方案的成本和效益,还应该在充分权衡各种方案的利弊的基础上,推荐一个较好的系统(最佳方案),并且制定实现所推荐的系统的详细计划。如果用户接受分析员推荐的系统,则可以着手完成本阶段的另一项主要工作。上面的工作确定了解决问题的策略以及目标系统需要哪些程序,但是,怎样设计这些程序呢?结构设计的一条基本原理就是程序应该模块化,也就是一个大程序应该由许多规模适中的模块按合理的层次结构组织而成。总体设计阶段的第二项主要任务就是设计软件的结构,也就是确定程序由哪些模块组成以及模块间的关系。通常用层次图或结构图描绘软件的结构。

2.3 详细设计

2.3.1 导航控制器

2.3.2 制导控制器

2.3.3 姿态控制器

2.3.4 数据流

总体设计阶段以比较抽象概括的方式提出了解决问题的办法。详细设计阶段的任务就是把解法具体化,也就是回答下面这个关键问题:“应该怎样具体地实现这个系统呢?”这个阶段的任务还不是编写程序,而是设计出程序的详细规格说明。这种规格说明的作用很类似于其他工程领域中工程师经常使用的工程蓝图,它们应该包含必要的细节,程序员可以根据它们写出实际的程序代码。通常用H I P O图(层次图加输入/处理/输出图)或P D L语言(过程设计语言)描述详细设计的结果。

2.4 编码和单元测试

这个阶段的关键任务是写出正确的容易理解、容易维护的程序模块。

程序员应该根据目标系统的性质和实际环境,选取一种适当的高级程序设计语言(必要时用汇编语言),把详细设计的结果翻译成用选定的语言书写的程序,并且仔细测试编写出的每一个模块。

第三章 仿真

3.1 软件仿真

3.1.1 软件准备

3.1.2 组成部分

3.1.3 数据流走向

3.1.4 效果展示

3.2 硬件仿真

3.2.1 硬件准备

3.2.2 组成部分

3.2.3 数据流走向

3.2.4 效果展示

3.3 综合测试

这个阶段的关键任务是通过各种类型的测试（及相应的调试）使软件达到预定的要求。最基本的测试是集成测试和验收测试。所谓集成测试是根据设计的软件结构，把经过单元测试检验的模块按某种选定的策略装配起来，在装配过程中对程序进行必要的测试。所谓验收测试则是按照规格说明书的规定（通常在需求分析阶段确定），由用户（或在用户积极参加下）对目标系统进行验收。必要时还可以再通过现场测试或平行运行等方法对目标系统进一步测试检验。为了使用户能够积极参加验收测试，并且在系统投入生产性运行以后能够正确有效地使用这个系统，通常需要以正式的或非正式的方式对用户进行培训。通过对软件测试结果的分析可以预测软件的可靠性；反之，根据对软件可靠性的要求也可以决定测试和调试过程什么时候可以结束。应该用正式的文档资料把测试计划、详细测试方案以及实际测试结果保存下来，做为软件配置的一个组成成分。

整体的思路方法，找到自己做的凝练的地方;系统化，软件工程思路;系统需求分析, 总体框架, 我的模块, 子模块功能及其实现流程;用到系统的硬件, 接口的口地址函数, 接口的类定义方法, 自己所做的东西内容