

第七章——结构化设计

结构化设计

结构化设计是将结构化分析得到的数据流图映射成软件结构的一种设计方法，强调模块化、自顶向下逐步求精、信息隐蔽、高内聚低耦合等设计准则

结构图

用结构图(Structure Chart)来描述软件系统的体系结构

描述一个软件系统由哪些模块组成，以及模块之间的调用关系

结构图的基本成分有:模块、调用和数据

模块

模块是指具有一定功能的可以用模块名调用的一组程序语句，包括函数和子程序。

一个模块具有其外部特征和内部特征

外部特征包括:模块的接口(模块名、输入/输出参数、返回值等)和模块的功能

内部特征包括:模块的内部数据和完成其功能的程序代码

调用和数据

调用(call):用从一个模块指向另一个模块的箭头来表示，其含义是前者调用了后者。为了方便，有时常用直线替代箭头，此时，表示位于上方的模块调用位于下方的模块

数据(data):模块调用时需传递的参数可通过在调用箭头旁附加一个小箭头和数据名来表示

!结构化辅助符号

结构化设计是将结构化分析的结果(数据流图)映射成软件的体系结构(结构图)

将数据流图分为变换型数据流图和事务型数据流图，对应的映射分别称为变换分析和事务分析

变换流

特征:数据流图可明显地分成三部分

输入:信息沿着输入路径进入系统，并将输入信息的外部形式经过编辑、格式转换、合法性检查、预处理等辅助性加工后变成内部形式

变换:内部形式的信息由变换中心进行处理

输出:然后沿着输出路径经过格式转换、组成物理块、缓冲处理等辅助性加工后变成输出信息送到系统外

数据流沿着输入路径到达一个事务中心，事务中心根据输入数据的类型在若干条动作路径中选择一条来执行

事务中心的任务是:接收输入数据(即事务);分析每个事务 类型;根据事务类型选择执行一条动作路径

!事务中心

变换分析

变换分析的任务是将变换型的DFD映射成初始的结构图，步骤如下:

- 1.划定输入流和输出流的边界，确定变换中心
- 2.进行第一级分解:将DFD映射成变换型的程序结构
- 3.进行第二级分解:将DFD中的加工映射成结构图中的一个适当的模块
- 4.标注输入输出信息:根据DFD，在初始结构图上标注模块之间传递的输入信息和输出信息

!统计成绩子图的输入、输出流边界

!进行第一级分解

!第二级结构分解

事务分析

将事务型DFD映射成初始的结构图

实例:银行业务中有存款、取款、查询余额、开户、转帐等多种事务，这种软件通常是接收一个事务，然后根据事务的类型执行一个事务处理的功能事务型的结构图包括:

主控模块:完成整个系统的功能

接收模块:接收输入数据事务

发送模块:根据输入事务的类型，选择一个动作路径控制模块

动作路径控制模块:完成相应的动作路径所执行的子功能

1.在DFD图上确定边界

事务中心

接受部分(包括接受路径)

发送部分(包括全部动作路径)

2.画出SC图框架

DFD图的三个部分分别映射为事务控制模块，接受模块和动作发送模块

分解和细化接受分支和发送分支

!划分DFD

!画出SC图框架

!细化SC图

结构图的优化

深度:程序结构图中控制的层数，例如图中所示的结构图的深度 是5

宽度:程序结构图中同一层次上模块总数的最大值，例如图中所示的结构图的宽度为7

扇出(fan out): 该模块直接调用的模块数目。例如，例如图中模块M的扇出是4模块A的是2模块B的扇出是1

扇入(fan in): 能直接调用该模块的模块数目。例如图中模块G的 扇入是1模块I的扇入是2，模块R的扇入是4

!结构图的相关度量

优化宗旨

1.降低耦合度，提高内聚度

2.避免高扇出，并随着深度的增加，力求高扇入

3.模块的影响范围应限制在该模块的控制范围内

控制域：自身和下级的模块；

作用域：受这个模块中的判定所影响的模块