

## 软件工程 第4-2章 软件设计-SD方法

#### 重庆理工大学

计算机科学与工程学院 李梁

qq:1255214405

liliang@cqut.edu.cn

智慧树课号: K3388095

智慧树网址: www.zhihuishu.com



9

## 第4-2章 软件设计-SD方法

- 3 结构化设计方法: 面向数据流设计方法(SD法)
- 3.2 面向数据流设计模型
- 3.3 面向数据流设计方法 软件体系结构模型-自学
- 5 详细设计
- 5.1 详细设计的基本任务
- 5.2 结构化程序设计方法
- 5.3 结构化详细设计工具(自学)
- 5.4 人机交互与界面设计



#### 3.1 结构化设计方法概述

- ●软件设计分为概要设计、详细设计两个阶段:
  - ▶概要设计也称总体设计,确定软件的结构以及各组成成分( 子系统或模块)之间的相互关系。<mark>概要设计</mark>通过数据流图来 确定系统的结构图,并且对这些结构图进行分析和细化。
  - >概要设计的主要任务是: 将系统划分成模块; 决定每个模块 的功能;决定模块的调用关系;决定模块的界面,即模块间 传递的数据。
  - 》详细设计就是在概要设计的基础上决定如何具体实现各模块 的内部细节,直到对系统中的每个模块给出足够详细的过程 描述。在编码实现阶段可以完全按照详细设计的细节过程来 映射到代码,最终实现整个系统。一般使用结构化程序设计 工具来描述。

M



# 3.2 面向数据流设计模型

- 结构化设计方法: SD设计方法包括一组概念、标准和指导思想 用模块结构图来表达程序模块之间的关系。结构化设计方法 有面向数据流的设计方法、Jackson系统开发方法等。
- SD法其目的是降低软件开发费用及维护费用,有利于修正软件 及实现新的软件需求。极大地增加了软件复用能力。
- SD设计方法的目标:将软件设计为功能单一的模块,建立系统 的模块结构图。它的主要表示方法是一种分层次的结构图。
- SD设计方法的主要思想
- ▶认为一个程序、一组程序或一个系统,无非是由<mark>一组功能操</mark> 作来构成的,并进而吸取了结构化分析的"黑盒子"的概念
- 软件设计者首先必须无视程序、模块或过程的内部情况,而 只对它们之间的关系进行分析。将系统看作是,逻辑功能的 一功能模块的集合。这样软件设计者能够有最大 抽象集合-的自由度设计系统结构。



#### 3.2 面向数据流设计模型

- SD设计方法的优点:
  - ① 减少设计复杂性。将大化小,使复杂问题简单化。
  - ② 结构独立。将程序划分成多个相对独立的模块。
  - ③ 模块功能单一化,可使软件设计获得最大的益处。
  - ④ 便于软件的修改。
  - ⑤ 易于开发和维护。
  - 加强了代码的可重用性。
- 结构化设计描述工具:
  - >结构化设计方法通过一种图形工具——结构图,进行软件结 构设计。
  - >结构图是精确表达程序结构的图形表示方法。它作为软件文 档的一部分,清楚地反映出程序中模块之间的层次调用关系 和联系。它不仅严格地定义了各个模块的名字、功能和接口 ,而且还集中地反映了设计思想。



## 3.2 面向数据流设计模型

#### SD法设计准备过程

变化部分

В

3

4

化的

①首先研究、分析和审查数据流图。 从软件的需求规格说 明中弄清数据流加工的过程,<mark>修改数据流图,明确新系统做</mark> 什么?确定新系统的逻辑模型(要点)

6.

6

卓非本质的因詞 这个处理逻辑是否必须这样的

这个数据存储是否一定需要 这个数据的流向是否合理? 将DFD分解成没有变化的部

分,和将有变化的部分 重新分析将有变化的部分 DFD

M

(需凭经验重新分解该处理 )确定人机边界



# 3.2 面向数据流设计模型

#### 系统结构图

- 传入模块 从下属模块取得数据,经过某些处理,再将 其传送给上级模块。它传送的数据流叫做逻辑输入数据流
- 传出模块 从上级模块获得数据,进行某些处理,再将 其传送给下属模块。它传送的数据流叫做逻辑输出数据流
- 变换模块 一 它从上级模块取得数据,进行特定的处理, 转换成其它形式,再传送回上级模块。它加工的数据流叫 做变换数据流。
- 协调模块 对所有下属模块进行协调和管理的模块。





# 3.2 面向数据流设计模型

- <mark>软件模型的本质特性:</mark> 数据信息必须以 "外部"信息(键盘输入的数据,鼠标交互的事件)形式进入软件系统,经过内部处理以后再以"外部"的形式(报表、界面显示等) 离开系统。
- ●SD方法是以数据流图为基础设计系统的模块结构。不同类型的数据流图设计的过程以及所导出的初始的软件结构不同。不管系统的数据流图如何庞大和复杂,一般均可分为变换型、事务型和变换/事务混合型等三种数据流类型:
  - ✓变换型数据流
  - ✓事务型数据流
  - ✓混合型数据流

重庆班工大学计算机科学与工程学院 李燮(liliang@cgut.edu.cn)

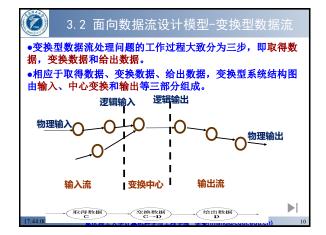
M

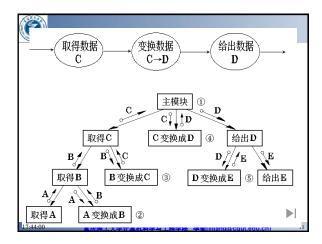


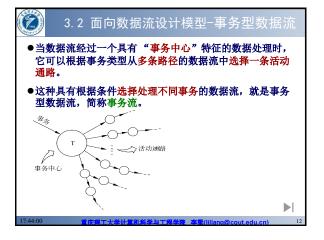
#### 3.2 面向数据流设计模型-变换型数据流

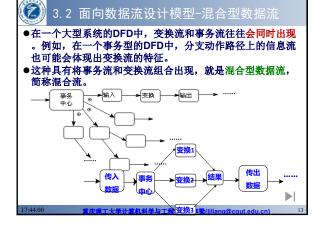
- ●信息可以通过各种路径进入系统,信息在"流"入系统的过程中由外部形式变换成内部数据形式,这被标识为输入流。
- ●在软件的核心,输入数据经过一系列加工处理,这被标识为 变换流。
- ●通过变换处理后的输出数据,沿各种路径转换为外部形式"流"出软件,这被标识为输出流。
- ●整个数据流体现了以输入、<sup>信息</sup> 变换、输出的顺序方式,沿 一定路径前行的特征,这就 是<mark>变换型数据流</mark>,简称变换流。

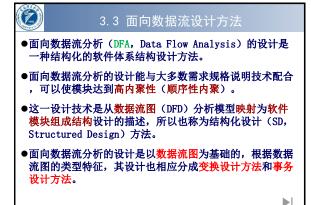
输入流 输出流 变换流

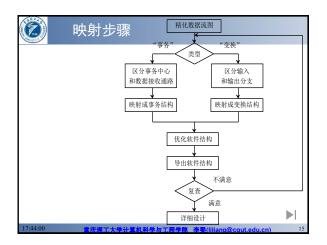


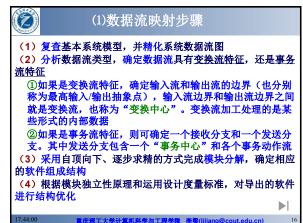


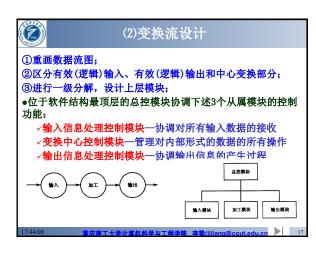


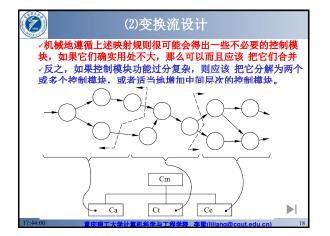




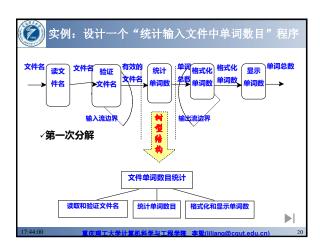




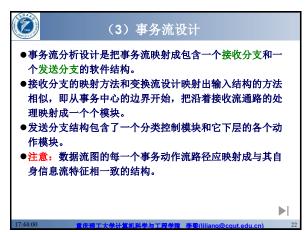


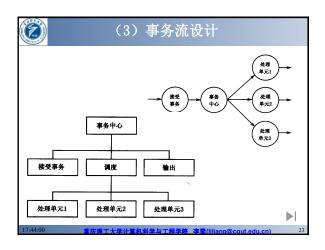


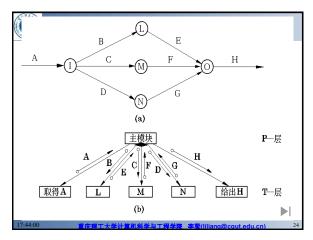


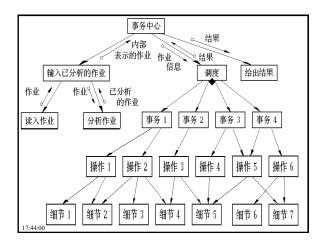


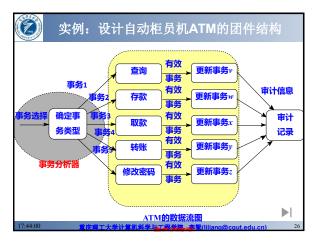


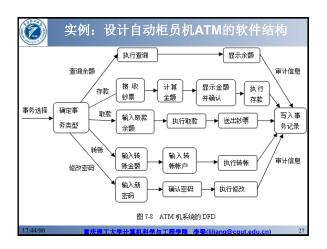


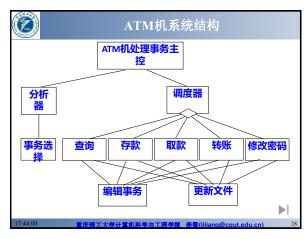


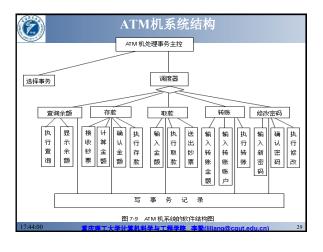


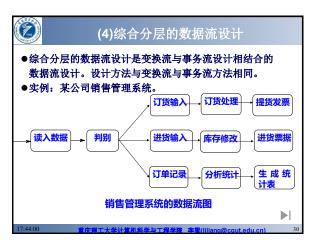


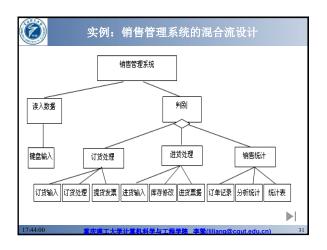


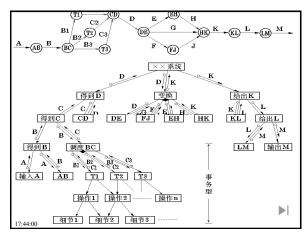












# **S**

# 设计后的处理-接口

- 为每一个模块写一份处理说明
- ·为每一个模块提供一份接口说明 ·确定全局数据结构和局部数据结构
- •指出所有的设计约束和限制
- ·进行概要设计的评审 ·进行设计的优化(如果需要和可能的话)
- 接口是提供给其他模块或者系统使用的一种约定或者规范。 因此接口必须要保证足够的稳定性和易用性。
- 接口的语义必须明确。接口一般包括接口调用方法、接口 名称、参数的类型和名称。
- 采用版本定义来区分接口的差异。比如提供接口版本查询 功能,接口的实现提供了接口版本的查询功能。
- 接口设计还要考虑规范性、健壮性、可移植性、安全性、 兼容性等。



#### 4 软件体系结构模型

●对于大规模的复杂软件系统来说,系统结构设计和规格说明和算法与数据结构相比,<mark>结构设计变得越来越重要了</mark>。

- ●软件体系结构是一种表达,使软件工程师能够分析设计是否满足 需求、选择合理的方案和降低风险。
- ◆大型软件系统总是被分解成一系列子系统,由子系统提供一些相 关的服务。
- 软件体系结构设计过程就是识别出这些子系统,并建立子系统控 制和通信的框架,最后给出软件体系结构的描述。
- 软件体系结构: 体系结构为软件系统提供了一个结构、行为和 属性的高级抽象。由构成系统的元素的描述(构件)、这些元素的相互作用、指导元素集成的模式以及这些模式的约束组成。
- 体系结构不仅指定了系统的组织结构和拓扑结构,并且显示了 一些设计 系统需求和构成系统的元素之间的对应关系,提供了-决策的基本原理。

M



## 4 软件体系结构模型-常见软件结构(架构)

1、客户机/服务器(Client/Server): 是一种分布式系统模型, 作为服务器的子系统为其他客户机的子系统提供服务,作为客户 机的子系统负责与用户的交互。 电子邮件,文件共享和在线应用 (1) 两层C/S结构: 服务器只负责数据的管理,客户机实现应用 逻辑和用户的交互。

胖客户端与瘦客户端:业务逻辑的划分比重,在客户端多一些还 是在服务器端多一些。

胖客户端:客户端执行大部分的数据处理操作。

瘦客户端: 客户端具有很少或没有业务逻辑。

(2) 浏览器/服务器体系结构(Browser/Server): 是随着对C/S 体系结构的一种变化或者改进。许多应用系统的用途都是从数据 库中检索数据,并将其显示给用户。在用户更改数据之后,系统 再将更新内容存储到数据存储中。因为关键的信息流发生在数据 存储和用户界面之间,所以一般倾向于将这两部分捆绑在一起 以减少编码量并提高应用程序性能。

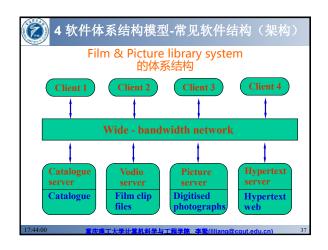


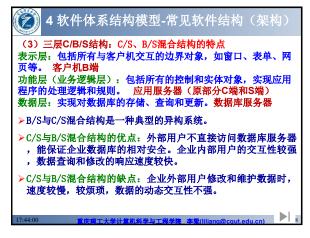
#### 4 软件体系结构模型-常见软件结构(架构)

#### B/S体系结构的不足之处

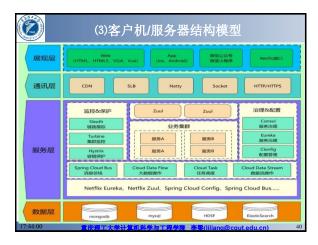
- ① 缺乏对动态页面的支持能力,数据库处理能力差。
- ② 系统扩展能力差,安全性难以控制。
- ③ 响应速度远低于C/S体系结构。 ④ 数据提交一般以页面为单位,数据的动态交互性不强,不利于 在线事务处理 (OLTP) 应用。

在基于Web的应用程序中,用户界面逻辑的更改往往比业务逻辑 频繁。如果将UI代码和业务逻辑组合一起并放在UI中,则每次更 改界面都可能引起对业务逻辑的修改。在某些情况下,应用程序 以不同的方式显示同一数据。与业务逻辑相比,用户界面代码对 设备的依赖性往往更大。设计美观而有效的用户界面与开发复杂 业务逻辑需要不同的编程技能。通常,为用户界面创建自动测试 比为业务逻辑更难、更耗时。

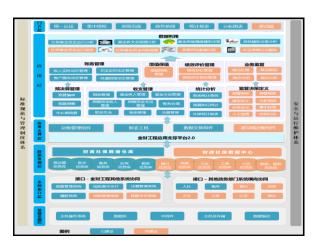


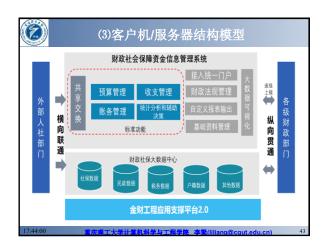


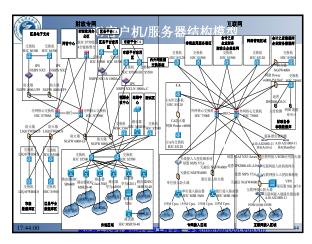


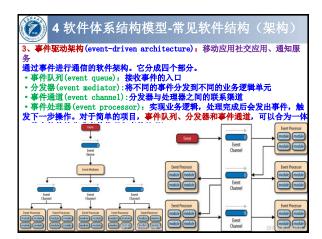


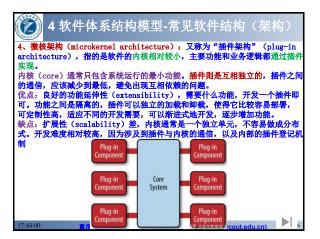


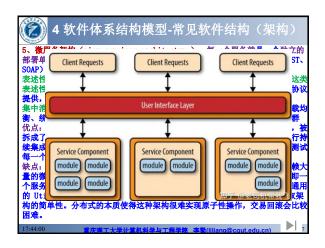


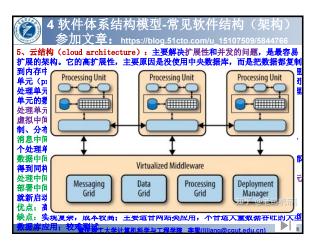














#### 小结

- 概要设计就是确定系统的模块以及模块之间的结构和关系,将软件 的功能需求分配给所划分的最小单元模块。
- 详细设计要确定数据结构、文件结构、数据库模式以及确定测试方 法与策略。
- 在概要设计阶段主要树立一个模块化的思想。将分析阶段的数据流 图,将数据流进行分析。是变换流还是事务流,或者是两种的结合 混合流。按照数据流到软件结构图的映射步骤将数据流图转化为系 统的软件结构图。再对结构图进行分析、细化得到合理的软件模块 结构。
- 在软件模块确定后,就需要考虑为软件结构图中的每一个模块确定 相应的算法和块内的数据结构,用结构化程序设计工具来描述。
- 结构化程序设计工具通常以图形语言来描述,然后用PDL语言来加 工。使得操作的步骤尽可能详细和清晰。在详细设计工具中主要有 程序流程图、盒图、PAD图和HIPO图等。
- 用面向数据结构的结构化程序设计方法JSD方法来详细描述程序的 结构



# 软件工程中的一些观念讨论

- 1、如果使用原型法,可否仅设计出系统总体结构后进行实施
- 2、有经验的系统设计人员采用先凭经验设计出一个模型,然 后根据系统分析进行修改,如何理解?
- 3、使用SD发进行设计,然后用OOD法进行实施是否可行?
- 4、系统设计中,模块间的接口设计到底是什么?
- 5、采用自顶向下设计好还是自底向上设计好。
- 6、系统需求分析人员是否可以作为系统设计人员。
- 7、系统设计中,是否应坚持"用户第一"的概念。
- 8、系统设计中如何考虑算法、资源,性能和功能间的关系?
- 9、采用面向对象的实现技术,系统设计到那一步为好?



# 软件工程中的一些观念讨论

- 10、系统设计时是否要考虑实现技术,分析、调查阶段是否考 虑实现技术?
- 11、系统设计时是否要考虑系统规模、资金、集成环境等,分 析阶段考虑否?
- 12、有没有不能完成系统设计的系统,有没有不能实现的系统
- 13、系统分析人员和系统设计人员谁的要求高?
- 14、如何理解"一致性、规范"和"个性"的关系,"漂亮"和 "简洁"的关系。
- 15、界面设计中主要考虑的因素是什么?
- 16、使用OOD开发还需要画程序框图否?

 $\triangleright$