



北京交通大学
BEIJING JIAOTONG UNIVERSITY



软件系统分析与设计 System Analysis & Design

M210007B [03]

Haiming Liu

Sunday, May 29, 2022

首先，回顾一下上一节课内容

Previously on System Analysis & Design

课程计划

软件工程的基础

1. 软件工程基础/软件生命周期
2. 软件过程模型
3. 软件项目管理

系统分析与需求描述 结构化设计方法

4. 系统分析方法与问题定义
5. 需求分析与需求获取
6. 用例建模与用例描述
7. 结构化系统分析与设计

面向对象的设计方法

8. 面向对象方法概述
9. 面向对象建模与UML
10. 软件系统动态建模
11. 软件系统静态建模
12. 数据库设计
13. 综合案例分析
14. 考前复习

/-1

过程建模

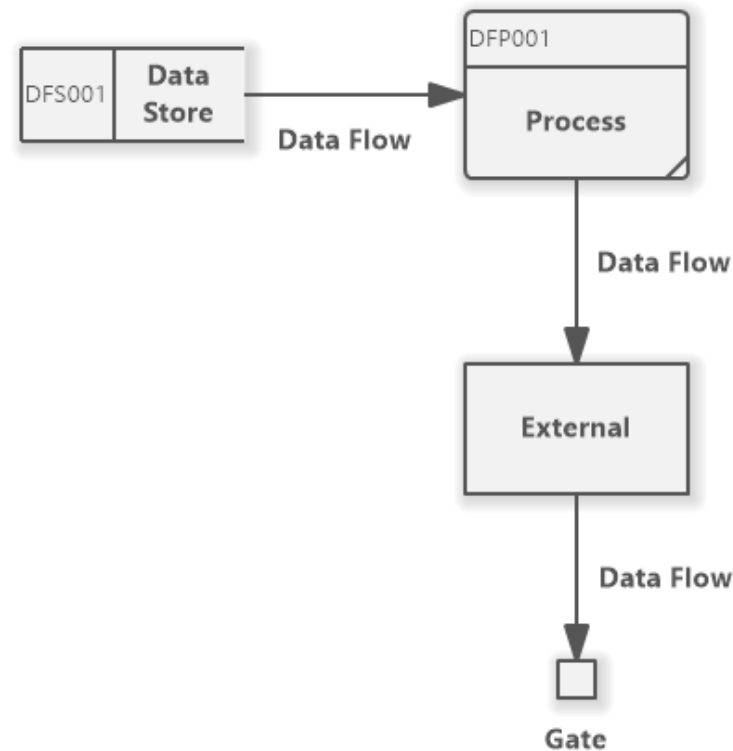
Process Modeling

逻辑过程建模技术

Logical Process Modeling

结构化方法：

- 一种 面向数据流的需求分析和设计方法，适用于分析和设计大型数据处理系统，简单、实用、应用广泛
- 从系统建模的视角来看，基于为 过程建模 而使用的 数据流图 (Data Flow Diagram, DFD) 技术
- 系统以 过程为中心，在功能分解活动中被分解为可管理的单元，同时将系统划分为 由数据流连接的业务过程



过程建模

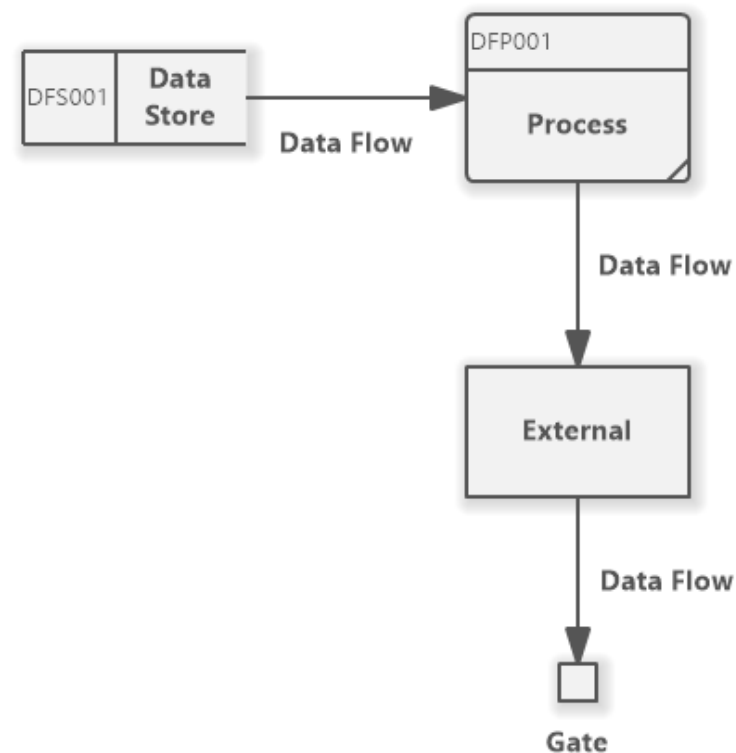
Process Modeling

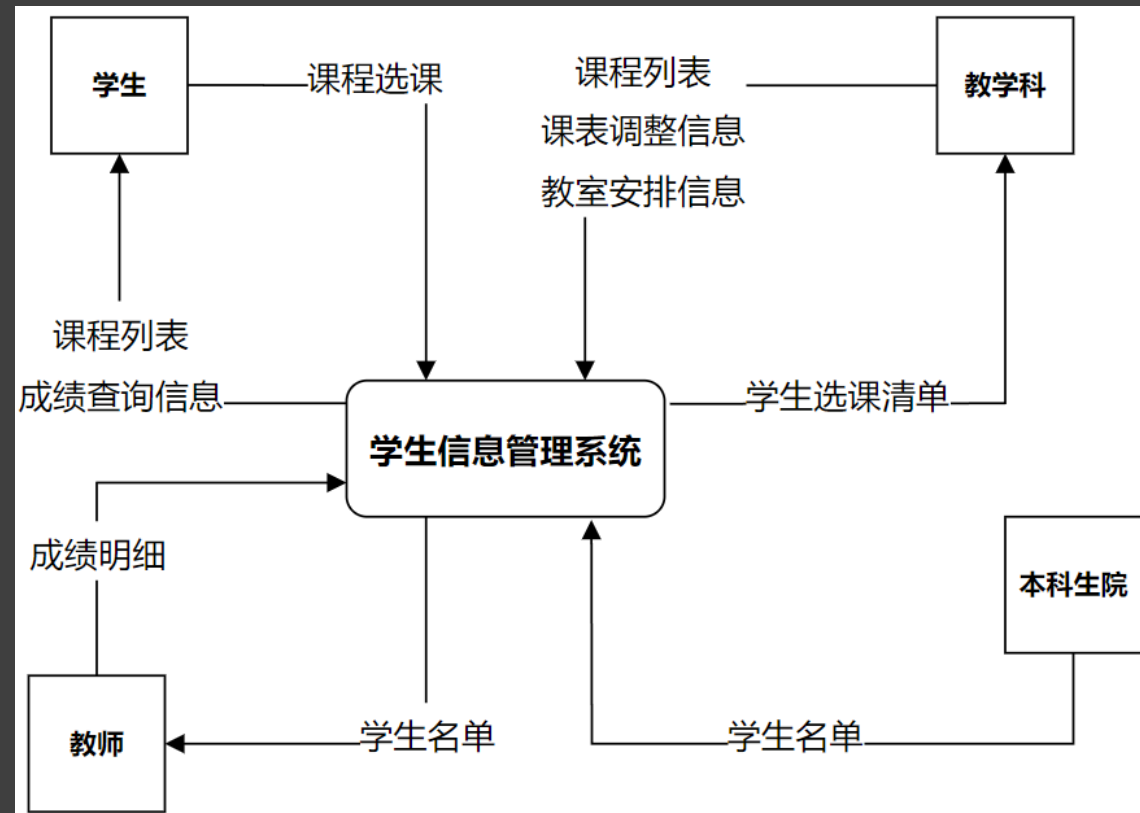
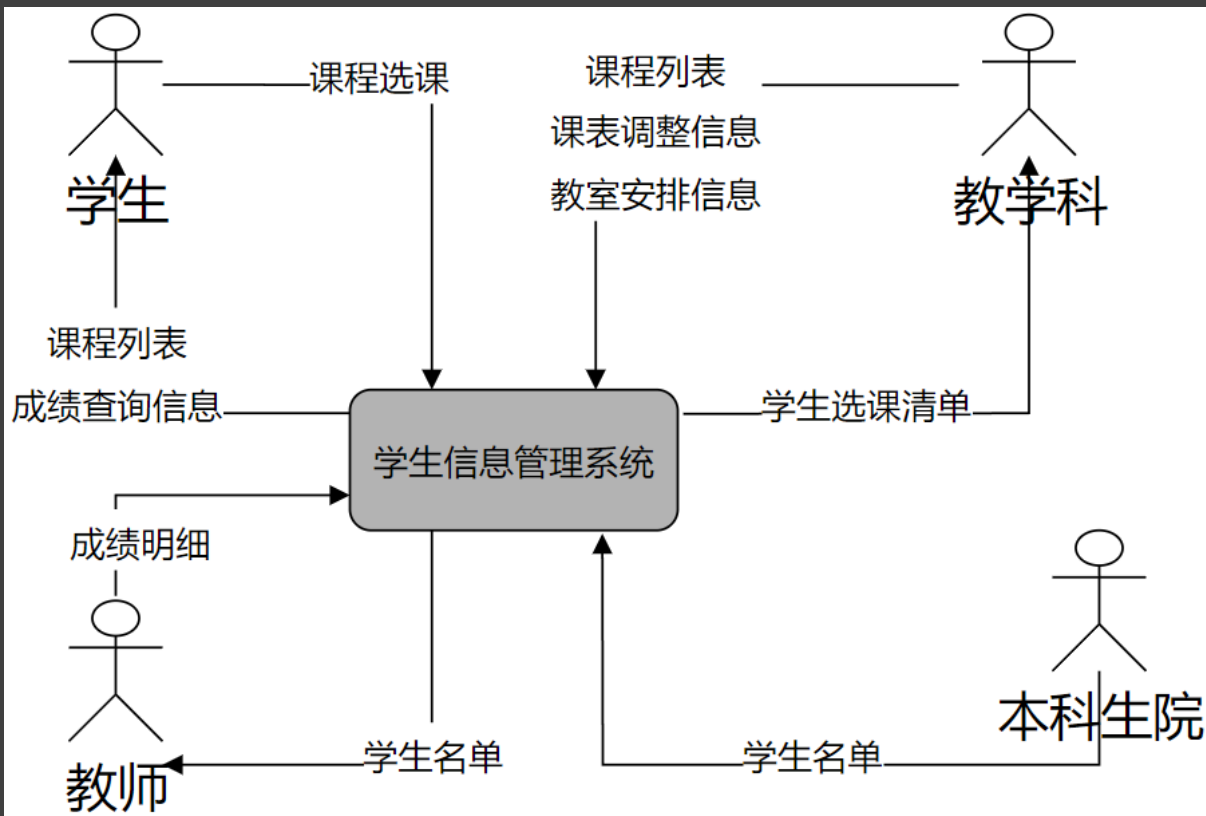
过程建模:

- 一种组织和记录数据的结构和流向的技术
- 记录系统的“过程”和/或有系统的“过程”实现的逻辑、策略和程序。

数据流图(Data Flow Diagram, DFD; 又称泡式图、转换图和过程模型):

- 一种描述通过系统的数据流以及系统实施的工作或处理过程的工具
- 优点: 容易阅读, 仅有三种符号和一种连接





数据流图

Data Flow Diagram

- 圆角矩形表示要完成的**过程**或者工作
- 正方形表示**外部代理**——系统的边界
- 箭头表示**数据流**——或者输入和输出，到过程和来自过程
- 开放的方框表示**数据存储**——有时称为文件或者数据库



外部代理

External Agent

外部代理又称**外部实体**，定义位于项目范围之外，但与正在被研究的系统交互的人、部门、其他系统或者其他组织

- 外部代理很少是固定的。如果系统范围扩大，**外部代理**可能会被认为是系统内部的一个**新过程**
- 用**正方形**表示，用描述性的单数名词命名，如：学生、供应商或信息系统
- 放置在页面的周围，与它的系统边界的定义相一致
- 为避免数据流线的交叉，允许使用重复的外部代理



Gane and Sarson shape

数据存储

Data Store

数据存储，又称**文件、数据库**。是一个数据的“仓库”，收集数据供日后使用

- 描述关于企业想存储数据的“事物”，包括：
 - 人 (或一组人): 学生、部门、客户
 - 地点: 销售地区、建筑物、校园
 - 对象 (或一类对象的说明): 图书、工具、软件包
 - 事件: 添加、删除、修改、查看
- **数据存储表示了数据实体的所有具体值**
- 用**末端开口的矩形（左端封闭，右端开口）**表示，用**名词**命名
- 为避免数据流线的交叉，允许使用重复的数据存储



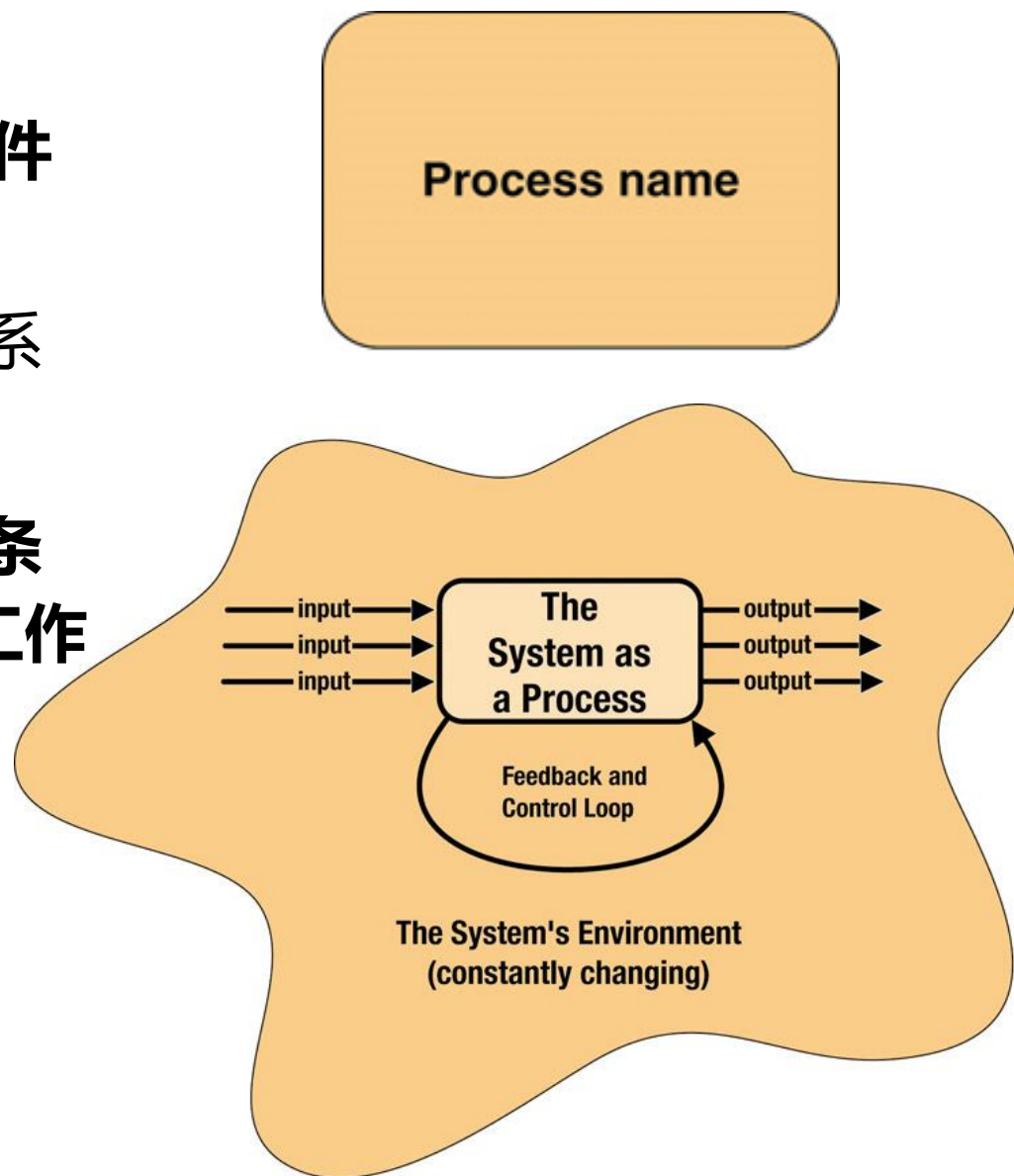
Gane and Sarson shape

过程概念

Process Concept

过程是一个基本的信息系统构件。**过程响应业务事件和条件，并将“数据”转换成有用的信息。**

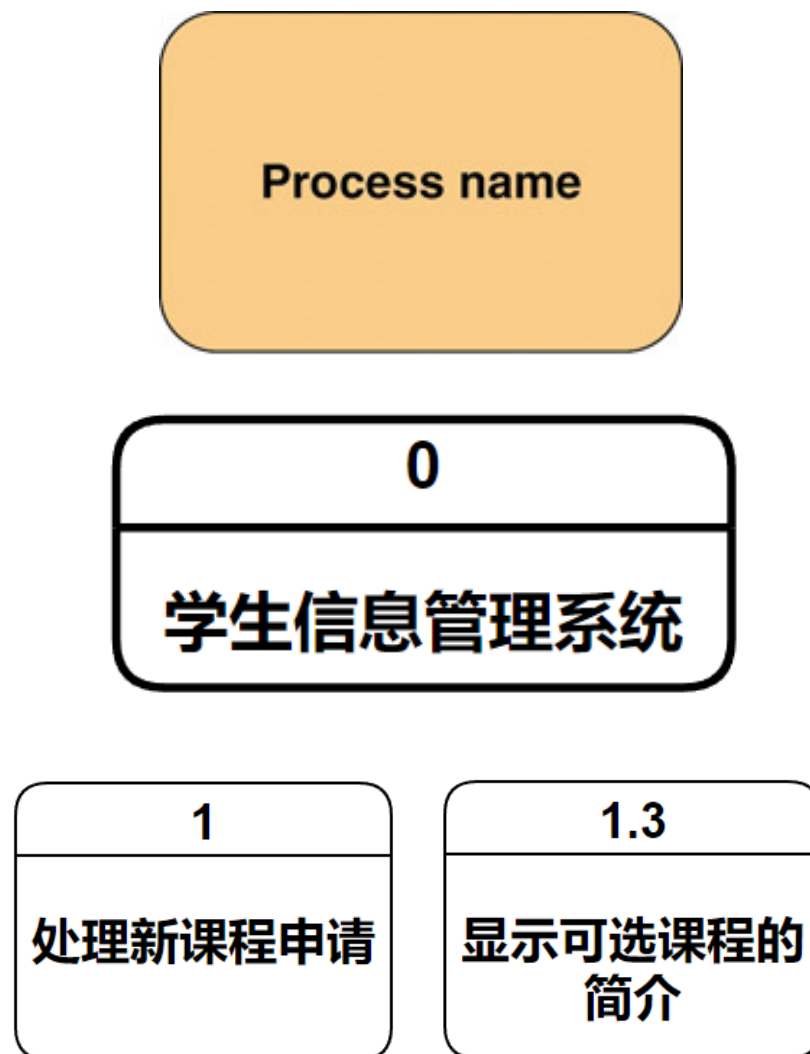
- 对**过程建模**帮助我们理解过程与系统环境、其他系统以及其他过程如何交互
- 使用**圆角矩形**表示过程，**过程是在输入数据流或条件上执行，或者对输入数据流或条件作出响应的工作**

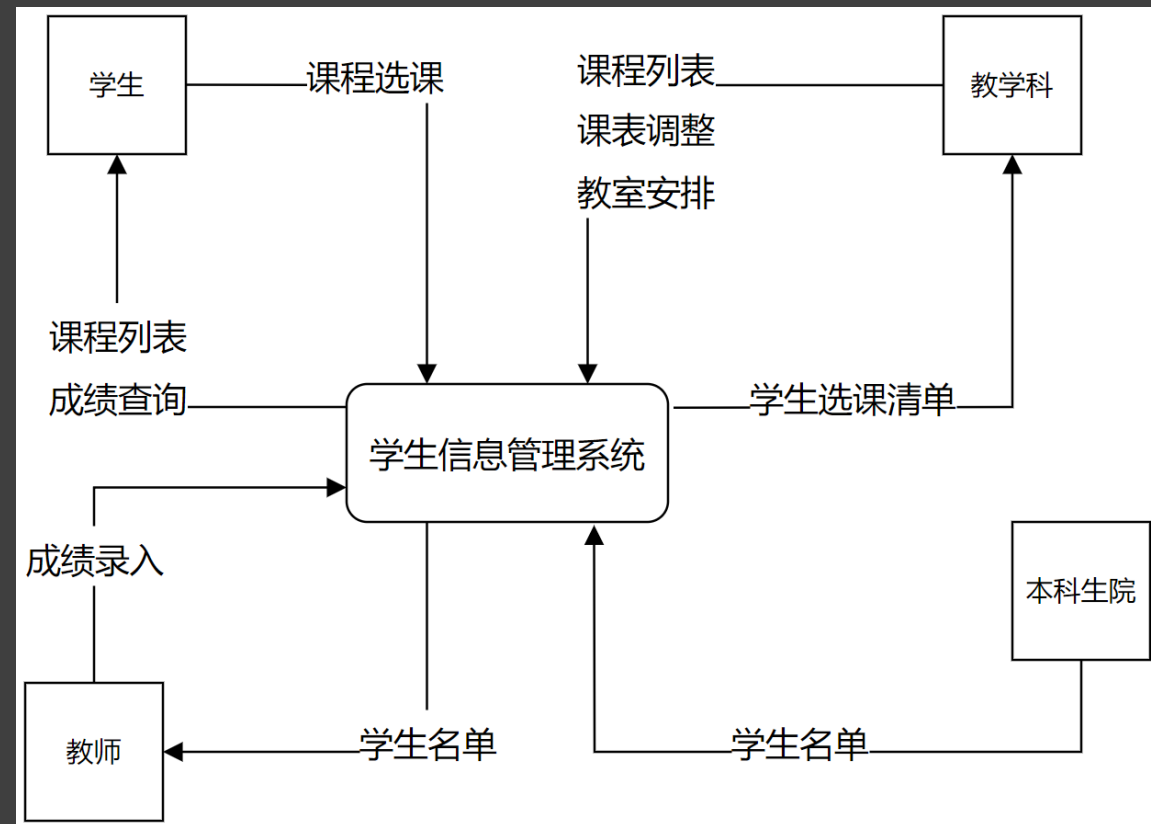
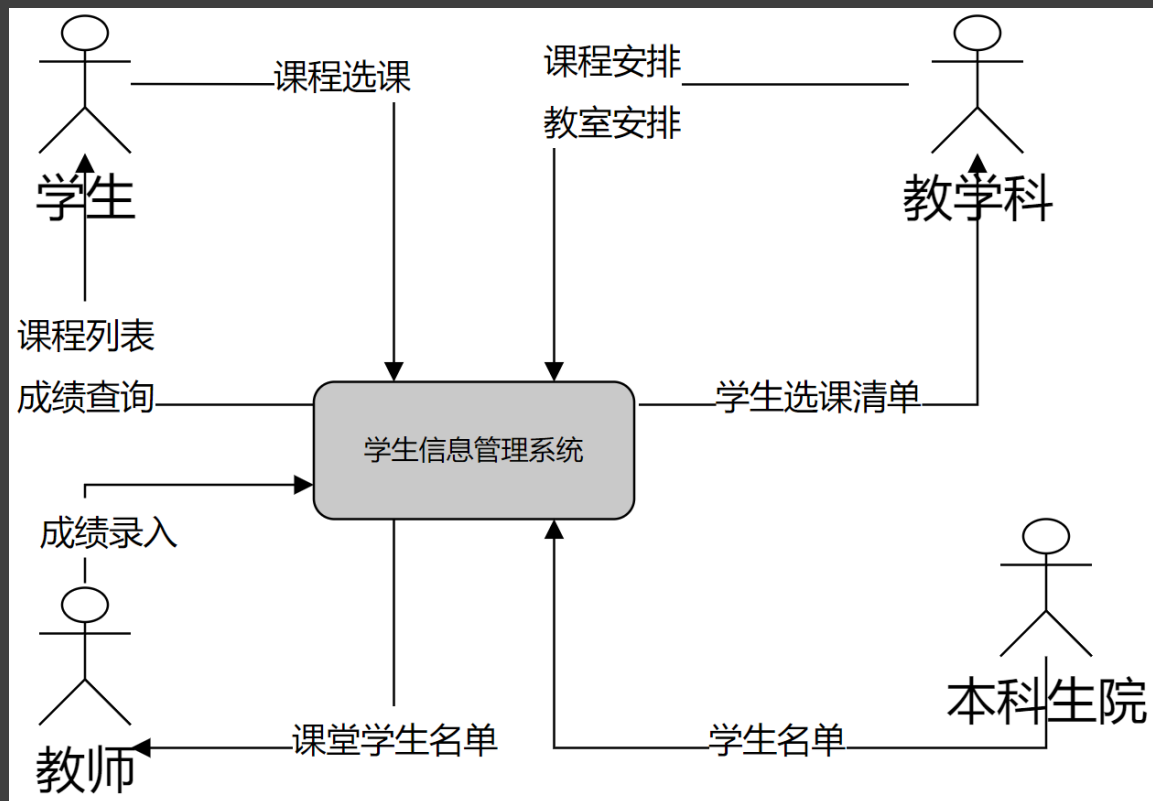


过程概念

Process Concept

- **命名高级过程**，以整个系统/子系统的名称命名
例如：学生信息管理系统、库存管理子系统
- **命名一般过程**，使用“动词-形容词-名词”格式
例如：验证账号沟的状态、添加订单的记录等
- **必须有唯一的标识号**，指出它在图中的层次





数据流图

Data Flow Diagram

- 圆角矩形表示要完成的过程或者工作
- 正方形表示外部代理——系统的边界
- **箭头表示数据流——或者输入和输出，到过程和来自过程**
- 开放的方框表示数据存储——有时称为文件或者数据库

数据流

Data Flow

数据流，表示到一个过程的数据输入，或者来自一个过程的数据（信息）输出。

- 数据流是运动中的数据
- 也用于表示在文件或数据库中创建、删除、修改或读取数据



控制流，表示触发一个过程的条件或非数据事件。

- 可以把它看作是系统工作时的一个监控条件

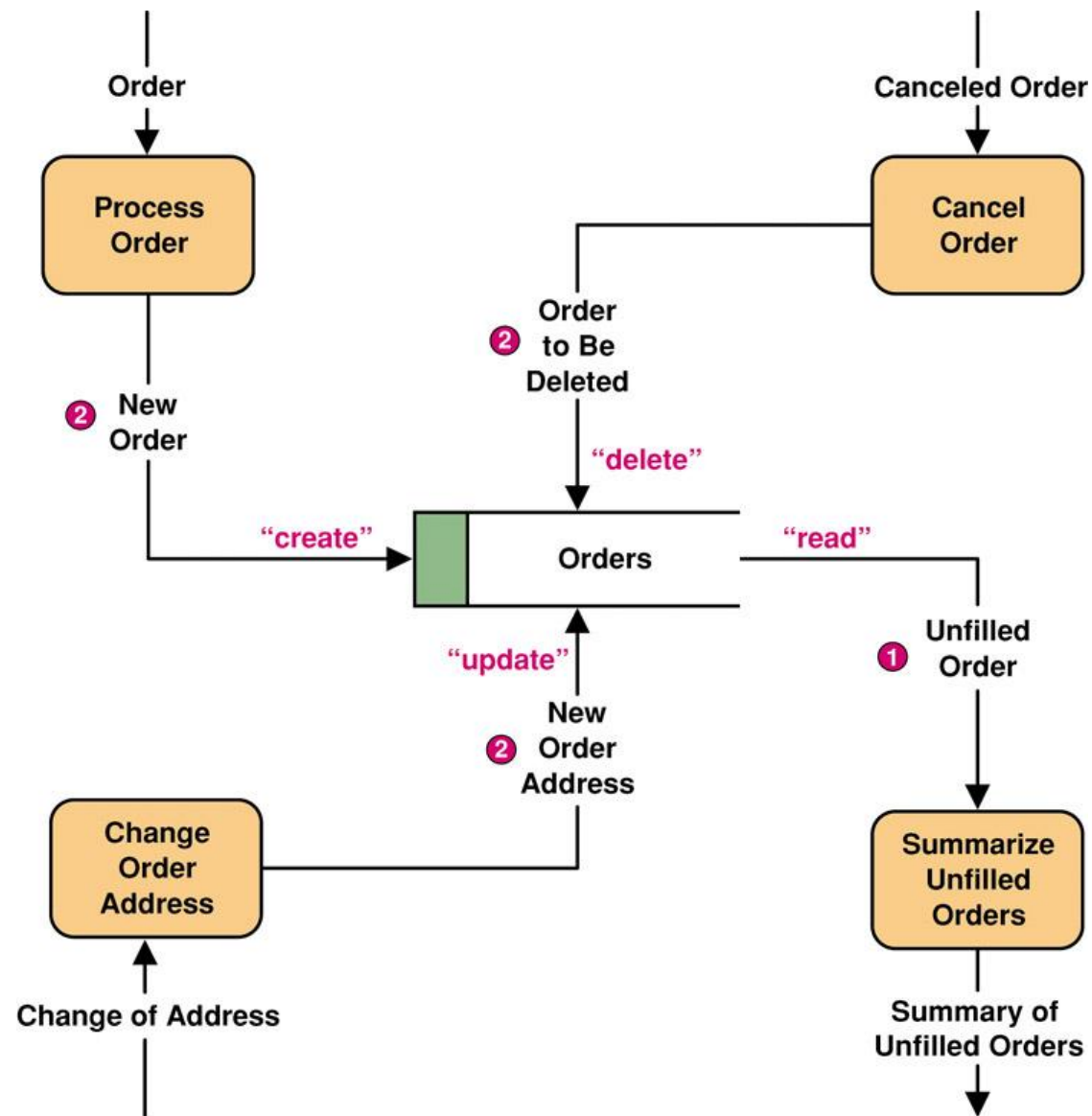
例如：时间、温度和高度等。报告生成过程可以由时序事件“每晚”、“月末”触发



- 使用**描述性名词命名**

数据流 Data Flows

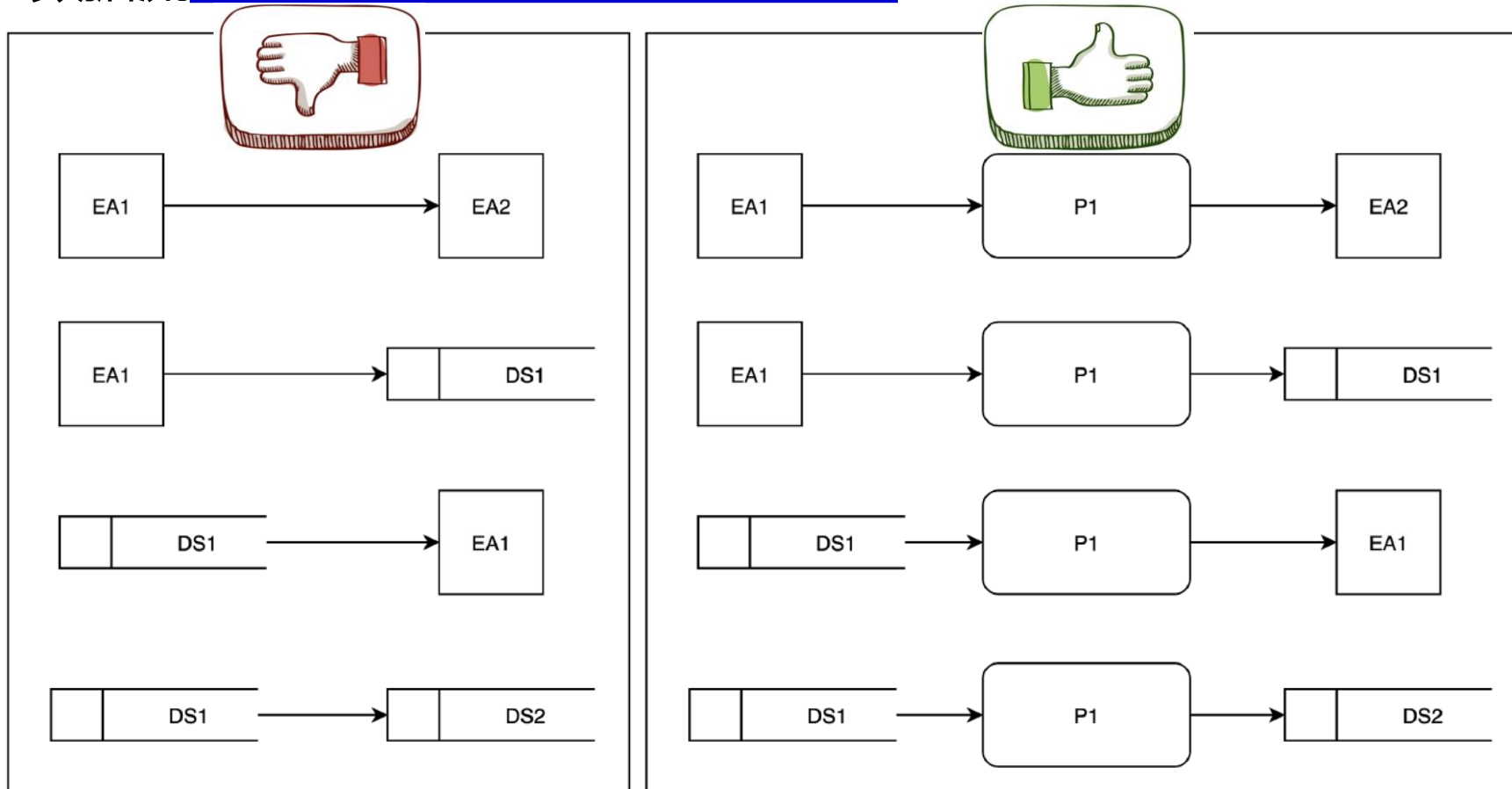
- ① 从一个数据存储到一个过程的数据流，清楚地指出读取了什么数据
- ② 从一个过程到一个数据存储的数据流，清楚的命名以反映执行的特定动作



往返于数据存储的数据流

数据流规则 Rules for Data Flows

- 数据流名称应该是唯一的，并采用描述性单数名词和名词短语
例如：一个过程的输入是“订单”，输出可以命名为“有效的订单”
- 数据流名称应该描述数据而不描述实现
- 数据流必须以一个过程开始和/或结束



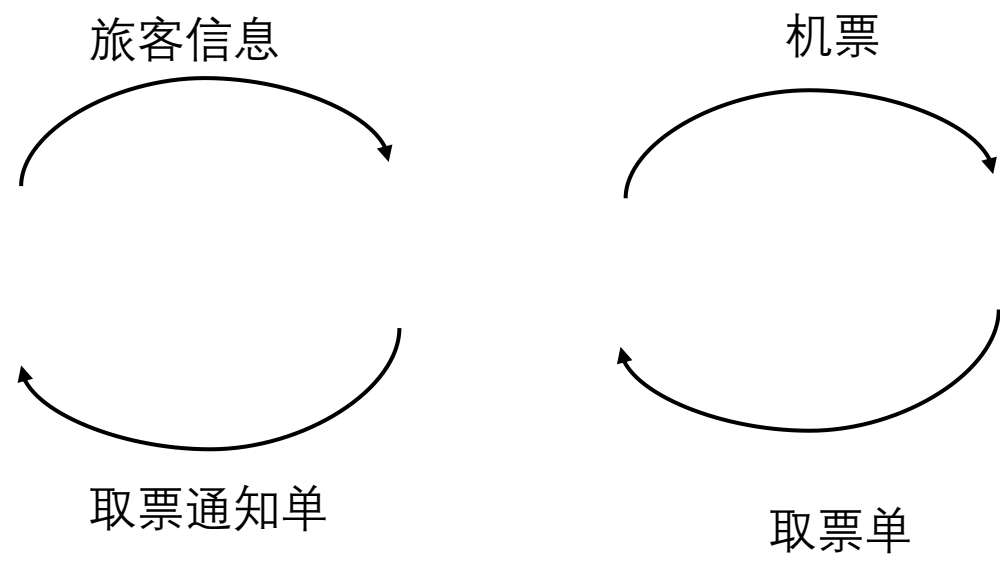
练习：机票预定系统顶层DFD

——只需要画上下文数据流图

系统功能：旅行社把预定机票的旅客信息输入机票预定系统；系统为旅客安排航班，打印出取票通知单，返给旅行社；旅客在飞机起飞前凭取票单取票，系统检验无误后，输出机票给旅客。

【腾讯文档】飞机票预定系统DFD

<https://docs.qq.com/form/page/DsGR6dm1qU25ab2xi>



练习：招生考试系统顶层DFD

——只需要画上下文数据流图

学校首先公布招生条件。考生根据自己的条件报名，之后系统进行资格审查，并通知学生资格审查信息；

对于资格审查合格的考生可以参加答卷，系统根据学校提供的试题及答案进行自动判卷，并给出分数及答题信息，供考生查询；

最后系统根据学校的录取分数线进行录取，并将录取信息发送给考生。

【腾讯文档】招生系统DFD

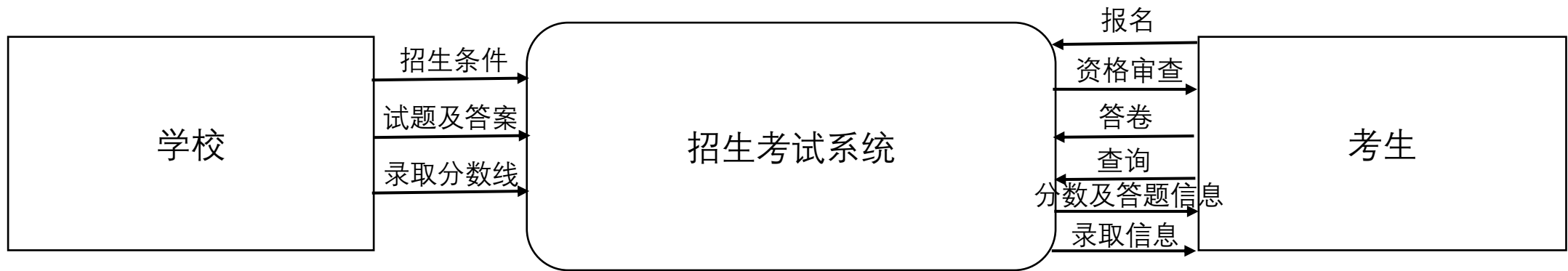
<https://docs.qq.com/form/page/DSE1UeXhEU3plcnlJ>

招生考试系统DFD

学校首先公布招生条件。考生根据自己的条件报名，之后系统进行资格审查，并通知学生资格审查信息；

对于资格审查合格的考生可以参加答卷，系统根据学校提供的试题及答案进行自动判卷，并给出分数及答题信息，供考生查询；

最后系统根据学校的录取分数线进行录取，并将录取信息发送给考生。



/-2

逻辑过程建模的过程 **Process Modeling**

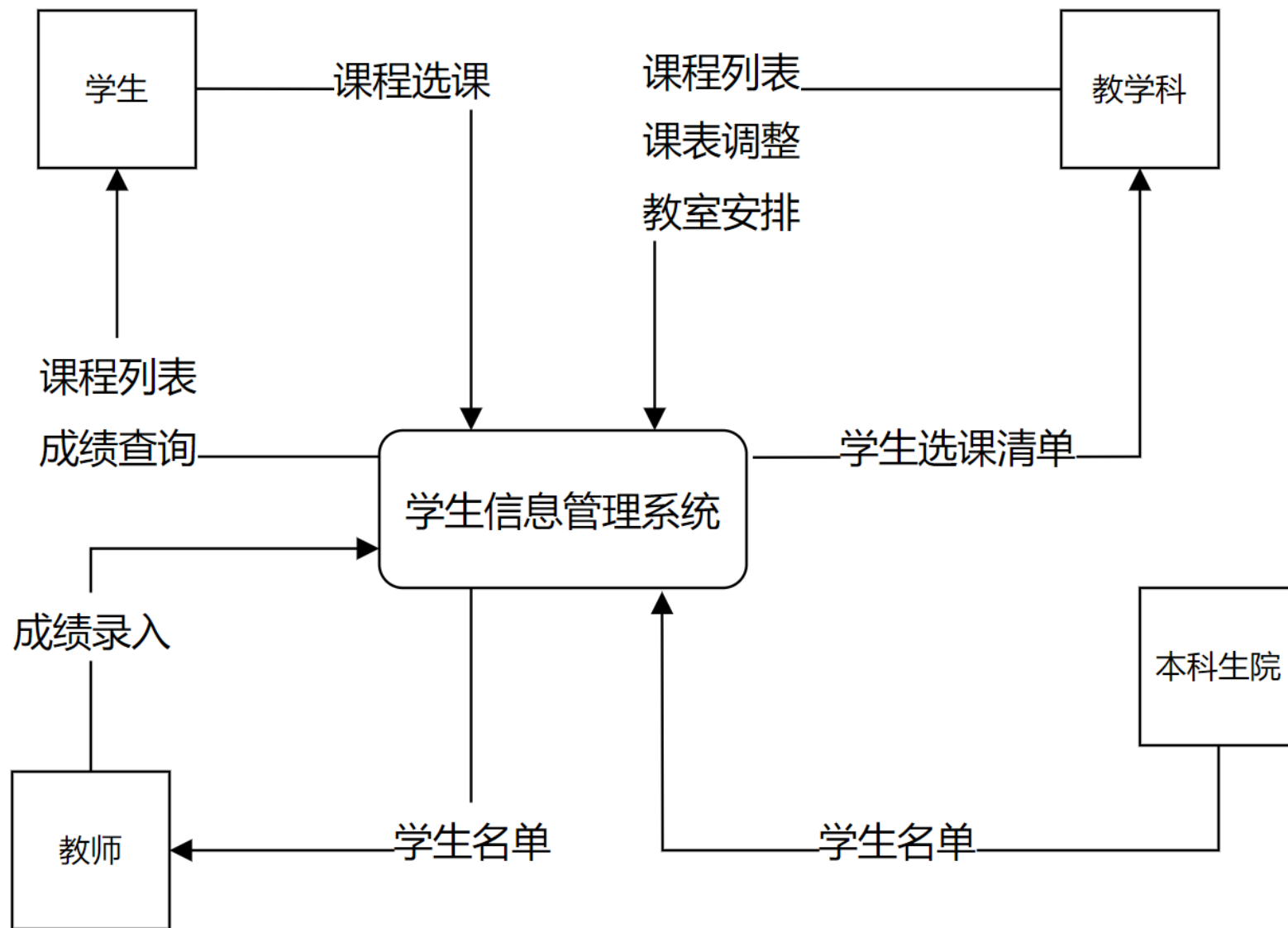
上下文数据流图

Context Data Flow Diagram

顶层数据流图

(Top-level data flow diagram)

描述了整个系统的作用范围,对系统的总体功能、输入和输出进行了抽象,反映了系统和环境的关系



自上而下结构

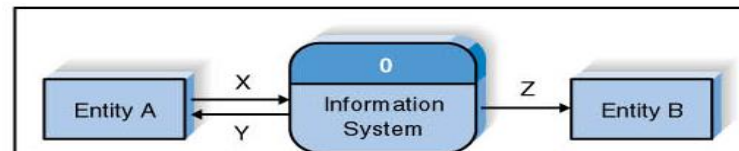
Top-Down Approach

顶层数据流图

(Top-level data flow diagram)

描述了整个系统的作用范围,对系统的总体功能、输入和输出进行了抽象,反映了系统和环境的关系

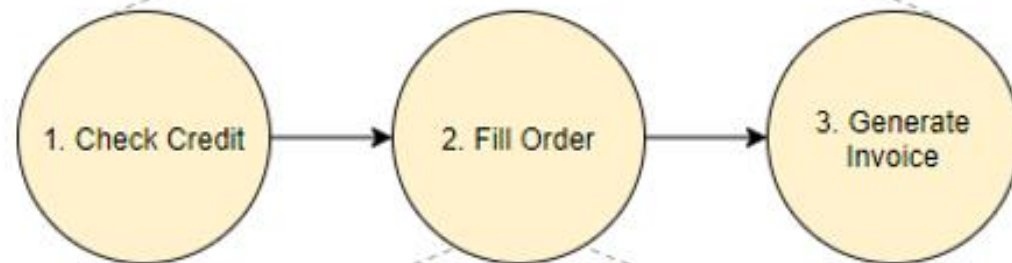
Context Diagram



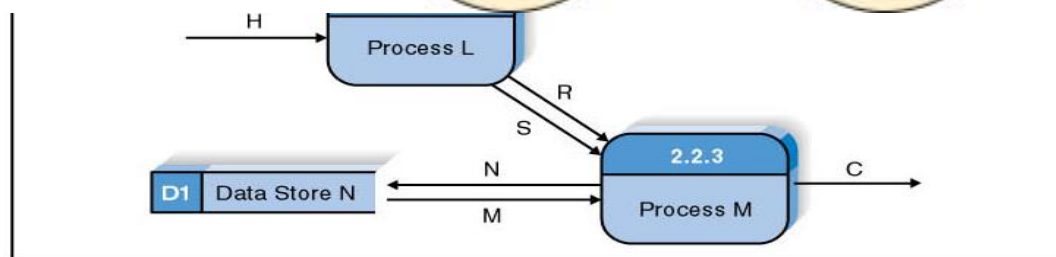
Context diagram (whole system)



Level 0 diagram

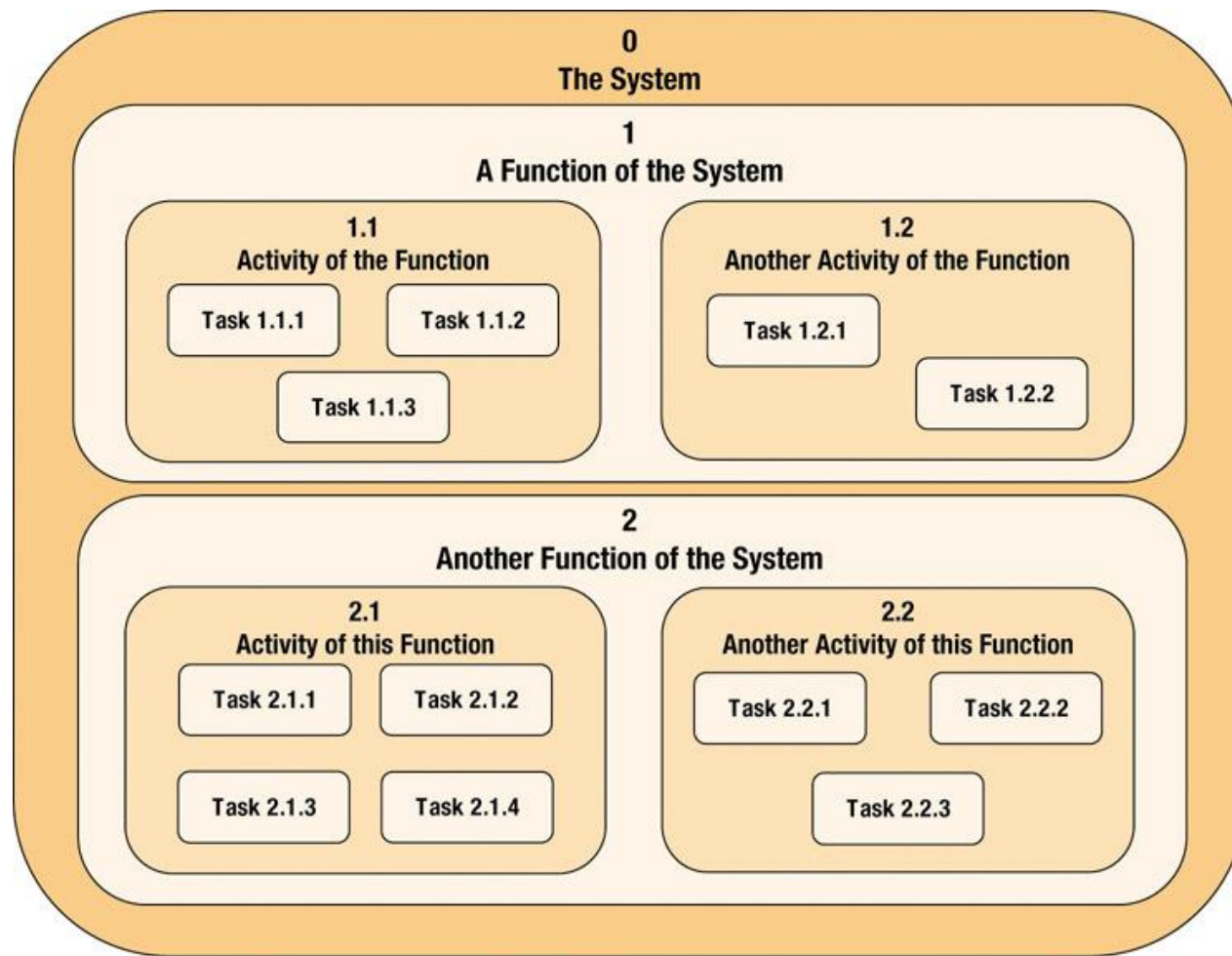


Level 1 diagram

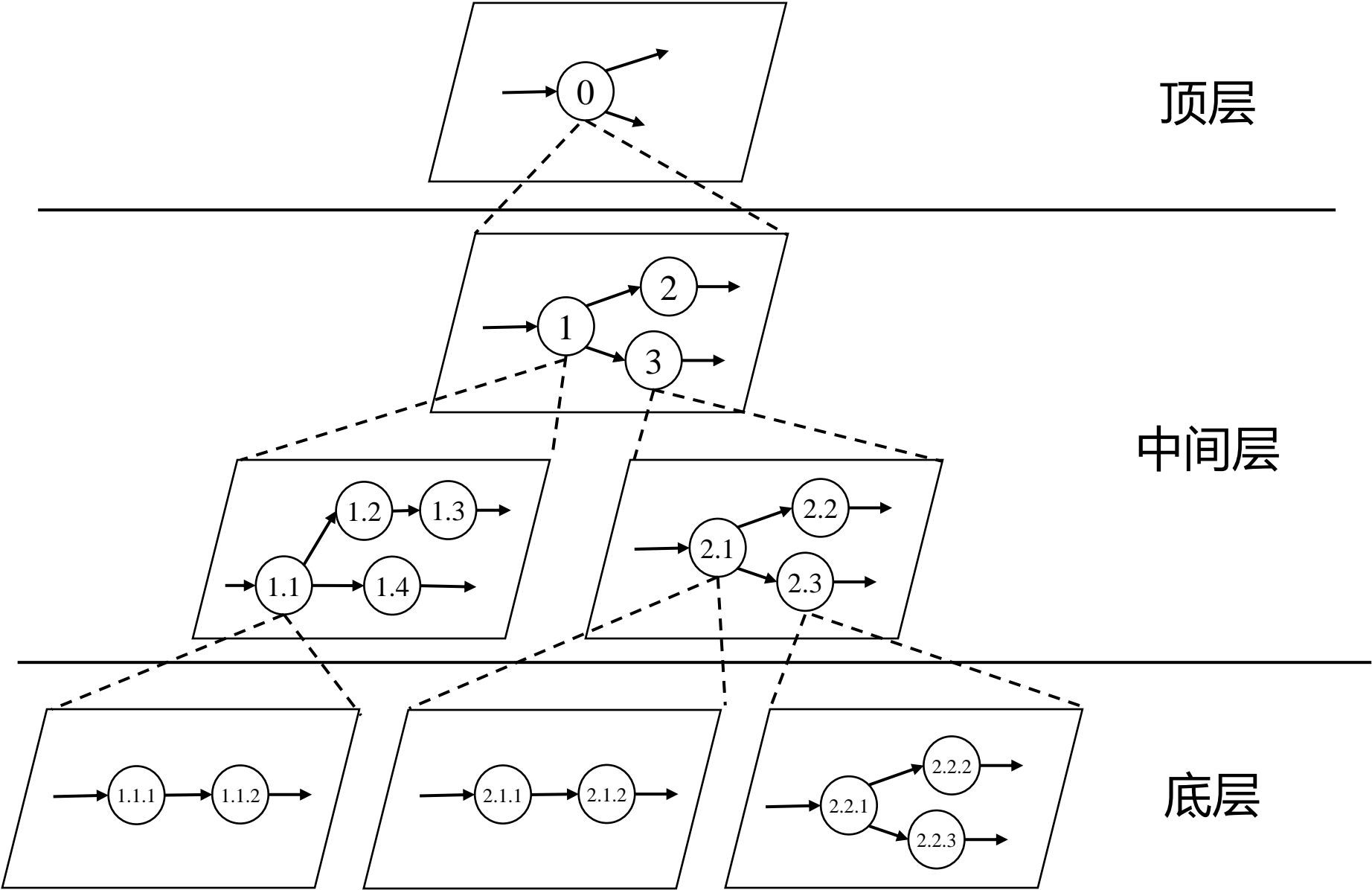


过程分解 Process Decomposition

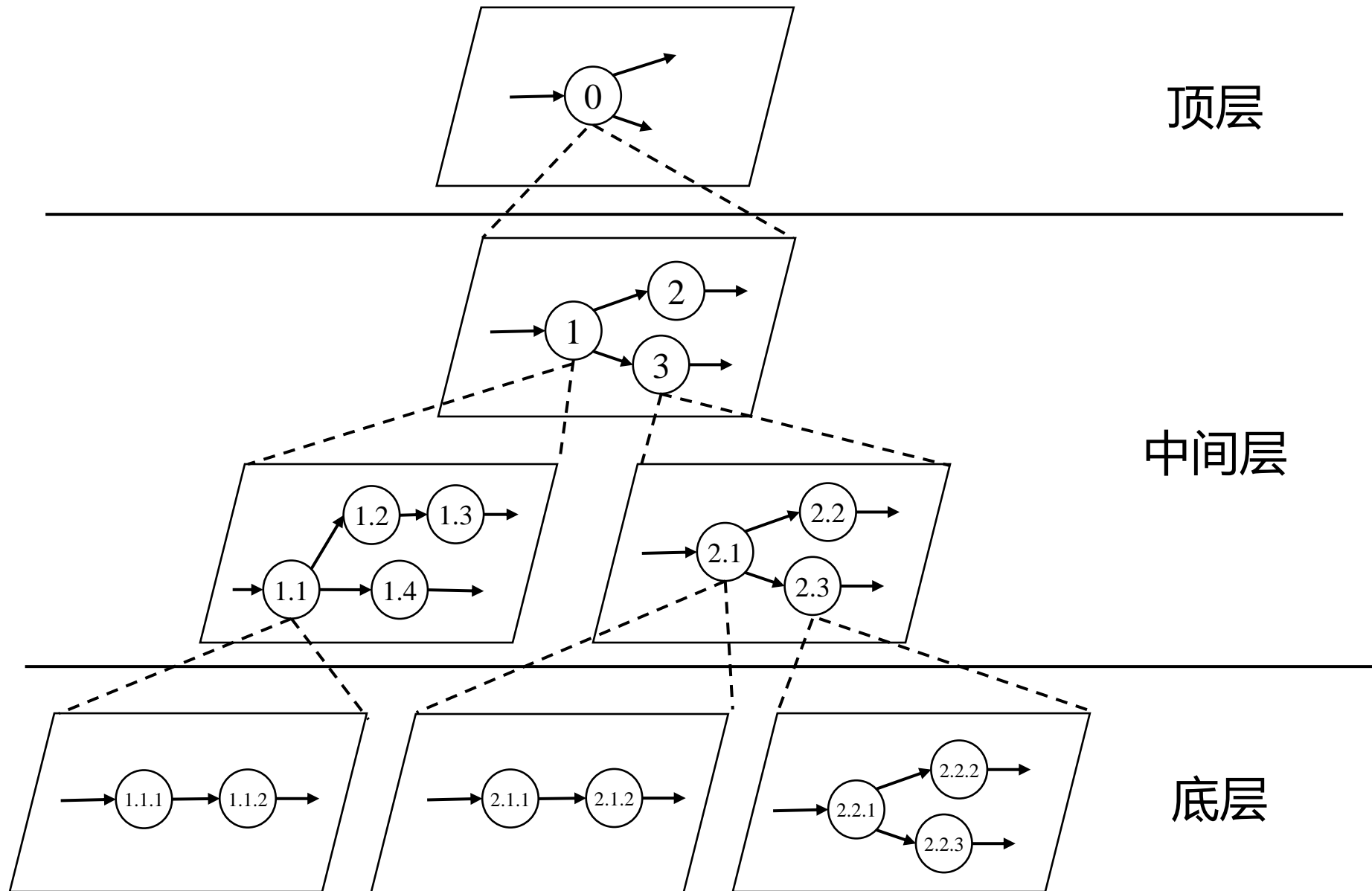
分解是将一个系统分解成它的组件子系统、过程和子过程的行动。每个层次的**抽象**都揭示了系统的细节



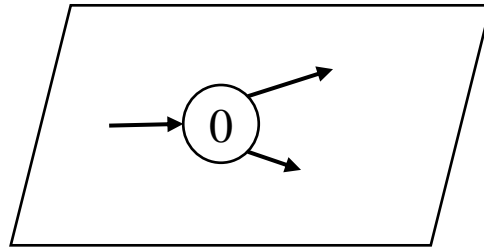
自上而下结构



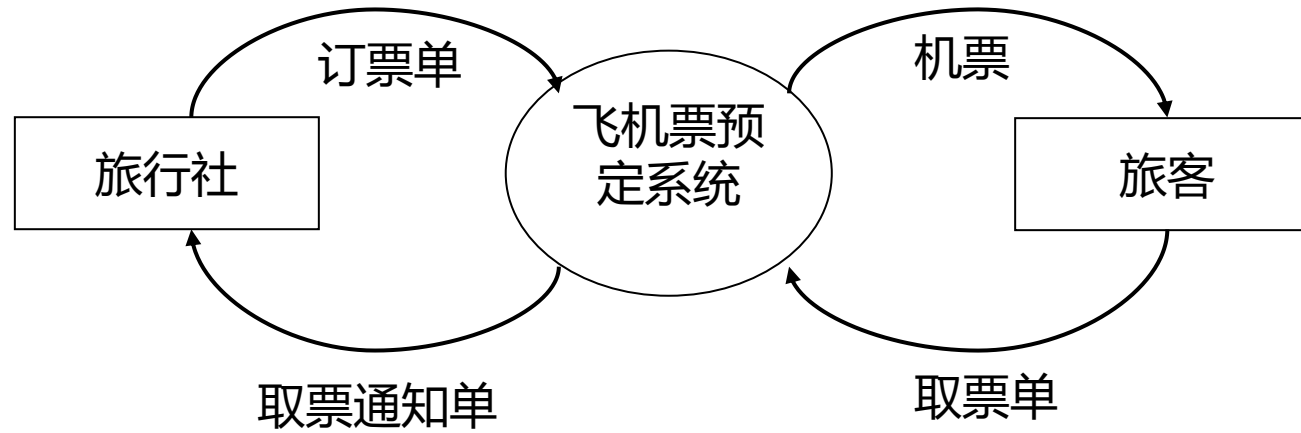
顶层



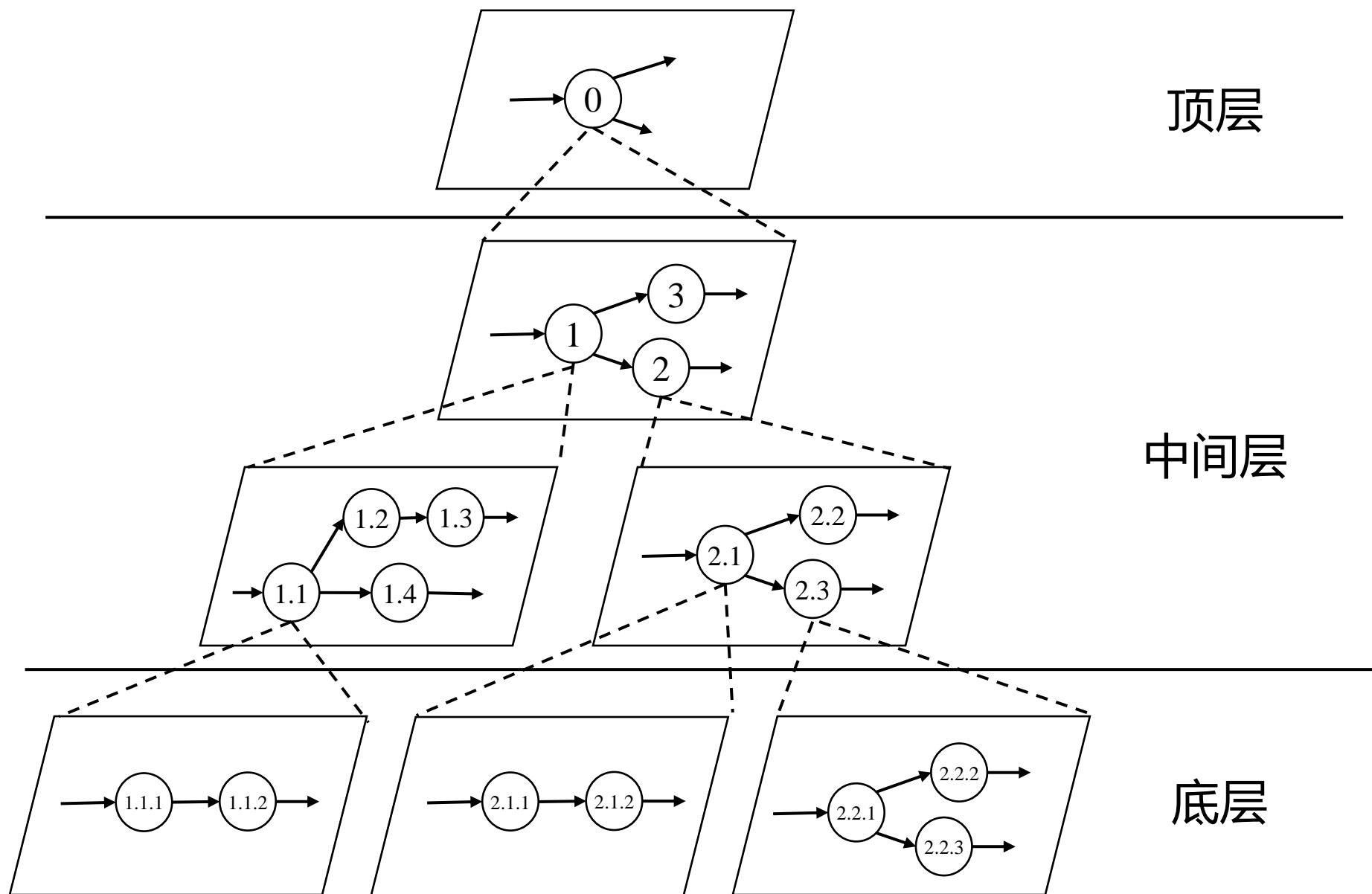
顶层

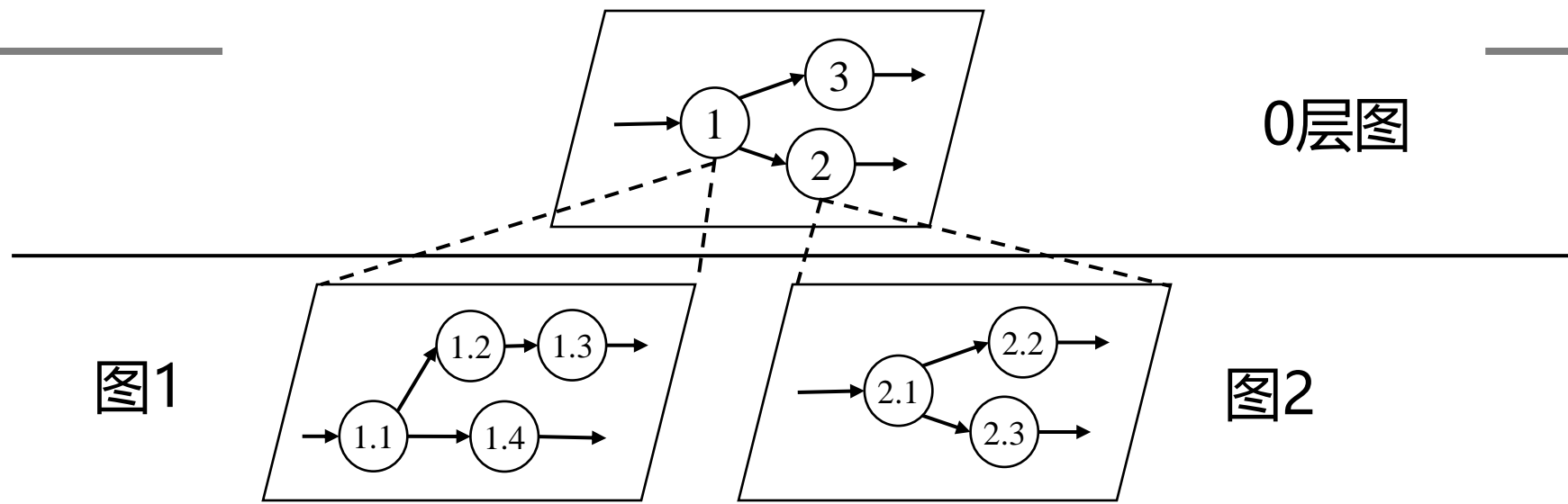


顶层



中间层





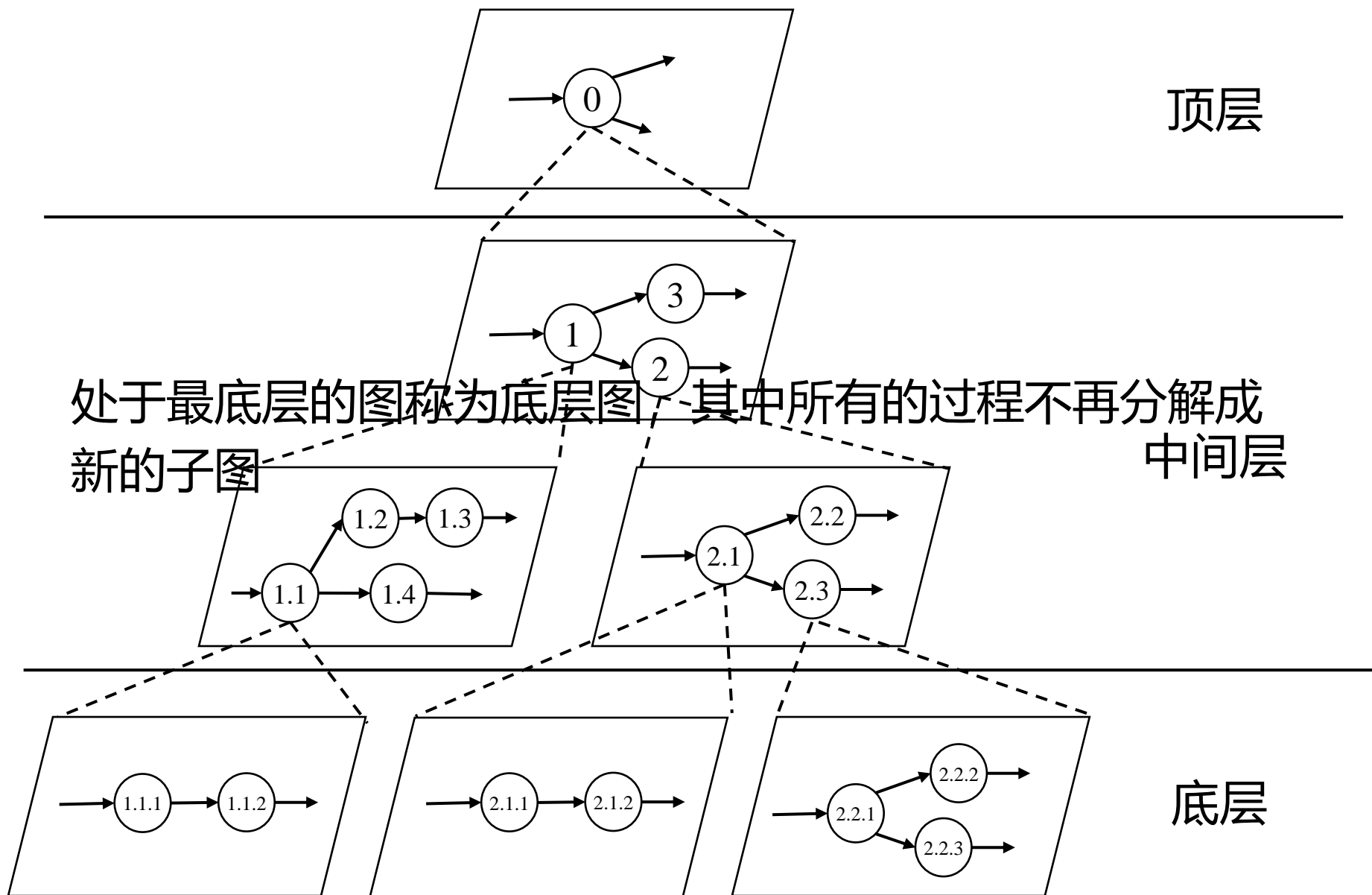
中间层图中至少有一个过程(也可以有多个)在下层图中分解成一张子图

顶层图中的过程经分解后的图称为0层图(只有1张),0层图中的过程编号分别为1, 2, 3, ...

子图号: 若父图中的过程编号x分解成某一子图, 则该子图号记为 "图x"

子图中过程的编号: 若父图中的过程编号为x的过程分解成某一子图, 则该子图中的过程编号分别为x.1、x.2、x.3...

底层



资格和水平考试的考务处理系统

—功能需求

对考生送来的报名单进行检查，返回不合格的报名单

对合格的报名单编好准考证号后将准考证送给考生，并将汇总后的考生名单送给阅卷站

对阅卷站送来的成绩清单进行检查，返回错误的成绩单，并根据考试中心制订的合格标准进行判定

制作考生通知单送给考生

进行成绩分类统计和试题难度分析，产生统计分析表，送给考试中心

资格和水平考试的考务处理系统

—功能需求

对考生送来的报名单进行检查，返回不合格的报名单

对合格的报名单编好准考证号后将准考证送给考生，并将汇总后的考生名单送给阅卷站

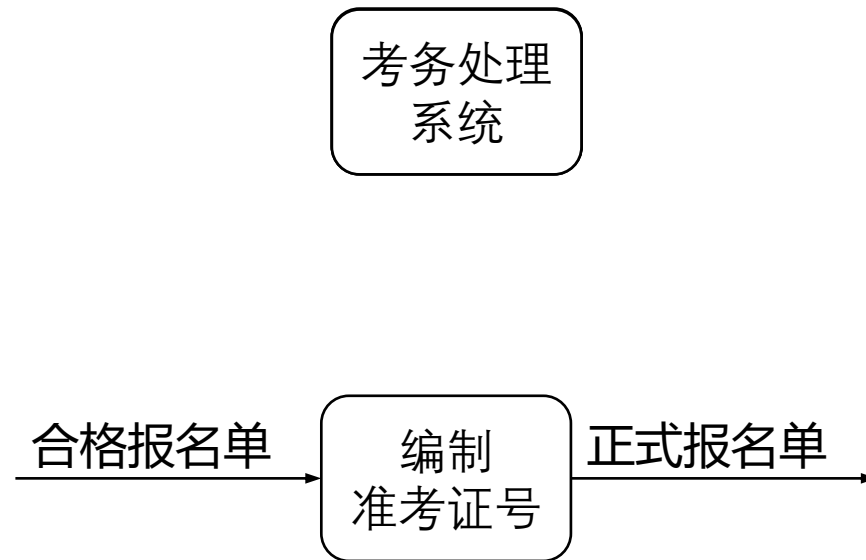
对阅卷站送来的成绩清单进行检查，返回错误的成绩单，并根据考试中心制订的合格标准进行判定

制作考生通知单送给考生

进行成绩分类统计和试题难度分析，产生统计分析表，送给考试中心

系统内部(0层图)

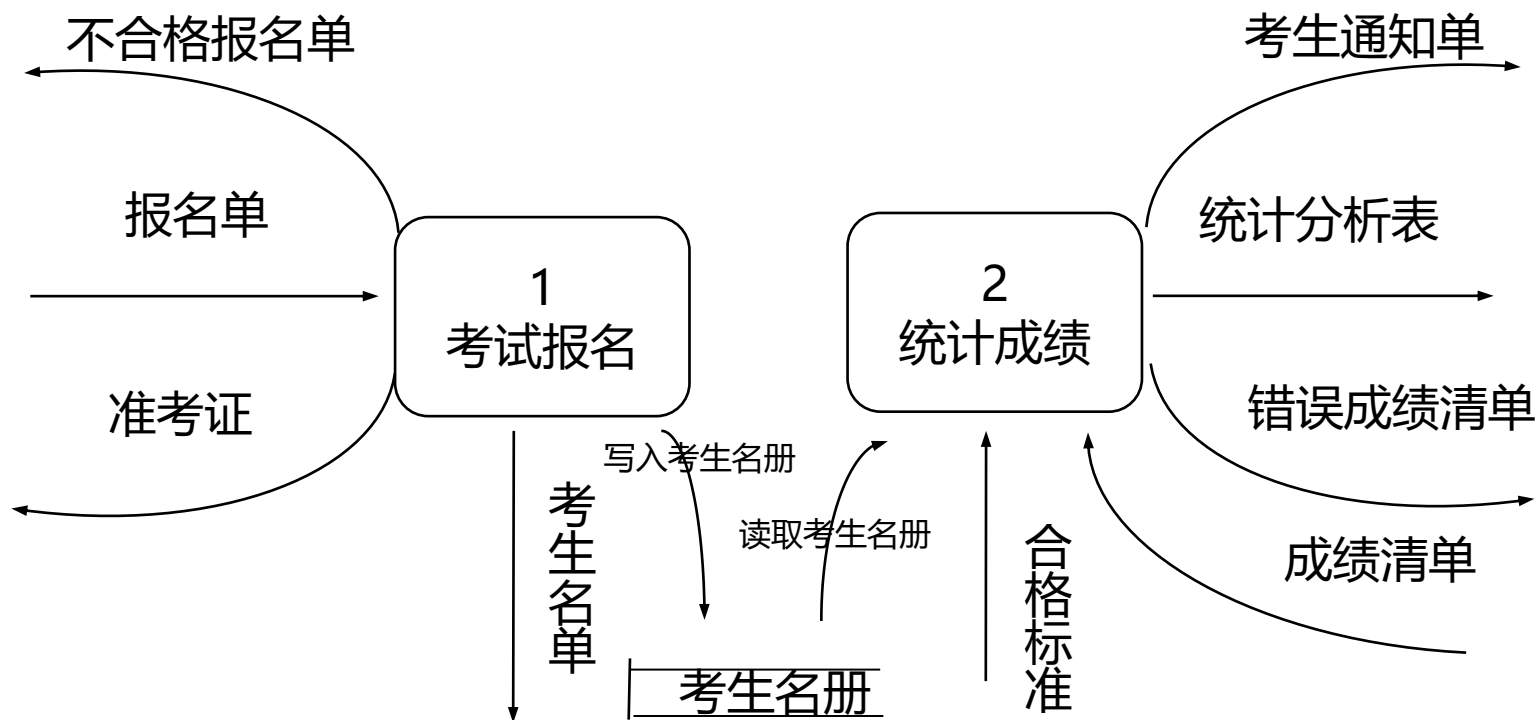
- 确定过程：将父图中某过程分解而成的子过程，根据功能分解或业务处理流程来确定过程



系统内部(0层图)

➤ 确定数据流

- ✓ 如果图图桌透程解而咸弱文图的数据图中烟相应输和输出数据流数据流都是图快是子图边界上的输入/输出数据流
- ✓ 在解解的图过程如果需要保存增添相应数据数据流表示的过程将这些数据数据成一个新的文件



过程内部(1...n层图)

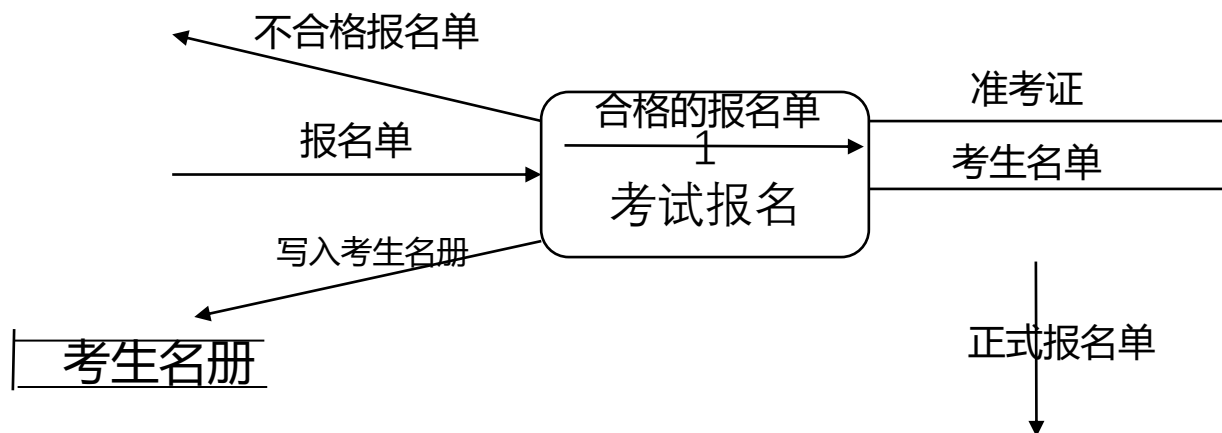
- 复杂的过程可以继续分解成DFD子图
- 分解方法
 - 将该过程看作一个小系统，该过程的输入/输出数据流就是这个假设的小系统的输入/输出数据流
 - 然后采用画0层图的方法，画出该过程的子图

以0层图中过程1(考试报名)为例

➤ 复杂的过程可以继续分解成子图，根据业务处理流程确定由过程的分解

➤ 与过程1相关的业务流程：首先检查考生送来的报名单并产生合格的报名单，然后根据报名单编准考证号并产生准考证，和正式报名单，最后登记考生产生的考生名单并存入考生名册(文件)

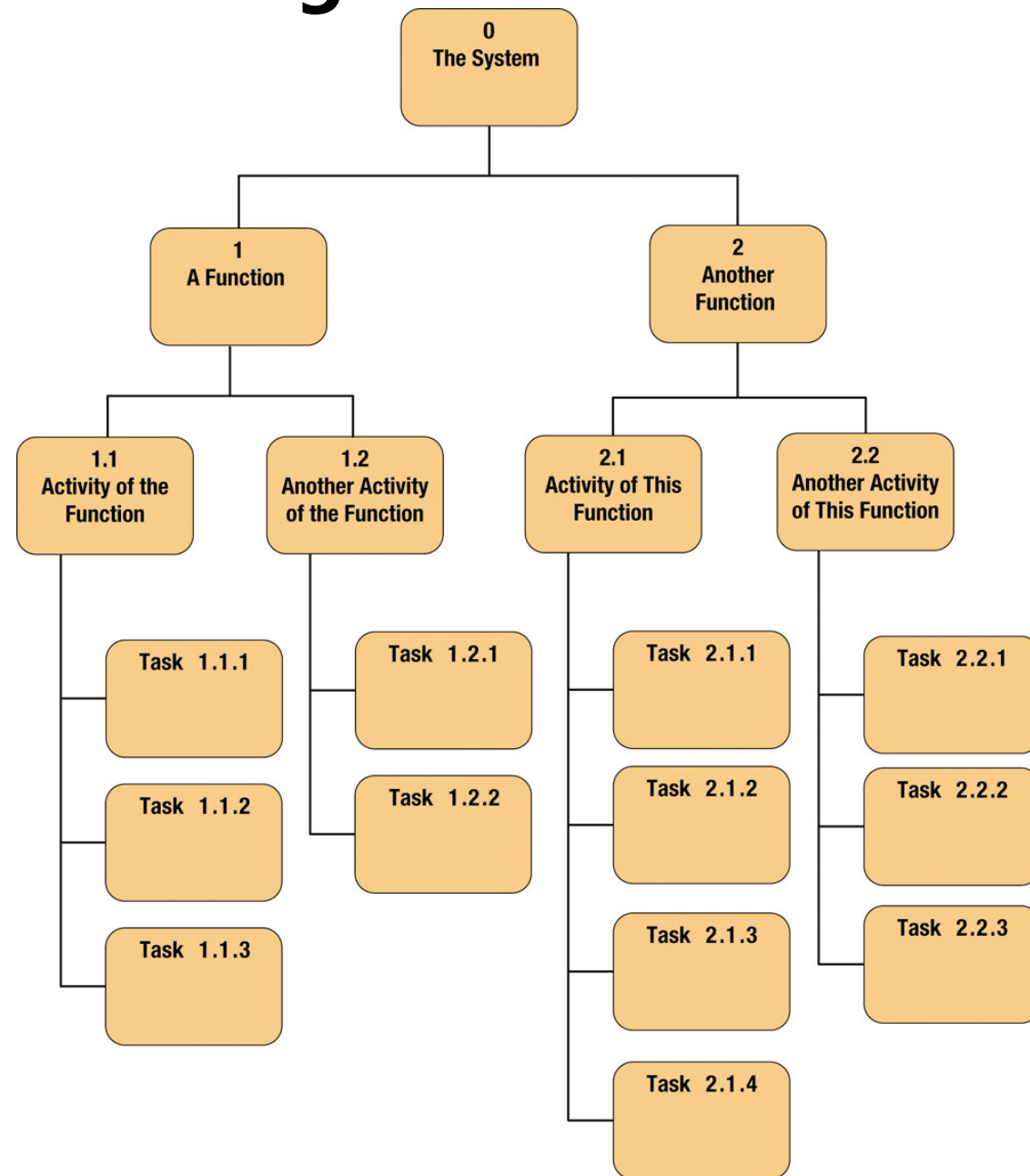
登记考生

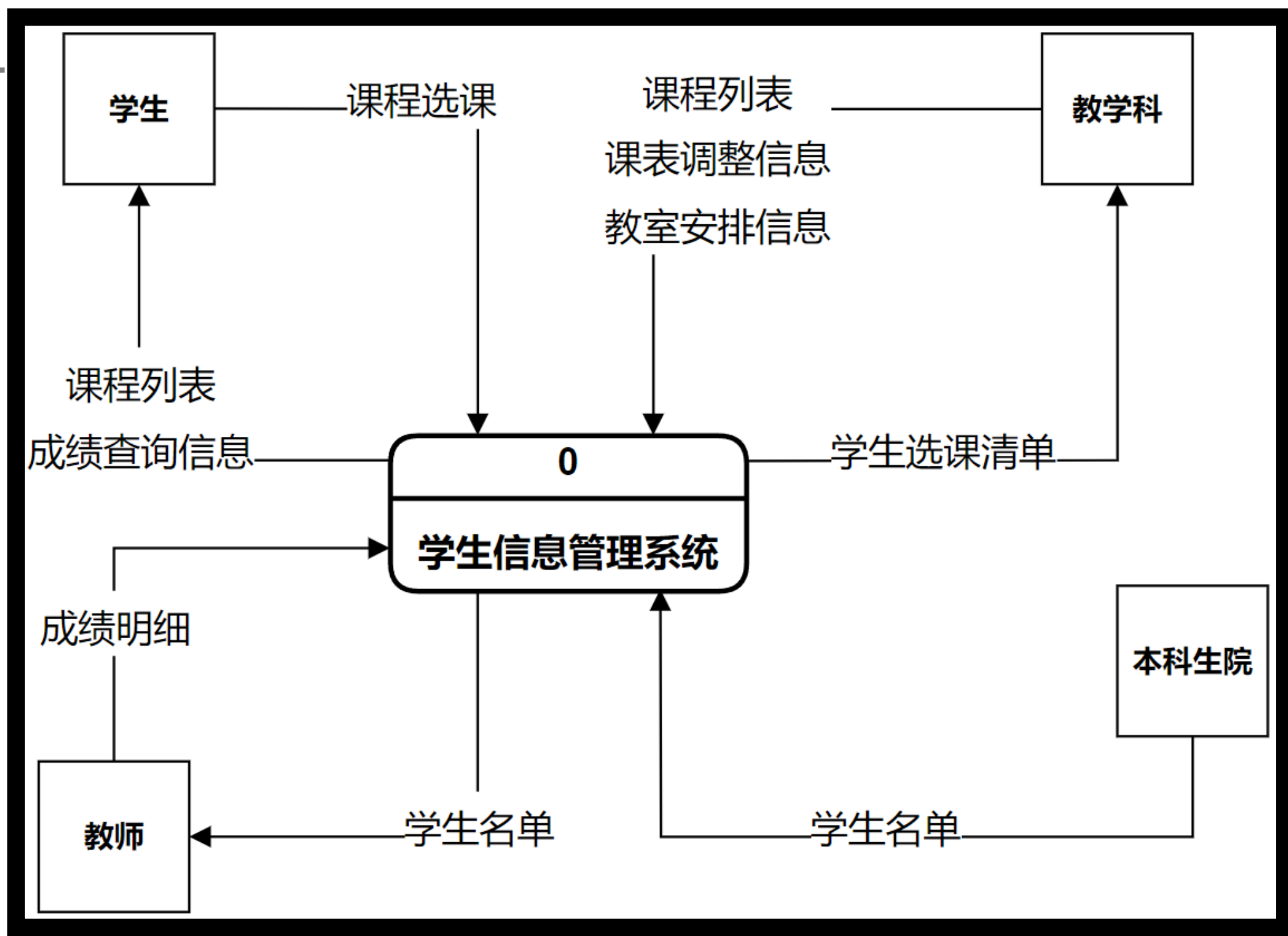


分解图 Decomposition Diagram

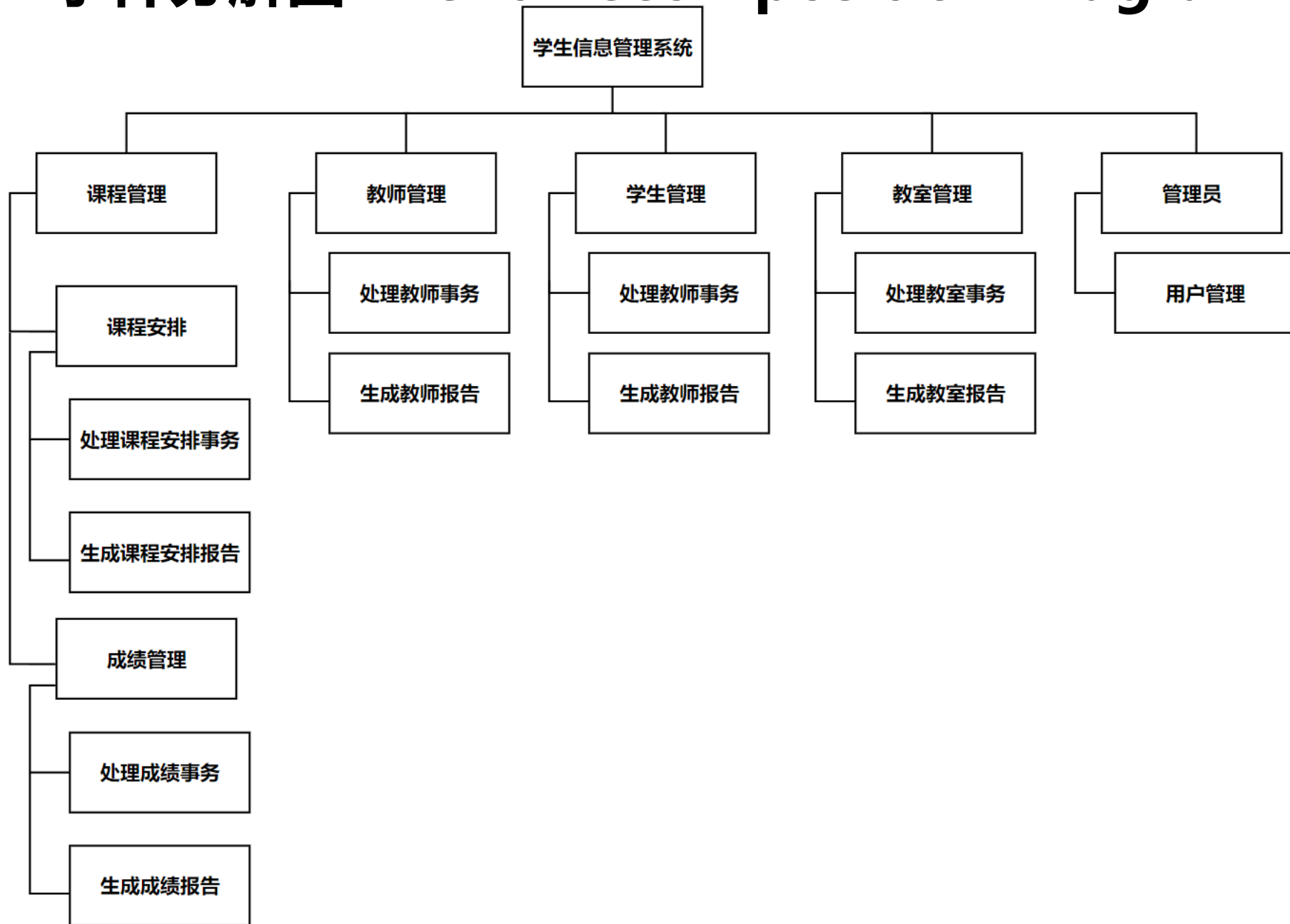
分解图，又称**层次图**，显示了一个系统自顶向下的功能分解和结构。

- 本质上是一种规划工具，用于更详细的过程模型
- 每个过程或者是**父过程**，或者是**子过程**，或者二者都是
- 父进程**必须**有两个或多个子过程
- 图中的连线不含箭头且没有命名





事件分解图 Event Decomposition Diagram



事件响应或用例清单

The Event-Response or Use-Case List

构建分解图后，需确定系统必须响应的业务事件

通常存在以下三种事件类型：

外部事件由外部代理发起，当事件发生时出现一个到系统的输入数据流

例：事件“客户发出一个新订单”来自于外部代理“客户”的输入数据流的形式而识别

时序事件以时间为基础触发过程，当事件发生时产生一个输入控制流

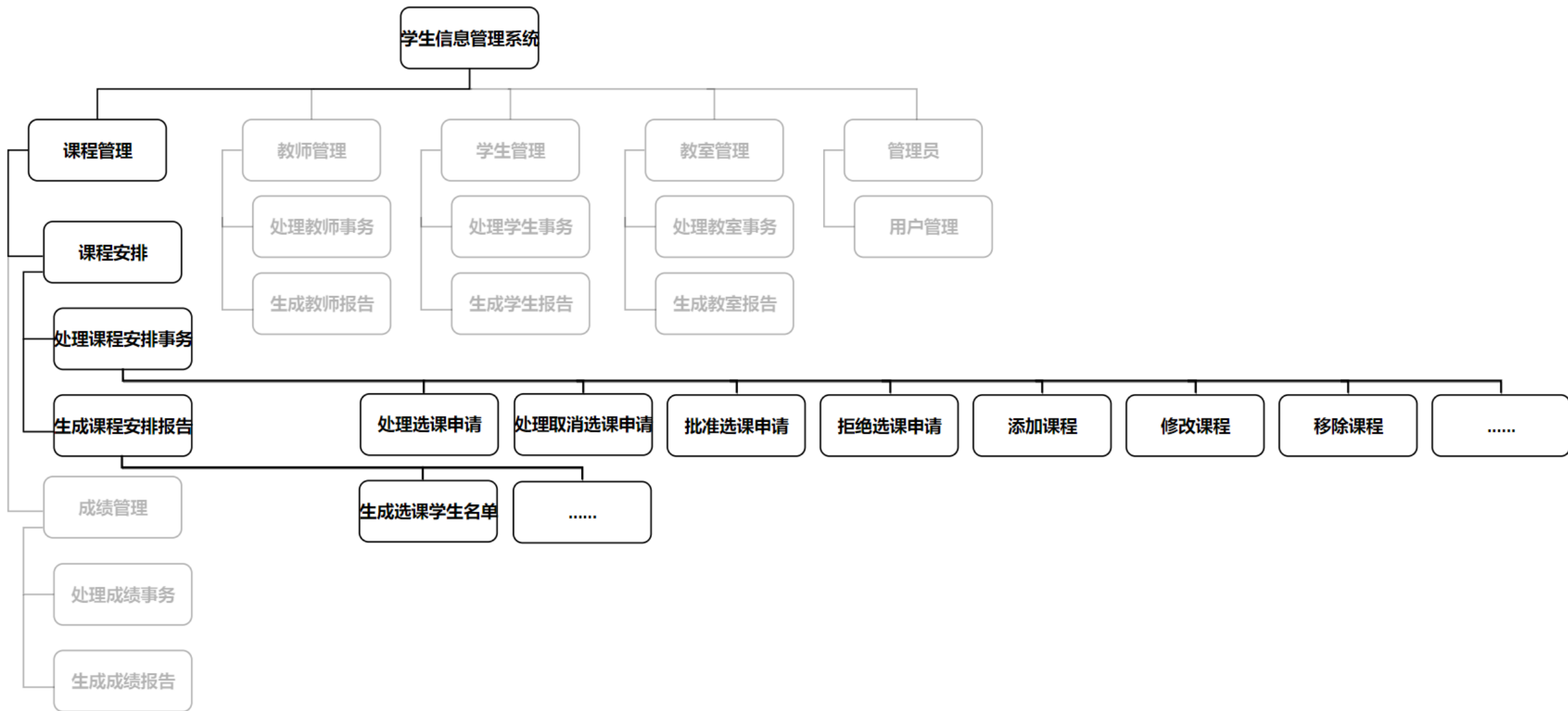
例：“月末”、“每晚”或“提醒客户支付订单的时间”

状态事件基于系统从一个状态或条件到另一个状态或条件的转变触发过程，用一个输入控制流表示

用例建模中的参与者等价于DFD中的外部代理

参与者(外部代理)	事件(用例)	触发器	响应
学生	提交一个选课申请	选课申请创建	生成“选课申请创建提醒” 在数据库中创建“选课申请”
学生	撤回一个选课申请	选课申请取消	生成“选课申请取消提醒” 在数据库中删除“选课申请”

事件分解图 Event Decomposition Diagram

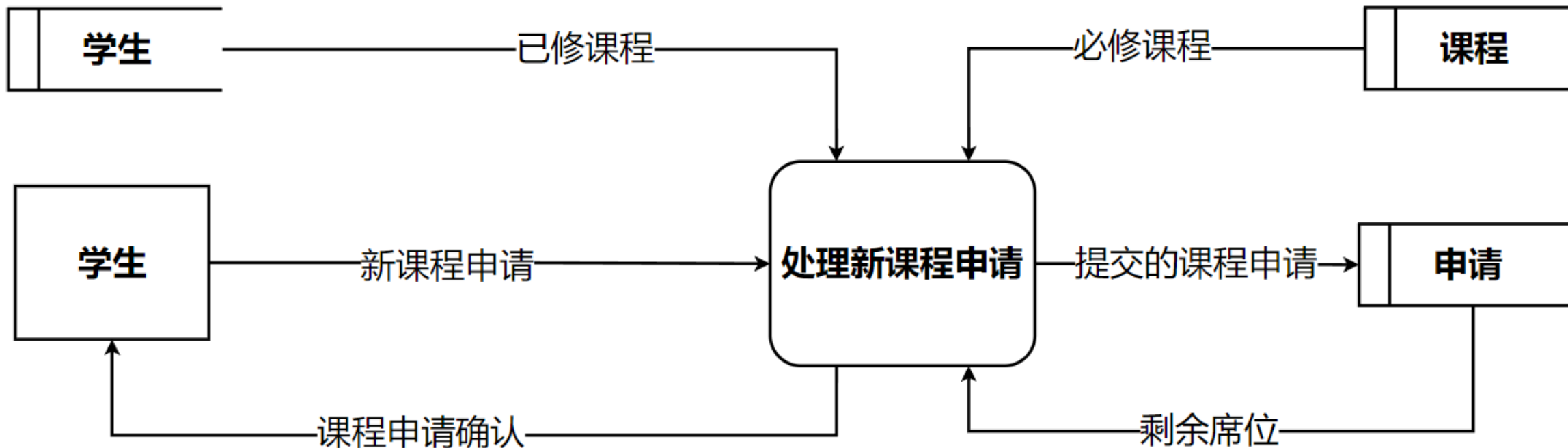


事件图 Event Diagram

以分解图为提纲，可以为每个事件过程绘制一个事件图。

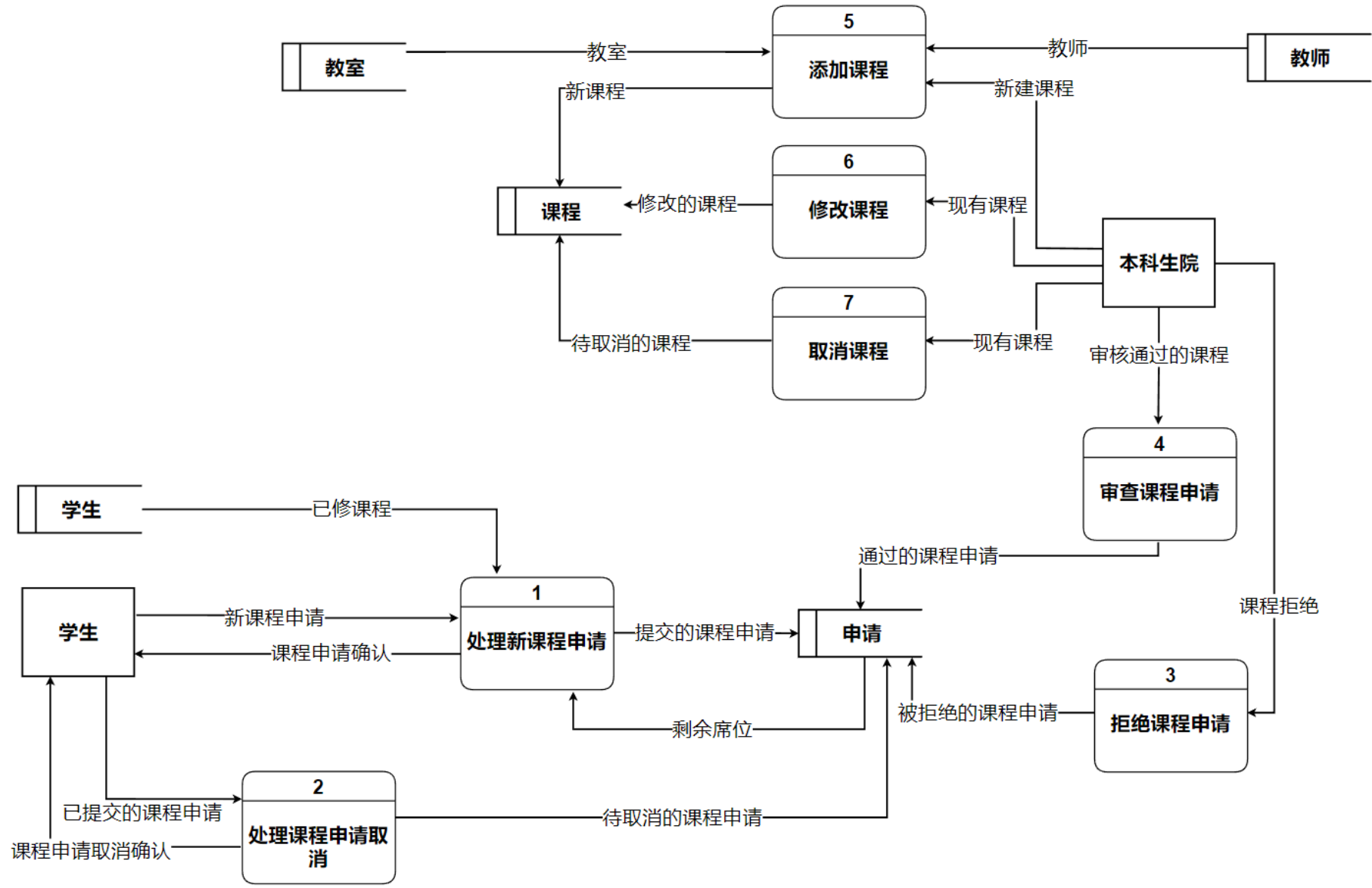
事件图是一个事件的上下文图，显示了事件的输入、输出和数据存储交互。

- 数据流名称应该是唯一的，并采用描述性单数名词和名词短语



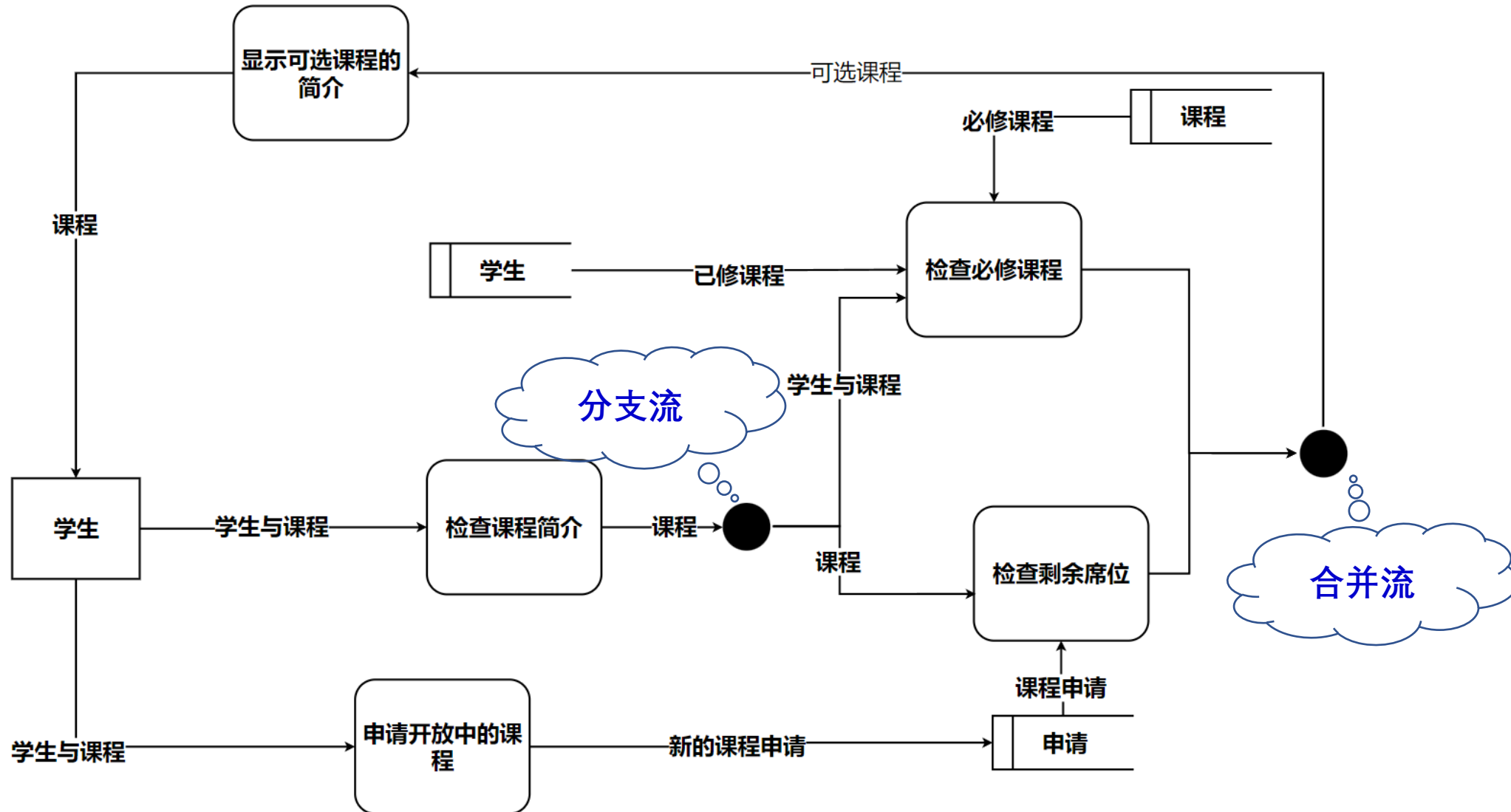
系统图 System Diagram

系统图是原始的上下文图，在单张图中显示了系统的所有事件。



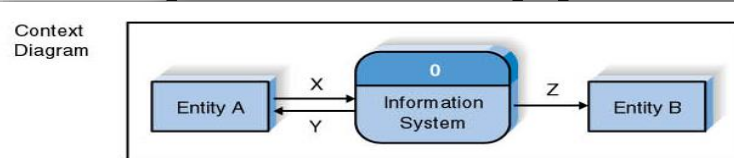
基本图 Primitive Diagram

系统图上的某些（复杂）事件过程可能会扩展成一个基本数据流图以显示该事件详细的过程需求。

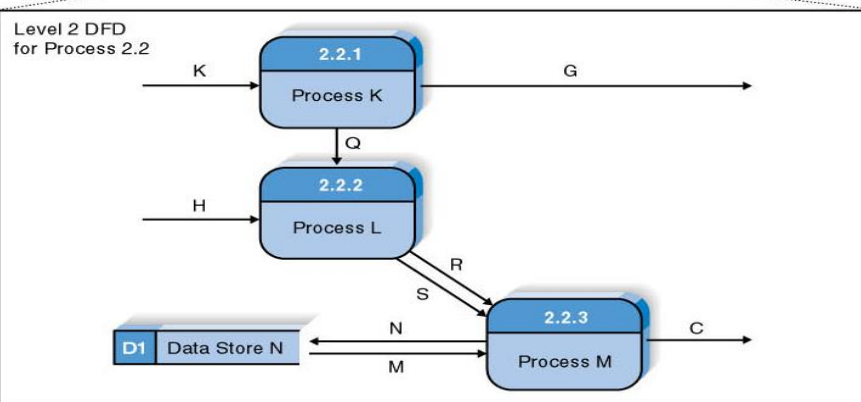
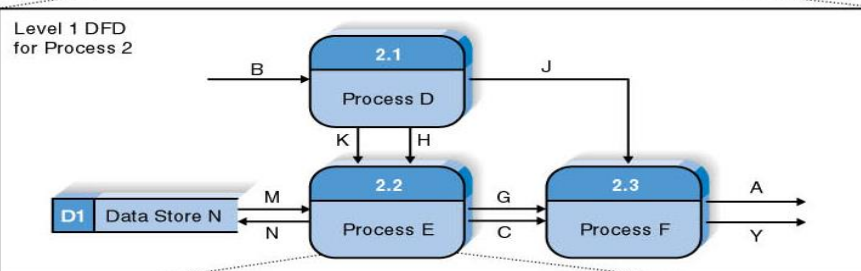
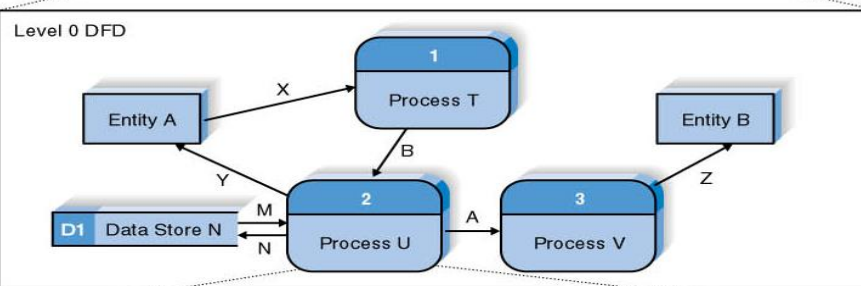


自上而下结构 Top-Down Approach

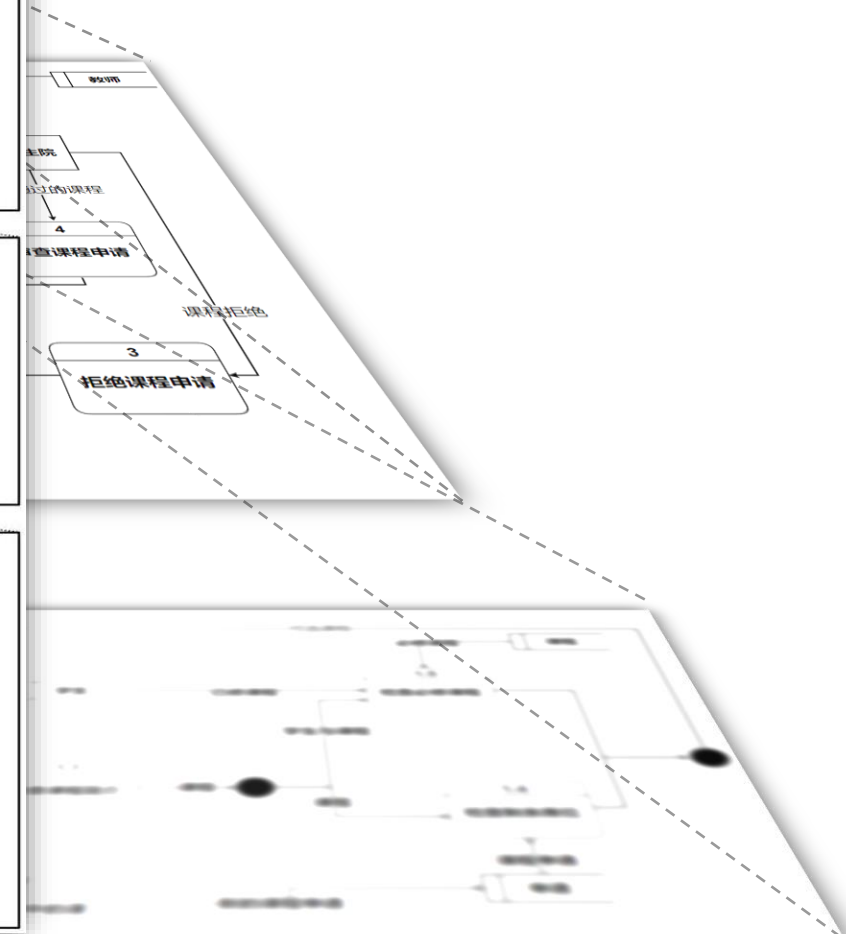
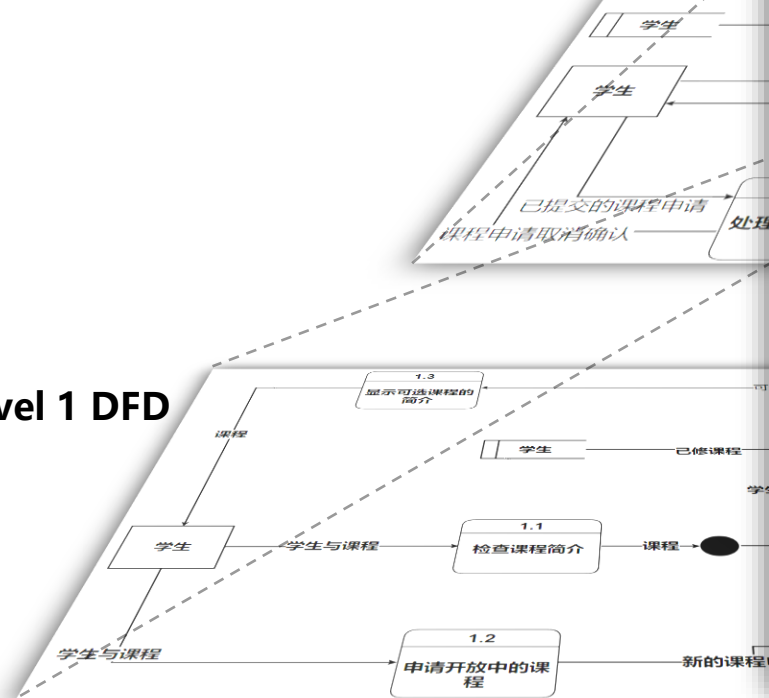
上下文数据流图 (Context DFD)



Level 0 DFD



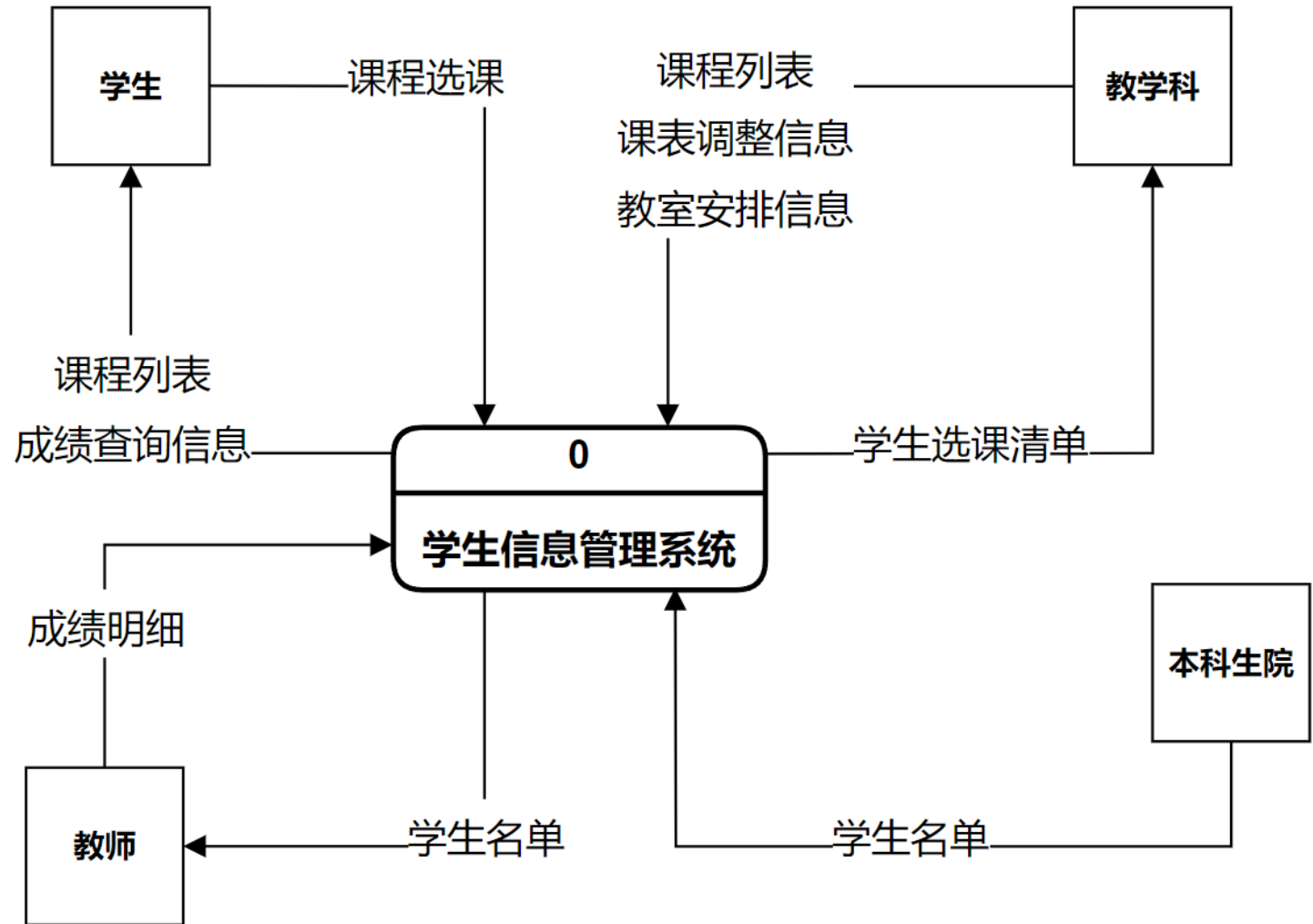
Level 1 DFD



事件驱动的过程建模策略 (1/4)

Event-Driven Process Modeling Strategy

自上而下结构 Top-Down Approach

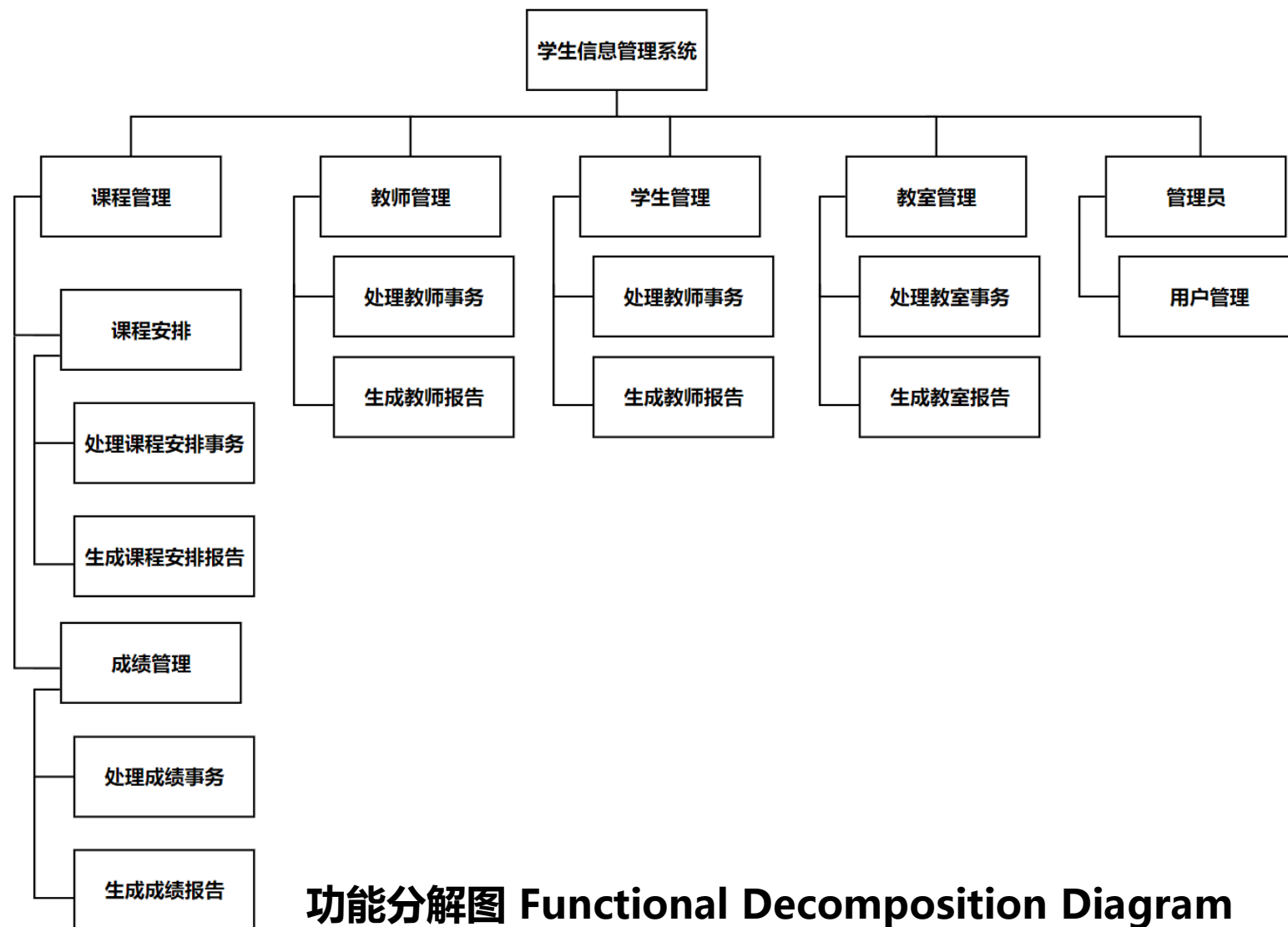


上下文数据流图 (Context DFD)

事件驱动的过程建模策略 (2/4)

Event-Driven Process Modeling Strategy

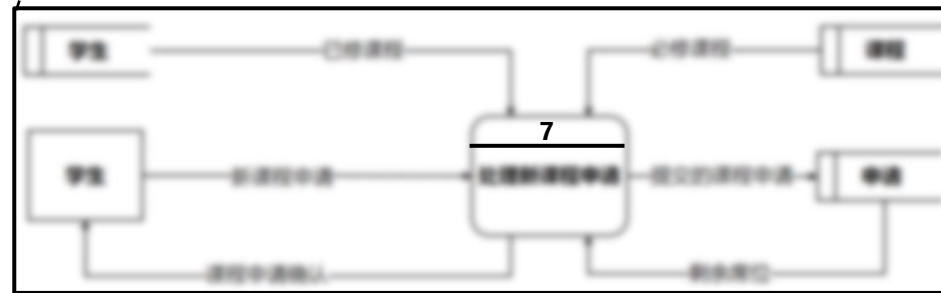
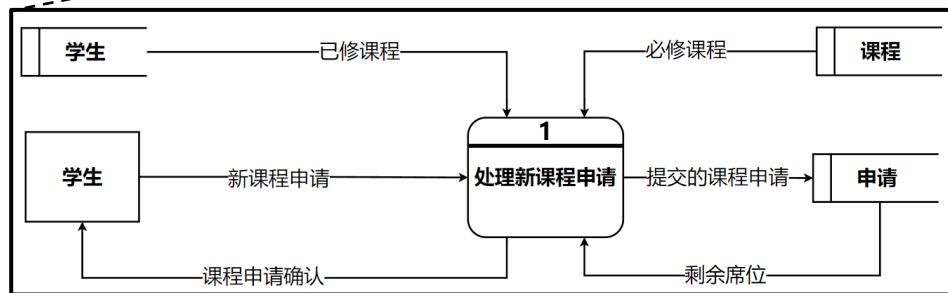
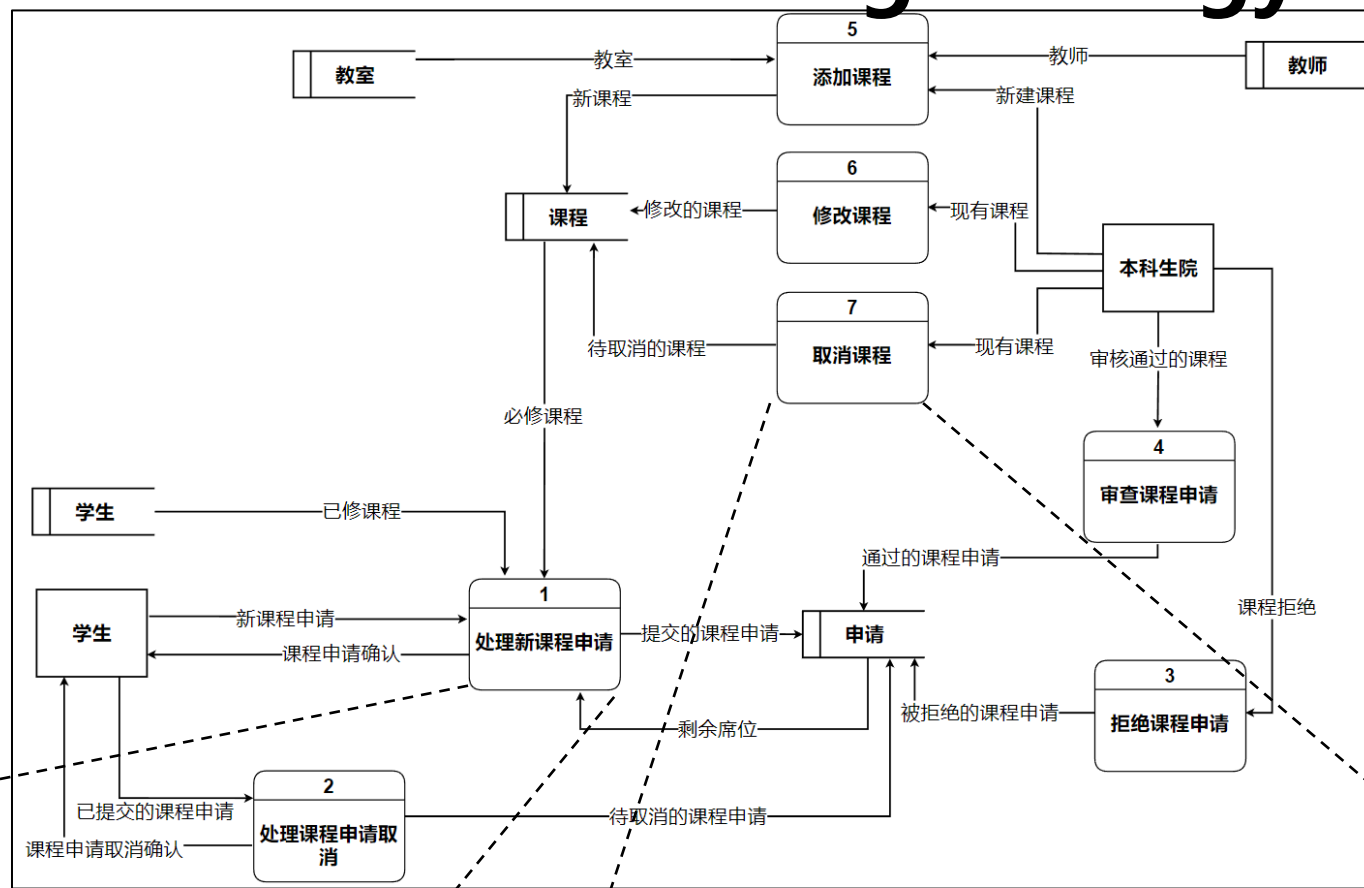
自上而下结构
Top-Down
Approach



事件驱动的过程建模策略 (3/4)

Event-Driven Process Modeling Strategy

事件图与系统图 Event Diagram & System Diagram

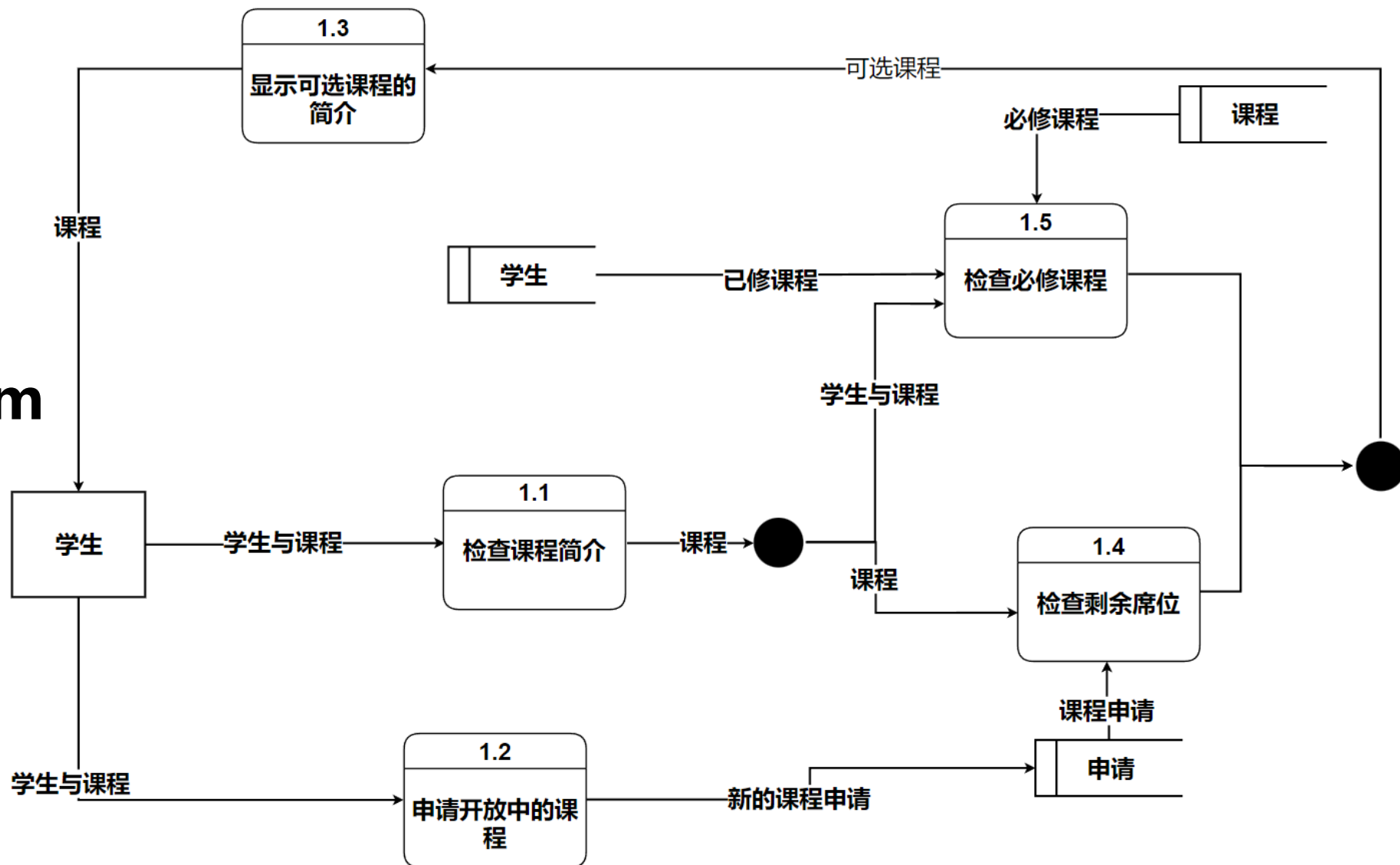


事件驱动的过程建模策略 (4/4)

Event-Driven Process Modeling Strategy

基本图

Primitive Diagram



事件驱动的过程建模策略

Event-Driven Process Modeling Strategy

- ① 构造系统上下文数据流图，建立初始的系统范围
- ② 绘制功能分解图，将系统划分成逻辑子系统和/或功能
- ③ 列出事件响应或用例清单以确定系统必须提供响应的业务事件
- ④ 为每个事件构造一个事件图
- ⑤ 通过合并事件图构造系统图（对于较大的系统可以是子系统图）
- ⑥ 对需要进一步处理细节的事件过程构造基本图，显示单个事件的所有基本过程、数据存储和数据流