

BUPT
TSEG

软件工程 模型与方法

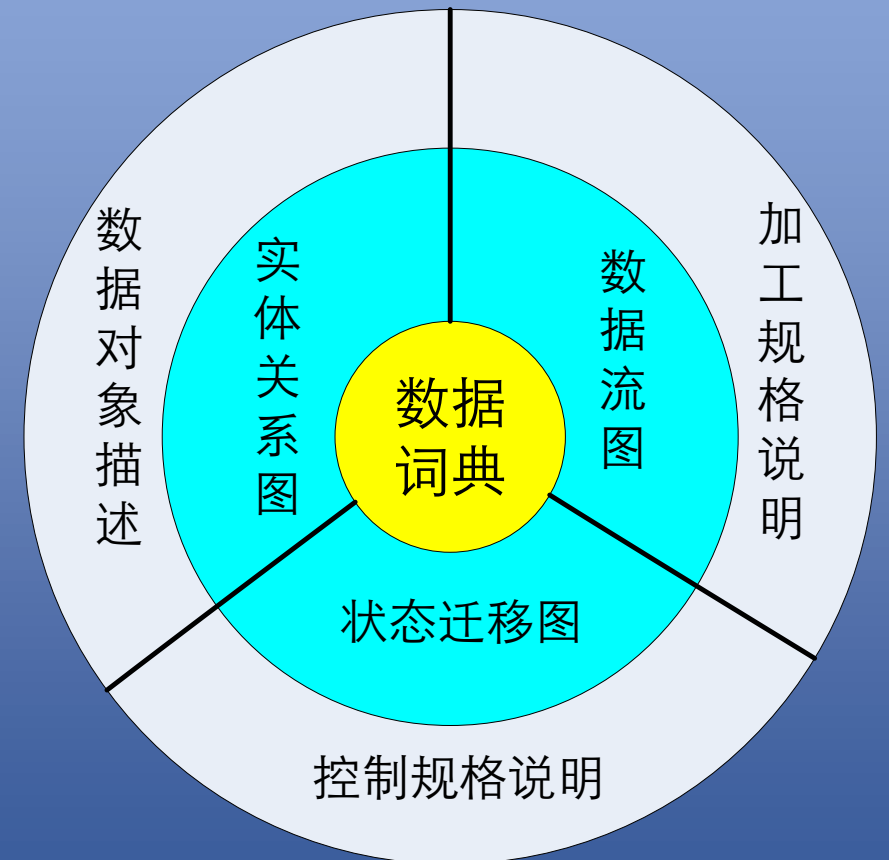
Models & Methods of SE

结构化需求分析方法

- 结构化需求分析的由来
- 结构化分析模型的组成
 - 数据建模
 - 功能建模
 - 行为建模
- 数据词典
- 软件需求规格说明书

- 分析建模开始于20世纪60-70年代，但结构化分析方法的第一次出现是作为另一个重要课题：“结构化设计”的附属品。
- 其目的是需要一种图形符号体系来表示数据和对数据进行变换的处理，这些处理最终能被映射到软件体系结构的设计中。
- 结构化分析这个词汇最初由Douglas Ross提出，由DeMarco进行了推广。

- 需求分析的分析模型必须达到三个主要目标：
 - 描述客户的需求；
 - 建立创建软件设计的基础；
 - 定义在软件完成后可以被确认的一组需求。
- 结构化分析模型的组成



- 概念性数据模型是一种面向问题的数据模型，是按照用户的观点来对数据和信息建模。其表示方法称为实体-关系 (Entity-Relation) 法，也称为实体关系模型。
- 它描述了从用户角度看到的数据，反映了用户的现实环境，但与在软件系统中的实现方法无关。
- 软件系统本质上是信息处理系统，即对数据进行处理系统，因此在开发过程中必须考虑以下两方面的问题：
 - “数据”
 - 需要有哪些数据？
 - 数据之间有什么联系？
 - 数据本身有什么性质？
 - 数据结构等
 - 对数据的“处理”
 - 对数据进行哪些处理？
 - 每个处理的逻辑功能是什么？

- 数据建模反映了与任何数据处理应用相关的一组特定问题：
 - 系统处理哪些主要的数据对象？
 - 每个数据对象的组成如何？
 - 哪些属性描述了这些数据对象？
 - 这些数据对象当前位于何处？
 - 每个数据对象与其他数据对象有哪些关系？
 - 数据对象和变换它们的处理之间有哪些关系？

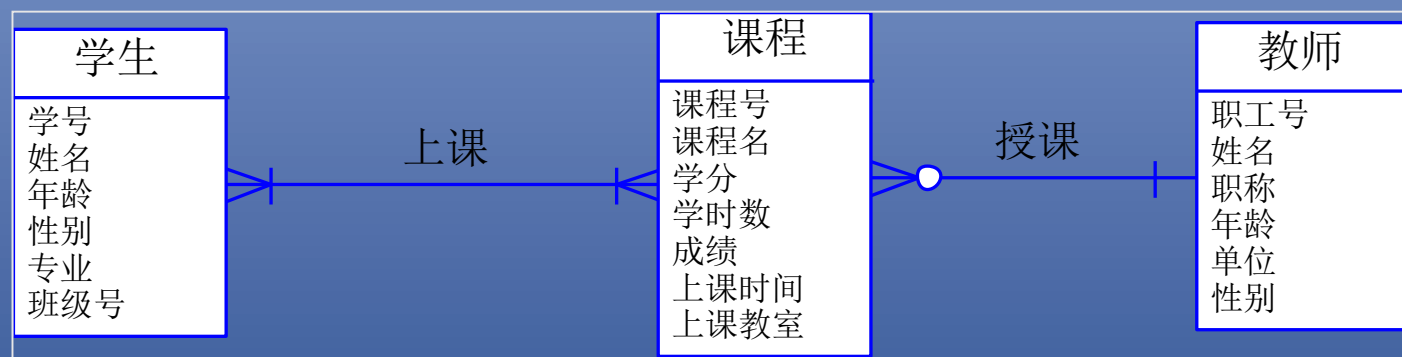
- 数据对象描述包括了数据对象的名称及其所有属性。通常将数据对象简称为“实体”，其具体表现可以是：
 - 外部实体：产生或使用消息的任何事物；
 - 事物：例如建筑物、汽车等物体；
 - 事件：例如警报；
 - 角色：例如老师、学生、销售等；
 - 组织单位：例如学校教务处、财务处等；
 - 地点：例如仓库、停车场等；
 - 结构：例如文件、档案等。

- 每个数据对象都具有一些区别于其他数据对象的特征和性质，这些特征称为数据对象的属性。它可用于：
 - 命名数据对象；
 - 描述数据对象实例；
 - 建立与其它数据对象的联系；
- 数据对象之间可以存在某种特定的连接，称之为数据对象的关系。
- 关系是由被分析问题的语境定义的。

- 数据建模的基本元素：数据对象、属性和关系提供了理解问题信息域的基础，但还必须了解数据对象之间出现的次数有无必然的联系，即实体-关系对的基数。
- 基数通常简单地表达为“一”或“多”。考虑到“一”和“多”的所有组合，两个实体可能的关联如下：
 - 一对一：例如人和身份证件的关系；
 - 一对多：例如父母与孩子的关系；
 - 多对多：例如学生和老师的关系；

以图形的形式表示

- ER图的主要目的是以图形的形式表示实体与实体之间的关系。
- ER图最初是由Peter Chen（陈品山）为关系数据库系统的设计提出的，标识了一组基本的构件：实体、属性、关系。
- 例如，教学管理系统中的教师、学生和课程之间的关系



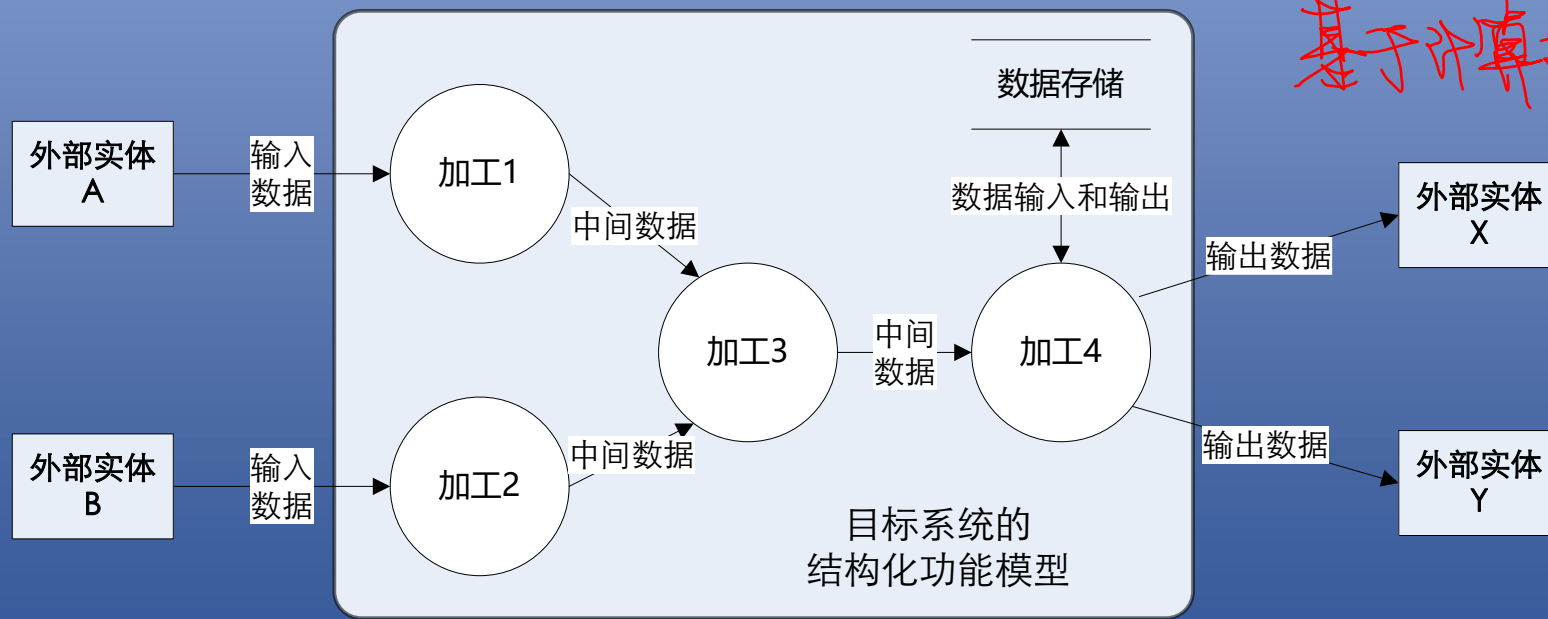
- 数据对象之间的基数关系，在关系型数据库中就会造成数据存储的冗余现象。为了消除数据冗余，就需要对实体-关系图中的数据进行规范化处理，即数据范式。
对实体-关系图中数据规范化处理
- 数据结构的规范化有以下好处：
 - 消除多义性：使关系中的属性含义清楚、单一；
 - 关系单纯化：让每个数据项只是简单的数或字符串，方便操作。使数据的插入、删除与修改操作可行且方便；
 - 使关系模式更灵活：易于实现接近自然语言的查询方式。

- 第一范式的数据冗余程度最大，随着范式提高数据冗余会随之变小。
- 然而范式级别越高，存储同样数据就需要分解成越多张表，因此“存储自身”的过程也就越复杂。
- 随着范式级别的提高，数据的存储结构与基于问题域的结构间的匹配程度也随之下降，为此在需求发生变化时数据的稳定性随之下降。除此之外，范式级别的提高则需要访问的表增多，因此性能或处理速度将下降。

不出现表中有表

- 第一范式 (1FN) : 关系中所有属性都是“单纯域”, 即不出现“表中有表”; 表的每一行只包含一个实例的信息。
- 第二范式 (2FN) : 非主属性完全函数依赖于关键字;
 - 首先必须满足1FN, 然后要求表中的每个实例或行必须可以被唯一地区分。
 - 需要添加一个实例的唯一标识列, 称为主关键字或主键、主码。
- 第三范式 (3FN) : 在满足第二范式的基础上, 要求非主属性相互独立, 即任何非主属性间不存在函数依赖。

- 当数据或信息“流”过计算机系统时将会被系统的功能所处理、加工或变换后再将处理或变换后的数据从系统输出。
- 基于计算机的系统可被表示为数据流图的基本结构：

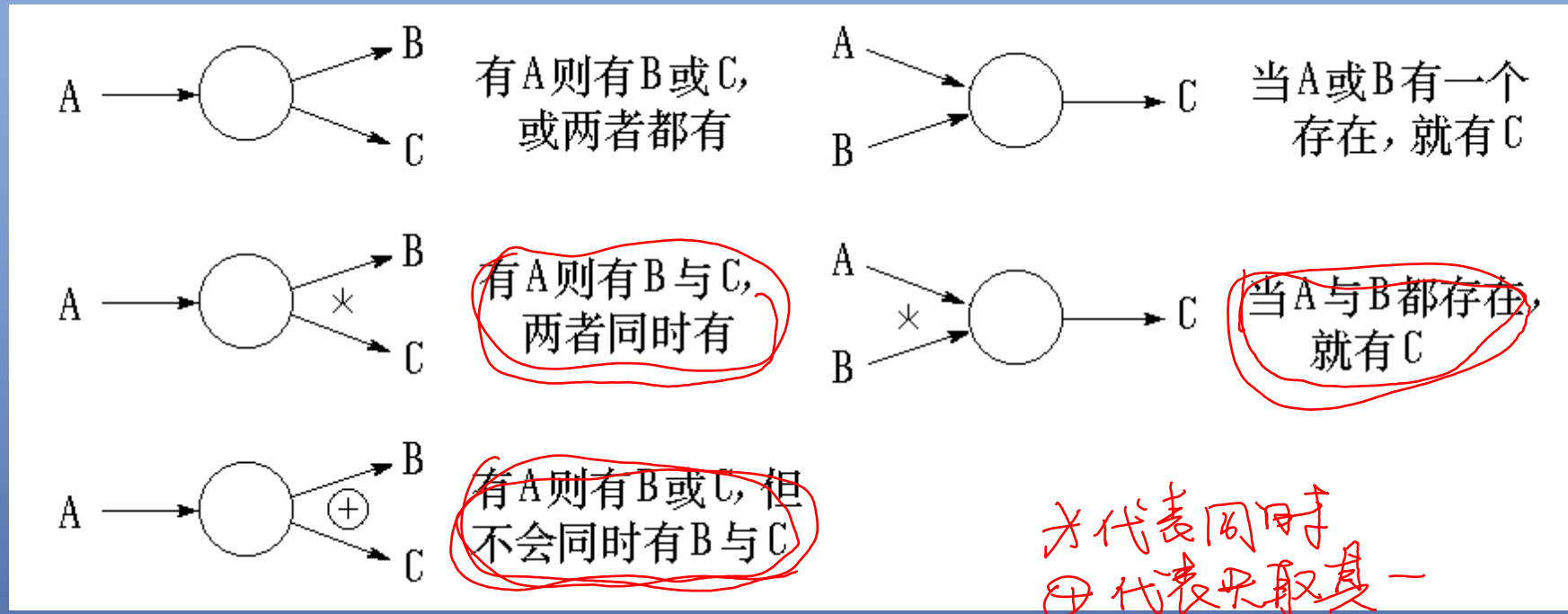


基于计算机的系统可被表示为
数据流图的基本结构。

- 数据流图可以被用来抽象地表示系统或软件，既能提供功能建模的机制。
- 也可提供数据流建模的机制，并可以自顶向下的机制表示层级的功能细节和数据变换细节。
- 从数据流图中可知，数据流图有四种基本元素：



基本元素



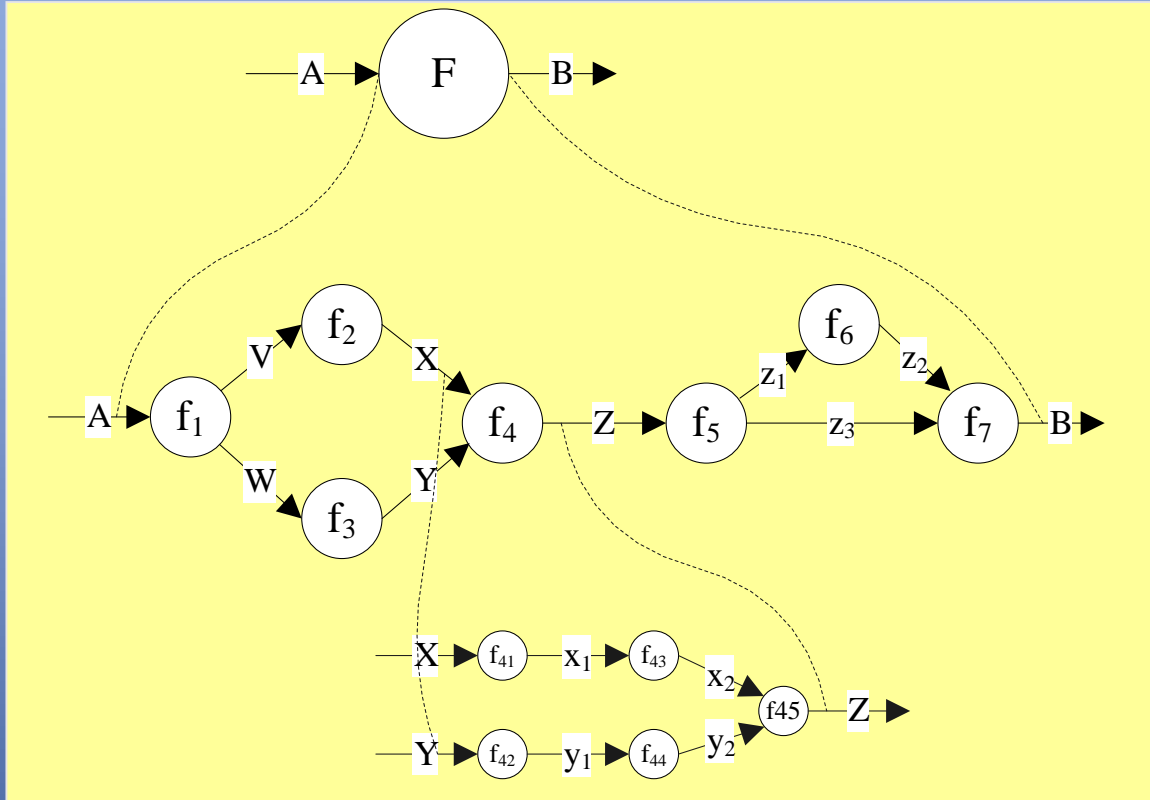
“*” 表示相邻的一对数据流同时出现;
“⊕” 则表示相邻的两数据流只取其一。

- 为表达复杂的实际问题（图形符号过多），需要按照问题的层次结构进行逐步分解，并以分层的数据流图反映这种结构关系。
 - 顶层数据流图：顶层流图仅包含一个加工，它代表被开发系统，其作用在于表明被开发系统的范围，以及它和周围环境的数据交换关系。
 - 中间层数据流图：表示对其上层父图的细化。它的每一加工可以继续细化，形成子图。中间层次的多少视系统的复杂程度而定。
 - 底层数据流图：是指加工不须再做分解的数据流图，称为“原子加工”。

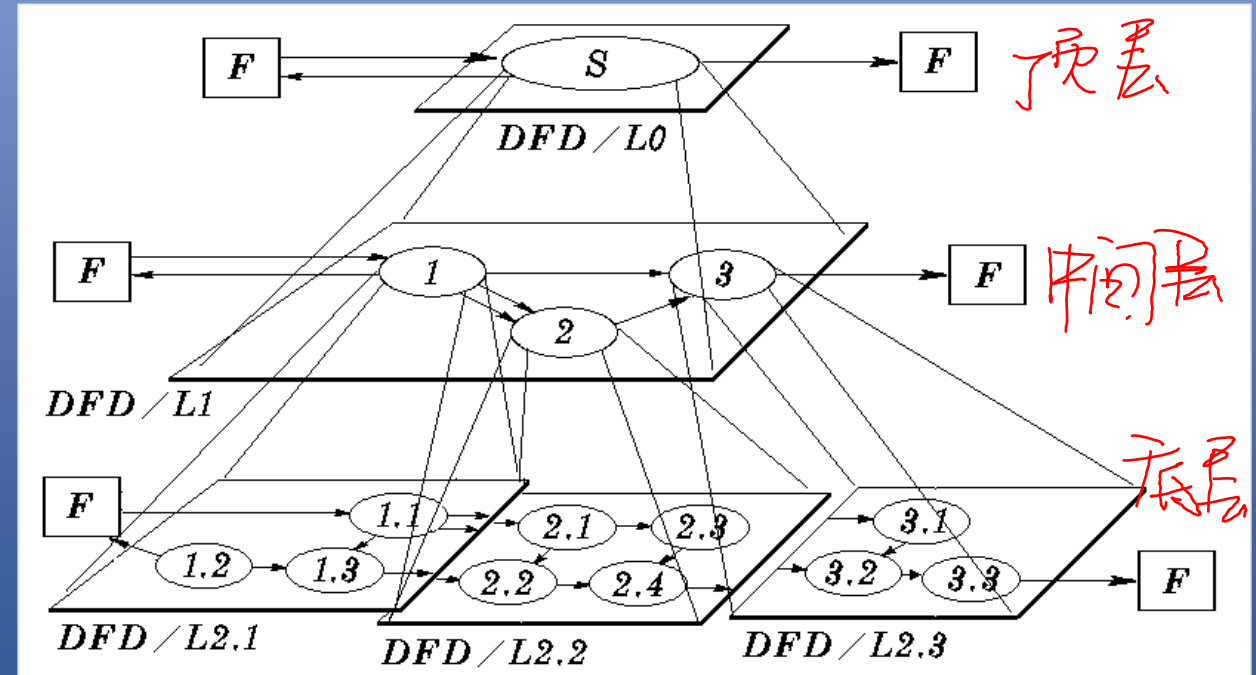
自顶向下的分析思路。

加工不须再做分解。

分层数据流图的两种表示



数据流图的“平衡”关系

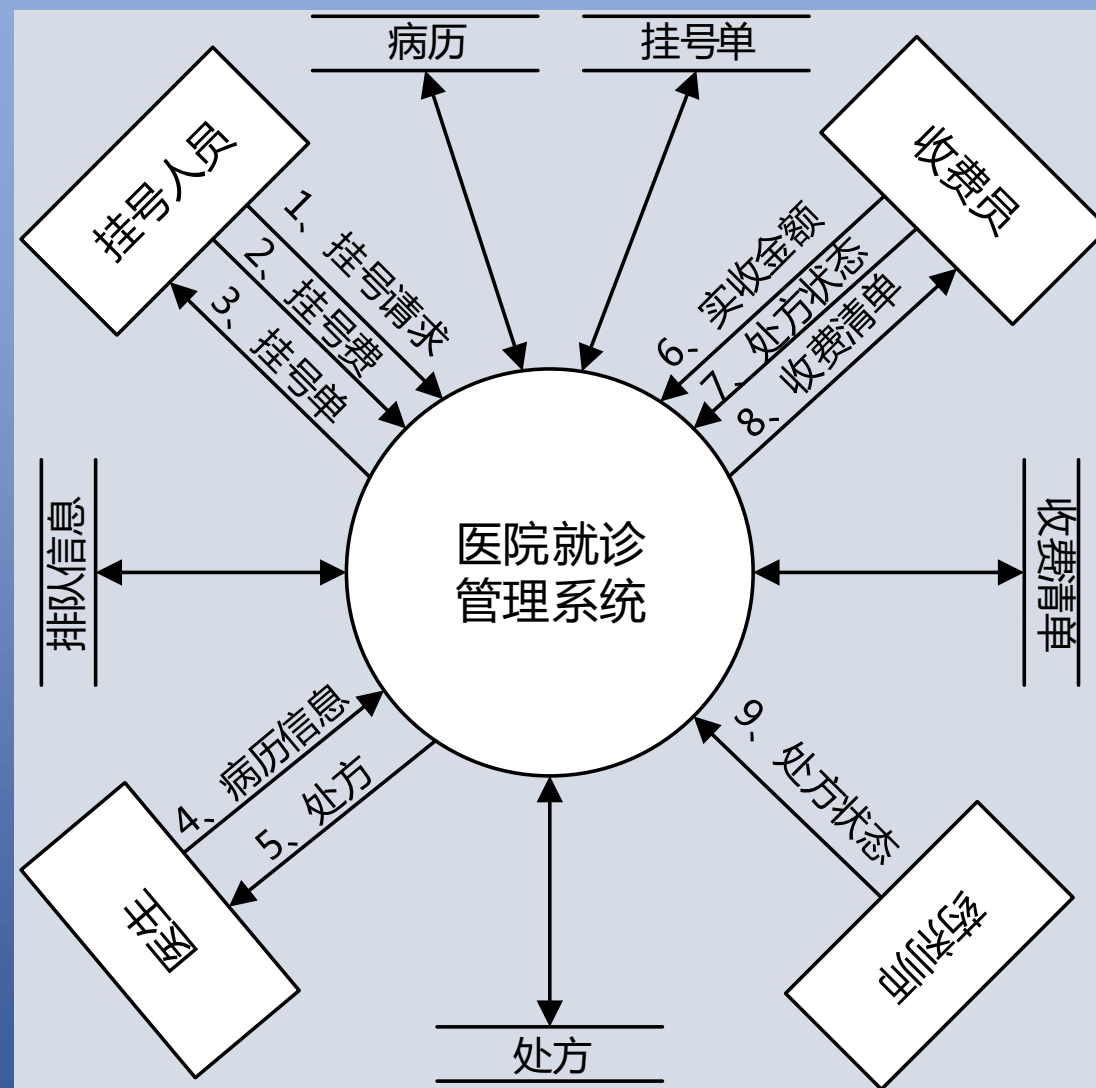


- 画数据流图的基本步骤概括地说，就是自外向内，自顶向下，逐层细化，完善求精。具体步骤可按如下来做：
 1. 先找系统的数据源点与汇点。它们是外部实体，由它们确定系统与外界接口。
 2. 找出外部实体的输出数据流与输入数据流。
 3. 在图的边上画出系统的外部实体。
 4. 从外部实体的输出数据流（即系统的源点）出发，按照系统的逻辑需要，逐步画出一系列逻辑加工，直到找到外部实体所需的输入数据流(即系统的汇点)，形成数据流的封闭。
 5. 按照下面所给的原则进行检查和修改。
 6. 按照上述步骤，再从各加工出发，画出所需的子图。

- 假设，该系统涉及医院的三个部门：挂号、问诊、交费和取药。
 - **挂号**：挂号处的挂号人员接受病人的就诊请求，根据门诊科室各医生的病人的排队情况，分配合适科室和医生，记录并打印挂号凭据，收取挂号费完成挂号请求。
 - **问诊**：医生根据挂号的次序对病人进行病情诊断，根据挂号单据及病历号获取该病人的历史病历，然后将问诊结果记录在病历当中并开具相应的处方（处方中可根据系统提供的药品进行选择），打印处方交给病人完成一次问诊。
 - **交费取药**：病人在收费处根据挂号单据和处方所列出的药品种类和数量进行交费，取得相应的药品完成一次看病流程。

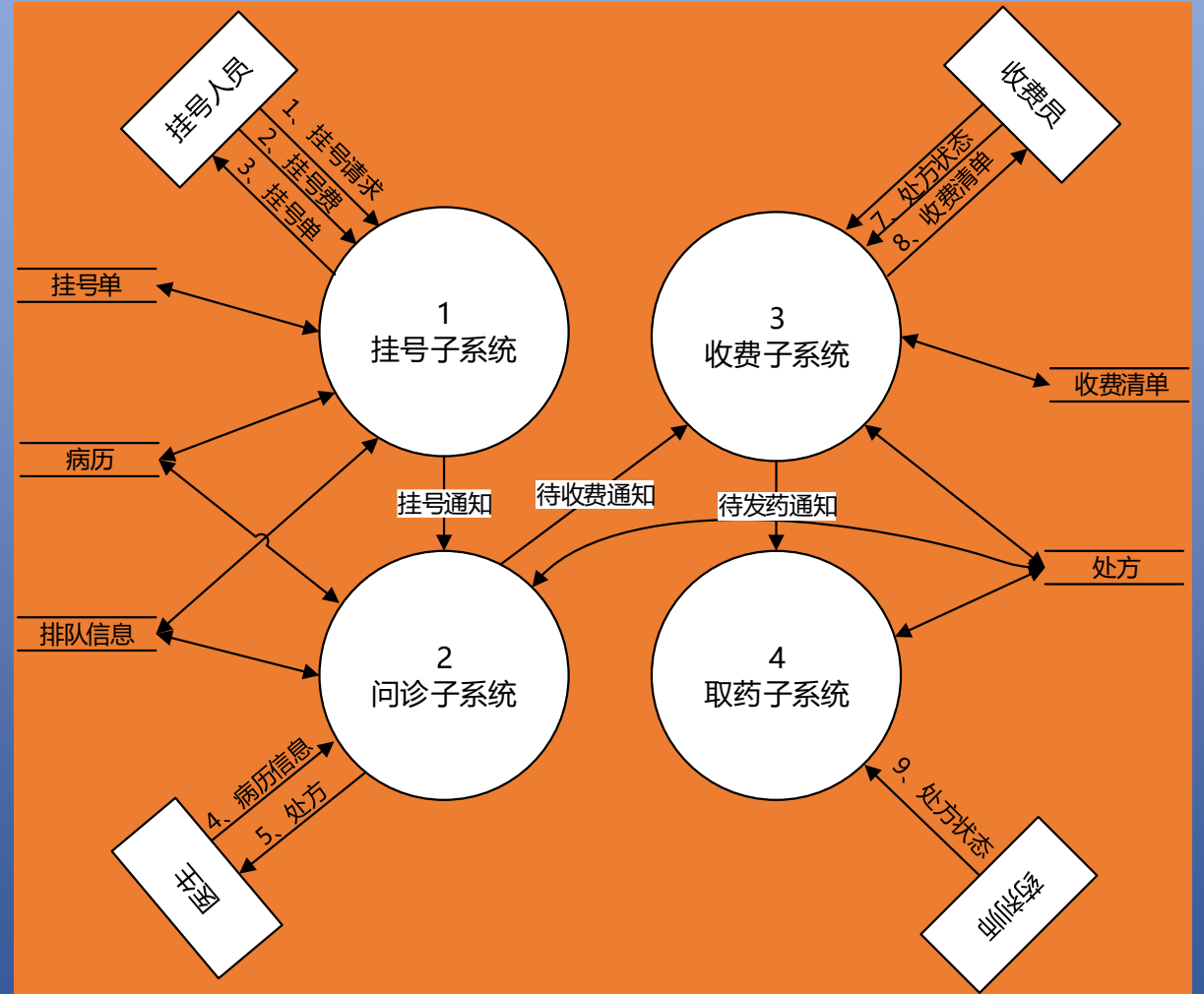
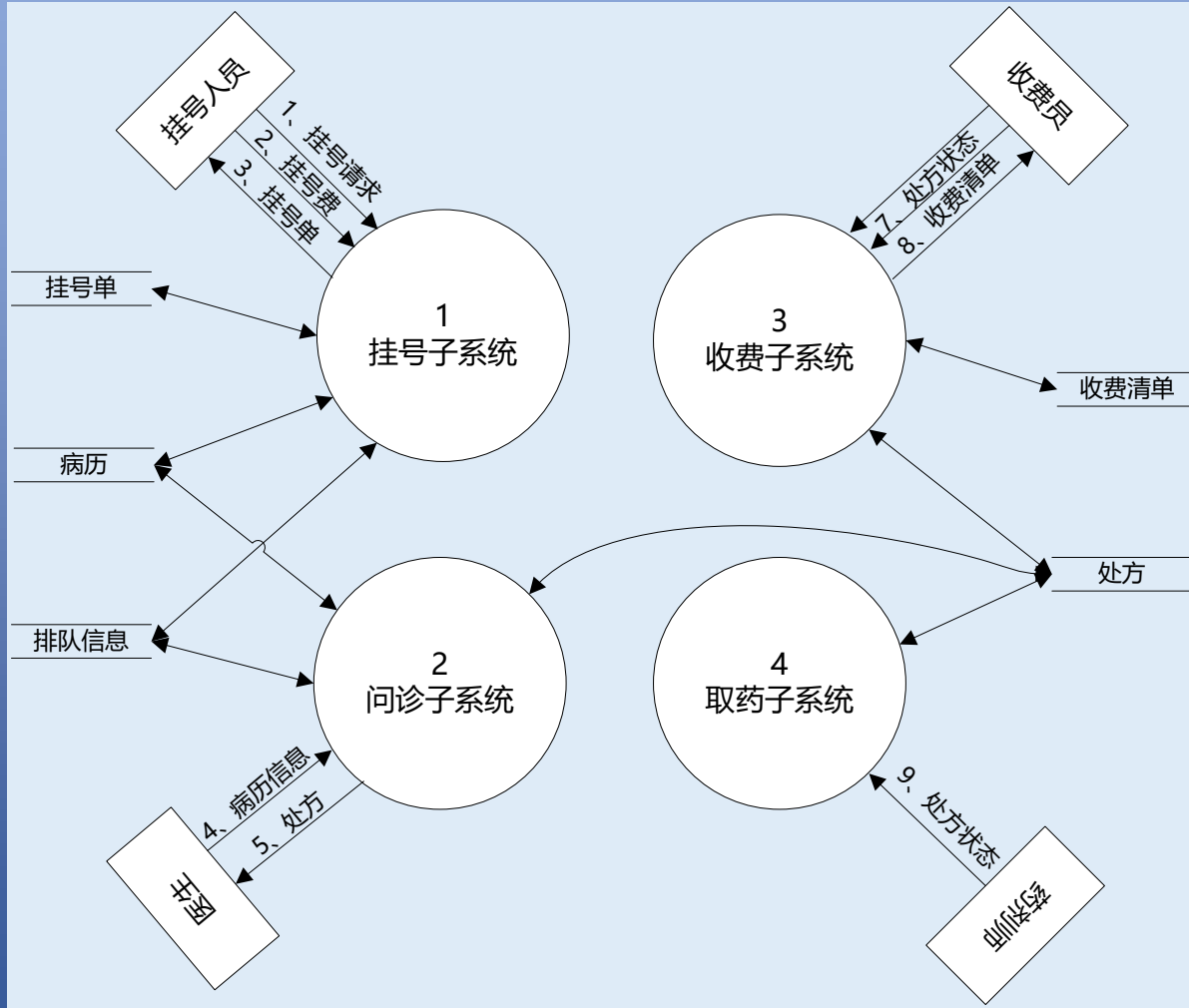
- 首先确定顶层的加工：待构建的软件系统：医院就诊管理系统；
- 根据掌握的业务知识通过分析确定外部实体：挂号人员、医生、收费员和药剂师等；
- 根据业务的操作流程确定外部实体与加工的数据流：1、2、...9；
- 逐步给出可能的数据存储；

数据流

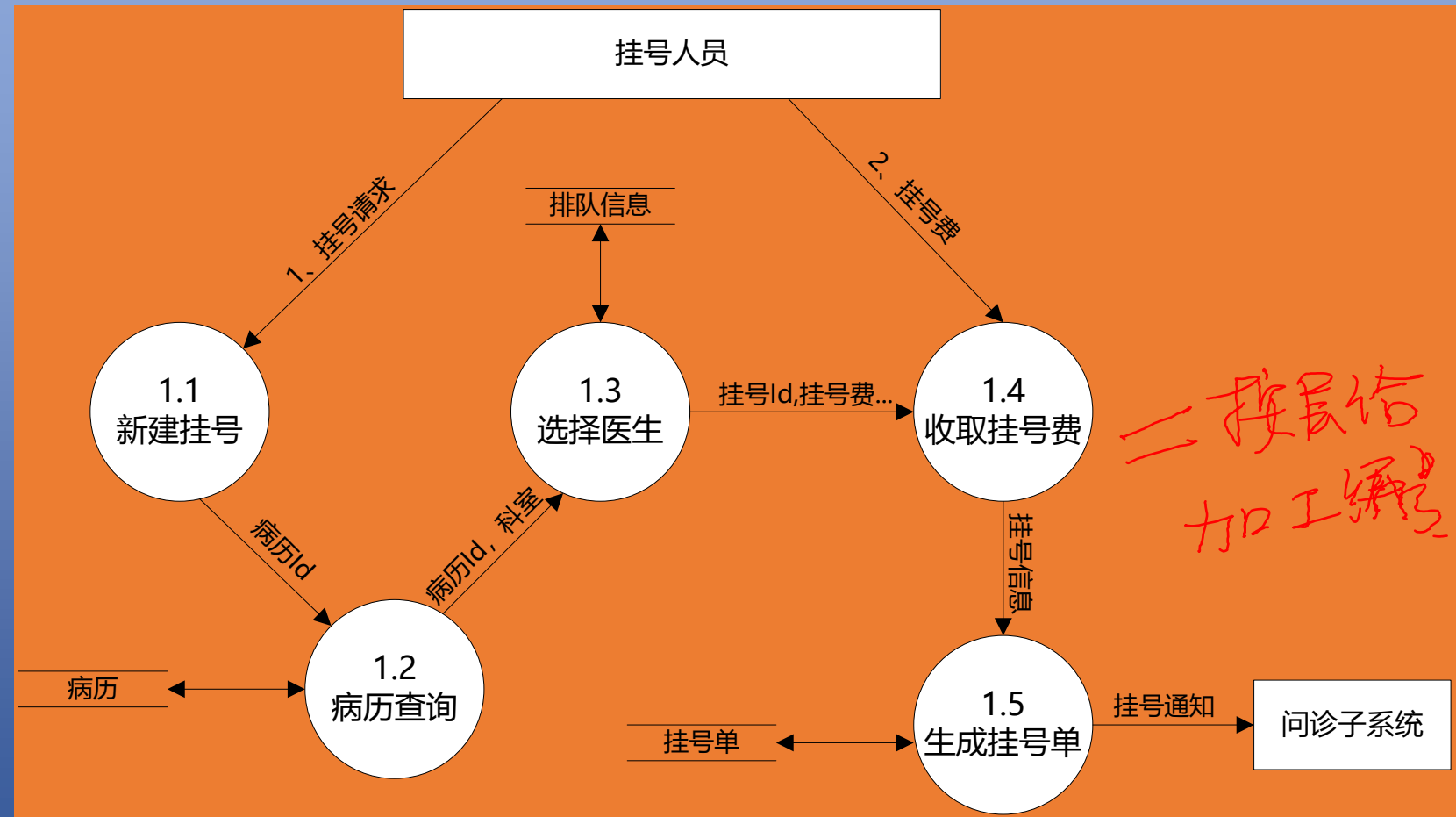


- 目标是要确定该层具有多少个子加工，以及子加工之间新增的数据流。
- 解释顶层数据流图中每一条数据流从外部实体流入到系统级的加工之后，这些数据流是如何被接收和处理的。
 - 子加工的分析：尽可能将系统分解为多个子系统，降低问题域的复杂程度，经分析得到：挂号处理子系统、问诊处理子系统等；
 - 子加工之间有无新增数据流：（示例）两个可能的解决方案

两个可能的方案



- 确定子系统内部具体的业务功能。
- 解释每一个数据流进入到加工后，系统内是否还存在一些加工：接收数据流、分解数据流、转换数据流、处理数据流直到存储必要的的数据信息进入数据文件存储，并产生需求规定的的数据数据流。



- 需要根据具体的场景进行分析和确定
- 新建挂号、生成挂号单是否需要进一步分解？
- 选择医生：是否需要考虑不同的分配方案？
 - 时间最短
 - 费用最低
 - 指定分配
- 由于某种原因，病人需要撤销挂号单，该如何考虑？
- 其他三个子系统的第二层数据流图如何构建？
- 以上问题作为课后练习进行思考。

我的天.

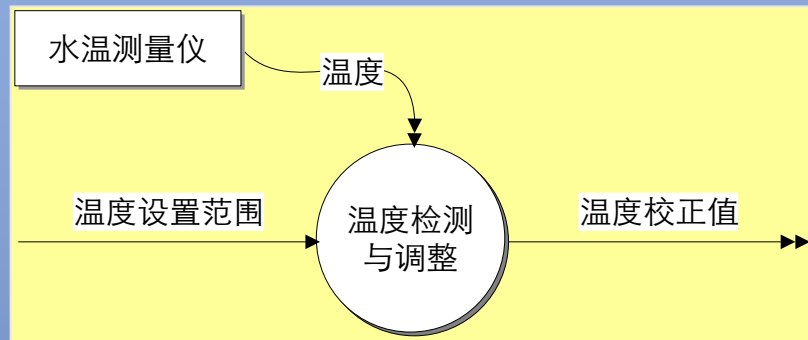
检查和修改数据流图的原则

- 数据流图上所有图形符号只限于前述四种基本图形元素，且必须包括前述四种基本元素，缺一不可。
- 数据流图的主图上的数据流必须封闭在外部实体之间，外部实体可以不止一个。
- 每个加工至少有一个输入数据流和一个输出数据流。
- 在数据流图中，需按层给加工框编号，表明该加工处在哪一层，以及上下层的父图与子图的对应关系。
- 任何一个数据流子图必须与它上一层的一个加工对应，两者的输入数据流和输出数据流必须一致。即父图与子图的平衡，表明在细化过程中输入与输出不能有丢失和强加。
- 图上每个元素都必须有名字。表明数据流和数据文件是什么数据，加工做什么事情。

必须有名字

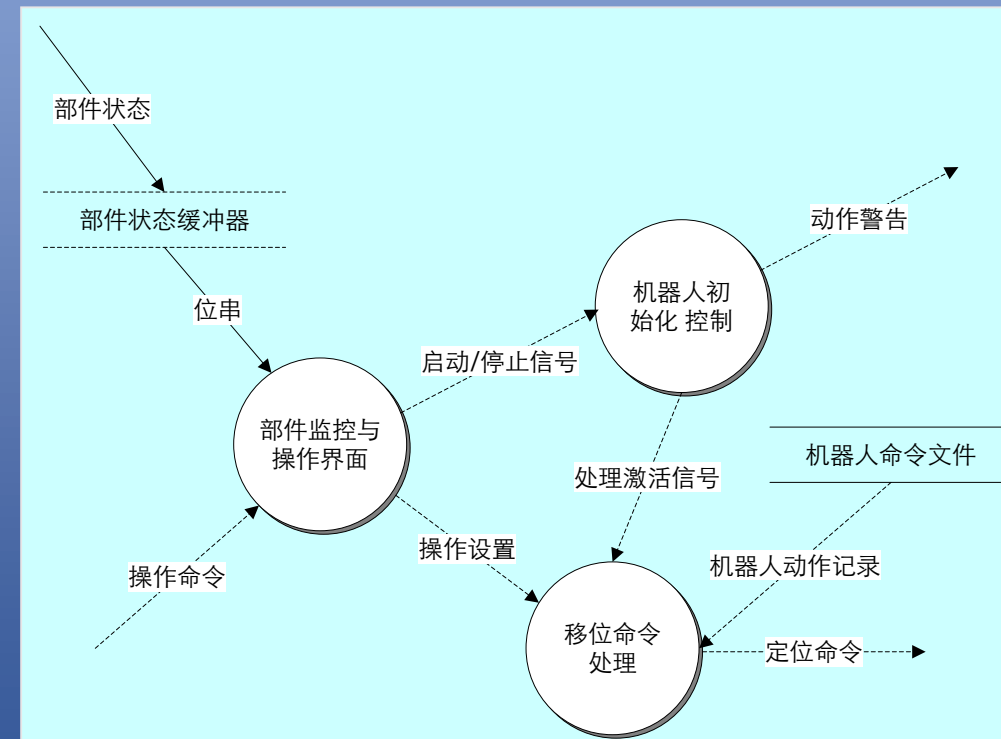
- Ward & Mellor对实时系统的数据流图进行相应的扩展，引入了控制流及连续的数据流等符号。该扩展可适应实时系统提出的要求：
 - 在时间连续的基础上接收或产生数据流；
 - 贯穿系统的控制信息和相关的控制处理；
 - 多任务的情况下可能会遇到同一个加工的多个实例；
 - 系统状态以及导致系统状态迁移的机制。





上图给出一个基于计算机的水温控制系统的处理。水温测量仪连续传送水温数据给温度监控加工模块，该加工将水温与允许波动范围进行比较，输出校正数据给温度调节装置。

下图给出一个制造车间的数据和控制流的顶层流图。事件可以作为普通数据加工的输入数据流，控制也可以接收普通的输入数据流。



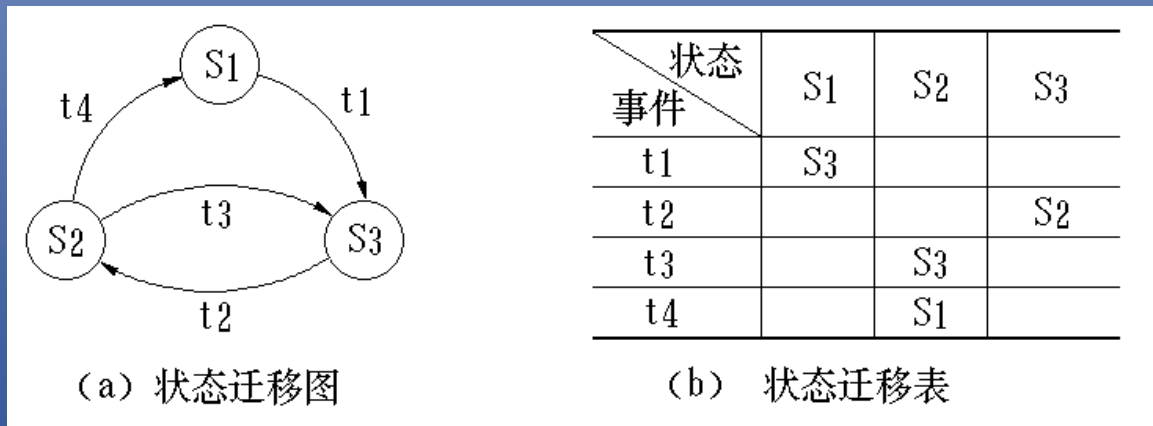
- 为了直观地分析系统的动作，从特定的视角出发描述系统的行为，需要采用动态分析的方法。
- 其中最为常用的结构化动态分析方法有：
 - 状态迁移图
 - 时序图
 - Petri网等。

状态迁移图.

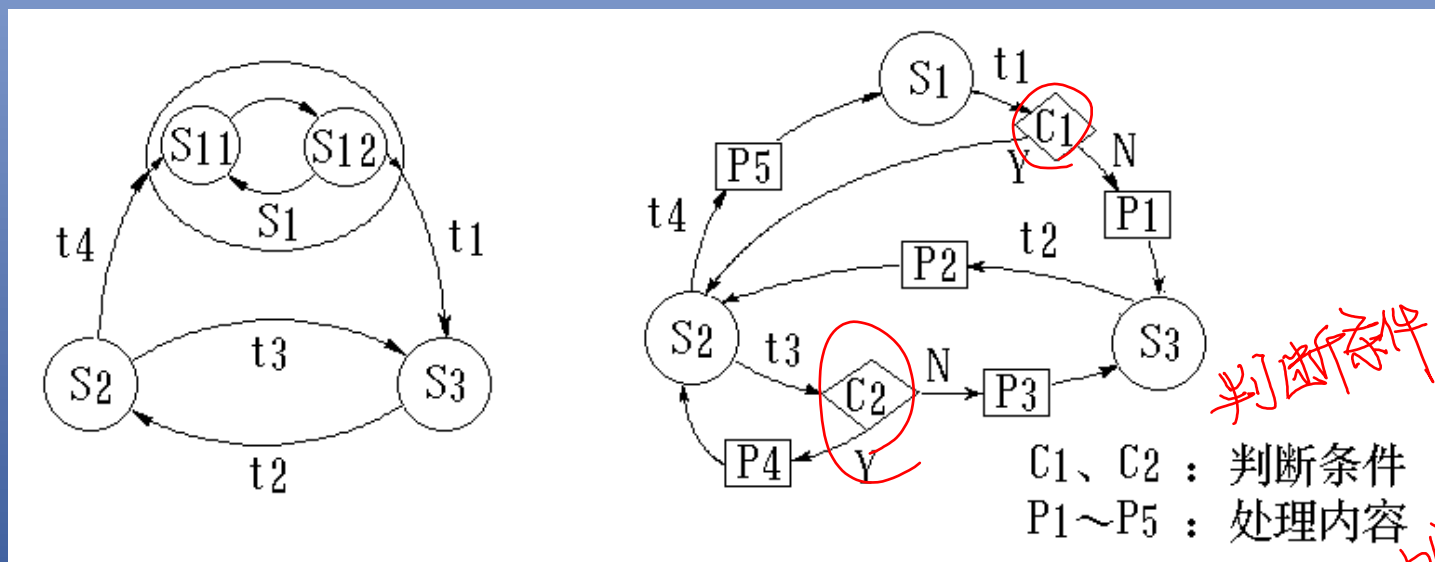
时序图.

Petri网.

- 利用状态迁移图来描述系统或对象的状态，以及导致系统或对象的状态改变的事件，从而描述系统的行为。
- 状态迁移图指明系统的状态如何相应外部事件进行推移。
- 在状态迁移图中，用圆圈“○”表示可得到的系统状态，用箭头“→”表示从一种状态向另一种状态的迁移。
- 状态迁移图指明了作为特定事件的结果（状态）。在状态中包含可能执行的行为（活动或加工）。



- 在状态迁移图中，由一个状态和一个事件所决定的下一状态可能会有多个。实际会迁移到哪一个是由更详细的内部状态和更详细的事件信息来决定的。



判断条件.
处理内容.

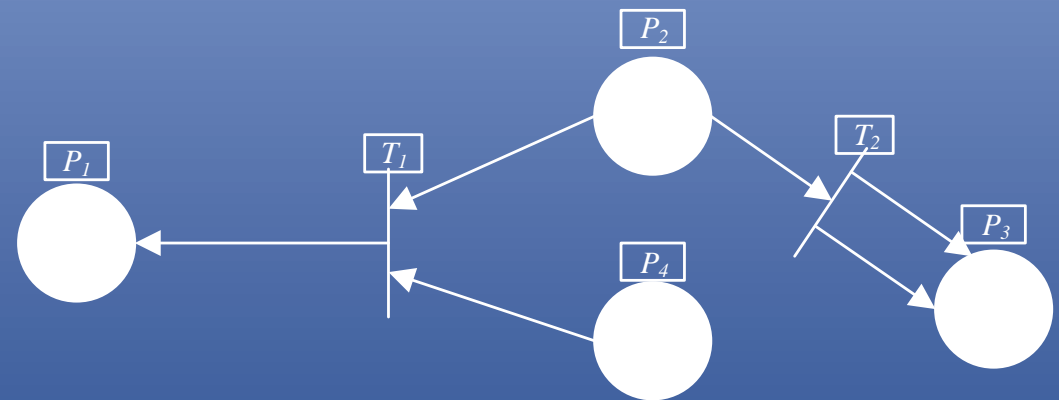
- Petri网是由德国人Carl Adam Petri于1962年提出,主要用于处理并发系统当中的同步问题、资源竞争问题以及死锁问题等。
- Petri网是一种有向图, 包含四种基本元素, 一组位置P、一组转换T、输入函数I以及输出函数O。

一组位置P为{P1, P2, P3, P4}

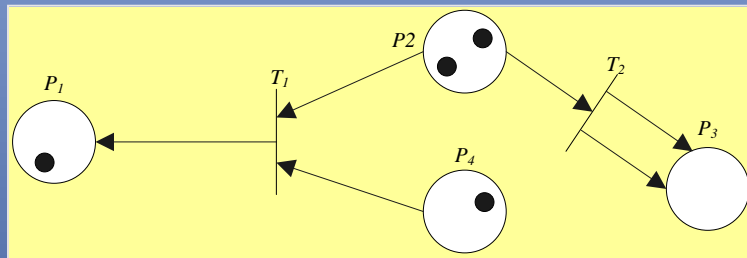
一组转换T{t1, t2}

两个输入函数, $I(t1) = \{P2, P4\}$; $I(t2) = \{P2\}$

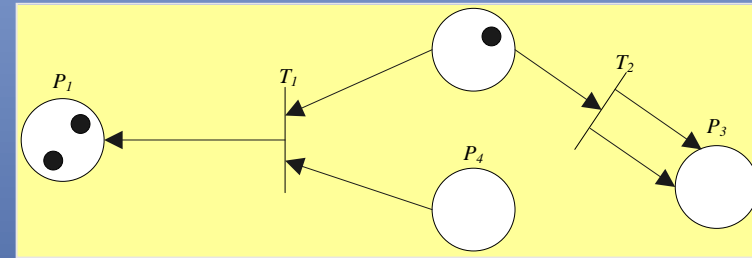
两个输出函数, $O(t1) = \{P1\}$; $O(t2) = \{P3\}$



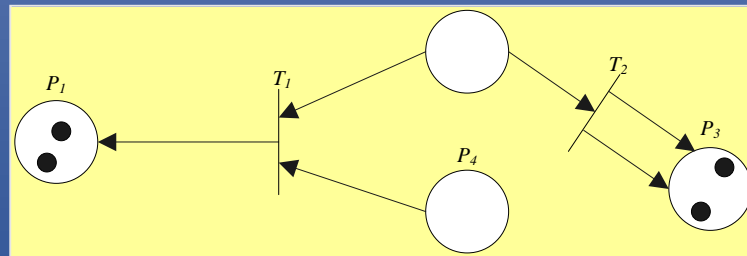
- Petri网位置中如果加了一个黑点，称之为标记（token）。标记在位置中的出现表明了处理要求的到来。
- 通常，当每个输入位置所拥有的标记数大于等于从该位置到转换的线数时，就允许转换。



标记向量表示为 $(1, 2, 0, 1)$

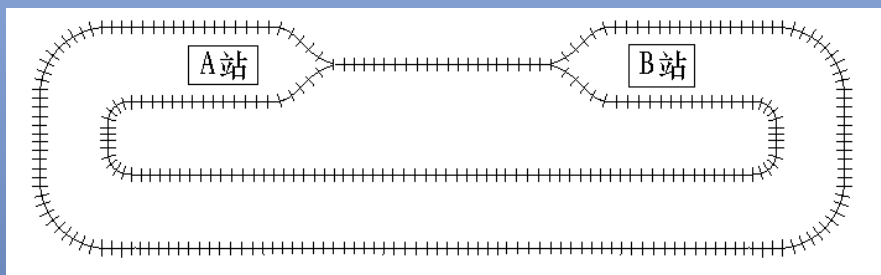


标记向量表示为 $(2, 1, 0, 0)$

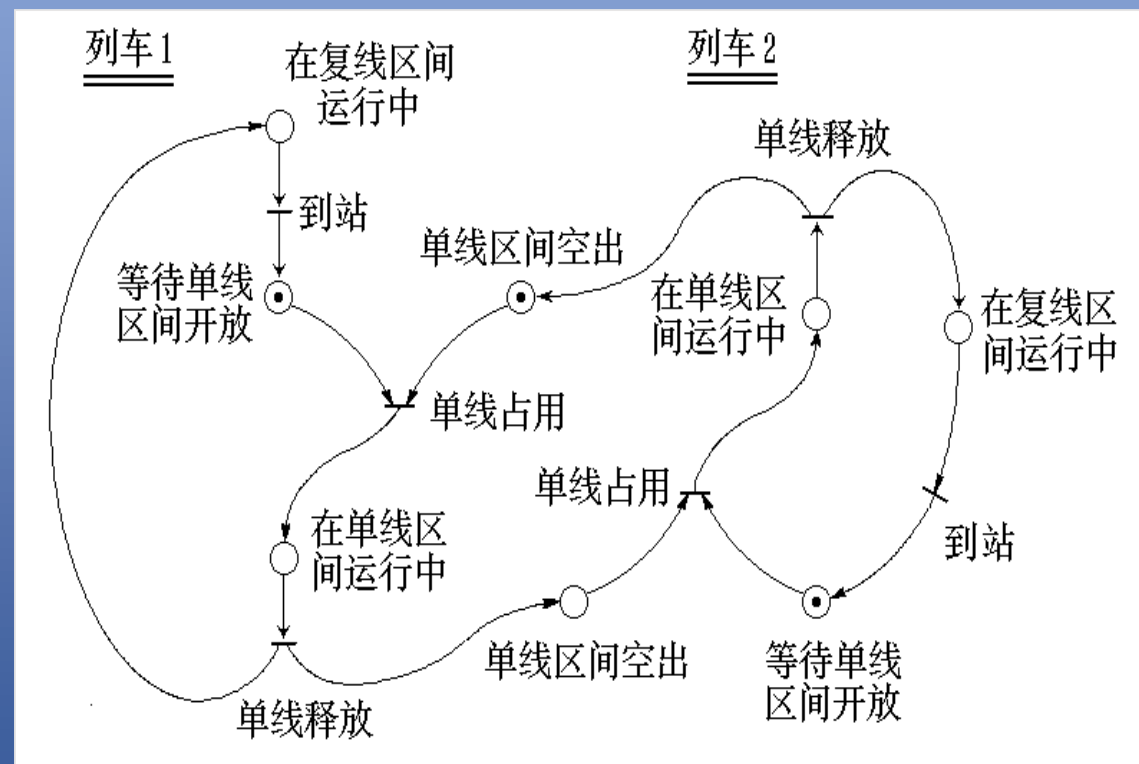


标记向量表示为 $(2, 0, 0, 2)$ 表示

- 如图有一个环形铁路，在A站与B站之间是单轨，在某一时刻只能走一列火车。但A站与B站都是双向运行的。



由于在A站和B站之间有一段单轨线路，在某一时刻只允许有一列火车通过，因此只有当单线上没有列车通过时，火车才能进入单线运行，否则火车只能在A站或者B站等候，等待单线空出来使用。



- 数据词典：对于数据流图中出现的所有被命名的图形元素加以定义，使得每一个图形元素的名字都有一个确切的解释。
- 其定义应是严密的、精确的，不可有半点含混并消除二义性，它由以下内容组成：
 - 数据流词条
 - 数据元素词条
 - 数据文件词条
 - 加工词条
 - 外部实体

数据流名称	唯一标识数据流的名称
简要描述	简要介绍该数据流的作用，即它产生的原因和结果
数据流来源	来源于何处
数据流去向	流向何处
数据流组成	描述该数据流的内部数据元素的组成
备注	需要的情况下描述数据流量和流通量等信息

数据元素名称	唯一标识数据元素的名称或编号
简要描述	简要描述该数据元素的作用，以及位于哪一个数据结构内
类型	数字、字符等类型
长度	该数据类型规定的取值范围，例如姓名的长度为60个字符
取值范围	该姓名的取值范围必须大于2个字符且小于60个字符
备注	

数据文件名称	唯一标识数据文件的名称
简要描述	简要描述该数据文件存放的是什么数据
输入数据	写入该数据文件的数据内容或数据结构
输出数据	从该数据文件读出的数据内容或数据结构
数据文件组成	描述该数据文件的数据结构组成
存储方式	数据文件的操作方式以及关键字
备注	

外部实体名称	唯一标识外部实体的名称
简要描述	指明该实体的性质以及与系统之间的关系
有关数据流	指明该外部实体与系统之间交互的数据流有哪些
备注	

户名	所号	帐号
开户日	性质	印密

日期 年月日	摘要	支出	存入	余额	操作	复核

存折 = 户名 + 所号 + 帐号 + 开户日 + 性质 + (印密) + 1{存取行}50

户名 = 2{字母}24

所号 = "001".. "999"

帐号 = "00000001".. "99999999"

开户日 = 年 + 月 + 日

性质 = "1".. "6"

印密 = "0"

存取行 = 日期 + (摘要) + 支出 + 存入 + 余额 + 操作 + 复核

日期 = 年 + 月 + 日

年 = "00".. "99"

月 = "01".. "12"

日 = "01".. "31"

摘要 = 1{字母}4

支出 = 金额

金额 = "0000000. 01".. "9999999. 99"

操作 = "00001".. "99999"

注：金额规定不超过9999999.99元

注：表明该存取是存?是取?还是换?

- 数据流图中的每一个加工除了要进行基本信息的描述之外，还须对加工的逻辑或规则进行描述，其方法有判定表、判定树或结构化英语等。

加工名称	唯一标识加工的名称
简要描述	描述加工逻辑和规则以及功能简述
加工编号	反映该加工的层次
输入数据流	描述进入该加工的一个或多个数据流
输出数据流	描述流出该加工的一个或多个数据流
加工逻辑	简述该加工的逻辑或规则（参见加工逻辑说明一节）
备注	

- 在书写基本加工逻辑的说明时，应满足如下的要求：
 - 对数据流图的每一个基本加工，必须有一个加工逻辑说明；
 - 加工逻辑说明描述基本加工如何把输入数据流变换为输出数据流的加工规则（参考业务规则）；
 - 加工逻辑说明必须描述实现加工的策略而不是实现加工的细节。

- 结构化英语也称为PDL，是一种介于自然语言和形式化语言之间的半形式化语言。
- 它是在自然语言基础上加了一些限制而得到的语言，是使用有限的词汇和有限的语句来描述加工逻辑。
- 其词汇表由英语命令动词、数据词典中定义的名字、有限的自定义词和控制结构关键词：
 - **IF_THEN_ELSE**
 - **WHILE_DO**
 - **REPEAT_UNTIL**
 - **CASE_OF**等组成。
- 其动词的含义要具体，尽可能少用或不用形容词和副词。

- 商店业务处理系统中“检查发货票” (Invoice) 功能使用结构化英语完成的加工逻辑描述

IF the invoice exceeds \$500 **THEN**

IF the account has any invoice more than 60 days overdue **THEN**

 The confirmation pending resolution of the

ELSE (account is in good standing)

 Issue confirmation and invoice

ENDIF

ELSE (invoice equal \$500 or less)

IF the account has any invoice more than 60 days overdue

 Issue confirmation, invoice and write message

 action report

ELSE (account is in good standing)

 Issue confirmation and invoice

ENDIF

ENDIF

IF 发货单金额超过\$500 **THEN**

IF 欠款超过了60天 **THEN**

 在偿还欠款前不予批准

ELSE (欠款未超期)

 发批准书，发货票

ENDIF

ELSE (发货票金额未超过\$500)

IF 欠款超过60天 **THEN**

 发批准书，发货票及赊欠报告

ELSE (欠款未超期)

 发批准书，发货票

ENDIF

ENDIF

- 在某些数据处理问题中，数据流图的加工需要依赖于多个逻辑条件的取值，即完成这一加工的一组动作是由某一组条件取值的组合而引发时，逻辑组合比较复杂，就需要用判定表来表示。

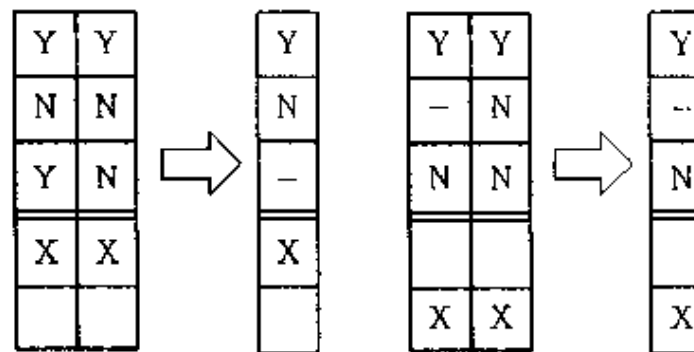
		1	2	3	4
条件	发货单金额	>\$500	>\$500	≤\$500	≤\$500
	赊欠情况	>60天	≤60天	>60天	≤60天
操作	不发出批准书	✓			
	发出批准书		✓	✓	✓
	发出发货单		✓	✓	✓
	发出赊欠报告			✓	

条件桩—左上部分：列出了各种可能的条件。

条件项—右上部分：给出各个条件的条件取值的组合。

动作桩—左下部分：列出了可能采取的动作。

动作项—右下部分：是和条件项紧密相关的，它指出了在条件项的各种取值的组合情况下应采取什么动作。

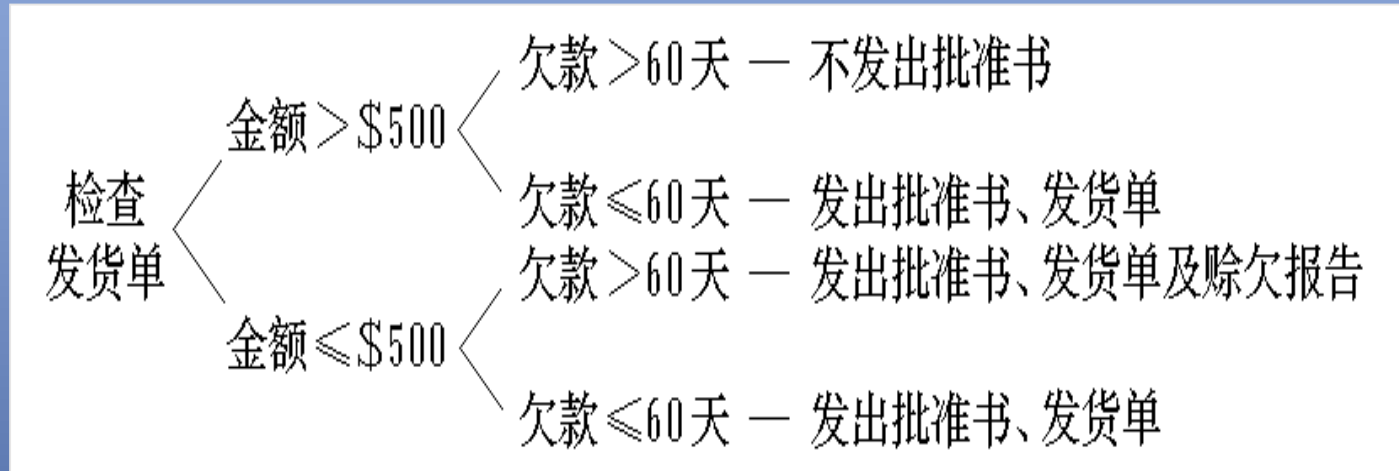


(a) 两条规则合并

(b) 进一步合并

图 3.22 动作相同的规则合并

- 判定树也是用来表达加工逻辑的一种工具。有时候它比判定表更直观。用它来描述加工，很容易为用户接受。



- 在表达一个基本加工逻辑时，结构化英语、判定表和判定树常常交叉使用，互相补充。
- 总之，加工逻辑说明是结构化分析方法的一个组成部分，对每一个加工都要加以说明。
- 应当以结构化英语为主，对存在判断问题的加工逻辑，可辅之以判定表和判定树。