



南京大學
NANJING UNIVERSITY



结构化分析方法

张天

软件工程组

ztluck@nju.edu.cn

2025年秋季



分析的位置

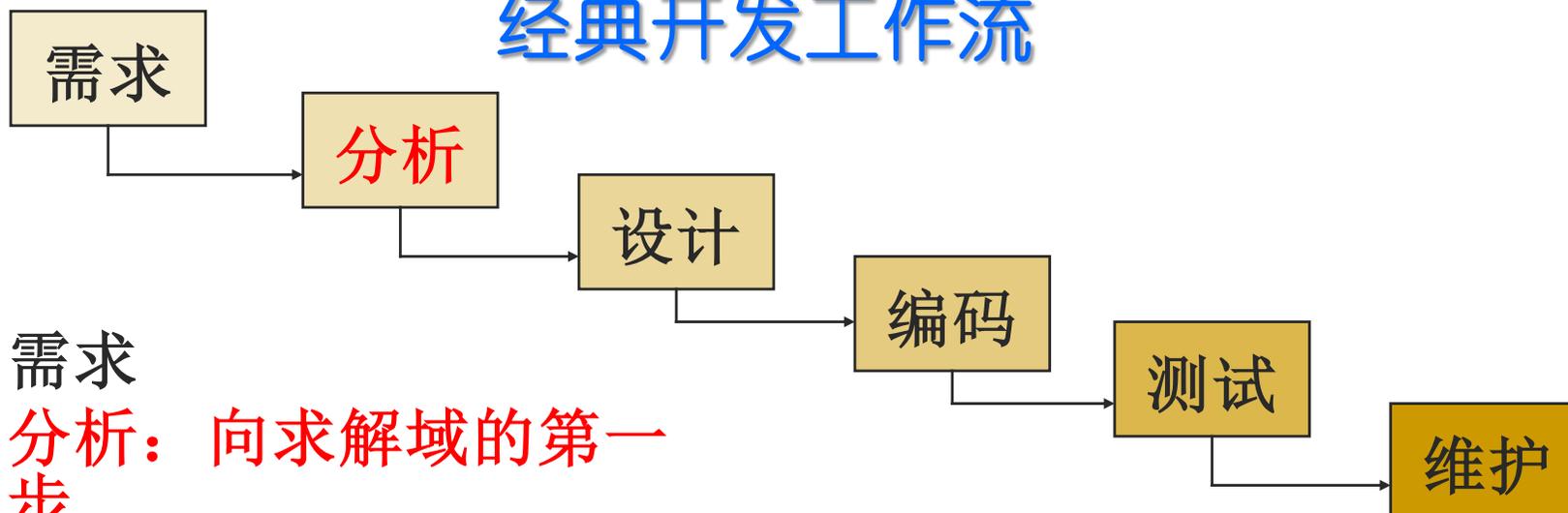


问题域

求解域



经典开发 workflow



- 需求
- 分析：向求解域的第一步
- 设计
- 编码
- 测试
- 维护

分析 workflow 的目标：
分析和提取需求，以获得正确开发软件产品和易于维护它所必须的需求。



分析 workflow



- 分析是从问题域向求解域迈进的第一步
 - 需求流的输出必须能够完全被客户所理解，即必须用客户的语言来表达（英语、汉语、法语……）
 - 分析流则使用更为精确的语言（向计算机的方向）
 - 分析活动的产生制品是应用域专家和计算机专家都可以看得懂的规格说明
 - **注意：边界其实是模糊的**



需求工程



- 需求的内涵 / 分类
 - 客户需求、用户需求
 - 变更
 - 功能
 - 速度、体验
- 任务 / 活动
 - 收集获取需求
 - 现有系统调研，抽象
 - 目标系统的逻辑模型
 - 需求分析
- 输出



需求分析



- 描述客户需要什么、构造规格说明
 - 系统处理什么数据？
 - 执行什么功能？
 - 显示什么行为？
 - 定义什么接口？
 - 有什么约束？



需求：案例



- 需要开发一款手机银行APP，使得用户可以通过该软件，可以对自己的银行帐户进行操作。
- 讨论并分析
 - 用户场景
 - 访问控制、转账、代缴费、查询（余额、账单）、理财推荐、公益捐赠
 - 申请贷款、信用卡还款、发红包、挂失、扫二维码、预约、绑定收集、短信通知
 - 功能及其关系
 - 数据及其关联
 - 用户账号、密码、余额、交易信息、卡号、



结构化分析方法（SA）



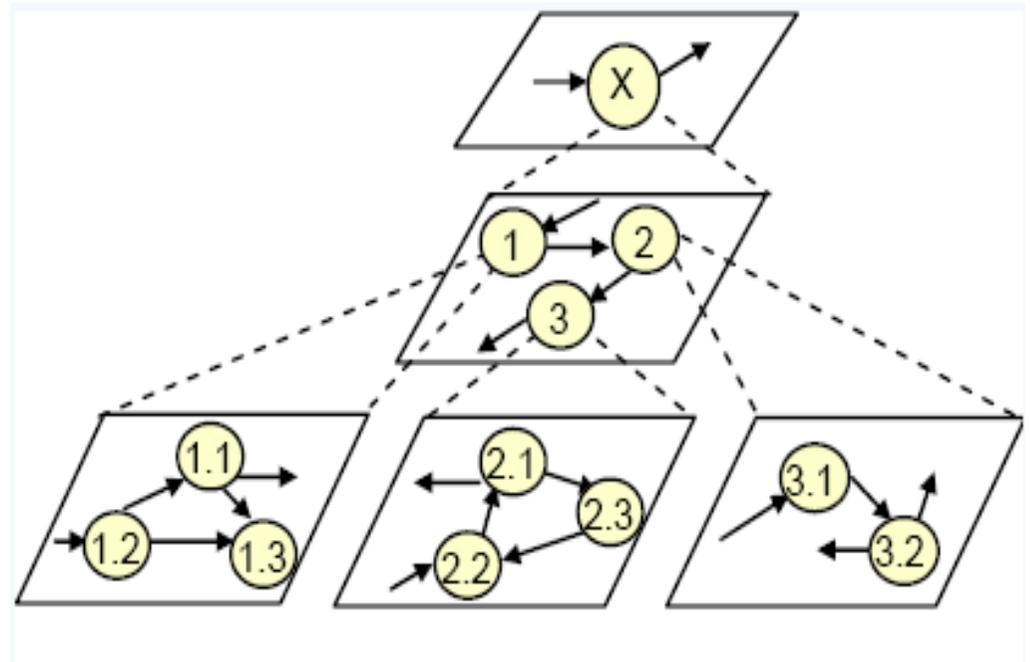
- 结构化分析方法（Structured Analysis,简称SA法）
 - 70年代Edward Yourdon, Tom DeMarco等提出
 - 将系统概念转换为用数据及其控制来表示
 - 基本思想：“自顶向下，逐步求精”，“抽象和分解”
 - 结构化方法：结构化分析 + 结构化设计 + 结构化程序设计
- SA方法采用“抽象”和“分解”两个基本手段，用抽象模型的概念，按照软件内部数据传递、变换关系，由顶向下逐层分解，直到找到满足功能需要的所有可实现的软件元素为止。
- SA方法采用“分解”的方式来理解一个复杂系统，“分解”需要有描述手段，[数据流程图](#)就是作为描述信息流程和分解的手段而引入的。



结构化分析方法(SA)



- 结构化分析 (Structured Analysis,简称SA法) 的**基本思想**: “分解” 和 “抽象”
- **分解**: 把系统的复杂性降低到可以掌握的程度, 把大问题分解成若干小问题, 然后分别解决。
- **抽象**: 即先考虑问题最本质的属性, 暂把细节略去, 以后再逐层添加细节, 直至涉及到最详细的内容。



自顶向下逐层分解



结构化分析



- 功能分解：功能 = 功能 + { 子功能 } + 功能接口
 - 直接地反映用户的需求，容易入手
 - 不能直接地映射问题域，难以检验
 - 不适应需求频繁变化
 - 局部的错误和修改会引起系统波动
- 结构化分析方法（数据流方法）：数据流法 = 数据流 + 数据处理（加工） + 数据存储 + 端点 + 处理说明 + 数据字典
 - 数据流和加工的数量太多，引起分析文档的膨胀
- 信息建模法：系统 = 实体 + 属性 + 关系



结构化分析



- 以分析数据在不同模块中流动的观点来看待一个系统，系统的功能用数据流的转换来表达，分析的结果是一组相关的图、程序描述、数据定义
 - 系统关系图：系统与外界相关系统的输入与输出
 - 数据流程图：表达信息系统中数据的流向方式
 - 信息结构图：表达系统的顶层架构、体现分解的过程
 - 数据字典：定义数据，描述了所有的在目标系统中使用的和生成的数据对象
 - 实体—关系图(ERD)：描述数据对象及数据对象之间的关系
 - 状态—迁移图(STD)：描述系统对外部事件如何响应，如何动作。
 - 加工规格说明：定义数据的处理



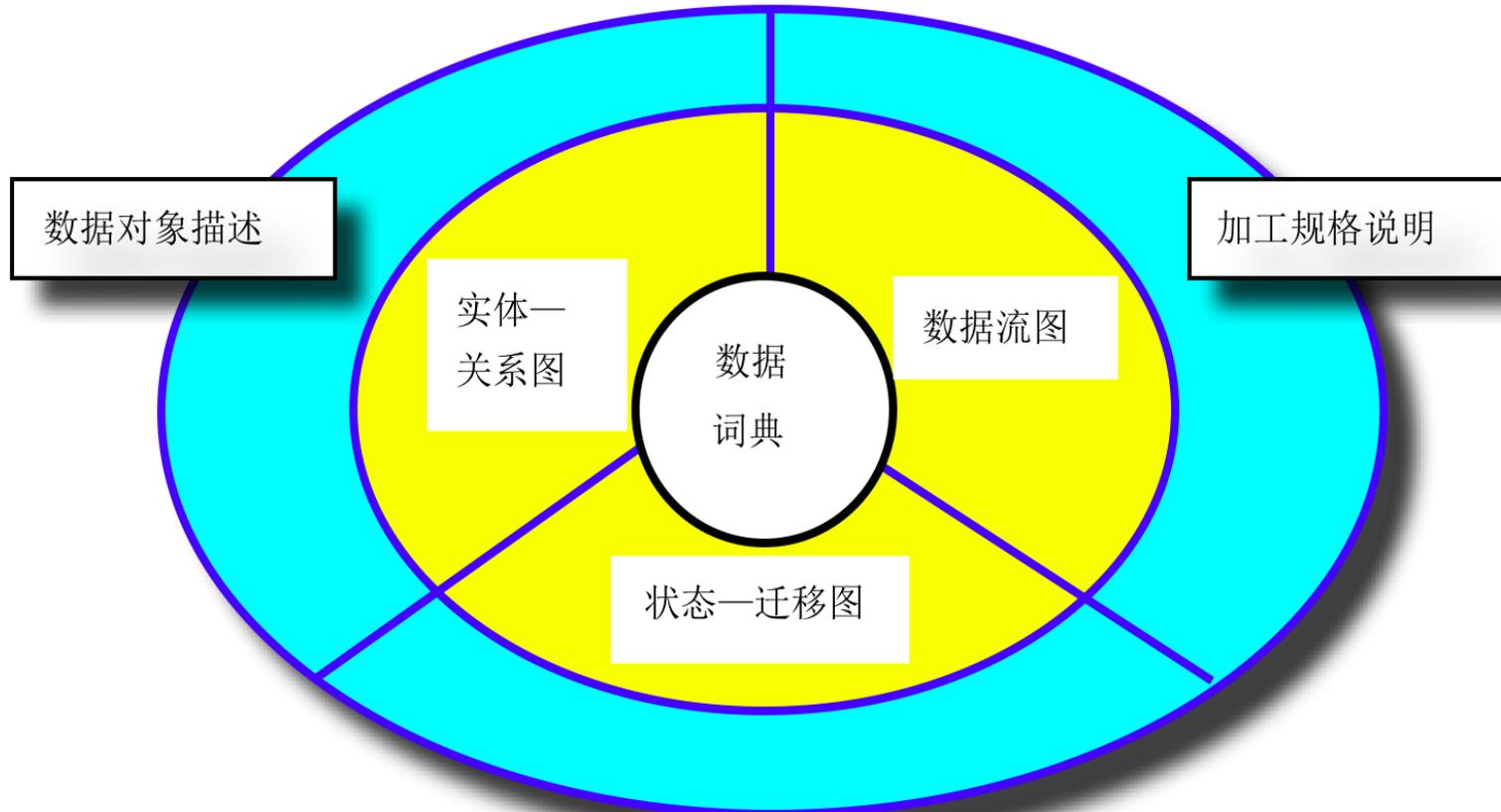
结构化分析



- 早期结构化分析方法仅讨论**数据流**建模，后期逐步完善形成体系
 - 实体关系图**ERD**用于数据建模
 - 数据流图**DFD**用于功能建模
 - 状态迁移图**STD**用于行为建模



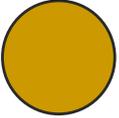
结构化分析模型





■ 数据流程图

➤ 流程图符号介绍

-  表示外部实体，代表数据源和数据池。
-  表示加工，代表接收输入，经过变换，继而产生输出的处理过程。
-  表示数据流，代表数据的流向和路径。
-  表示数据存储，代表系统加工的数据所存储的地方。



➤ 数据流程图的特点:

- 可以表示任何一个系统（人工的、自动的、或混合的）中的数据流程；
- 每个表示加工的圆圈可能需要进一步分解以求得对问题的全面理解；
- 着重强调的是数据流程而不是控制流程。

● 分层

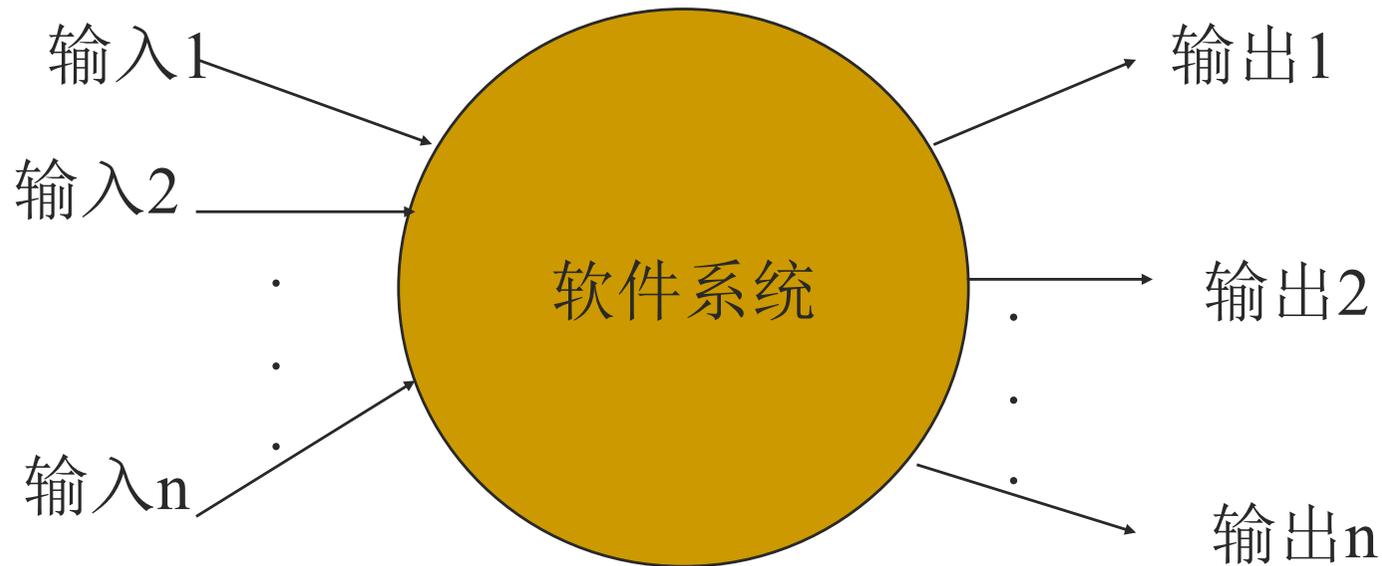
- 从高层到低层
- 分解前后的数据流必须一致

● 命名

- 数据流
- 处理



基本系统模型

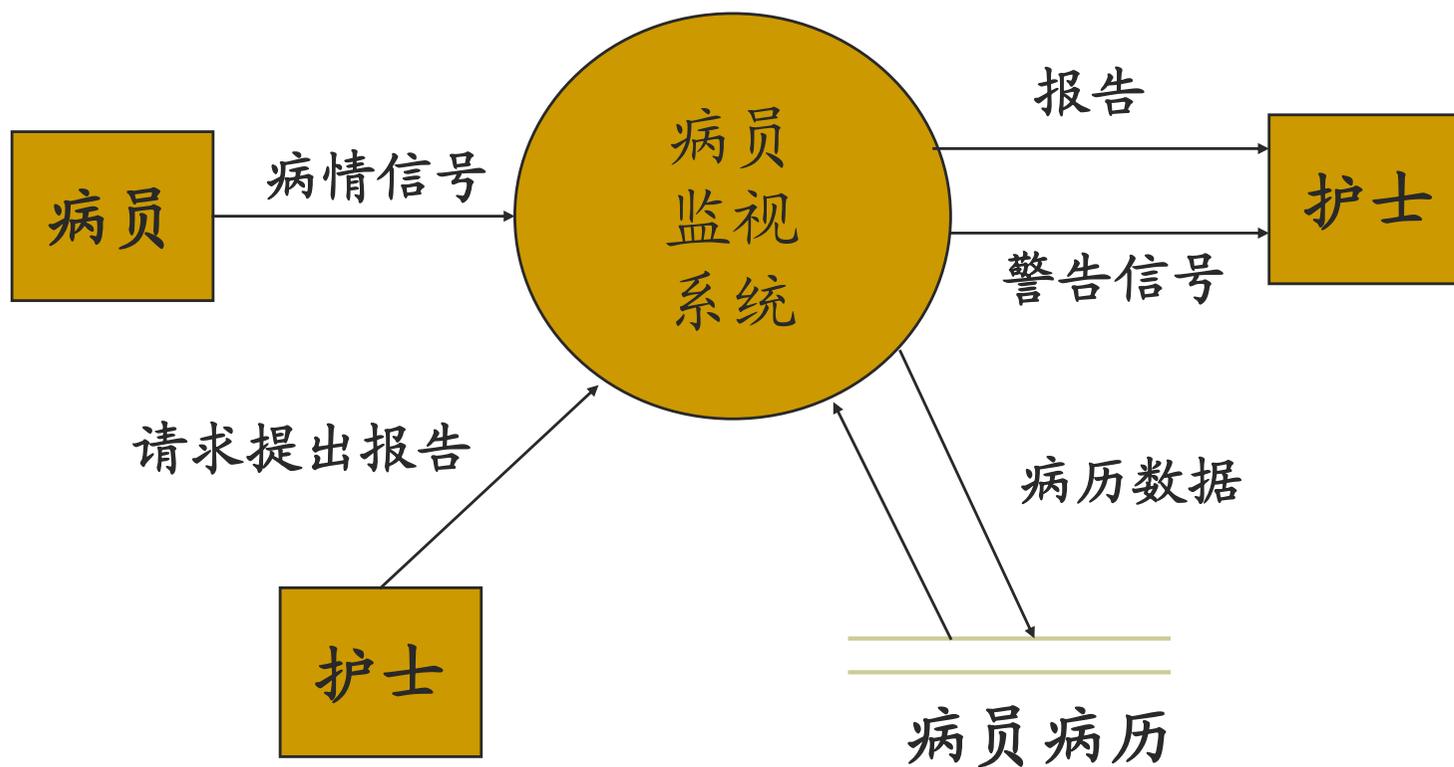


软件系统的全部功能被表示成一个单一的信息变换过程



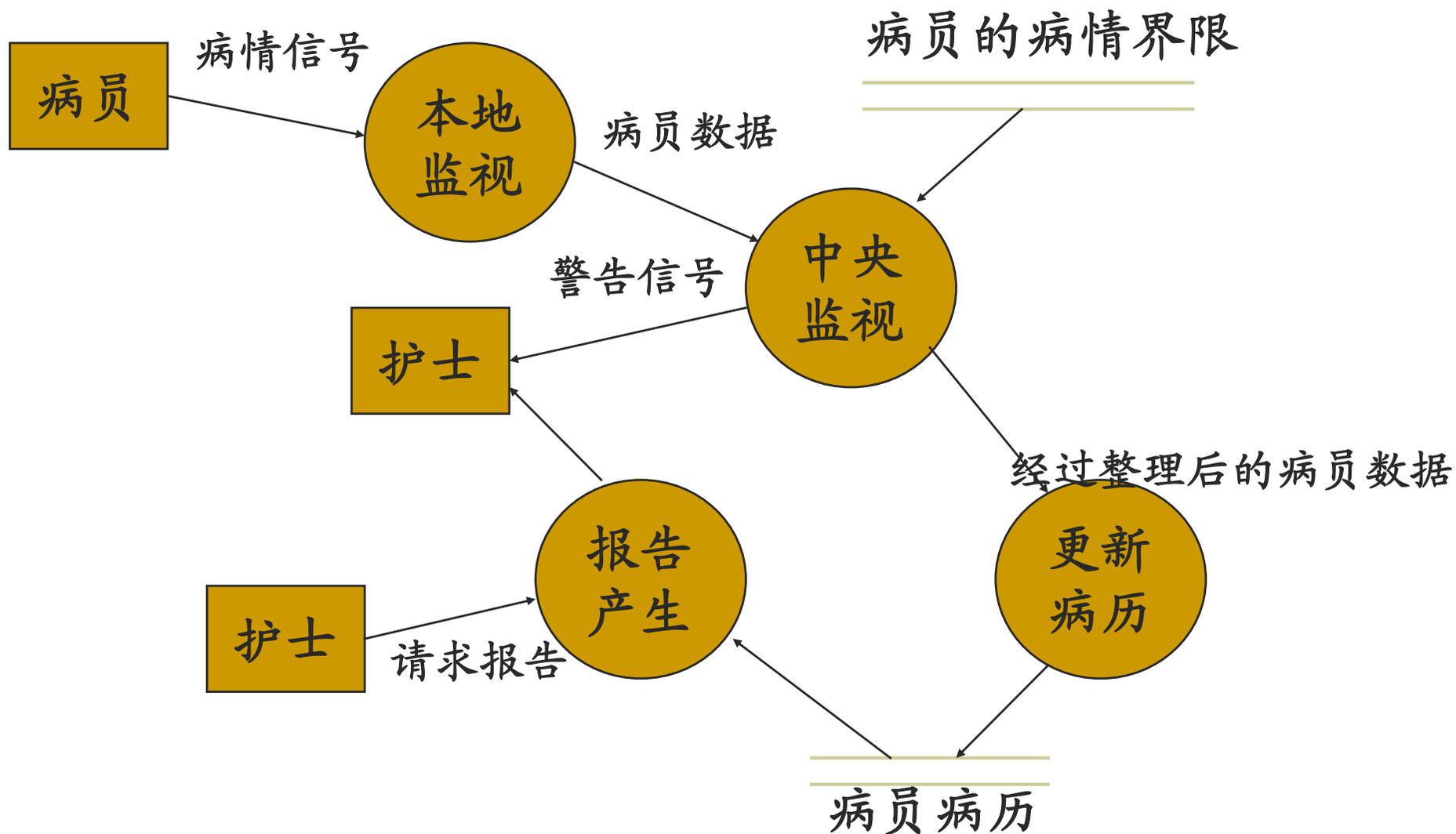
例：病员监视系统

基本模型



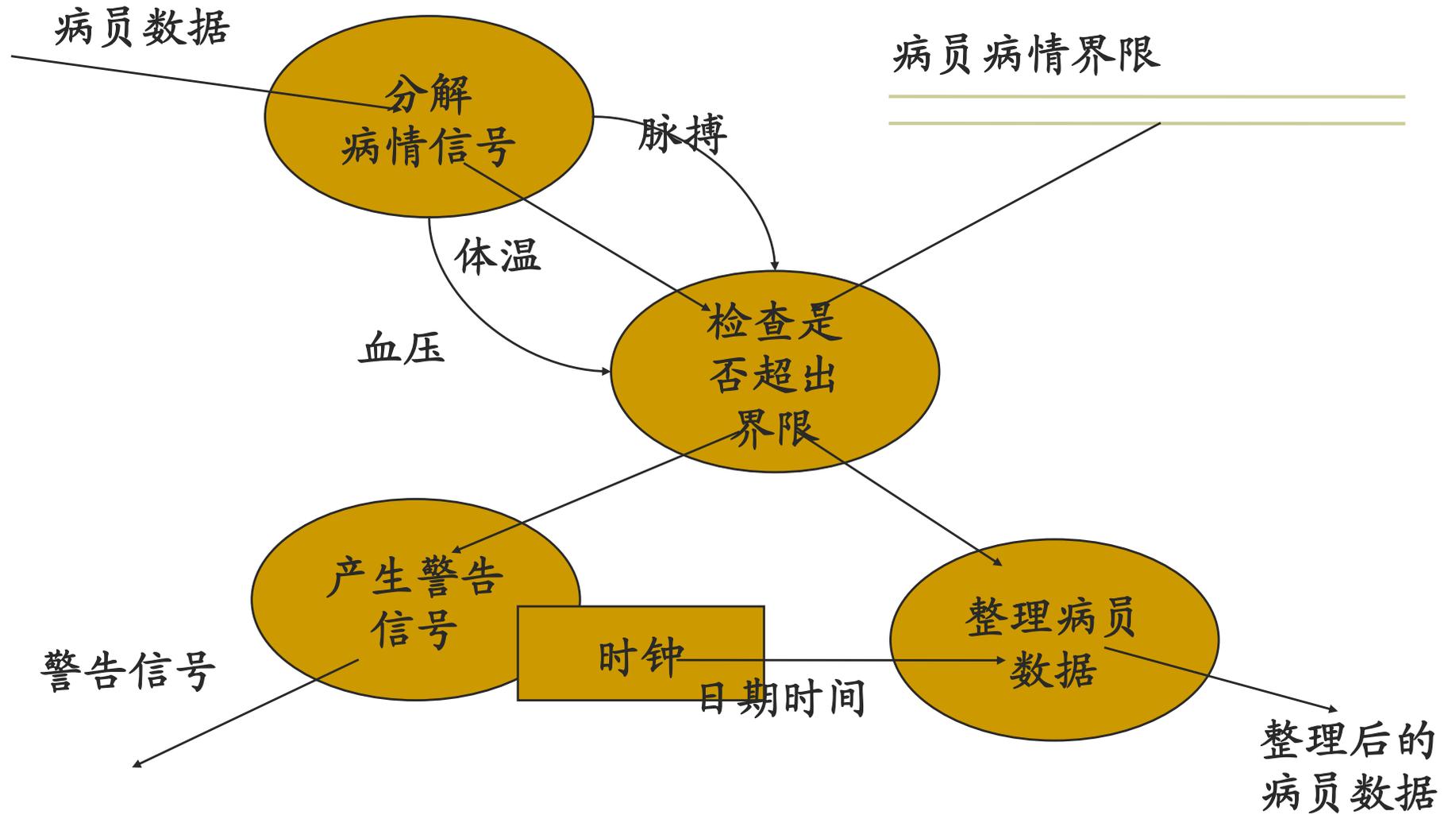


需求分析





需求分析





➤ 推导数据流程图的简单准则:

- 第一层数据流程图应当是基本的系统模型;
- 应当仔细说明原始的输入/输出文件;
- 所有箭头和圆圈均应当加上标注 (使用有意义的名字);
- 必须保持信息的连续性;
- 每次只加工一个圆圈。



功能需求分析



- 加工说明PSPEC
 - 说明DFD中的每个加工
- 描述工具
 - 结构化语言
 - 判定表
 - 判定树



功能需求分析



■ 处理方法：事件列表与功能列表

- 事件就是要求系统执行某项功能的请求
- 业务事件与产品事件
- 对复杂的业务任务采用任务说明、用例说明或数据流图等方法进行解释。
- 对复杂的功能采用数据流图、算法描述、活动图、数学说明等进行解释。



功能需求分析



- 事件及功能列表的优点
 - 主要作为核对清单，以说明应开发什么。其中对这些功能的详细说明构成了功能需求的主要部分。
 - 开发人员可以方便的检查产品是否实现每一个功能。
 - 用户能够在某种程度上确认业务事件和任务列表。
- 通过一致性检查确定列表是否完备。



数据需求分析



- 数据字典（DD）：
 - DFD中所有元素的定义的集合
- 内容
 - 数据流
 - 数据流分量
 - 数据存储
 - 处理（一般不用DD描述）
- 定义数据的方法
 - 自顶向下分解数据



数据字典



- 数据元素的组合方式
 - 顺序: $A+B$
 - 选择: $[A|B]$
 - 重复: $1\{A\}5$
 - 可选: (A)
- DD的用途
 - 分析阶段的交流工具
 - 包含控制信息
 - 数据库设计的基础



■ 数据字典

- 数据流程图中，所有的图形元素都进行了命名，所有名字的定义集中起来就构成一本数据字典。
- 数据字典最重要的用途是作为分析阶段的工具。在数据字典中建立的一组严密一致的定义有助于改进分析员和用户之间的通信，因此将消除许多可能的误解。对数据的这一系列严密一致的定义也有助于改进在不同的开发人员之间或者不同开发小组之间的通信。如果要求所有开发人员都根据公共的数据字典描述数据或设计模块，则能避免许多麻烦的接口问题。



数据字典



- 据词典精确地、严格地定义了每一个与系统相关的数据元素，并以字典式顺序将它们组织起来，使得用户和分析员对所有的输入、输出、存储成分和中间计算有共同的理解。
 - 词条描述
 - 名称：数据对象或控制项、数据存储或外部实体的名字。
 - 别名或编号。
 - 分类：数据对象？加工？数据流？数据文件？外部实体？控制项（事件/状态）？
 - 描述：描述内容或数据结构等。
 - 何处使用：使用该词条（数据或控制项）的加工。
 - 内容描述：



■ 信息结构

- 信息结构是各个数据成分之间逻辑关系的一种表示方法。
- 数据结构决定信息的组织、存取方法、结合性程度以及不同的处理方案。
- 典型的数据结构包括标量项、顺序向量、 n 维空间、链接表等。
- 分层数据结构表示方法
- 实体关系图



▶ 分层数据结构表示法:

- 分层框图
- Warnier图



- 分层框图

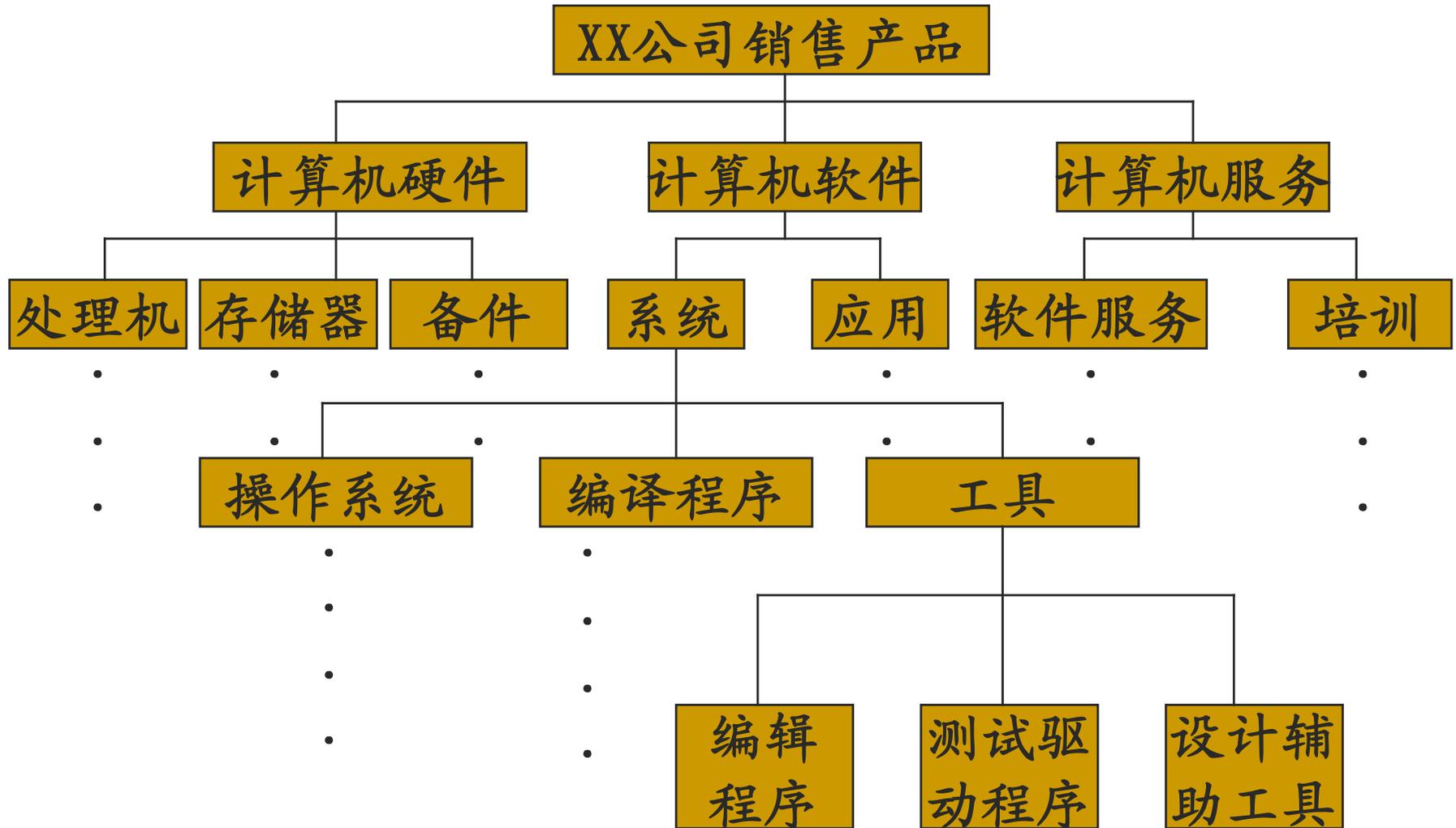
分层框图把信息用多层方框按照树形结构组织起来。在结构的顶层，用一个方框代表整个结构。下面各层由表示不同信息类别的方框组成，它们可以看成是上一层方框的子集。在该图的最低一层，每个框包含单独的数据实体。



需求分析



分层框图示例





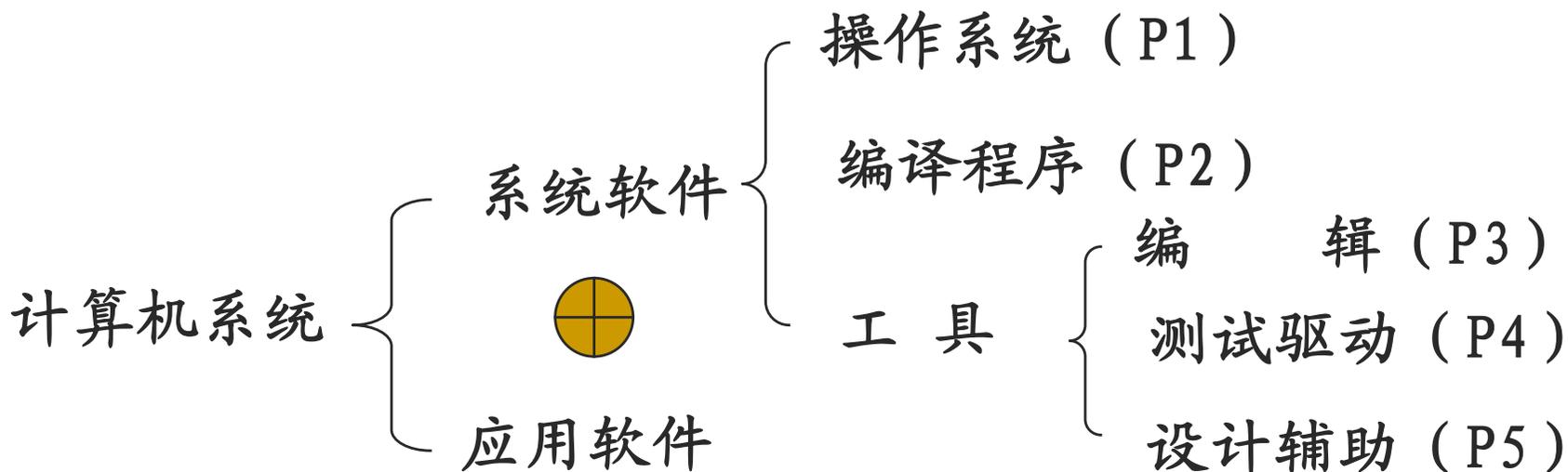
- Warnier图

Warnier图把信息表示成一种树形数据结构。可以规定某些信息种类或信息量是重复性的，也可以说明在某一种类中信息是有条件出现的。

- 1974年 J.D. Warnier提出，仅考虑输入数据结构



Warnier图示例

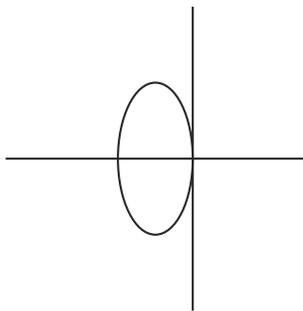




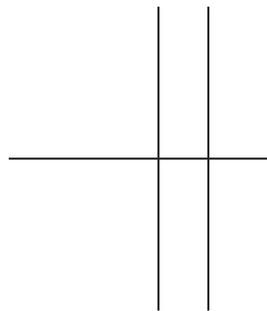
E-R图



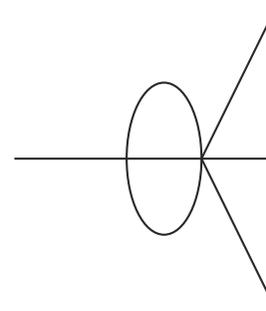
- 用于对复杂数据的数据分析和建模
- 实体、属性和关系
- 组成符号



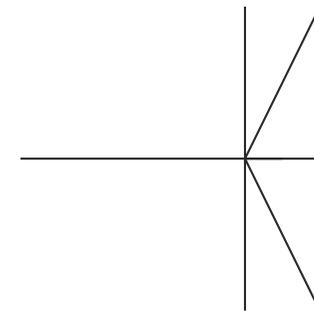
0: 1



1: 1



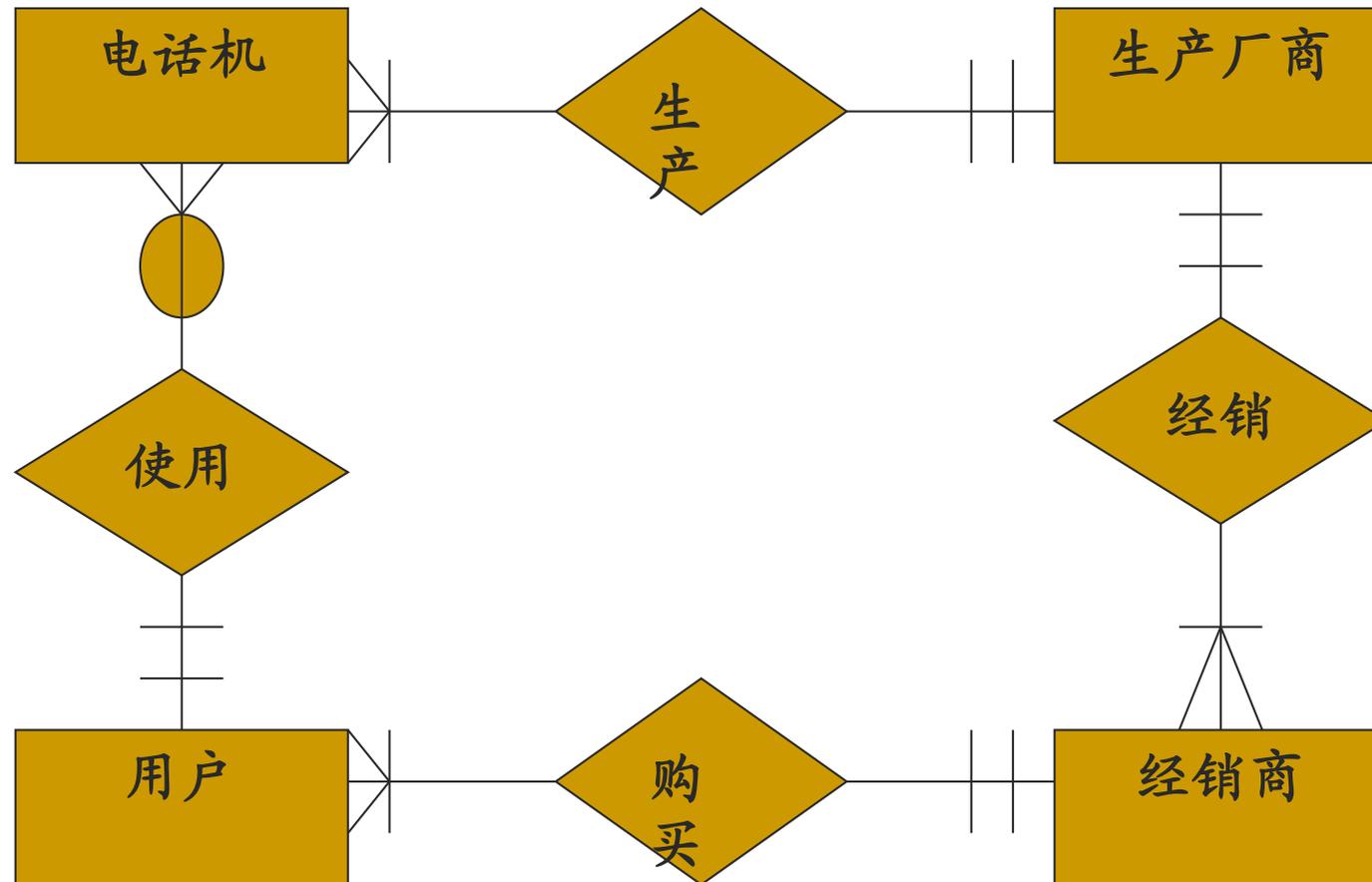
0: m



1:m



E-R图例子





数据需求与功能需求的区别



- 数据需求指定了系统的存储数据。
- 功能需求则说明数据的用途，以及如何记录、计算、转换、修改及传输数据等。



状态迁移图

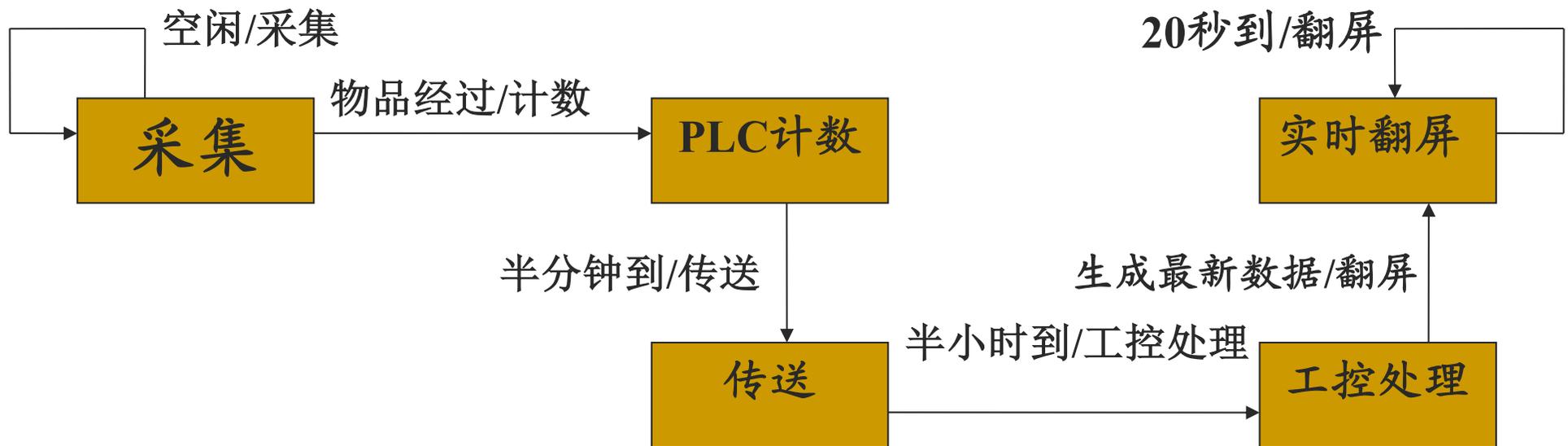


- 状态转移图STD (State Transition Diagram)
- 描述软件状态变迁及其触发事件
- 符号表示
 - 矩形---系统状态
 - 箭头---状态转变方向
 - 规则表达式---事件/触发行为





STD





结构化分析过程



- 获取软件需求
- 运用抽象和分解的技术，提供一组经验和规则以指导需求分析
 - 分析系统的功能和行为，用合适的模型表达
 - 采用自顶向下、逐步求精的策略
 - 分层建立数据流图
 - 完整、详尽、一致
 - 用数据字典、说明作为补充
- 描述需求
- 验证需求



结构化分析辅助工具



- MS VISIO
- Statemate
- 功能
 - 建模(数据流图, 数据字典, . . .)
 - 模型的存储, 显示和检索
 - 模型之间, 数据条目之间的一致性检查