目录

2 基于活动的过程建模方法

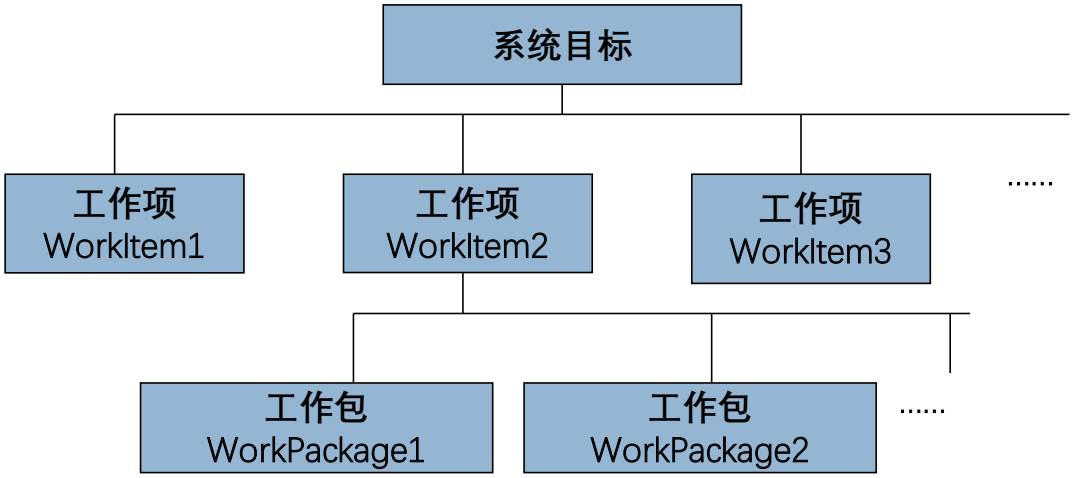
- 基于任务的流程建模，即基于功能的建模方法，是最为基本的业务建模描述方法。
- 其本质是按照处理时序，从顶向下，按照先后早晚，作为过程建模的出发点。
- 主要包括：业务任务划分，软件功能结构设计，面向执行的活动系列以及控制逻辑的描述 。
  - 任务划分（业务层面的任务规划）
  - 功能设计（软件层面的功能分解）
  - 时序描述（执行层面的时序逻辑/活动流描述）
  - 活动控制（活动执行时的活动处理及控制逻辑 ）

2.1 任务分解方法

- 任务分解方法主要采用工作分解结构（简称WBS）；
- 按照项目→任务→工作→活动分解：
  - 把一个项目，按一定的原则分解，项目分解成任务，任务再分解成一项项工作，再把一项项工作分配到每个人的日常活动中，直到无法分解。
- WBS的分解强调按照项目的目标进行分解，具体可采用的主要分解方式包括：
  - 按产品的物理结构分解
  - 按产品或项目的功能分解
  - 按照实施过程分解
  - 按照项目的地域分布分解
  - 按部门/职能分解

工作分解结构（WBS）方法

- 层次化树状结构图的WBS：
  - 层次清晰，非常直观，结构性很强，但不是很容易修改，对于大的、复杂的项目也很难表示出项目的全景。
- WBS将项目的“交付物”自顶向下逐层分解到易于管理的若干元素，以此结构化地组织和定义了项目的工作范围。



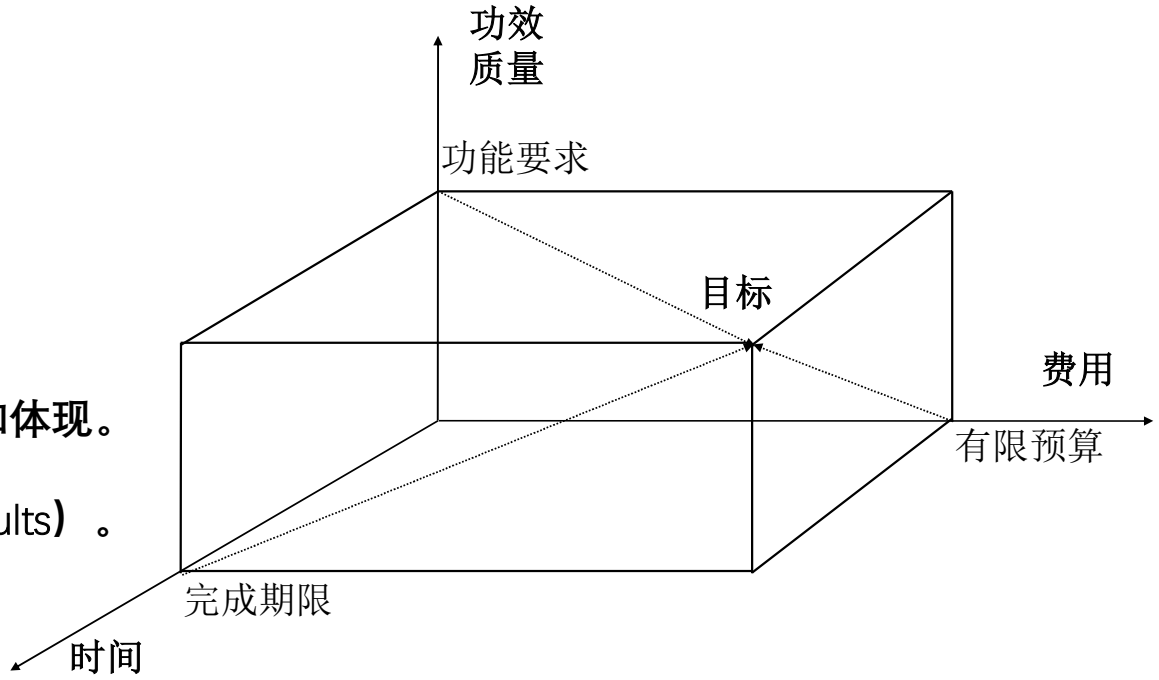
其他常见分解结构包括:
- 组织分解结构（OBS）
- 资源分解结构（RBS）
- 材料清单（BOM）
- 项目分解结构（PBS）

业务目标驱动的任务分解

目标的特点：

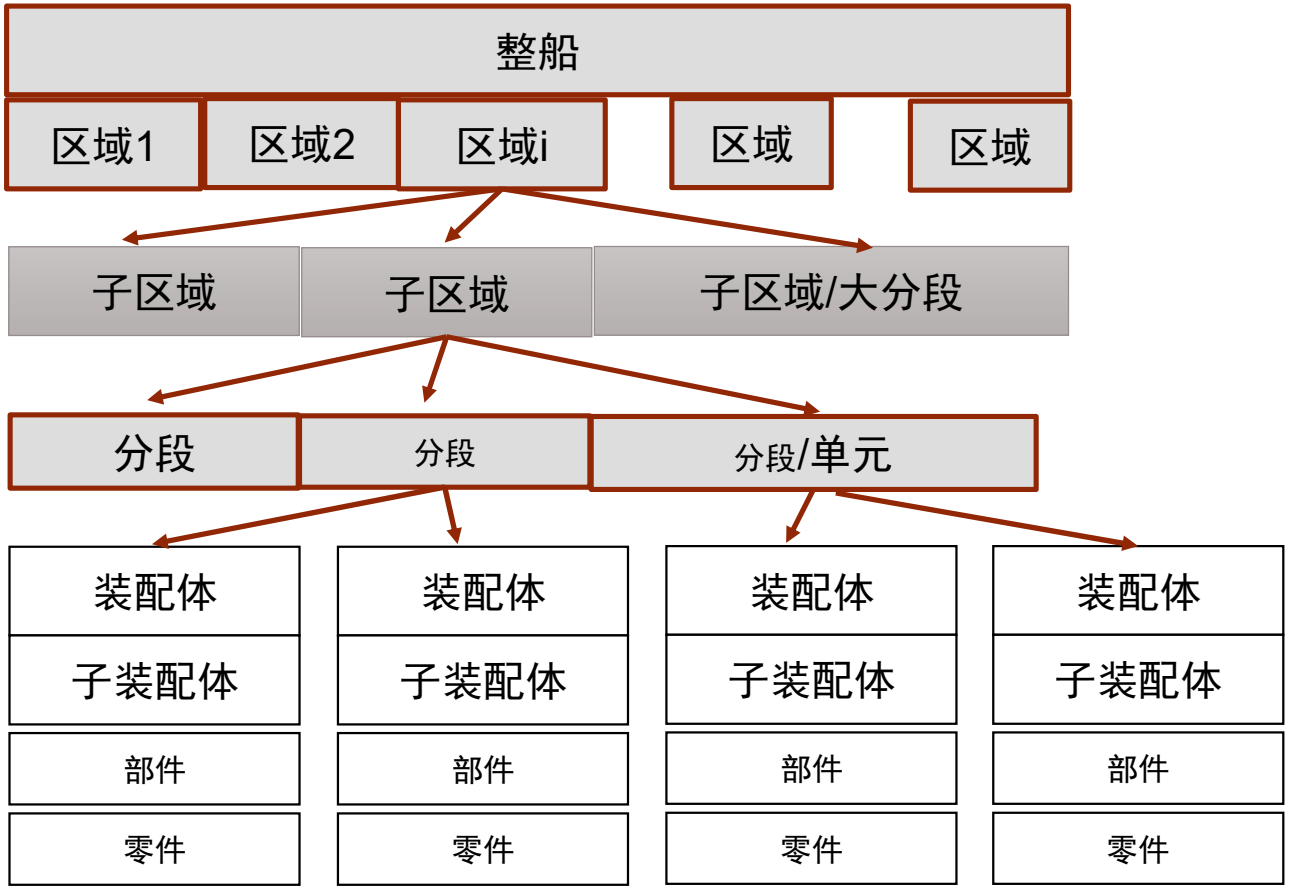
- 多目标性：TQCS等目标的对立统一
- 层次性：目标系统的递阶层次结构
- 一致性：各层次的目标具有一致性
- 优先性：关键结果对应的目标优先

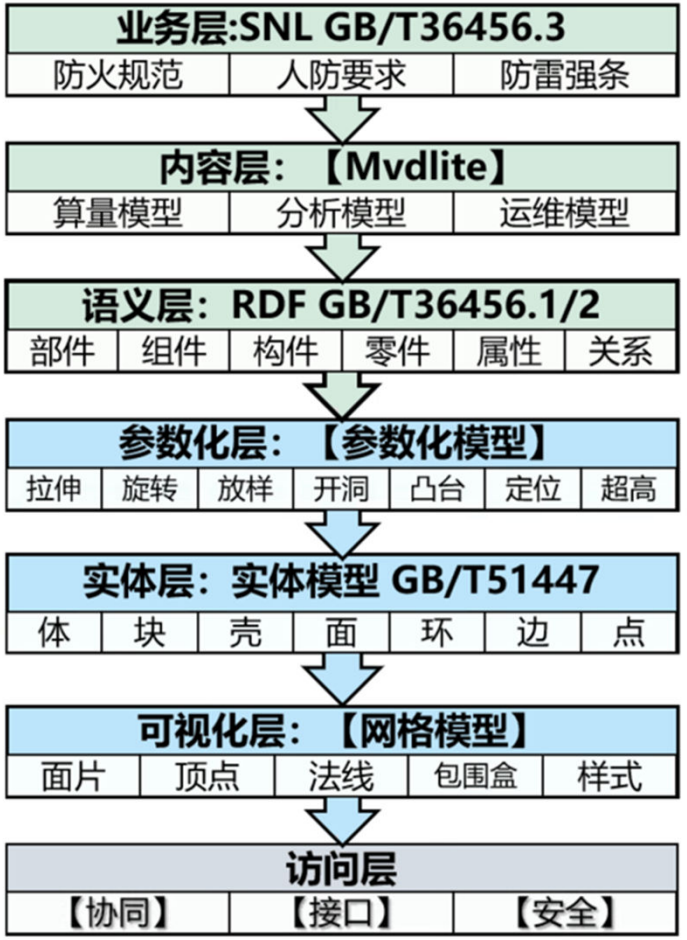
- 目标是方向，而关键成果则是过程进展的衡量和体现。
- 确定目标的优先级是成功的关键。
- OKR目标与关键成果法（Objectives and Key Results）。



WBS分解实例

- 现代造船主要采取分段制造或者分道制造模式；
- 对应的产品结构主要包括：





可信BIM软件标准协议栈

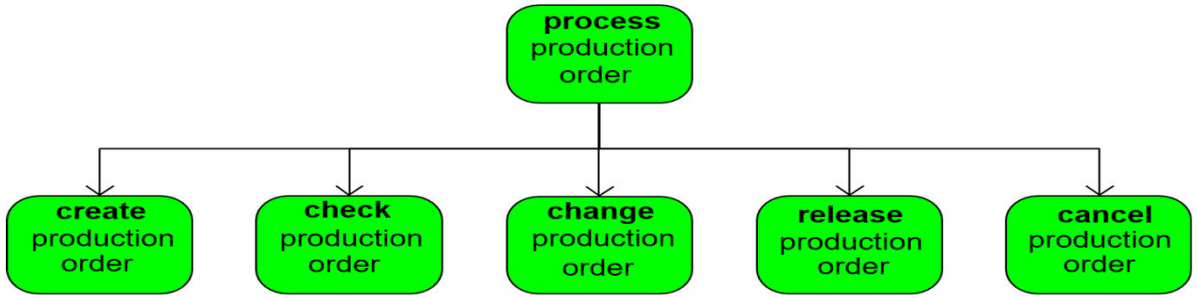
2.2 软件功能设计

- 业务任务确定之后，便可以开展软件功能的设计，即软件模块划分。
- 软件功能设计的基本方法：
  - 根据角色活动进行划分（以角色的任务区分为主）
    - 主要划分方式，减少交互，符合高内聚低耦合原则。

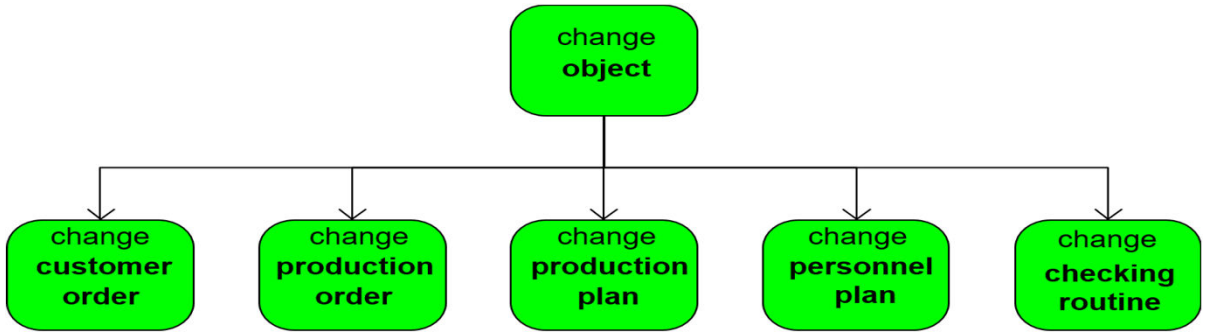
- 具体来说，软件系统模块划分方法还有：
  - 根据信息对象的划分（面向信息服务的开发方式）
  - 根据操作的任务划分（具体功能的划分）
  - 根据流程的任务划分（子系统的划分）

功能模块的划分方法

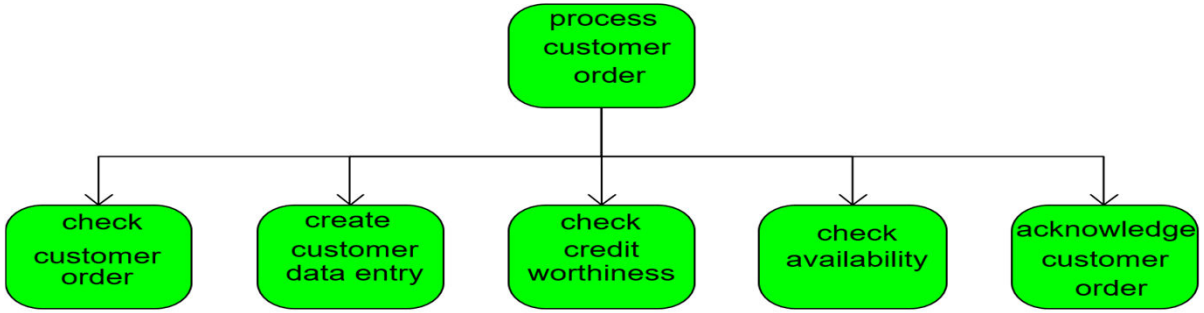
基于信息对象的模块划分



基于操作的模块划分

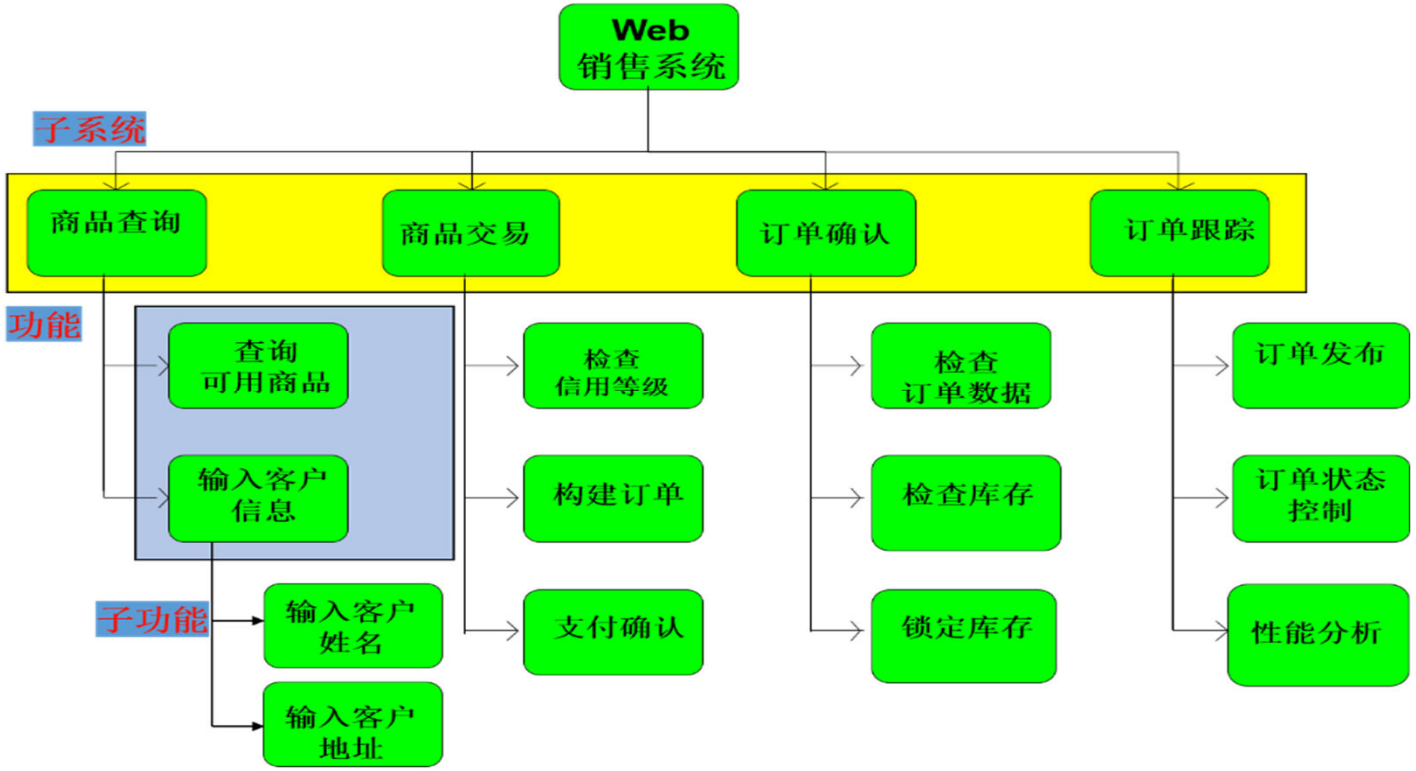


基于处理流程的模块划分



功能分解

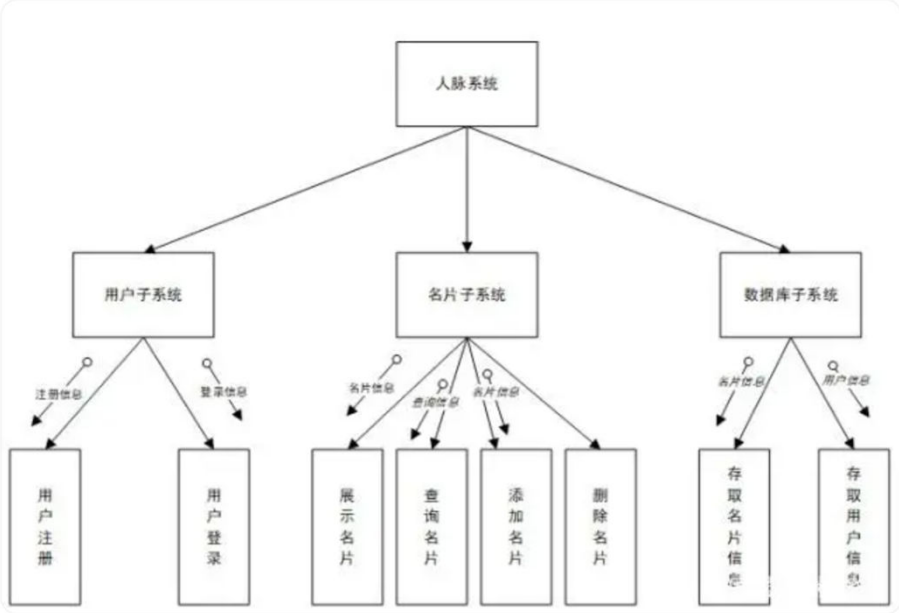
- 体现到软件系统中，任务划分主要对应功能的分解；
- 功能分解图的本质是从顶向下将一个复杂的系统分割成小的模块，分层组织功能模块
- 模块按照高内聚、低耦合方式设计并组织。



- 模块划分质量指标的主要是根据划分后的组件或者模块的耦合度和内聚性两方面
  - 耦合度指的是一个功能组件与其他功能组件的相关程度。
  - 内聚性指的是功能组件内部的凝聚程度。
- 高内聚、低耦合原则的设计实现。

- 微服务架构软件和多层架构软件如何满足这样的原则?

- 功能分解图描述了系统中的功能和子功能以用于软件业务分析。
- 类似软件设计中的模块结构图
  - 每个模块需要完成某一特定功能，高层模块对低层模块有调用关系
  - 各模块均有控制逻辑和错误处理逻辑
  - 在端点或叶子节点上的模块包含执行程序功能的确切算法
  - 一个低层的模块永远不会调用高层模块



基于数据聚类的模块划分方法

- 基于数据聚类的软件模块划分也是一种可行方法，主要过程包括：
  - 任务相关性矩阵构建
  - 构造聚类谱系图
  - 分解演化相关任务，重新建立任务关系
- 案例
  - 现有某软件系统相关的一批不同粒度功能任务1-11待处理，具体任务名称如下：
  - (0)模块集成，(1)报表插件开发，(2)生产报表生成功能，(3)制造流程开发，(4)领料功能开发，(5)库存报表开发，(6)产品物流开发，(7)安全存储功能，(8)数据库设计，(9)数据调用接口，(10)用户验证，(11)数据库元数据。
  - 另外0号任务为总体集成任务，由企业自行完成，其余任务则需要外协到其他外部企业进行实现。

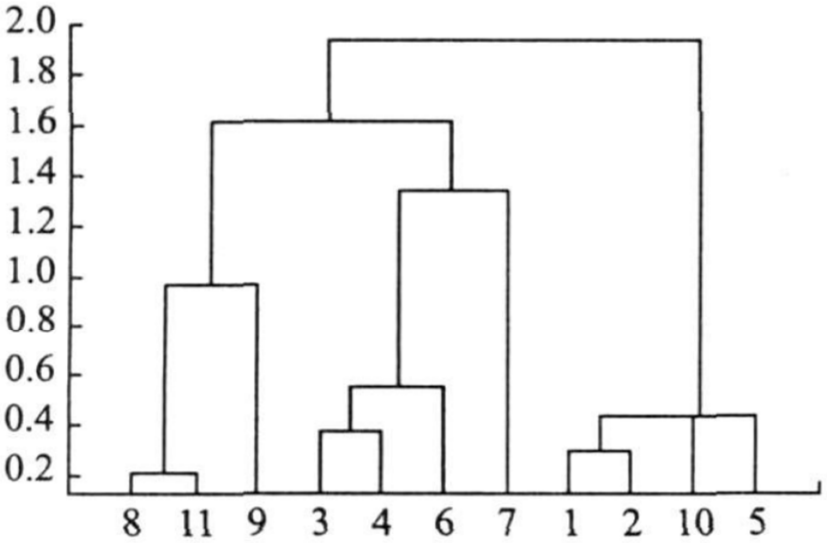
- 任务相关矩阵的构建，基于两两任务之间的相关性

$R_{Tij}=$

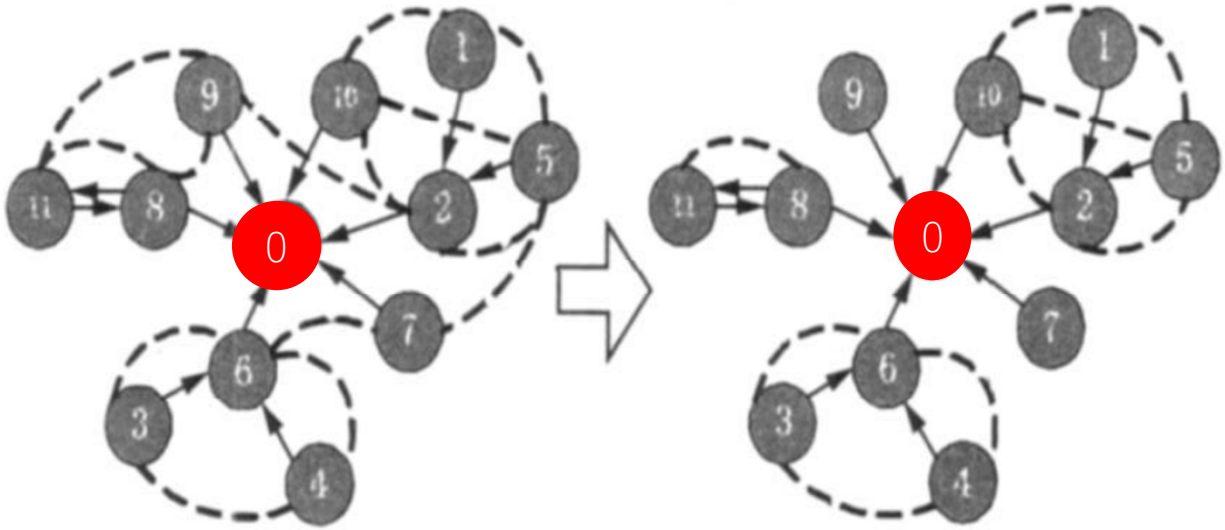
	MT_0	MT_1	MT_2	MT_3	MT_4	MT_5	MT_6	MT_7	MT_8	MT_9	MT_{10}	MT_{11}
MT_0	0 | 1											
MT_1		0 | 1.00	1 | 0.85			| 0.70					| 0.70	
MT_2	1 |	| 0.85	0 | 1.00			| 0.85					| 0.85	
MT_3				0 | 1.00	| 0.75		1 | 0.85					
MT_4				| 0.75	0 | 1.00		1 | 0.75					
MT_5		| 0.70	1 | 0.85			0 | 1.00					| 0.50	
MT_6	1 |			| 0.85	| 0.75		0 | 1.00	| 0.5				
MT_7	1 |						| 0.50	0 | 1.0				
MT_8	1 |								0 | 1.00	| 0.5		1 | 0.95
MT_9	1 |								| 0.50	0 | 1.0		| 0.30
MT_{10}	1 |	| 0.70	| 0.85			| 0.50				0 | 1	
MT_{11}									1 | 0.95	| 0.3		0 | 1.00

任务划分过程实例

- 任务的聚类谱系图

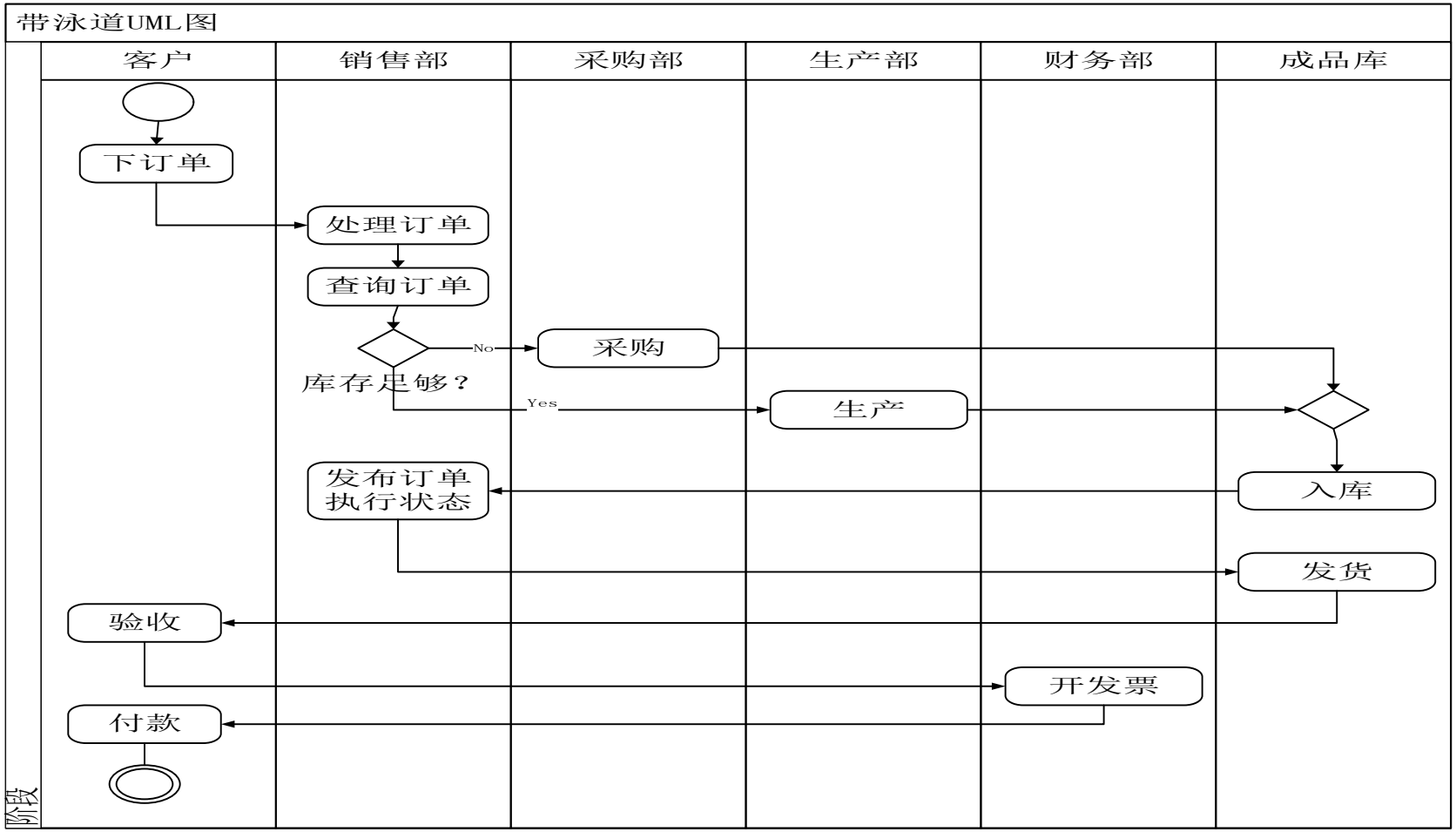


- 任务分解示意图



2.3 活动的时序描述

- 活动图描述的是对象活动的顺序关系所遵循的规则，描述对象之间传递的操作，多个组织或者角色的描述方式，一般采用带泳道的活动图进行描述。



角色行为图

- 角色行为图RAD（Role Activity Diagram) 是一种常见的流程图形化描述方法，主要用在组织建模中。
- RAD基于活动的执行时序描述不同组织的交互活动。
- RAD的基本思想是组成流程的活动按角色聚类，以便能从各角色执行的活动集中把握它们的责任，而且可以把注意力集中在角色之间的交互活动上。
- 它特别擅长强调流程中的角色职责。

角色行为图

- RAD元素符号说明：

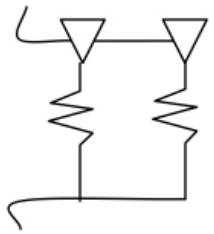
角色（Role）

状态（State）

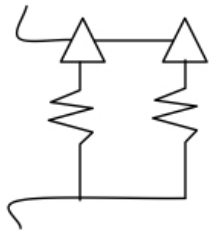
状态描述
(State
Description)

活动
(Activity)

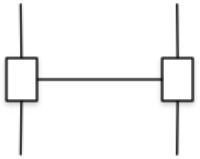
外部触发事件
(External
Event Occurs)



选择路径
(Alternative
Paths)



并行路径
(Alternative
Paths)

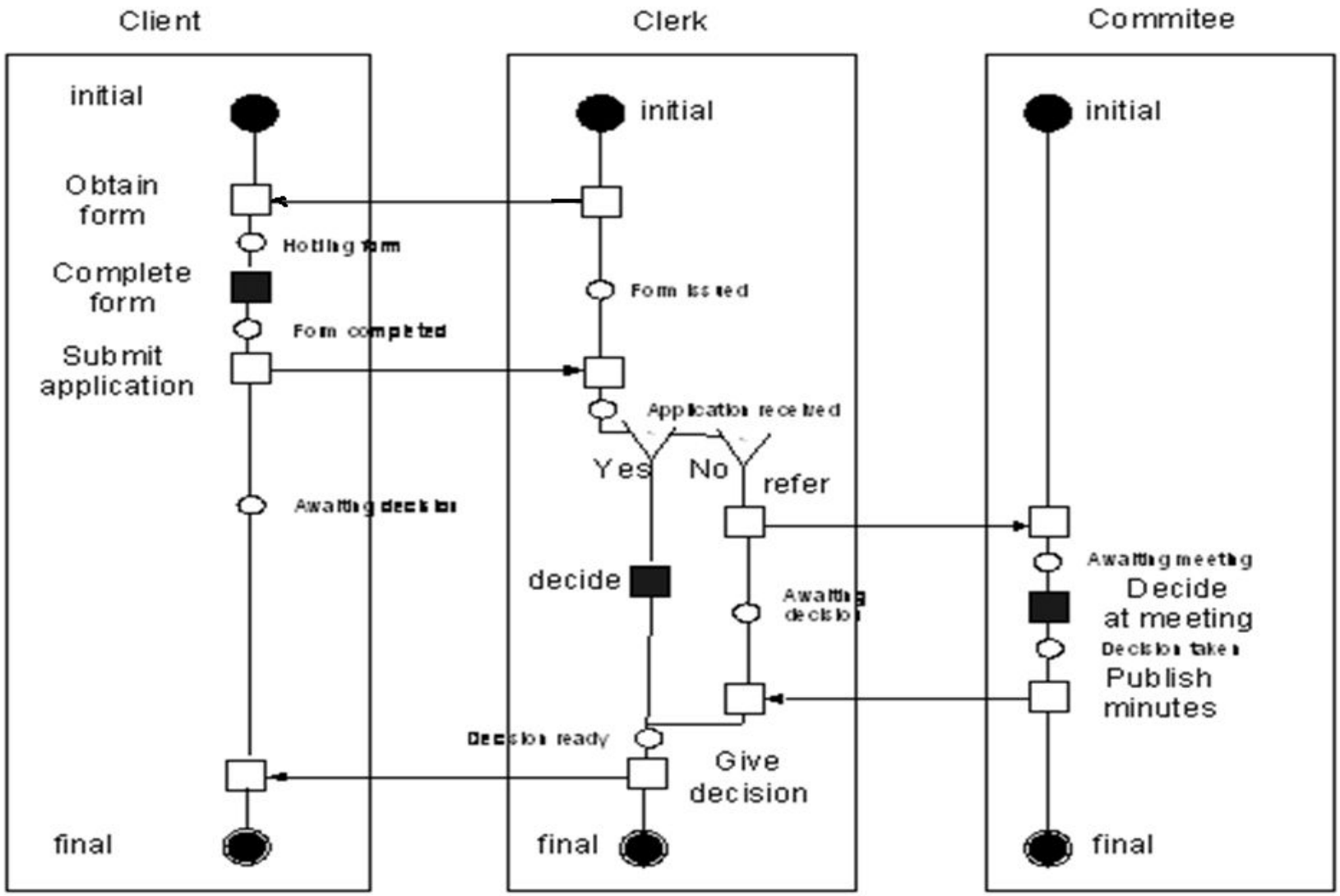


角色间相互作用
(Interaction
between roles)

RAD图基本概念

- RAD图的五种基本模型及概念：
  - 角色：通常是由担负一定职责的角色所完成的一组活动；
  - 目标：通过交互过程期望达到的最终状态或状态集合；
  - 活动：前置状态和后续状态，确定了各个活动的发生顺序，形成一种时间流序列；
  - 交互：并不表明每个交互的物理形式，而是作用的意义，包括同步、对象的传递和互换；
  - 规则：企业用什么样的约束条件规定人们能做什么和应该怎么做，RAD包括了顺序、决策和并行等将它们联系在一起的经营规则。

一个银行贷款的RAD图实例



RAD主要特点

- **按角色分解流程**：角色通常是由担负一定职责的个人或部门所完成的一组活动。与角色相关联的是对应任务的资源。流程中的角色是相对独立的，利用各自的资源集完成相应的活动，并通过与其他角色的交互实现工作协同。
- **基于角色状态描述的流程目标**：目标表述了一种功能性的状态，即流程试图达到的一种状态或状态集合。
- **按时序执行的活动**：按照时序连接，依次执行。
- **关注角色的交互作用而不是形式**：在交互方面，RAD 并不表明每个交互作用的物理形式，我们关心的是作用的意义。
- **体现企业的业务规则**：通过路径控制，实现企业经营规则。角色、目标、活动和作用这些概念在RAD中都有专门的符号表示，而顺序、决策和并行性等将它们联系在一起，体现经营规则的路径控制也有两种标识符，分别是路径选择和并行路径。

RAD图评价

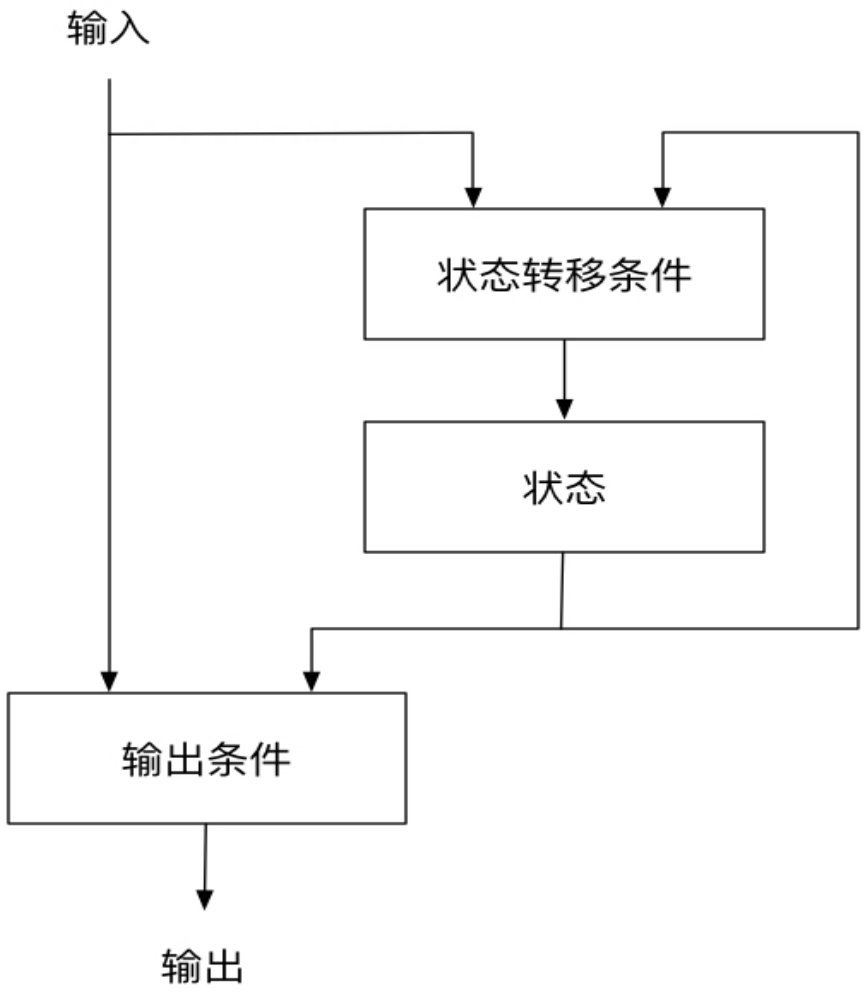
- RAD可以由传统基于活动的流程图派生而来。从组成元素上看，除活动外，RAD还包括状态和控制等元素，状态表明活动的原因和结果，控制表示路径选择等。
- RAD表示法扩展了基于活动的流程表示，虽然使角色的交互明晰化，但也增加表示的复杂性，因此不便于表达交互特别复杂的流程。
- RAD方法主要不足之处在于不具有模型分解的能力，用于模型流程总览适合，但在一些情况下，无法支持深入的流程描述。
- 面向复杂性问题，RAD的解决方式：
  - 非层次化划分，通过嵌套，一个自有的活动可扩展为新的RAD图，相关角色和活动可以重新展开。
  - 这与很多传统系统分析工具采用层次化的结构分解不同。不过，有时企业经营过程并不太适合结构化分解，它更趋向于多维网络结构。
  - 因此，RAD图在一定场合是适用的。

2.4 任务活动的执行控制

- 任务执行控制的理论支撑是有限状态机（Finite State Machine），又称有限状态自动机或简称状态机。
- 有限状态机FSM：有限个状态以及在这些状态之间的转移和动作等行为的数学模型
  - 状态：反映从系统某一时刻的条件。
  - 转移：指示状态的变更，用转移发生的必须条件来描述。
  - 动作：在给定时刻要执行活动的描述，动作的类型有：
    - 进入动作（Entry action）：在进入状态时进行
    - 退出动作（Exit action）：在退出状态时进行
    - 输入动作（Input action）：依赖于当前状态和输入条件进行
    - 转移动作（Transition action）：在进行特定转移时进行

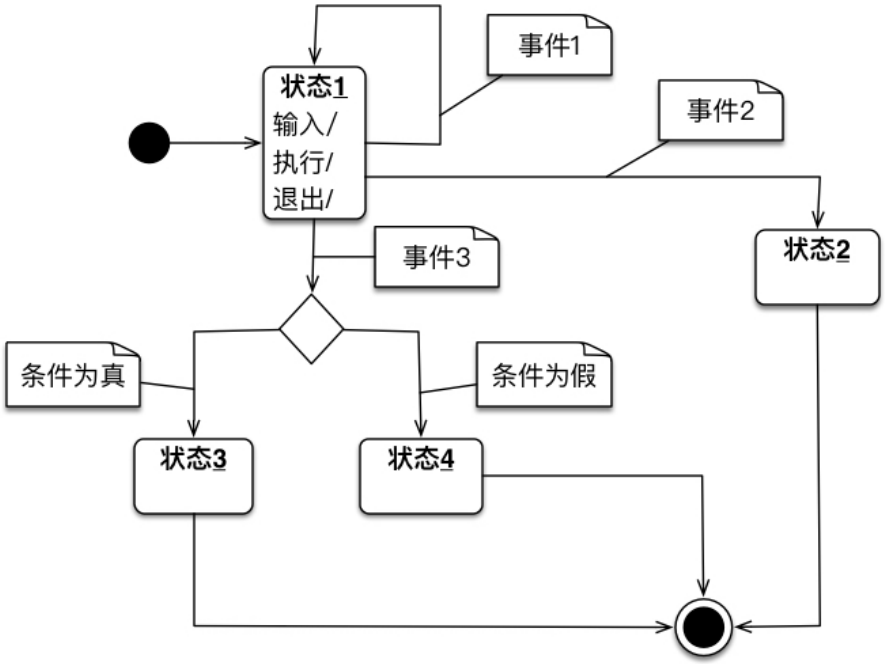
有限状态机逻辑图

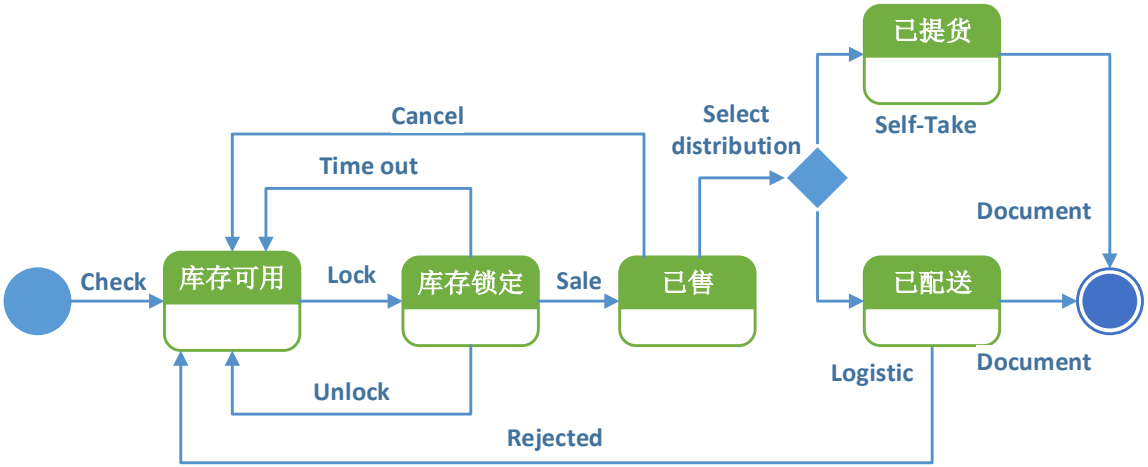
- 状态机专门用于定义依赖于状态的行为，根据模型元素所处的状态而有所变化的行为。
- 状态机由状态组成，各状态由转移链接在一起。
- 状态是对象执行某项活动或等待某个事件时的条件。



有限状态机符号

- 状态图（State Chart Diagram）描述了一个信息实体对于事件反应的动态行为，显示了该实体如何根据当前所处的状态对不同的事件做出反应。





状态转移表

- 状态转移表：描述资源状态变化的另一种方式。反映了什么状态在什么事件驱动下可能转移到什么状态，还为资源状态转移矩阵的构建进行了初步分析 。

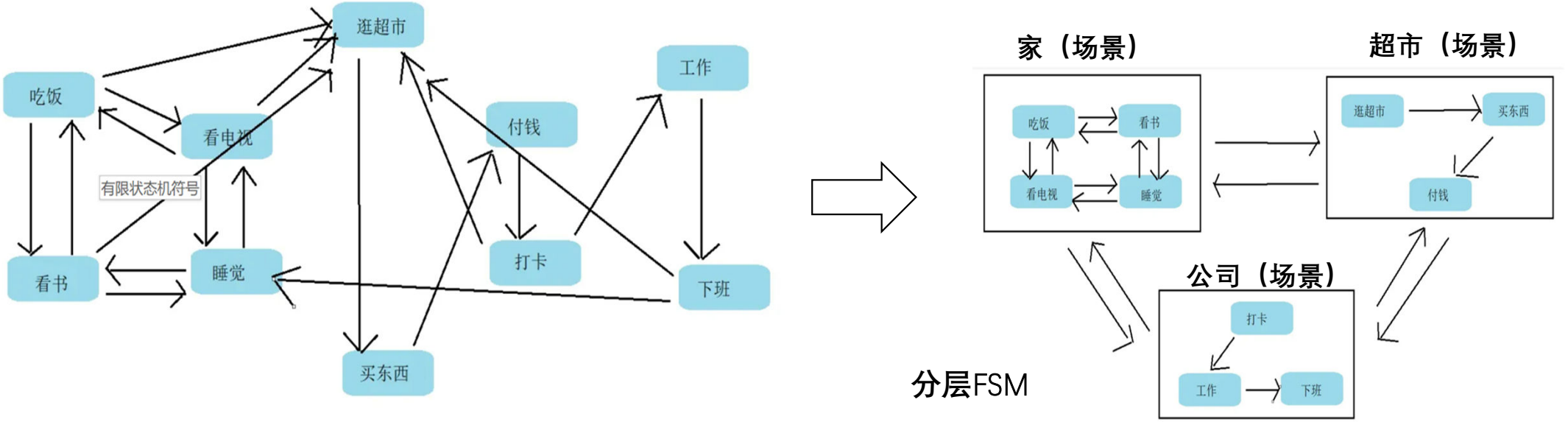
当前状态 → 条件 ↓	状态 S_1	状态S_2	状态S_3	状态S_4
条件C_0	…	…	…	状态S_1
条件C_1	状态 S_2	…	…	…
条件C_2	…	状态S_3	…	…
条件C_3	…	…	状态S_4	…
条件C_4	…	…	…	状态S_0

从Agent到AI agents

多Agent系统

- 由分布在网络上的多个问题求解器松散耦合而成的大型复杂系统，每一主体具有有限信息资源和问题求解能力，系统不存在全局控制。
- 知识与数据都是分散的；计算是异步执行的。
- 主要有：集中控制，层次控制，网络控制（控制结构动态可变）等方式。

活动执行的复杂状态控制



分层FSM