

信息系统分析与设计

Lecture 1

业务建模概述

蔡鸿明

hmcai@sjtu.edu.cn

软件学院信息系统技术实验室

lst.sjtu.edu.cn



Agenda

1. 业务模型
2. 企业业务建模体系
3. 业务流程描述方法
4. 小结



1 业务模型

信息系统实施开发的三种模式：

■ 定制开发

- 不考虑时间成本和质量等问题的话,可解决所有信息需求;

■ 基于套件的实施

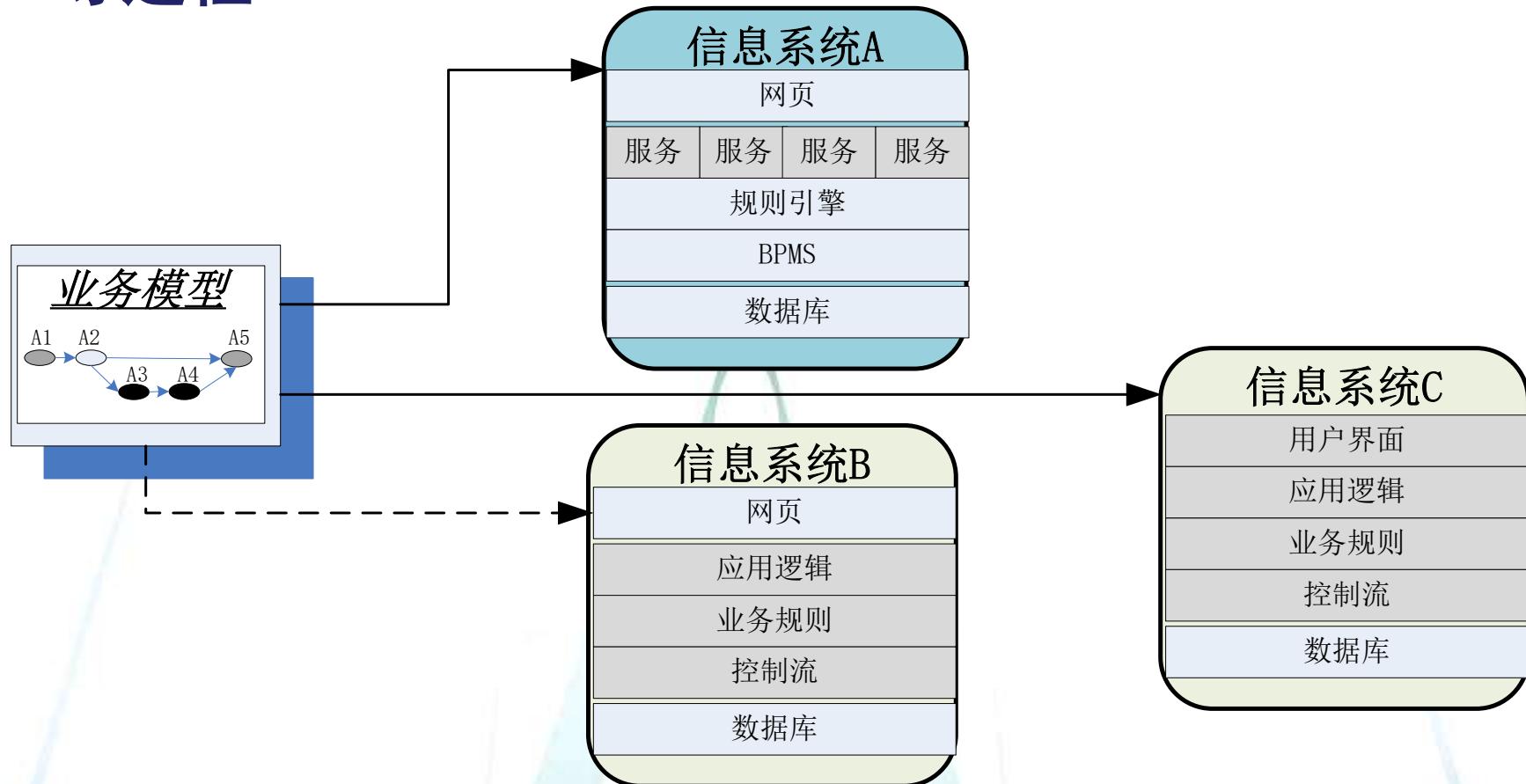
- 例如在 SAP R/3, ERP2005, Oracle EBS;

■ 业务模型驱动IT架构的开发实施

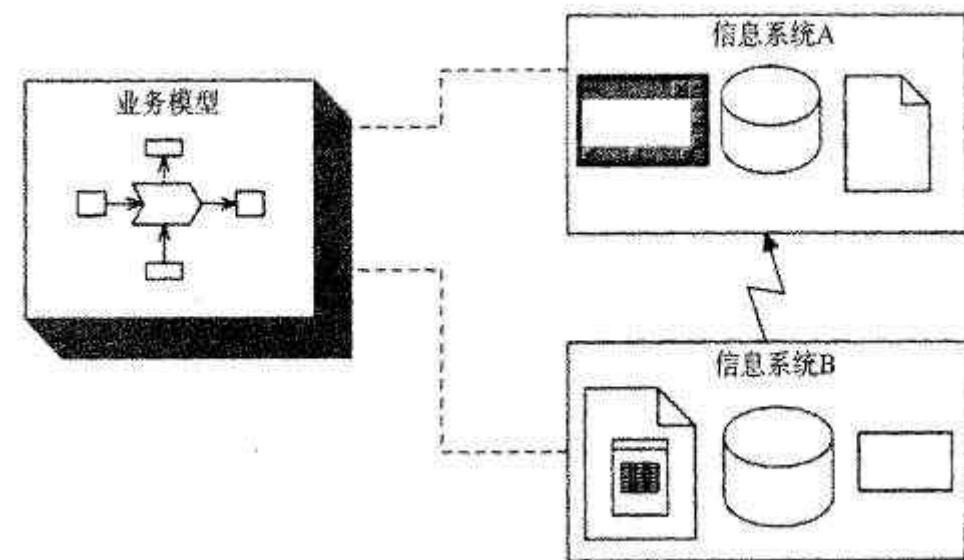
- 基于服务,基于构件,业务转换;



业务建模,模型转换,系统配置,服务监控, 持续治理等过程



- 业务建模的目的是**描述**出反映企业核心业务功能的复杂现实，构架业务领域到技术领域的**桥梁**，从而更好地理解业务的关键机制、或者作为信息系统实施的业务基础、开展业务改进、发现新业务机会、明确外购需求等。
- 理想情况下，业务模型中所描述的对象可以转变或者映射为信息系统中的对象。通常情况下，这并不是一一映射。
- 信息系统中有许多在业务模型中完全没有涉及的对象，反之亦然。





业务模型

- 面向企业业务分析
- 属于业务层面
- 层次较高：组织、业务概念，流程...
- 支持框架：ARIS, Zachmann, UML业务建模框架



系统模型

- 面向软件系统开发
- 属于技术层面
- 可能比较底层，数据类、功能、进程...
- 支持框架：SSH, RUP



什么是信息建模？

1. Model & Prototype

Model 模型是

- 简化，假设
- 抽取感兴趣的部分属性

Prototype 原型是

- 一种初始的类型、形式或例证，是作为其后期阶段的基础
- 相对完整

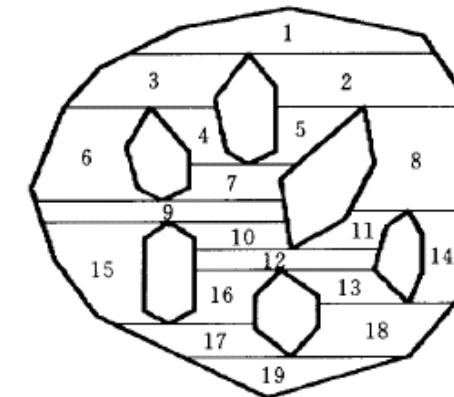
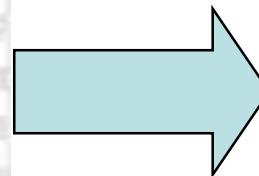
2. 信息建模&数学建模



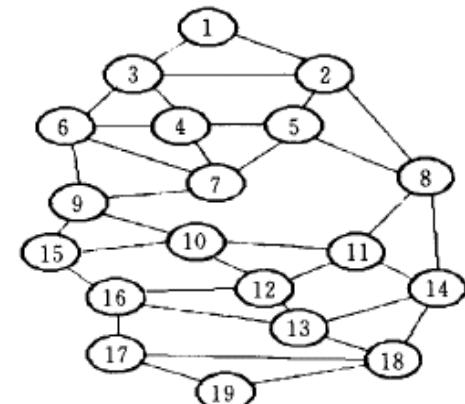
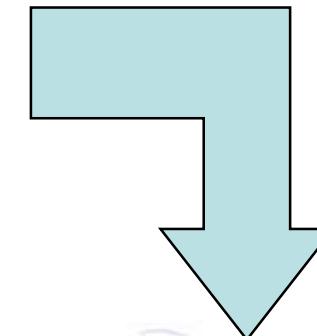
一个建模实例（绣花问题的算法建模）



图案实例

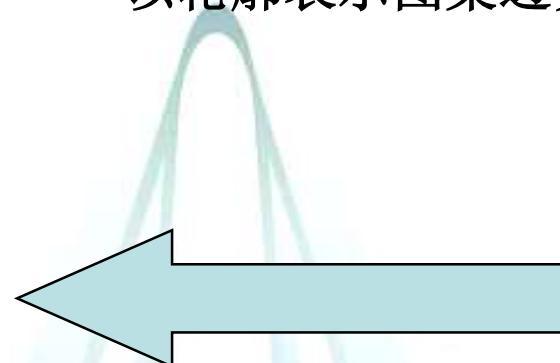


以轮廓表示图案边界



图的表现形式

图的遍历问题





哪些模型能用于描述企业?

- 功能:

检查信用度, 输入客户订单, ...

- 数据:

商品, 客户, 材料, 供应商, ...

- 组织单元:

销售, 采购, 会计, 生产部门, ...

- 事件:

客户订单已到达, 发票已寄出, ...

- 资源:

PC, 文件, 车床, ...

- 服务/产品:

BPR咨询, 芯片, 电路板, PC



这些不同对象不是孤立的,
存在相互关系。

- 层次关系
- 关联关系
- 时序关系
-

企业业务模型

在IT领域，标准而规范的描述手段，就是建模。

业务模型是描述、开发、实现、优化企业信息系统的方法基础：

- 模型是具有完整语义的一组信息，包括两方面：一方面，对现实的简化或者对实现的化简；另一方面，认知主体的视角和抽象层次。
 - 认知活动的主体决定了特定的视角，可以理解为简化的动机（视图差异）
 - 认知活动的原则决定了特定的抽象层次，可以理解为简化的水平（粒度差异）

企业模型具有多层次、多视角、复杂关联、时序相关等特点！



业务模型分类

从抽象的角度

- 概念模型
- 逻辑模型
- 物理模型

从内容的角度分

- 流程模型
- 功能模型
- 信息模型
- 数据模型
- 资源模型
- ...

从表示形式

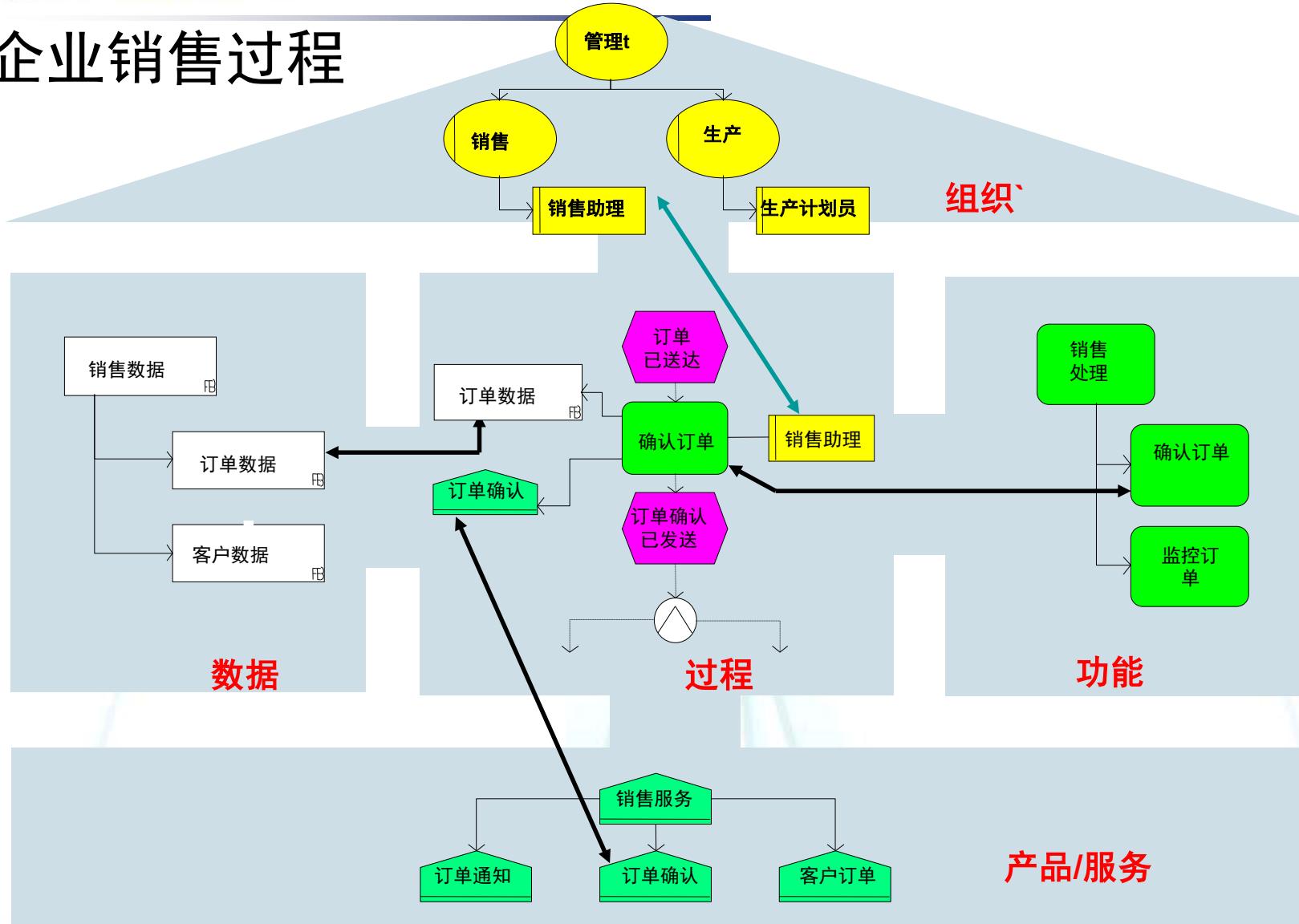
- 可视化模型
- 形式化模型

从（工作流）模式来分

- 控制模型
- 资源模型
- 数据模型
- 异常模型

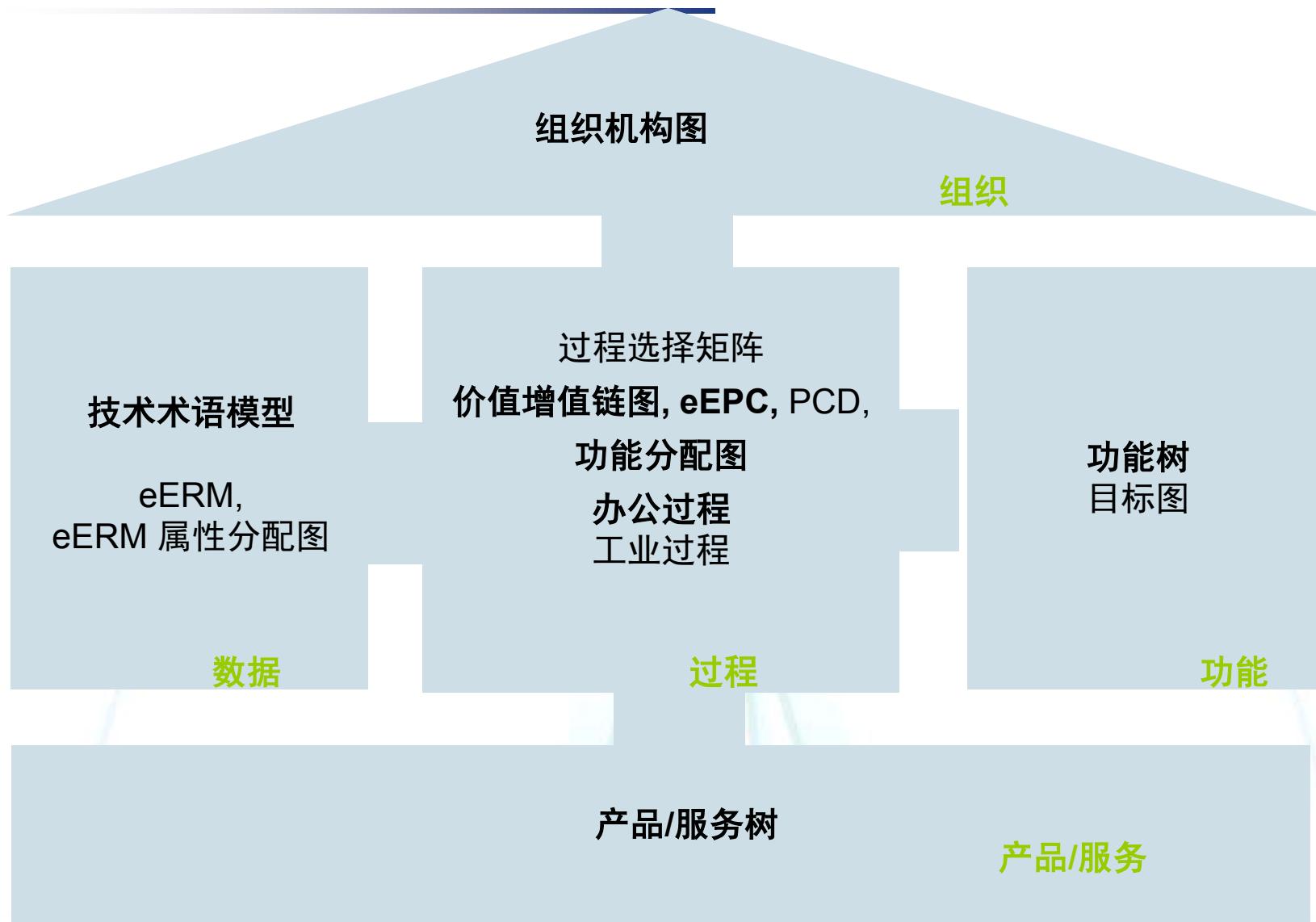
模型实例

企业销售过程



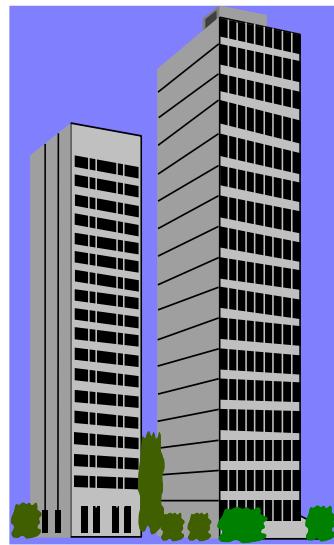


ARIS 模型类型

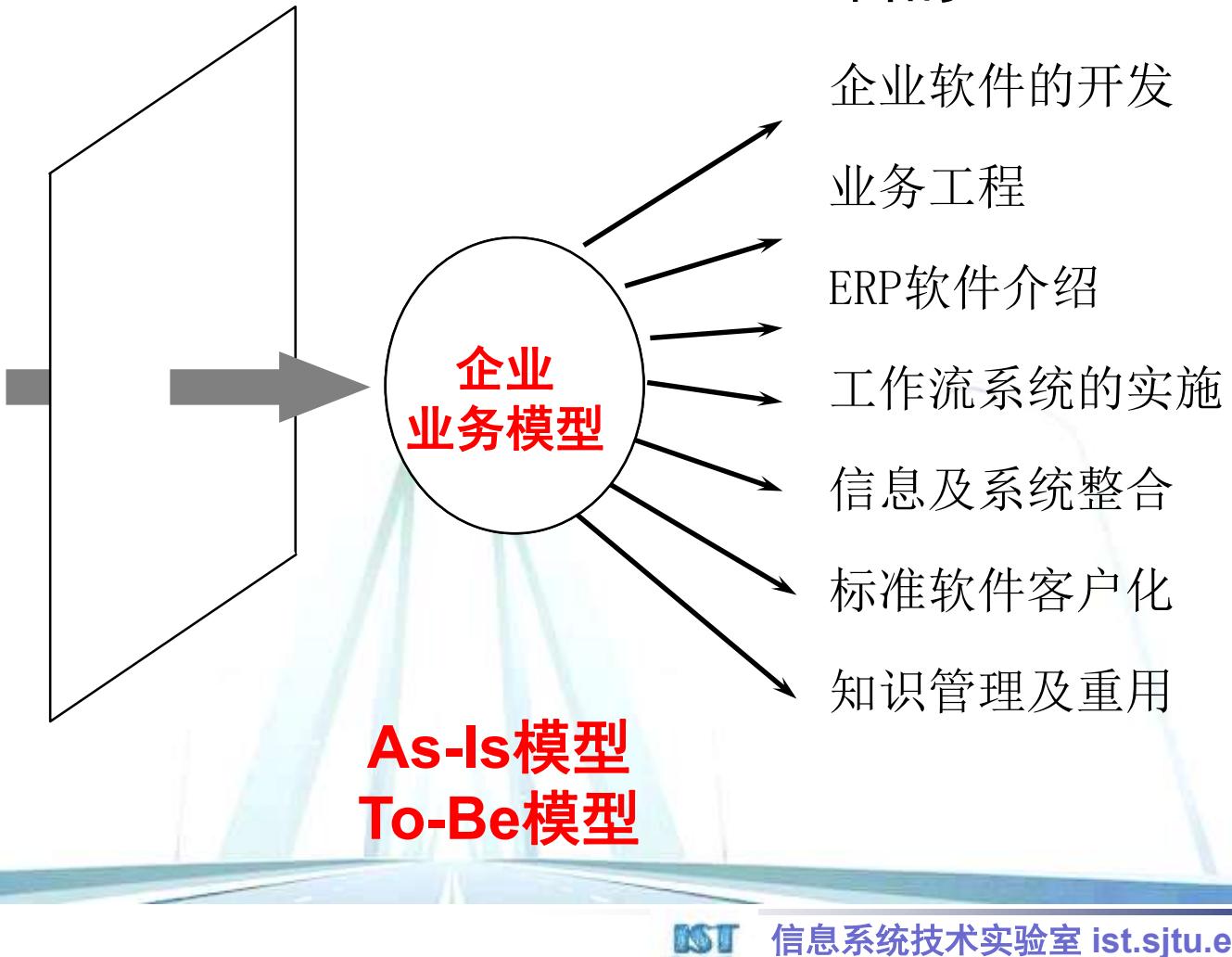


业务模型的用途

企业业务状况



- 今天(现状)
- 明天(目标)





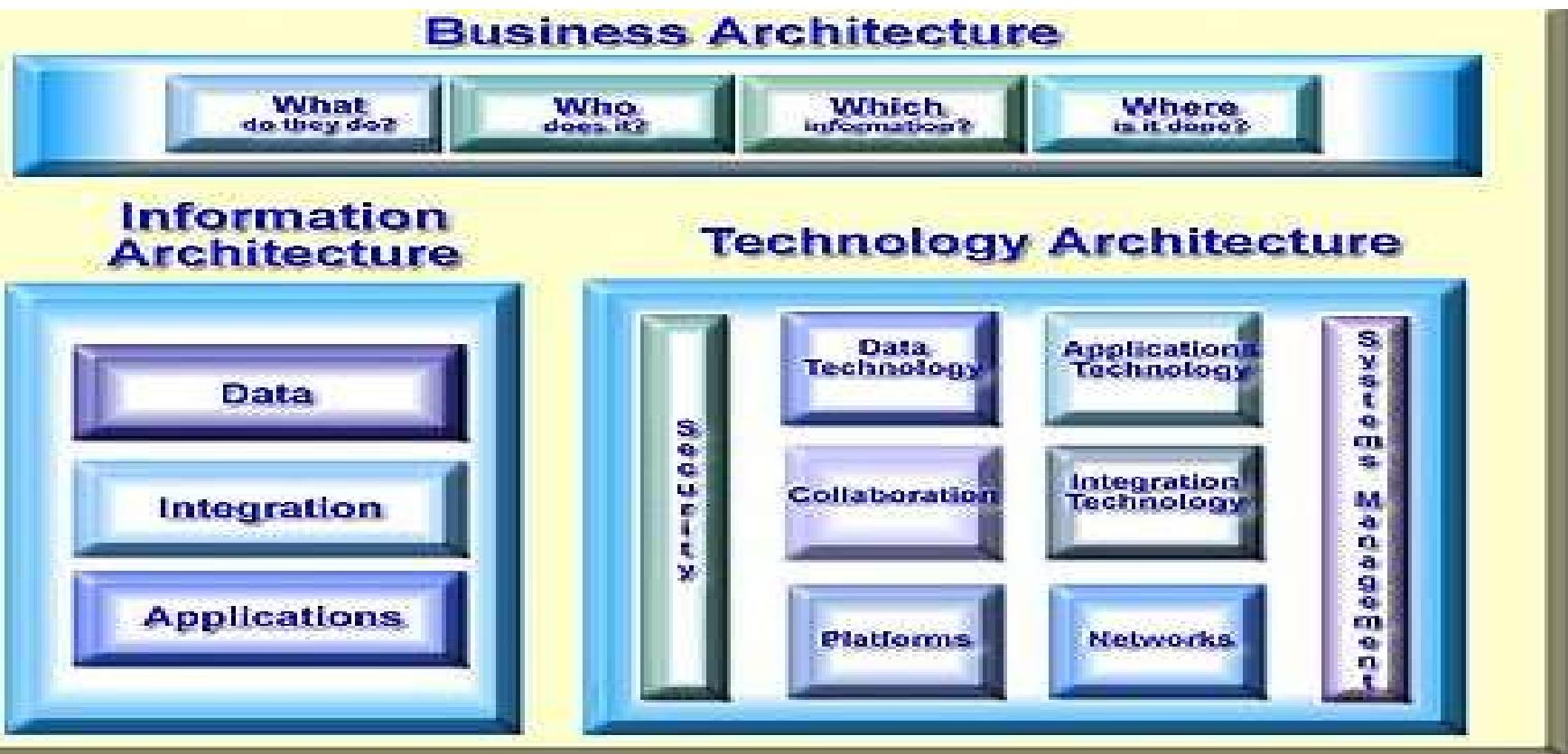
Agenda

1. 业务模型
2. 企业业务建模体系
3. 业务流程描述方法
4. 小结



2 企业业务建模体系

- 企业业务模型是一组具有特定目的的互补模型的一致集合，这些特定模型描述了企业某些用户的具体需求的不同侧面。
- 针对不同的需要，国内外开展了许多研究，提出了多种企业级建模框架。然而，到目前为止，现有的建模框架建立的模型与理想情况还有差异。



Enterprise architecture

Enterprise architecture provides a comprehensive approach for **designing, planning, implementing, and governing** an enterprise information architecture.

An architecture framework is a set of tools which can be used for developing a broad range of different architectures. It should:

- describe a method for defining an information system in terms of a set of **building blocks**
- show how the building blocks fit **together**
- contain a set of **tools**
- provide a common **vocabulary**
- include a list of recommended **standards**
- include a list of compliant **products** that can be used to implement the building blocks

目前的一些企业框架

- AGATE French Délégation Générale pour l'Armement Atelier de Gestion de l'ArchITecture des systèmes d'information et de communication.
- ArchiMate an open and independent modelling language for enterprise architecture
- ARCON - A Reference Architecture for Collaborative Networks - not focused on a single enterprise but rather on networks of enterprises [9][10]
- DoDAF United States Department of Defense Architectural Framework.
- CSC Catalyst  CSC Catalyst
- DYI framework Sogeti Framework.
- EIF European Interoperability Framework - Enterprise architecture at the level of EU Member States
- IDABC Interoperable Delivery (of European egovernment services to public) Administrations, Business and Citizens
- Integrated Architecture Framework (IAF) created by Capgemini.
- FEA United States Office of Management and Budget Federal Enterprise Architecture.
- MIKE2.0 (Method for an Integrated Knowledge Environment) which includes an enterprise architecture framework called SAFE (Strategic Architecture for the Federated Enterprise)
- MODAF United Kingdom Ministry of Defence Architectural Framework.
- Model-driven architecture (MDA) Object Management Group's Model Driven Architecture.
- OBASHI (The OBASHI Business & IT methodology and framework.
- The Operations Systems Computing Architecture (OSCAR), initially developed in 1986 by Bell Communications Research (the forerunner of the current Telcordia Technologies) to guide development of enterprise systems for the Regional Bell Operating Companies (RBOCs).
- PROMIS Framework  The PROMIS Enterprise Architecture Framework integrated into the EA tool EVA Netmodeler
- SABSA a comprehensive framework for Enterprise Security Architecture and Service Management.
- SAP Enterprise Architecture Framework is extension of TOGAF to better support Commercial off-the-shelf and Service-Oriented Architecture
- Zachman Framework IBM Framework from the 1980s.

2. 1 企业业务建模体系

方法类：

基于功能、基于过程、基于工作流、基于服务…

框架体系：

- 1. Zachman 框架：
- 2. CIM-OSA (Computer Integrated Manufacturing-Open System Architecture) 框架：
- 3. ARIS (Architecture of Integrated Information System) 框架
- 4. PERA (Purdue Enterprise Reference Architecture) 框架；
- 5. 动态企业建模 (DEM, Dynamical Enterprise Modeling) 方法：

工具：

- ✓ ARIS 的ARIS Toolset
- ✓ DEM的BAAN IV ORGWARE 工具

方法体系

(1) 基于功能活动的建模方法

- 以基于功能的建模方法为主导，SADT，IDEF为代表的功能分解法成为90年代企业建模的主要方法。

存在的主要问题：

- 由于功能分解法的基本组件仅有一个，因此该方法虽然有非常好的通用性，但缺乏丰富的语言描述能力。
- 功能分解法的严格的递阶关系很容易导致企业的组织结构，而且在不同的组织单元间造成交流障碍。形成新的自动化孤岛。

(2) 基于过程的建模方法

- ④ 过程定义成为一组活动的偏序集，过程与子过程间形成递阶关系。活动间有一定顺序、活动的执行有时间触发。
- ④ 可以将基于过程的控制流、物料流和信息流有效的集成。
- ④ 面向过程的建模方法有还可以进一步按照基本要素划分：
 - (1) 基于**数据**的业务流程建模
 - 以数据流图为主，将活动按照数据流向连接。
 - (2) 基于**活动**的业务流程建模。
 - 以活动图为主，将业务活动通过时序关系连接。
 - (3) 基于**事件**的业务流程建模。
 - 以EPC为主，将业务活动通过事件连接。
 - (4) 基于**状态**的业务流程建模。
 - 以PetriNet主，将业务活动通过状态事件连接。

(3) 多视图集成建模方法

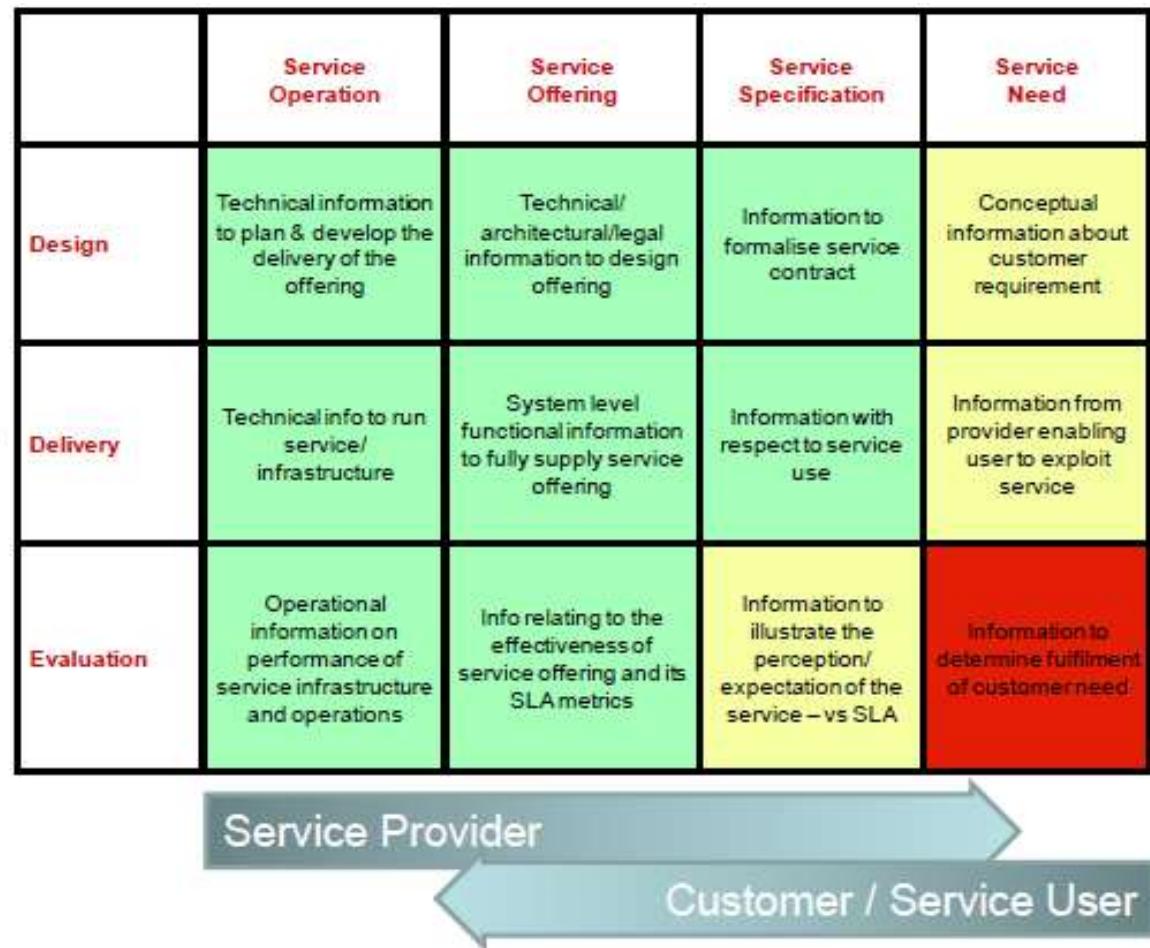
- 采用以过程模型（工作流模型）为核心，其他视图（功能视图、信息视图、组织视图、资源视图）模型为辅助来实现集成化建模。

- 过程视图建模方法：集成的核心
- 组织视图建模方法：
- 资源视图建模方法：
- 功能视图建模方法：

- ...

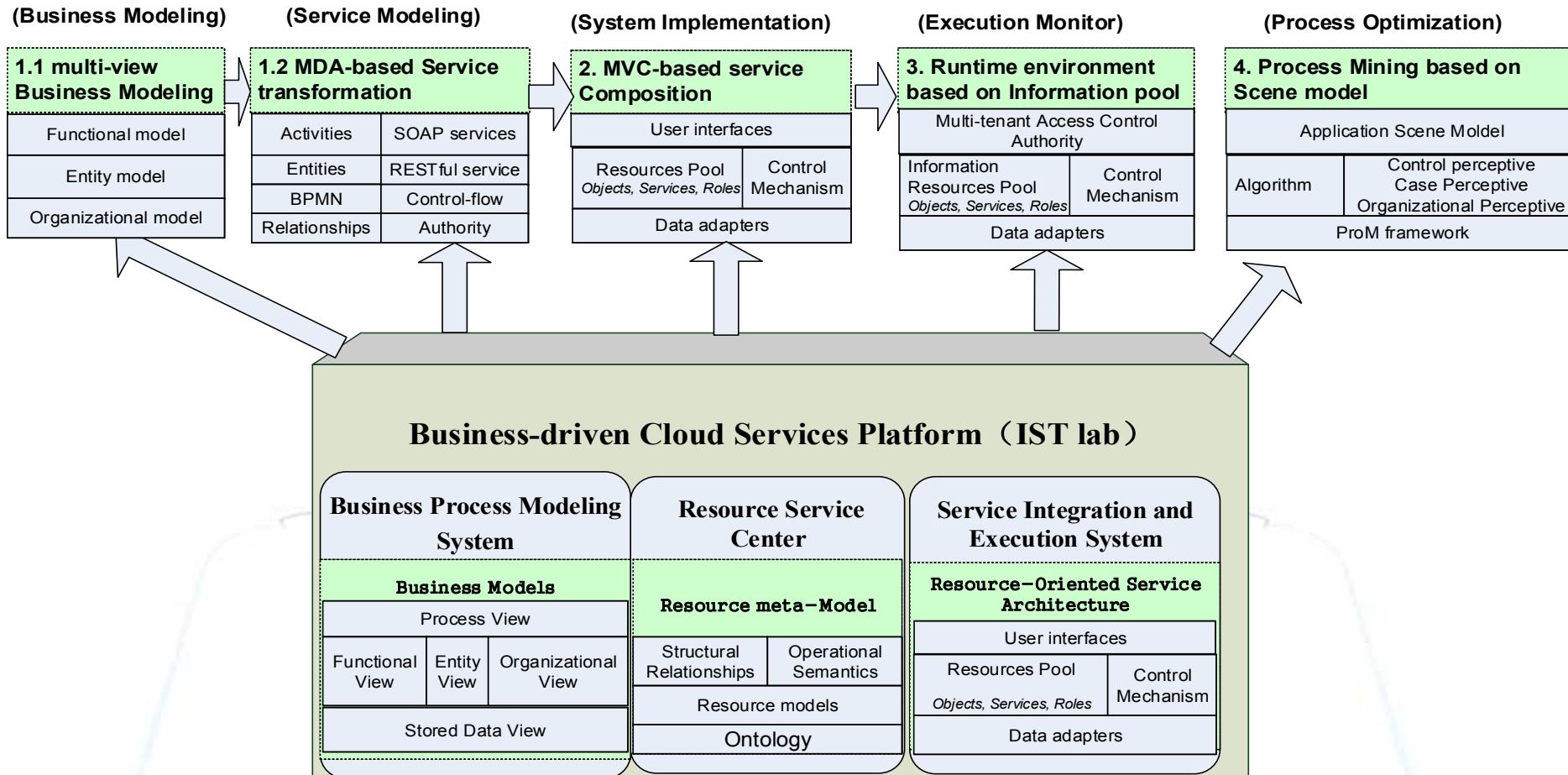
(4) 基于服务的建模及开发

SOA & RESTful



正逐渐发展完善成新兴方法体系！

业务模型驱动的云服务平台（IST lab）



（建模） 业务驱动需求（分析）；

（系统） 需求驱动技术（架构）；

（前沿） 技术推动业务（提升）；



2.2 建模方法论

模型建立的思路有两种：

- 自顶向下、逐步求精
如SAP ARIS 构建方式

- 自底向上、综合集成。
如基于服务组合的系统构建方式

业务模型的常见的表现形式有

形式化描述方法：

- 非常精确、严谨，易于系统以后的实现，
- 学习难度大，难以掌握和理解，
- 模型可读性差，难于推广。

图示化方法：

- 直观、自然，易于描述系统的层次结构、功能组成，
- 简单易学，
- 通常还有工具软件支持，因而成为信息系统的主要描述工具，
- 但这种方法的精确性和严谨性不够

企业业务建模的基本原则

- 企业业务模型具有分层次、多视角、复杂相关性等特点，建模更需要遵循一定原则！
- 根据**Ross**的提法，任何建模技术都具有以下基本原则：
 - ◆ 定义模型的**目的**：建模是为了什么。
 - ◆ 定义模型的**范围**：说明模型覆盖的领域和范围。
 - ◆ 定义模型的**视角**：从现实世界哪些方面的特征，有哪些特性被忽略。
 - ◆ 定义模型的**粒度**：模型的精度和模型的细致程度颗粒度。



企业建模的 8 大原则

- █ 分离的原则：
- █ 功能分解的原则：
- █ 分层的原则：
- █ 一致性原则：
- █ 模块化原则：
- █ 通用性原则：
- █ 功能与行为分离原则：
- █ 活动与资源解耦原则：



企业建模的8大原则（1）

- 分离的原则：由于企业模型较复杂，将整个企业考虑为一个整体进行研究显然是不合适的，因此，必须对每个功能逐个进行分析，如产品设计过程、主生产计划、制造系统。
- 功能分解的原则：根据经营目标的方法，将企业功能逐层分解，逐步细化方法。
- 分层的原则：在企业模型构筑的不同阶段，有不同的开发人员，在建模的过程中，必须有不同的模型，以不同的抽象程度，反映系统的不同侧面。
- 一致性原则：需要各组件在语义及语法上保持一致，这是目前企业建模中最重要也最难以满足的要求。



企业建模的 8 大原则（2）

- 模块化原则：有利于管理企业的变化，方便模型的维护。
- 通用性原则：需要提高模型的通用化程度，通过定义构件、通用模块化等方法，将模型中的共性问题统一进行表示。



企业建模的8大原则（3）

- 功能与行为分离原则：功能考虑做什么，行为关心如何做，区分功能和行为，有利于提高企业组织的柔性。

- 活动与资源解耦原则：活动描绘需要做的事情，资源描述了执行这个活动的人或设备，活动与资源解耦可以有效提供企业执行的柔性。

2.3 框架体系

框架体系即企业业务模型建立、管理、应用的框架，是企业业务要素的组合方式：

如：以流程，功能，信息等关系的**建模结构矩阵**

| | 流程 | 功能 | 业务对象/信息 | 组织/任务 |
|----|------|------|---------|-------|
| 流程 | 流程模型 | | 状态转换图 | |
| 功能 | 流程图 | 功能模型 | | 组织权限图 |
| 信息 | 顺序图 | 数据流图 | 信息模型 | |
| 组织 | | 用例图 | 协作图 | 组织模型 |

典型框架体系

- 1. Zachman 框架: Zachman 框架被认为是企业建模领域中的权威, 其他许多框架都派生于它。
- 2. CIM-OSA 方法:完整、开放、标准和形式化的面向CIM系统生命周期(建模、设计、实施、运行和维护等各个阶段)的CIM参考体系结构。
- 3. ARIS (Architecture of Integrated Information System) 方法:集成化的信息系统模型框架, 描述了企业的组织视图、数据视图、过程视图和资源视图。
- 4. PERA: 美国Purdue 大学于1990 年开始开发的, 为基于CIM 的企业提供一套企业建模的框架。
- 5. 动态企业建模方法(DEM):针对项目型、流程型和离散型企业特点, 利用动态企业建模DEM策略, 建立ERP应用系统。



Planner

Owner

Designer

Builder

Sub-contract

Product

What

How

Where

Who

When

Why

| | 重要业务对象列表 | 业务过程列表 | 业务执行地点列表 | 重要组织单元列表 | 重要事件列表 | 业务目标列表 |
|------------------|--------------------------|------------------------|--|-------------------------------------|-------------------------|------------------------------------|
| 企业规划 规划者 | 重要业务对象列表 实体=业务对象类 | 业务过程列表 功能=业务过程类 | 业务执行地点列表 节点=主要业务地点 | 重要组织单元列表 组织=主要组织单元 | 重要事件列表 时间=主要业务事件 | 业务目标列表 目标=主要业务目标 手段=成功要素 |
| 企业模型 业主 | 如:语义模型 | 如:业务过程模型 | 如:业务分布模型 | 如:工作流模型 | 如:主进度表 | 如:业务规划 |
| 系统模型 设计者 | 如:逻辑数据模型 | 如:应用系统体系结构 | 如:分布式系统体系结构 节点=处理器/存储器/等 连接=线路属性 | 如:员工接口体系结构 组织=任务 工作=交付的成果 | 如:处理结构 | 如:业务规则模型 目标=结构声明 手段=行动声明 |
| 技术模型 建立者 | 如:物理数据模型 | 如:系统设计 | 如:技术体系结构 | 如:描述体系结构 | 如:控制结构 | 如:规则设计 |
| 各部分的详细描述 子承建者 | 如:数据定义 | 如:程序 | 如:网络体系结构 | 如:安全体系结构 | 如:时限定义 | 如:规则说明书 |
| 具体实现 | 如:数据 | 如:功能 | 如:网络 | 如:组织 | 如:进度表 | 如:策略 |



框架的规则



Rule 1:

Columns have no order

Rule 2:

Each column has a simple, basic model

Rule 3:

Basic model of each column is unique

Rule 4:

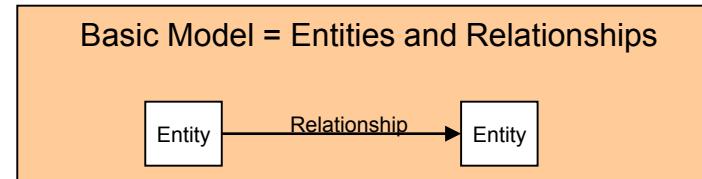
Each row represents a distinct view

Rule 5:

Each cell is unique

Rule 6:

Combining the cells in one row forms a complete description from that view



| | What | How | Where | Who | When | Why | |
|-------------|------|-----|-------|-----|------|-----|-------------|
| Contextual | | | | | | | Contextual |
| Conceptual | □○□ | →○→ | ○○○ | △△△ | ←○← | ○○○ | Conceptual |
| Logical | □□□ | →○→ | ○○○ | △△△ | ←○← | ○○○ | Logical |
| Physical | □□□ | □□□ | □□□ | □□□ | ←○← | ○○○ | Physical |
| As Built | □□□ | □□□ | □□□ | □□□ | □□□ | □□□ | As Built |
| Functioning | ❖❖❖ | ❖❖❖ | ❖❖❖ | ❖❖❖ | ❖❖❖ | ❖❖❖ | Functioning |
| | What | How | Where | Who | When | Why | |



企业信息架构中的 Zachman 框架

■ Row 1 – Scope

External Requirements and Drivers

Business Function Modeling

■ Row 2 – Enterprise Model

Business Process Models

■ Row 3 – System Model

Logical Models

Requirements Definition

■ Row 4 – Technology Model

Physical Models

Solution Definition and Development

■ Row 5 – As Built

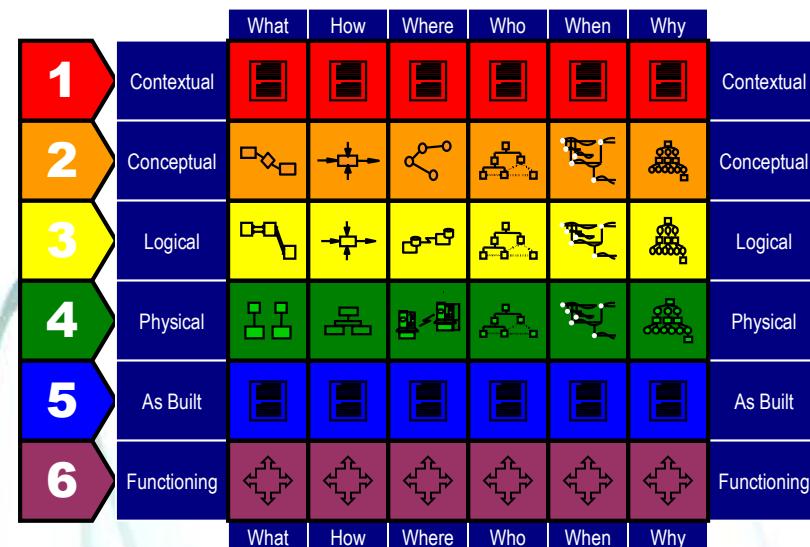
As Built

Deployment

■ Row 6 – Functioning Enterprise

Functioning Enterprise

Evaluation





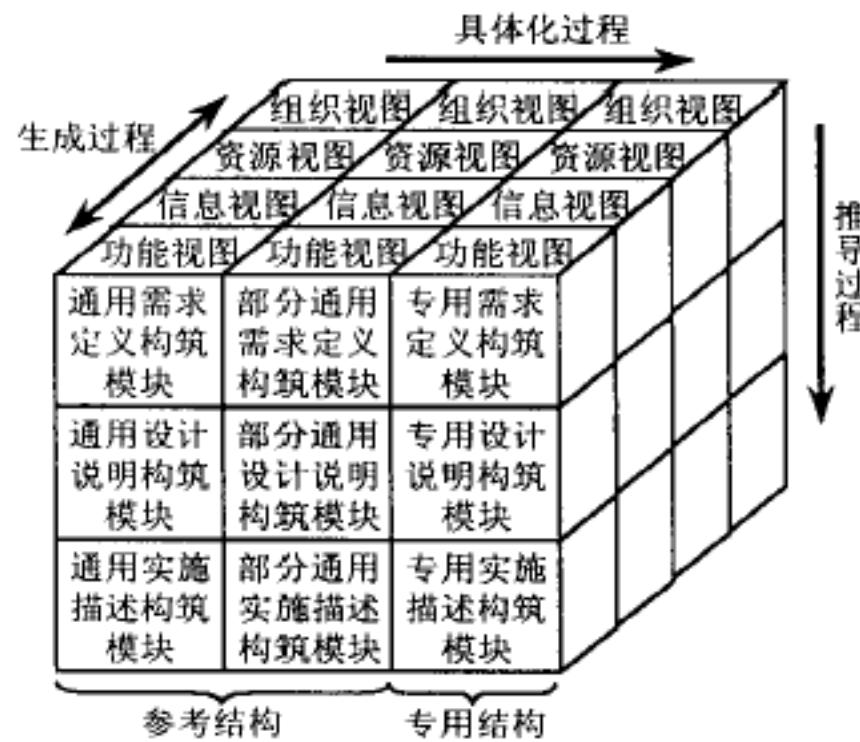
| | Why | How | What | Who | Where | When |
|------------|---------------------|--------------------------------|---------------------------|---|-----------------------------|---------------------|
| Contextual | Goal List | Process List | Material List | Organisational Unit & Role List | Geographical Locations List | Event List |
| Conceptual | Goal Relationship | Process Model | Entity Relationship Model | Organisational Unit & Role Relationship Model | Locations Model | Event Model |
| Logical | Rules Diagram | Process Diagram | Data Model Diagram | Role Relationship Diagram | Locations Diagram | Event Diagram |
| Physical | Rules Specification | Process Function Specification | Data Entity Specification | Role Specification | Location Specification | Event Specification |
| Detailed | Rules Details | Process Details | Data Details | Role Details | Location Details | Event Details |

CIM – OSA 框架

- CIM-OSA (Computer Integrated Manufacturing-Open System Architecture)是欧共体ESPRIT计划开发的，CIM-OSA是一种面向企业CIMS生命周期的体系结构。

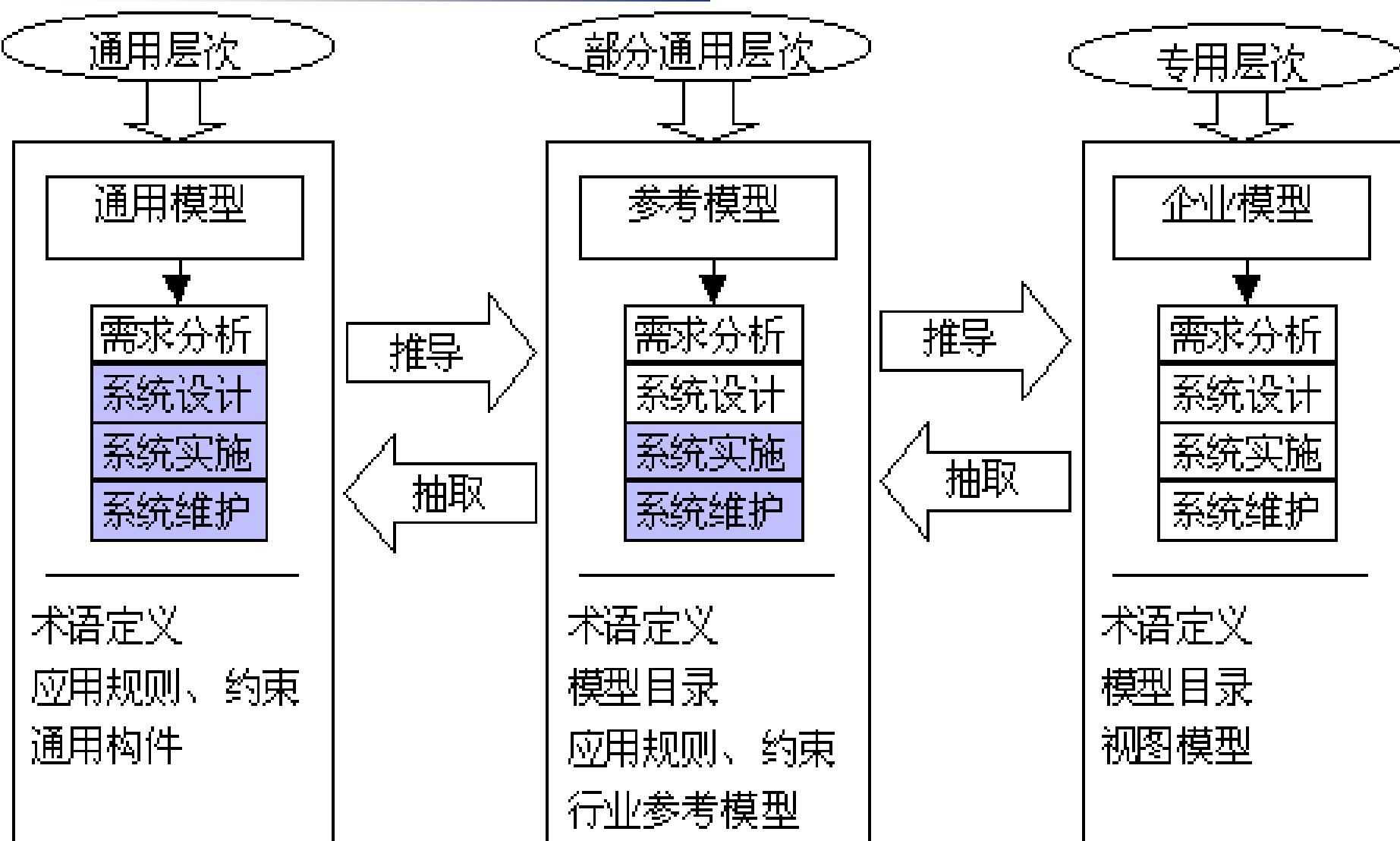
其主要特征：

- 1 建立了一个**体系结构框架**，用来把所需模型分类；
- 2 通过**CIM-OSA 集成基础结构**来协调企业工程环境和运行环境；





(1) 模型从通用到专用的发展过程





(2) 建模过程

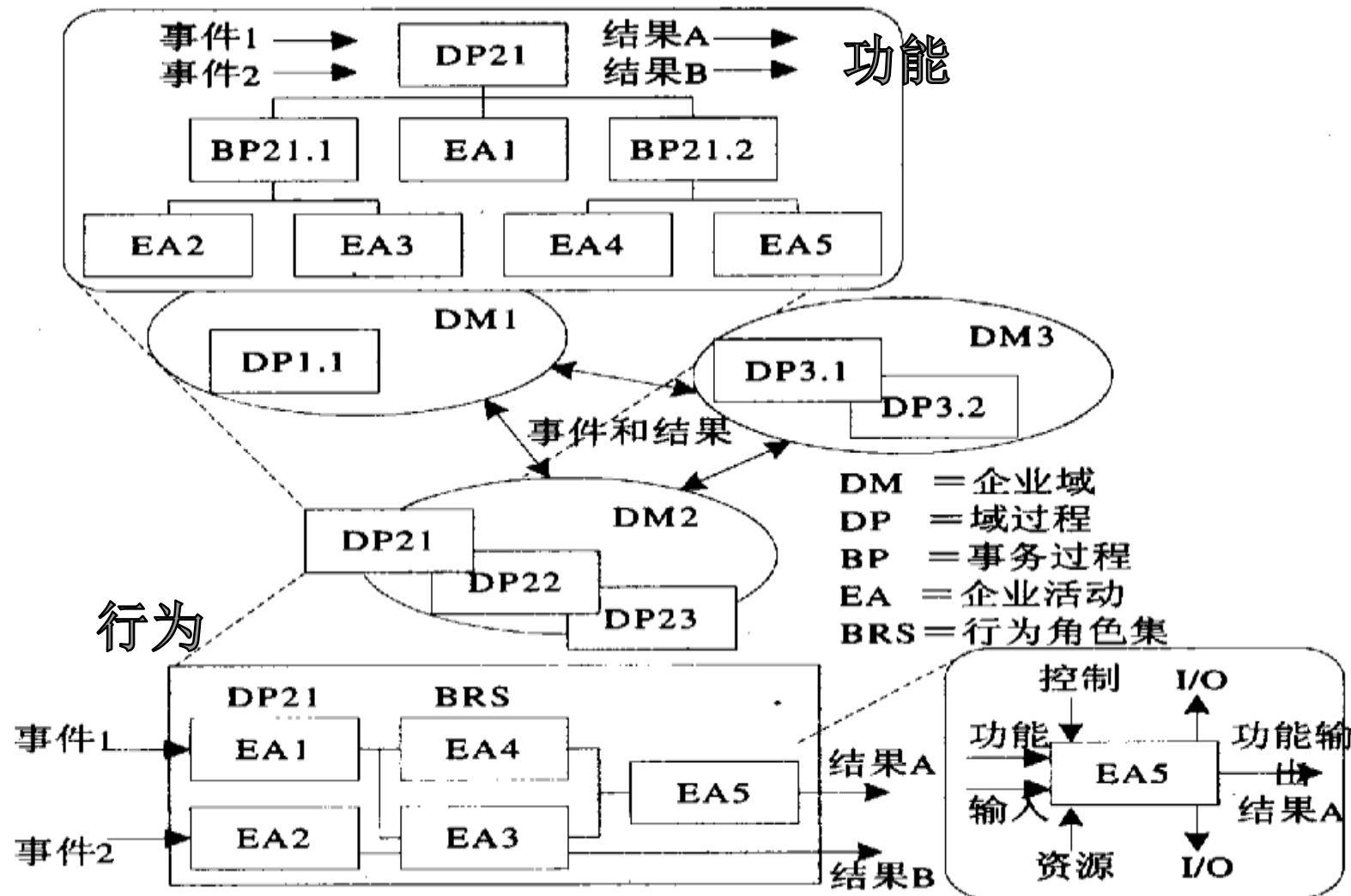
CIM-OSA建模方法的核心是对企业行为和功能的递阶分解，
如需求定义层的5层概念：

- DW企业域
- DP域过程
- BP事务过程
- EA 企业活动
- BRS行为角色集

如进入系统设计层，EA要进一步分解为功能操作（Functional Operation），还要确定每个FO操作的资源，即功能实体（Functional Entity）。



建模过程





(3) CIM-OSA集成基础结构

前端服务类

人前端

应用前端

机器前端

过程服务类

过程控制

活动控制

资源管理

信息服务类

系统数据

数据管理

通信服务类

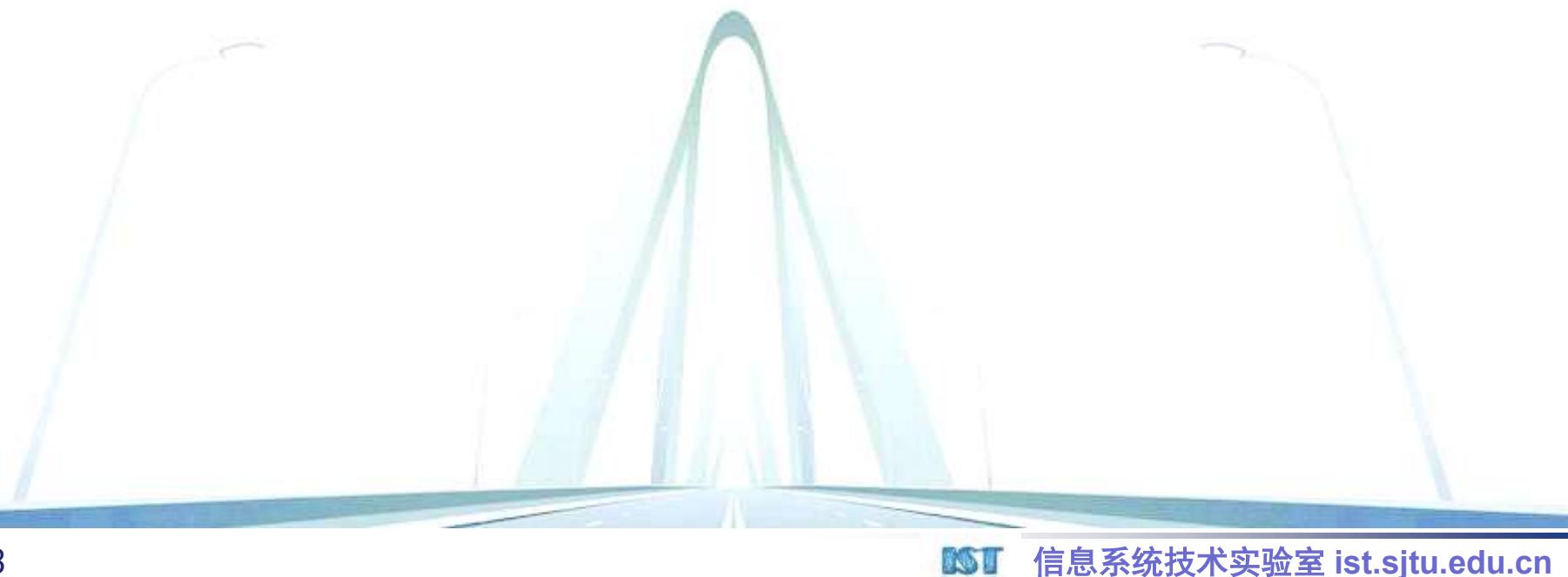
数据交换

通信管理



CIM-OSA评价

- ④ 优点：在于它具有全面性、完整性、开放性、标准化和形式化，并成为一种预标准，影响其它的建模框架。
- ⑤ 缺点：过于强调形式化，使得一般用户较难理解和掌握，限制了其应用；过于强调形式化，使得目前尚没有一种完全遵循CIM-OSA的商用软件。



ARIS 框架

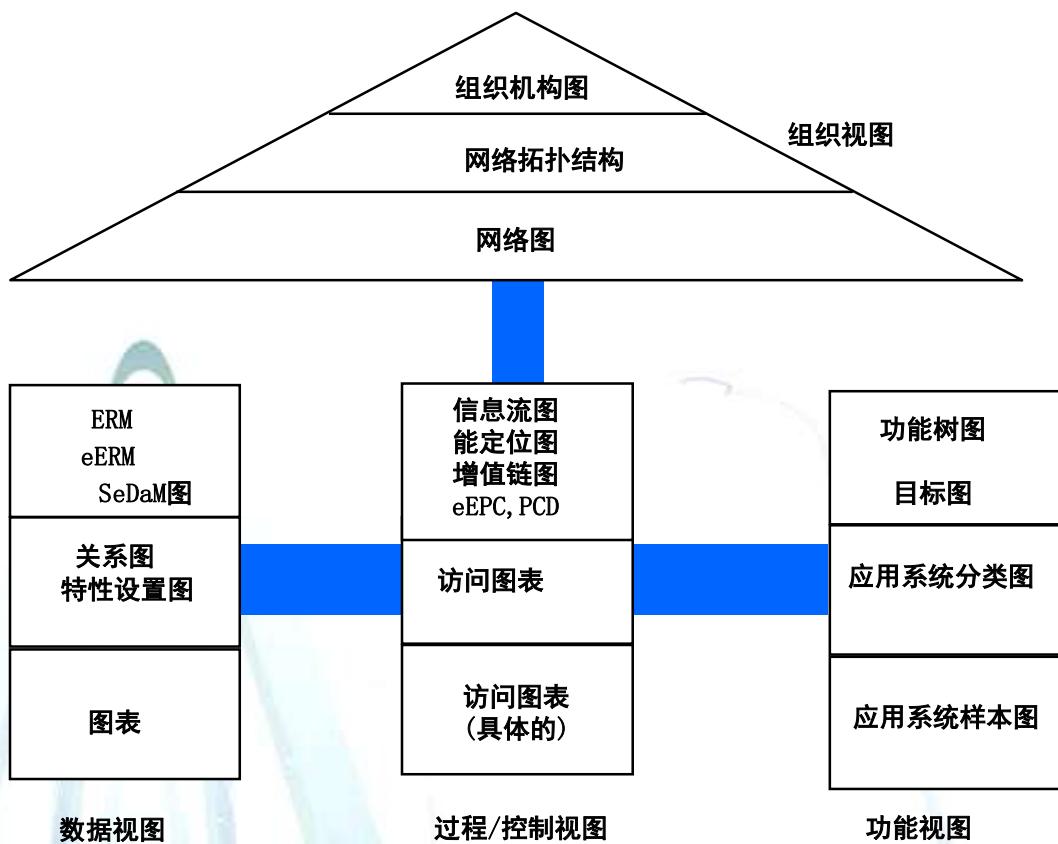
4个视图（**功能视图、数据视图、组织视图、控制视图**）

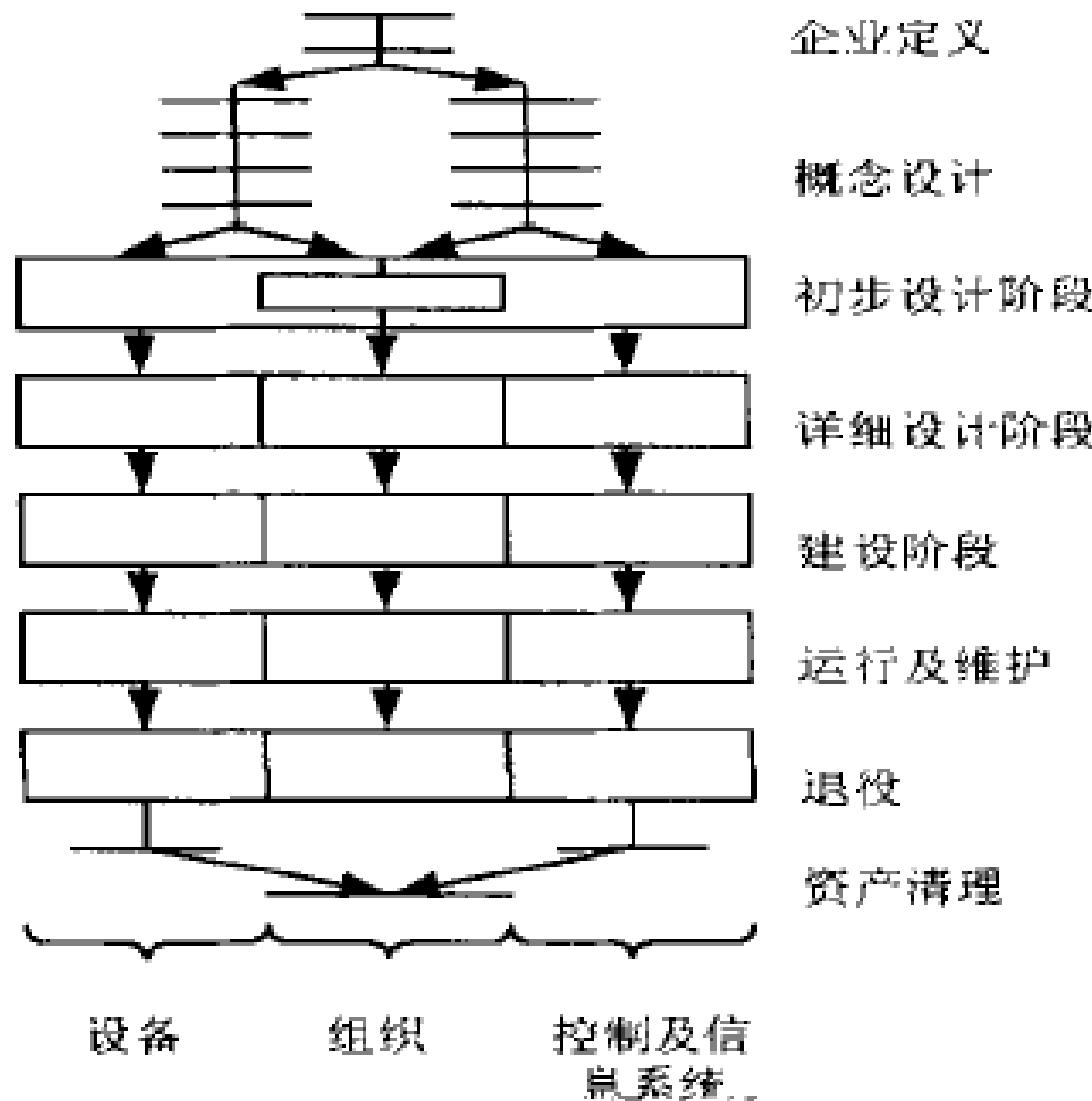
5个阶段（现行系统分析、**需求定义、设计说明、实施描述、运行维护**）

独特的控制视图打破了相对孤立地从各方面描述企业的各个视图之间的界限，使它们连结成了一个整体；

ARIS支持大量的建模方法，适用于不同阶段的建模；

同时ARIS是基于过程的三维模型结构，它有一套标准的可以互相集成的软件工具集来支持ARIS建模及仿真







PERA框架评价

- PERA体系结构是基于任务的建模方法，把生命周期图的结构按覆盖企业全部生命历程的任务阶段进行分层；
- PERA最重要的特点是考虑了对人的行为活动的建模；
- 与其它参考模型相比， PERA覆盖了CIM系统实施的最完整的生命周期。
- 作为一种非形式化描述方法， PERA最容易被没有计算机知识的用户理解。
- 缺点是：由于描述的非形式化， PERA的可执行性非常差； PERA缺乏对体系结构进行计算机建模所需的数学建模技术；没有支持建模的支持工具，不能进行仿真优化和冲突检验等。



框架的比较

| | | Zachman 框架 | CMOSA | PERA | ARIS |
|----|---------|------------|-------|--------|------|
| 结构 | 维数 | 二维 | 三维 | 二维 | 三维 |
| | 覆盖的视图 | 完全 | 基本完全 | 不太完全 | 基本完全 |
| | 视图之间的联系 | 松散 | 松散 | 松散 | 紧密 |
| | 覆盖的阶段 | 建立期 | 建立期 | 整个生命周期 | 建立期 |
| | 开放性 | 强 | 强 | 强 | 强 |
| 应用 | 参考模型 | 多 | 较多 | 少 | 多 |
| | 可操作性 | 强 | 较弱 | 较弱 | 极强 |

理想企业建模框架应该具备的特性：

- ① 具有“具体化过程”，即模型抽象层次这一维度；
- ② 业务要素的覆盖性，应包括流程、信息、功能、组织等；
- ③ 框架应划分视图，具有层次结构，满足不同层次的建模需要，覆盖企业中的所有重要业务对象；
- ④ 视图之间具有较强的集成性；
- ⑤ 业务框架要贯穿企业整个生命周期；
- ⑥ 具有开放性,能够持续治理;
- ⑦ 具有建模工具的支持和建模流程模型的指导,具有可操作性

2. 4 DEM业务建模体系

DEM的提出原因——ERP实施中的问题；

- ERP软件太复杂，不灵活，实施费用昂贵且实施周期太长。实施成功率低。
- ERP软件维护困难；
- ERP的业务模型僵化、无法灵活满足新的业务需求。
- 与工作流结合，与OA结合的需要；

针对ERP实施中的问题，荷兰Baan公司提出了下一代ERP，即动态企业建模（Dynamic Enterprise Modeling, DEM）的思想，并有BAAN IV ORGWARE工具将DEM付诸实现

DEM在如下几个关键方面区别于通常的ERP：

- DEM能进行高质量的设计（直观方式建模）；
- DEM支持软件结构的重组（预定义的模型框架）；
- DEM支持连续的业务过程改进（模型可自动转化成应用）；
- DEM允许客户选择自己的界面和工作平台，并能很方便地与其他工作平台集成。（支持CORBA等方式集成）

DEM中体现了两个思想：

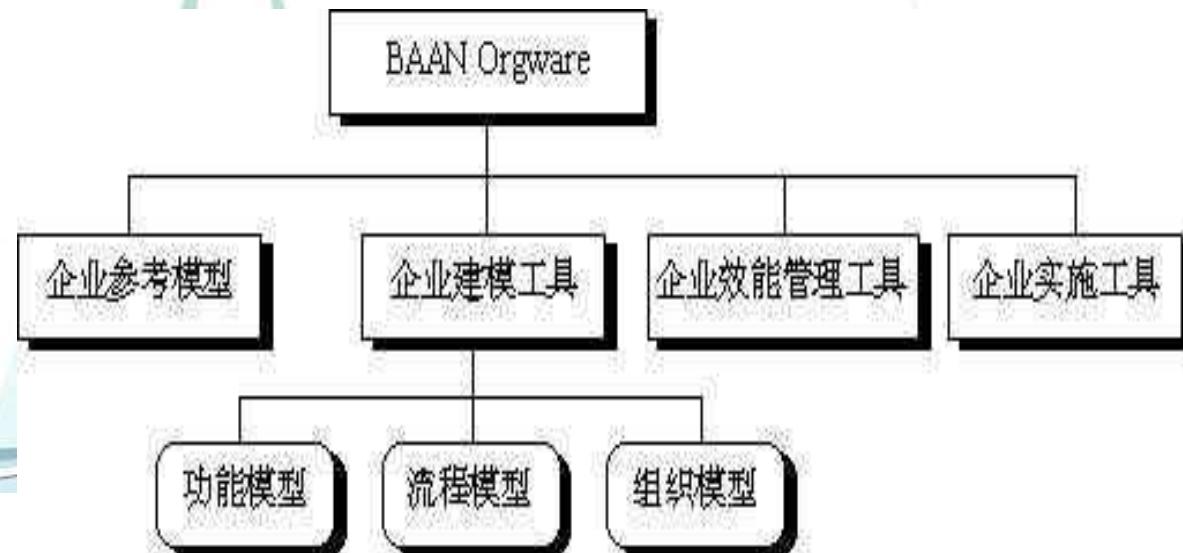
- ① 一是在生成企业特定的模型时，充分利用最好的实例知识和实践经验，表现在基于企业参考模型来建立特定的企业模型；
- ② 二是在适应市场而不断进行动态重组的企业中，企业的信息模型能快速适应应用环境、业务流程的变化。这表现在DEM的基于企业模型的的快速信息系统实施方式。

- ③ 从软件工程的角度来讲，DEM的本质特征是通过是用动态的管理模型来建立一个具有适应性的企业信息系统。



BAAN IV Orgware组成

- ① **企业建模工具：**它使公司可以用图形描述业务模型的组织结构、功能特点以及业务过程的先后顺序。
- ② **企业参考模型：**这些模型为自动化、电子、项目工程等领域的企业，提供了建立本企业模型的参考。
- ③ **企业效能管理工具：**这是ORGWARE性能管理工具，包括一套应用BAANIVORGWARE所有模块的预定义性能指标。即评价模块。
- ④ **企业实施工具：**在DEM概念框架内，BAAN通过三个预定义模型的连续赋值来实现，每个模型表示结构的不同方面，而它们组合起来就表示了整个企业。





Agenda

1. 业务模型
2. 企业业务建模体系
3. 业务流程描述方法
4. 小结

3 业务流程描述方法

- ① 业务流程描述及建模是系统开发和实施的关键模型；
- ② 业务流程建模是确定和描述系统的行为特性，然而业务流程往往与很多因素相关，也是应用中的难点所在；
 - 没有单一的图表可以有效地完全描述业务流程；
 - 大多数业务模型应用很多不同的整合的图表来描述业务流程。

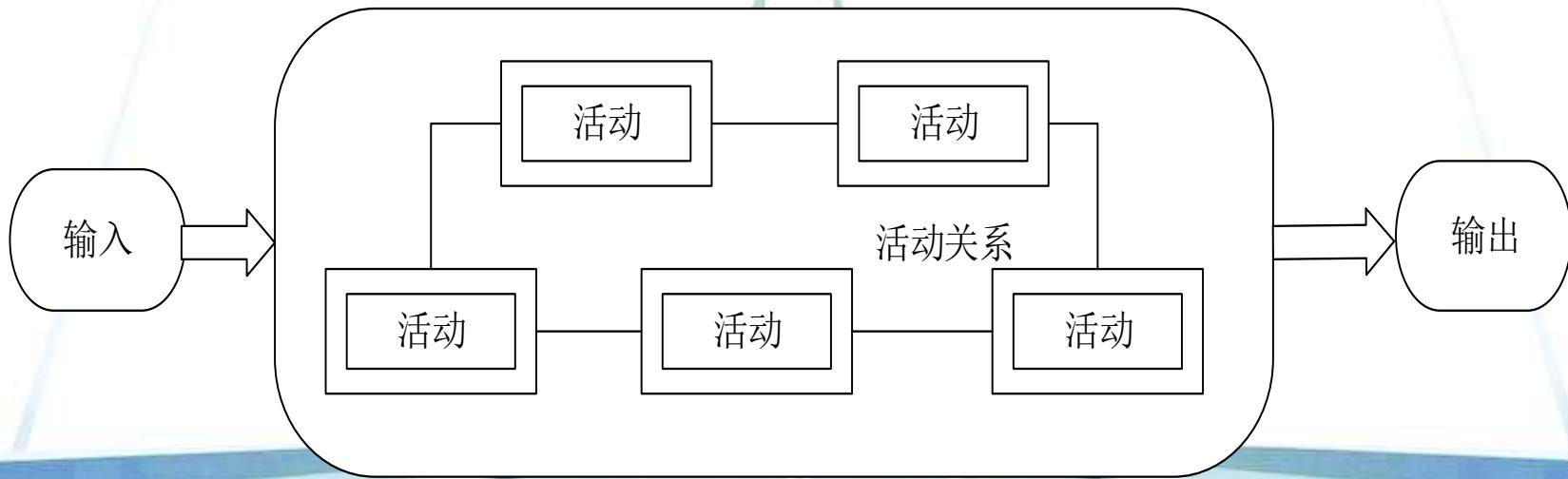
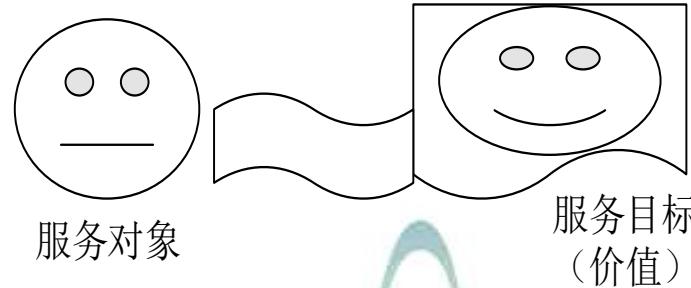
业务流程定义—— Davenport (1993)

业务流程就是一组能够为客户创造价值的相关联的活动进程。

a **business process** is simply a structured, measured set of activities designed to produce a specified output for a particular customer or market.

业务流程的六要素

- 输入，输出
- 活动，活动关系
- 服务对象，服务目标





从价值链看企业的核心业务活动

辅助活动



质量
活动

3.1 流程建模基础

- 建立流程模型通过多种不同的方式展开，主要可归纳为几大类方法：
 - 自顶向下 vs. 自底向上
 - 集中式 vs. 分布式
 - 自由式的 vs. 结构化的



■ 自顶向下的发现方法是从大型的流程开始，将它们划分成规模较小的流程。

- 比方说，我们可以从一个“Order to Cash”流程开始，将它分解成多个小的流程，比方说管理客户合同，输入订单，处理订单，给客户发送账单，收款。以上的每一个流程依旧是较大的流程，所以我们可以对它们进行进一步的分解。

■ 这样，我们可以获得流程的层次结构

- 最终，我们将一个流程分解到活动的层次，去执行简单的不可再分的活动。

■ 优缺点：

- 这个方法的主要好处是它确保了范围的广度。
- 缺点就是我们可能会漏掉一些细节。自顶向下的方法无法考虑到那些不知道其存在性的活动和流程。可能会因此停止在不准确的地方，或者漏掉一些步骤。

自底向上的方法开始于现有流程，将所做的工作的局部信息收集起来，组装到一个流程的上下文中。

优缺点：

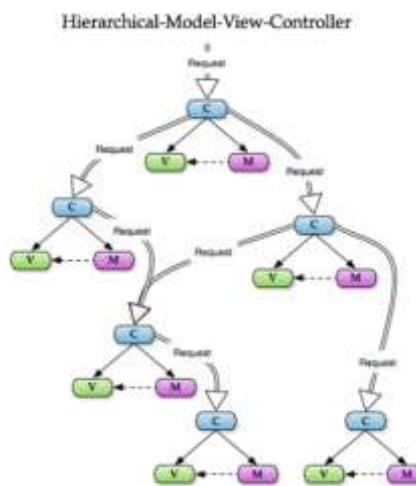
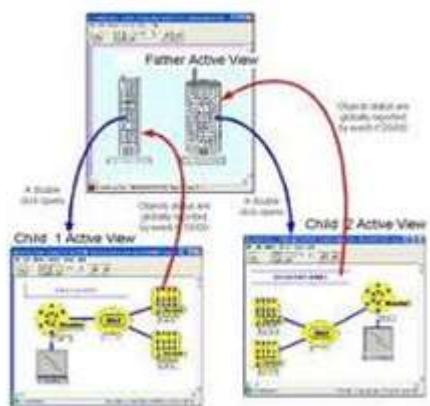
- 自底向上方法的一个主要优点是包含丰富的细节。
- 方法缺点在于如何判定了解了足够多的流程，因而**完整性**是一个问题。
- 另外，如何从所有的局部信息中创建一个流程的上下文，关联性可能丢失，这实际上就是一个把所有细节放进一个可被理解的流程集合的任务。

流程表现的视图

- 如果能从多个不同的角度出发，视图可以提供很强大的表现力。
- 以下是几个视图的例子：
 - 层级视图Hierarchical View
 - 端对端视图End to End View
 - 泳道视图Swim-lane Views
 - 路径分析Path Analysis
 - Highlighting

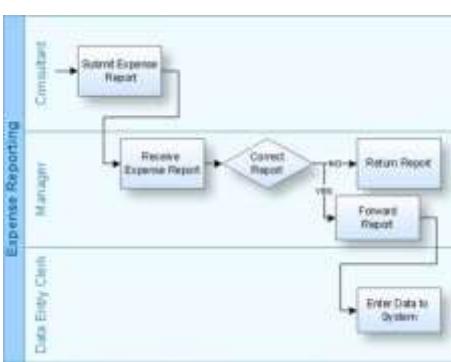


- 可以从不同等级的分辨率来定义流程。
- 层级视图允许以一连串子流程的形式来展现一个流程。
- 每个块表示的是一个子流程或一个相关活动的集合，而不是一个活动。.



- 端对端关注客户需求实现的全过程，实现真正的客户导向。
- 需要查看完整意义上的所有基本活动，而不是子流程。
- 需要显示流程端对端的流。
 - 比如：一个流程由7个子流程组成，每个子流程包括6个活动。
 - 因此，端对端视图将显示42个活动。

泳道视图



它将流程图结构化的放置在一系列水平或竖直的泳道中，每个泳道代表活动由某个特定的角色执行。.

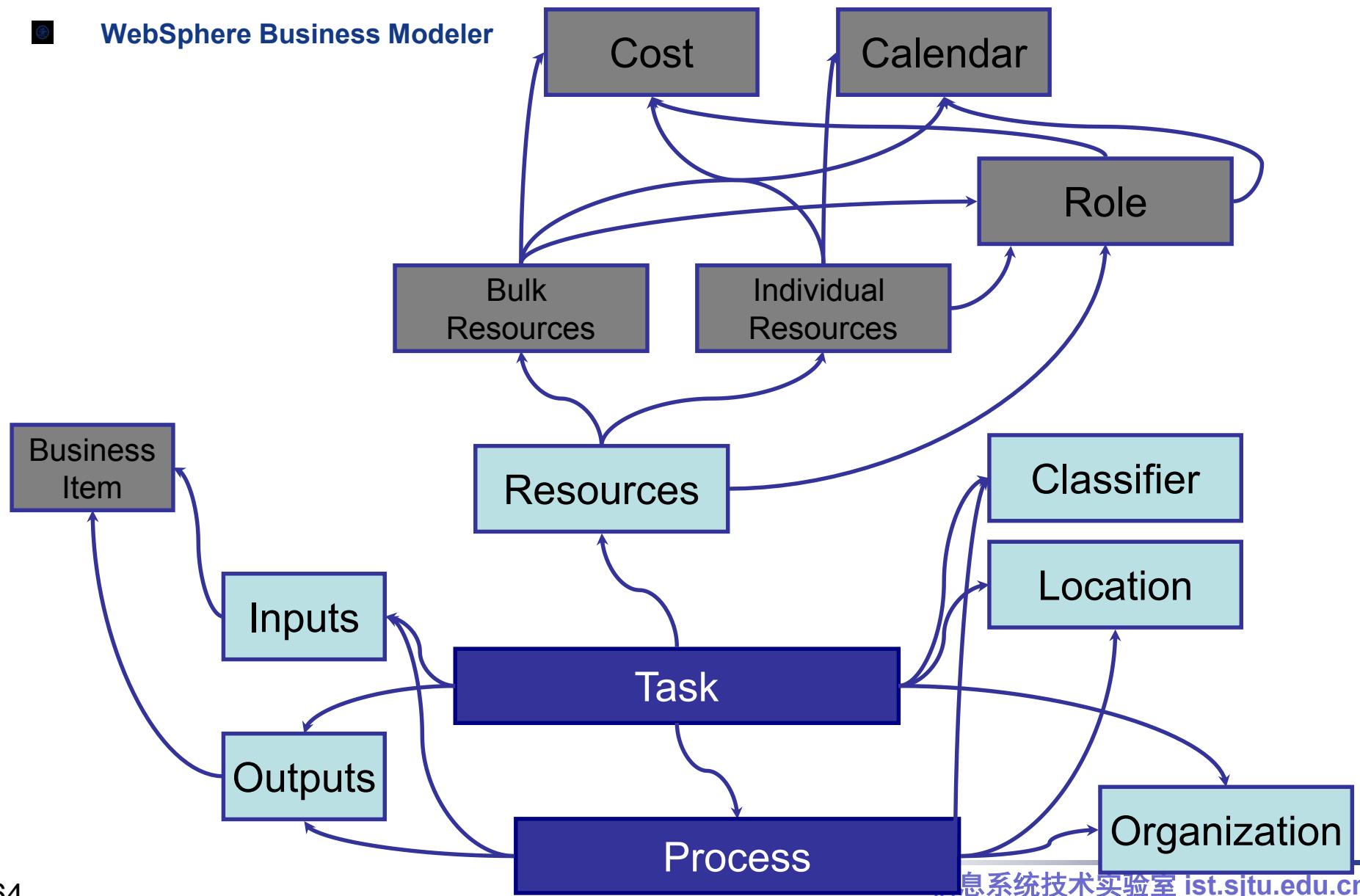
比方说，第一个泳道表示活动由客户执行，第二个泳道表示活动由客户服务代表执行，等等。

改变泳道的模式通常会起到不错的效果。我们可能会想用泳道来表示组织而不是角色。

- 或者我们可以用泳道来表示所使用的技术
- 表示活动执行的地点
- 表示增值或非增值的活动



WebSphere Business Modeler





Process comes with time and money,

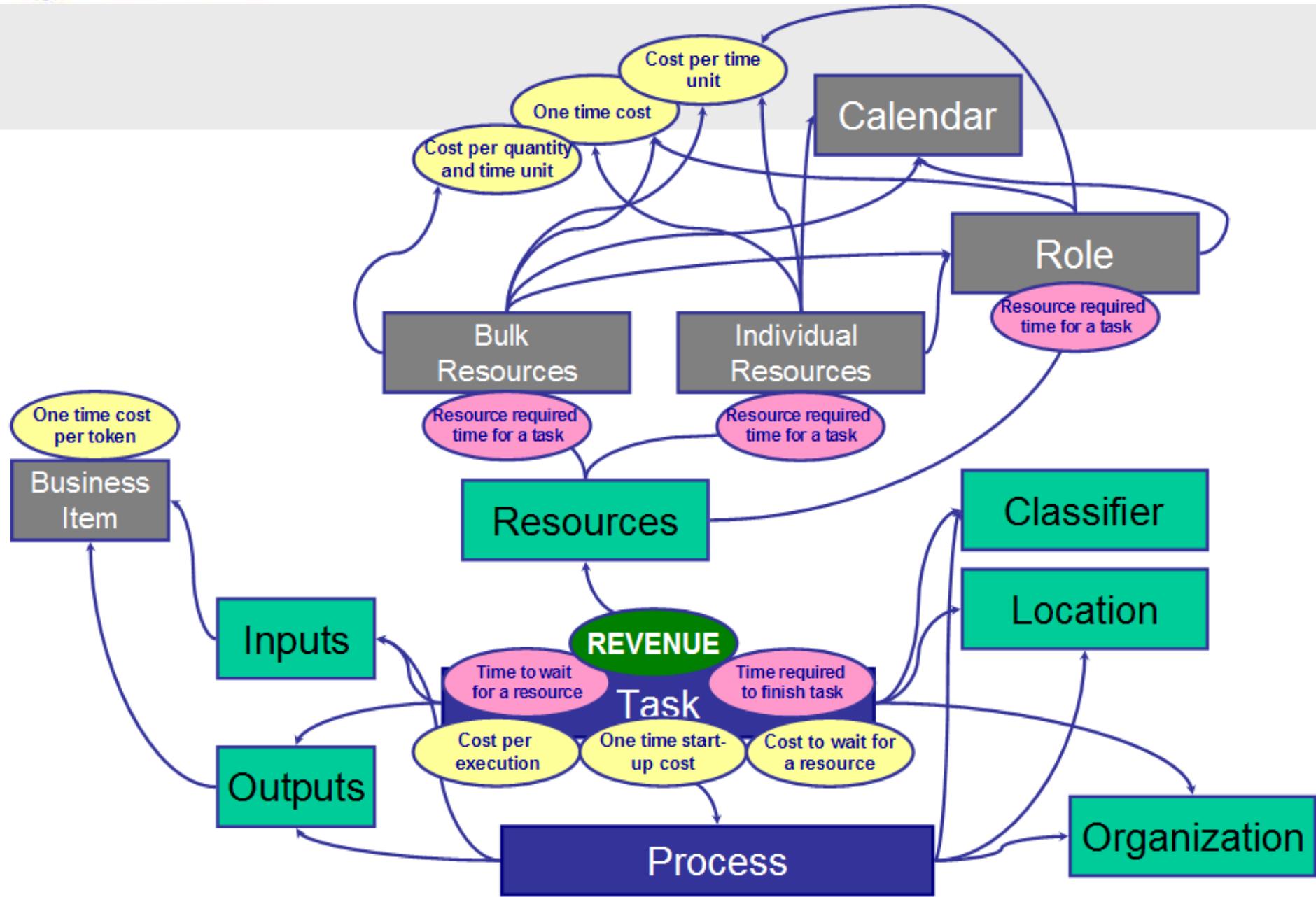
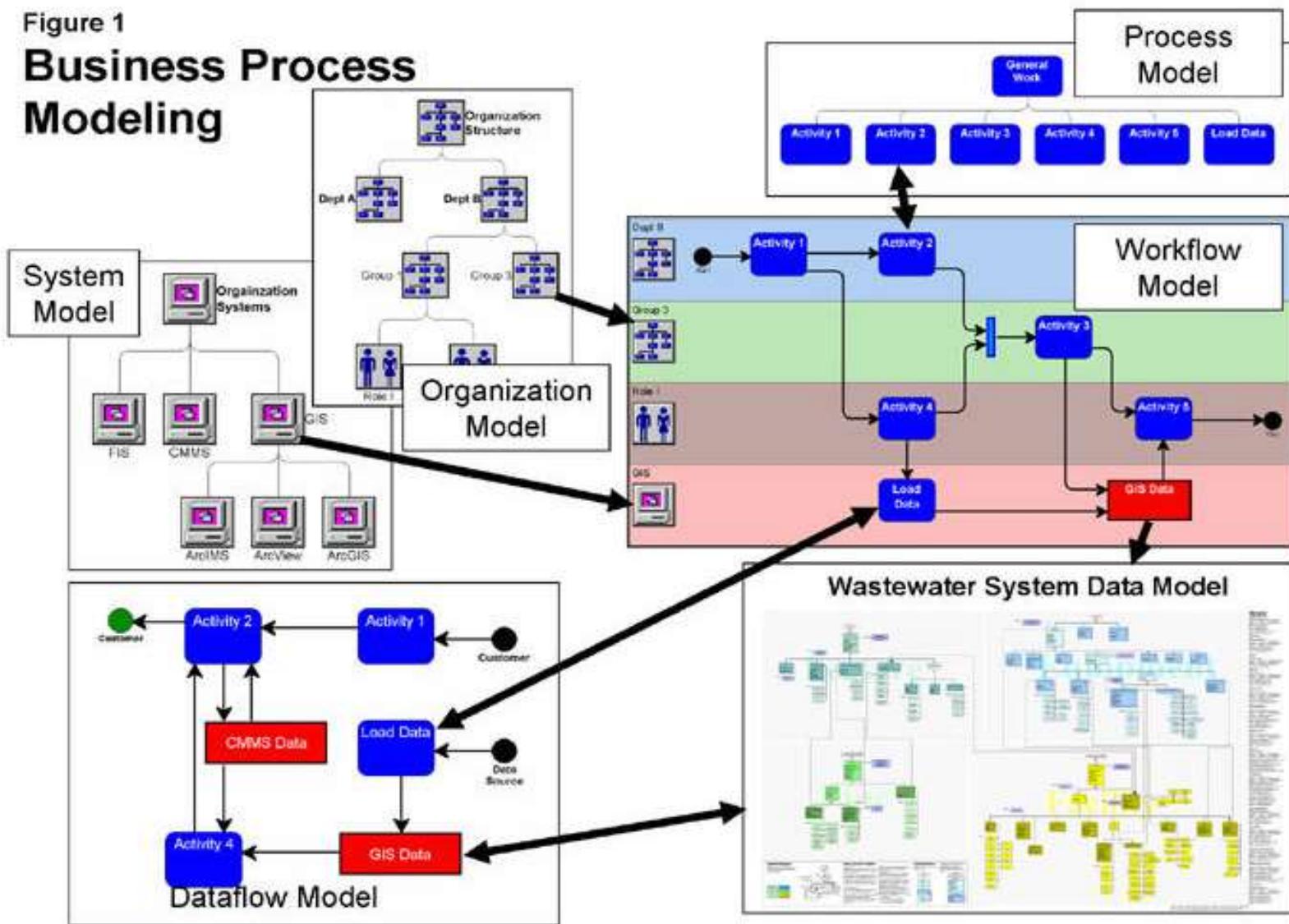




Figure 1
Business Process Modeling



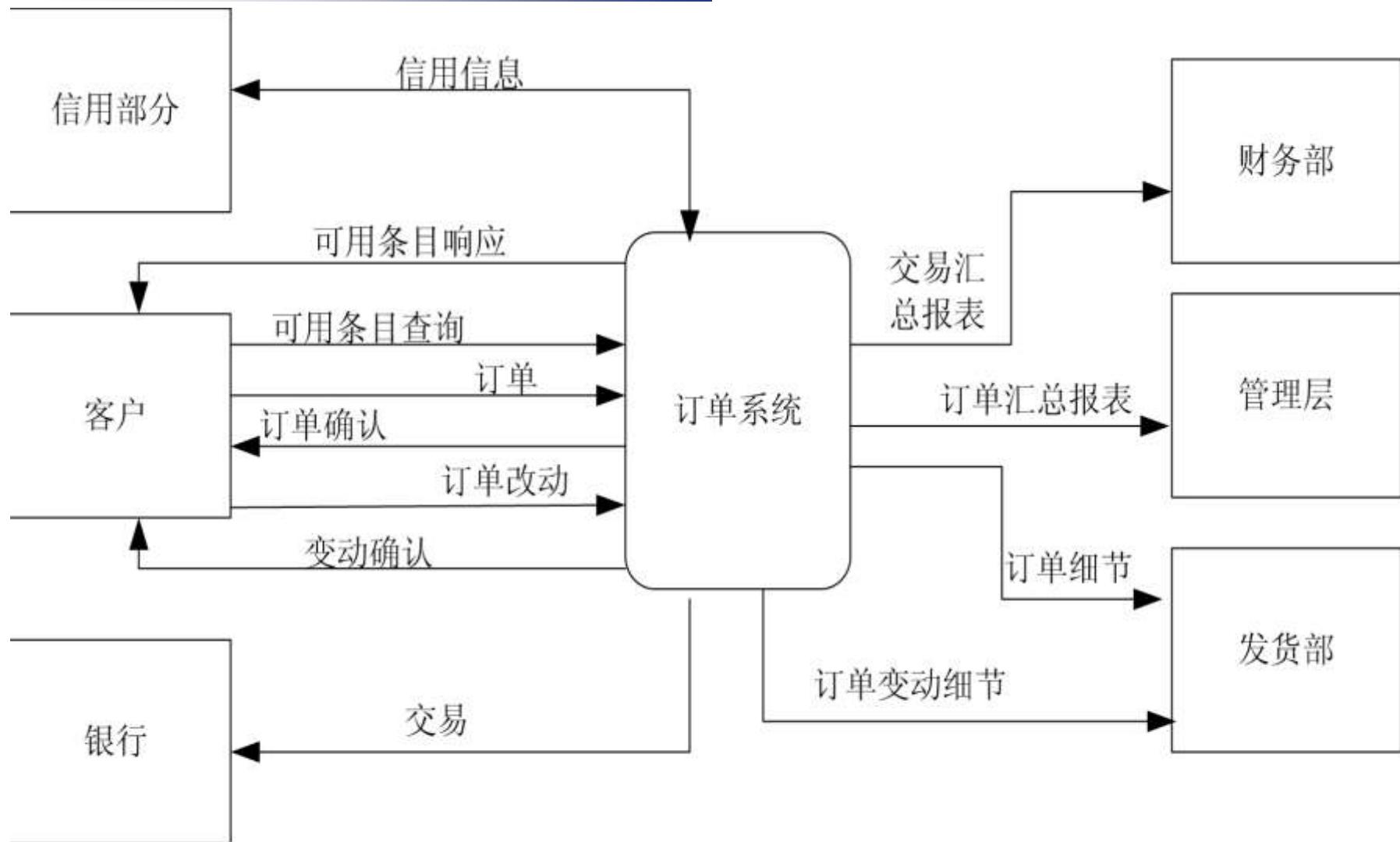
3. 2 几种典型的流程描述方法

从外而内，逐步细化：

- ④ 实体交互图：从系统或用户交互基础上发现信息系统所需的活动、进而构造业务流程

- ④ 业务流程图（TFD）：处理、组织单元、数据；
- ④ 角色行为图（RAD）：活动，活动的关系、角色；
- ④ IDEF动态图：活动，对象转化网络；
- ④ EPC模型：事件，活动，规则连接，其他；
- ④ UML的活动图（活动、活动的控制流）
 - 状态图（状态，状态变换）。
 - 顺序图（对象、消息）。

3.1.1 实体交互图



事件表

事件：引起系统执行某项操作的事件。

来源：引起系统执行的外部实体。

响应：系统产生的输出结果。

| 事件 | 触发器 | 来源 | 活动/用例 | 响应 | 目的地 |
|-------------|------|----|--------|--------|-----|
| 客户想确定可用商品信息 | 商品查询 | 客户 | 查询可用商品 | 可用商品细节 | 客户 |

触发器：系统动作的触发条件。外部事件，就是进入的数据，临时事件就是定义好的时间点。

活动/用例：事件发生时，系统的操作。

目的地：得到输出的外部实体

事件表

| 事件 | 触发器 | 来源 | 活动/用例 | 响应 | 目的地 |
|---------------|--------------|----|----------|--------|-------|
| 1 客户想确定可用商品信息 | 商品查询 | 客户 | 查询可用商品 | 可用商品细节 | 客户 |
| 2 客户发送订单 | 新订单 | 客户 | 生成新订单 | 实时连接 | 信用卡部门 |
| | | | | 订单确认 | 客户 |
| | | | | 订单细节 | 发货部门 |
| | | | | 交易处理 | 银行 |
| 3 客户修改或取消订单 | 订单修改请求 | 客户 | 修改订单 | 修改确认 | 客户 |
| | | | | 订单修改细节 | 发货部门 |
| | | | | 事务处理 | 银行 |
| 4 生成订单汇总报表的时刻 | 周末、月末、季度末、年末 | 无 | 生成订单汇总报表 | 订单汇总报表 | 管理部门 |
| 5 生成交易汇总报表 | 每天下班时 | | 生成交易汇总报表 | 交易汇总报表 | 会计 |

3. 2. 2 业务流程图

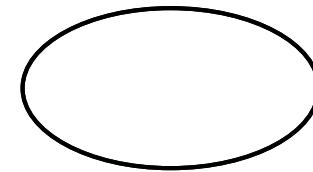
业务流程图 (Transaction Flow Diagram , TFD)

- 就是用一些尽可能少的规定的符号及连线来表示某个具体业务处理过程。
- TFD易于阅读和理解，是分析业务流程的重要方法。
- 业务流程图基本图形符号尚无统一的标准，但在同一系统开发过程中所使用的基本图形应是一致的。

业务流程图的基本符号



业务处理
功能描述



业务处理单位



表格 / 报表制作



数据 / 文件存档



收集数据

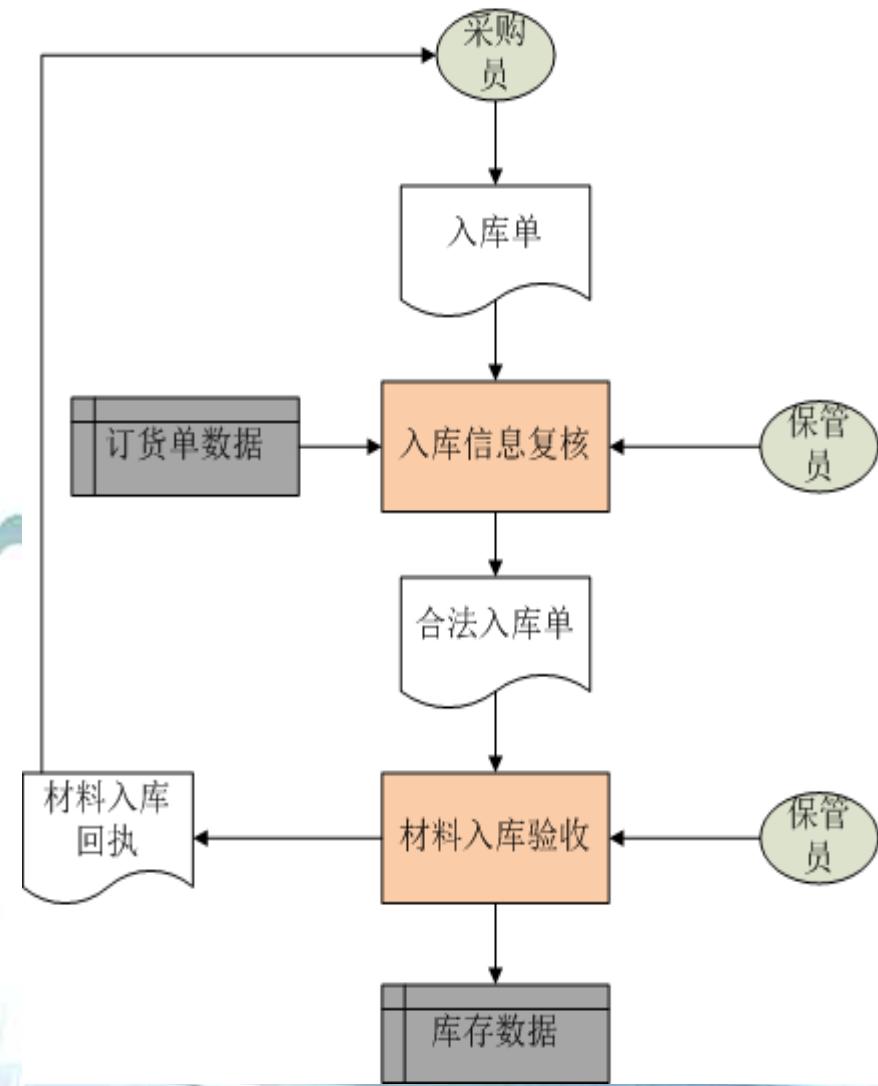


信息传递过程



实例：入库流程

- 处理活动为核心
- 文档流
- 数据的读写支持
- 几种变形



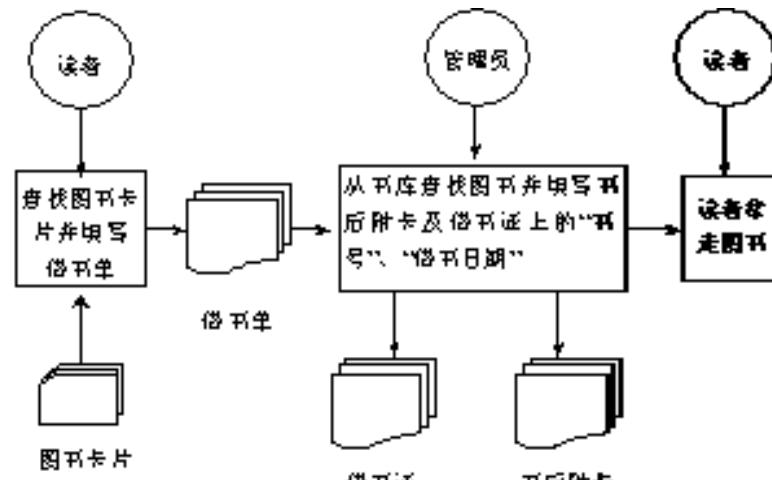
TFD 和 DFD 的关系

- ① TFD 和 DFD 都是描述业务数据处理过程的图形工具。
- ② TFD 强调业务过程中每一项处理活动和具体业务部门的关系。而 DFD 更注重描述业务内数据间的关系及业务的“系统”特征，标识业务通过外部实体与其环境交换信息。
- ③ 从使用者的角度来看，用 TFD 描述企业各项业务的数据处理过程更容易与用户进行交流。DFD 较 TFD 抽象，描述的是业务处理过程的数据处理模式，但难于描述系统的控制流。

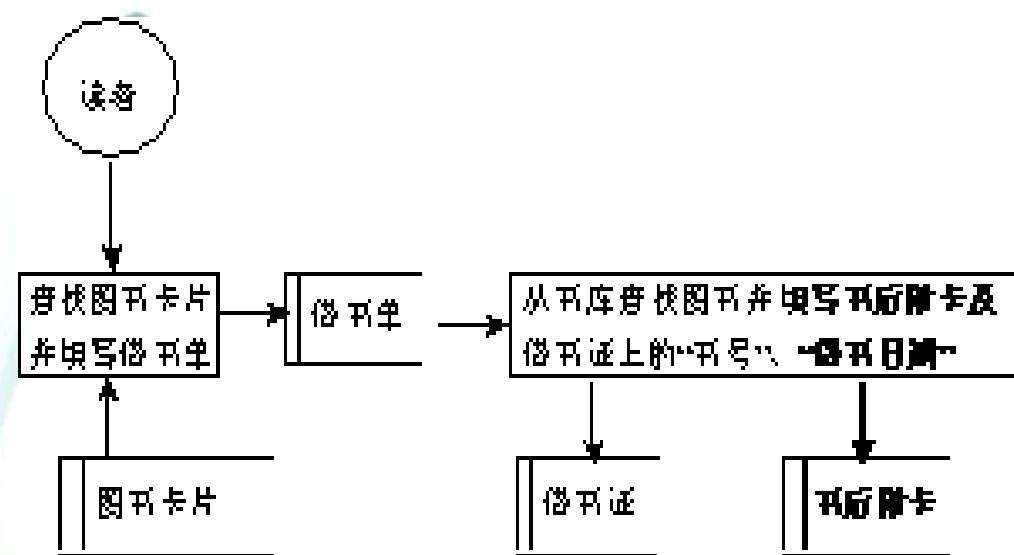


“图书借阅”业务

业务流程图



数据流图





3. 2. 3 角色行为图RAD

- ④ 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。
- ④ 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。
- ④ 它特别擅长强调流程中的角色职责。行为被表示为垂直的一串节点，水平线则表示人的参与。

RAD中采用了五种概念作为模型建立的出发点：

- **角色：**按角色划分过程，与角色相关联的是为完成这一角色而必需的资源；
- **过程目标：**表述了试图达到的状态集合；
- **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；
- **角色间的交互作用：**并不表明每个交互作用的物理形式，而是作用的意义，包括同步、对象的传递和互换。
- **经营规则：**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；



RAD元素符号说明

角色 (Role)



状态(State)



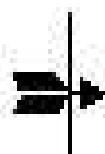
状态描述
(State description)



活动(Activity)

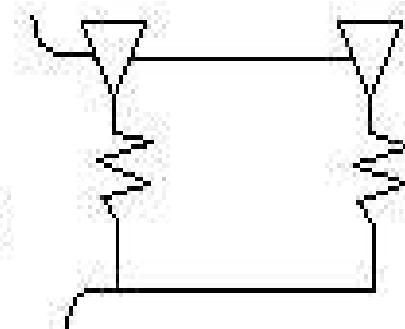


外部触发事件
(External event occurs)



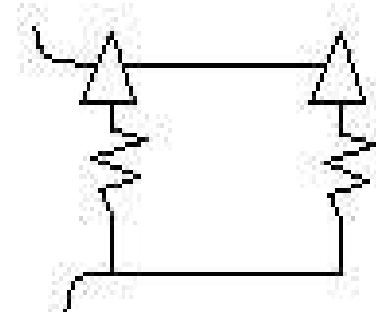
路径选择

(Alternative Paths)



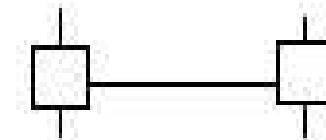
并行路径

(Concurrent Paths)



角色之间的相互作用

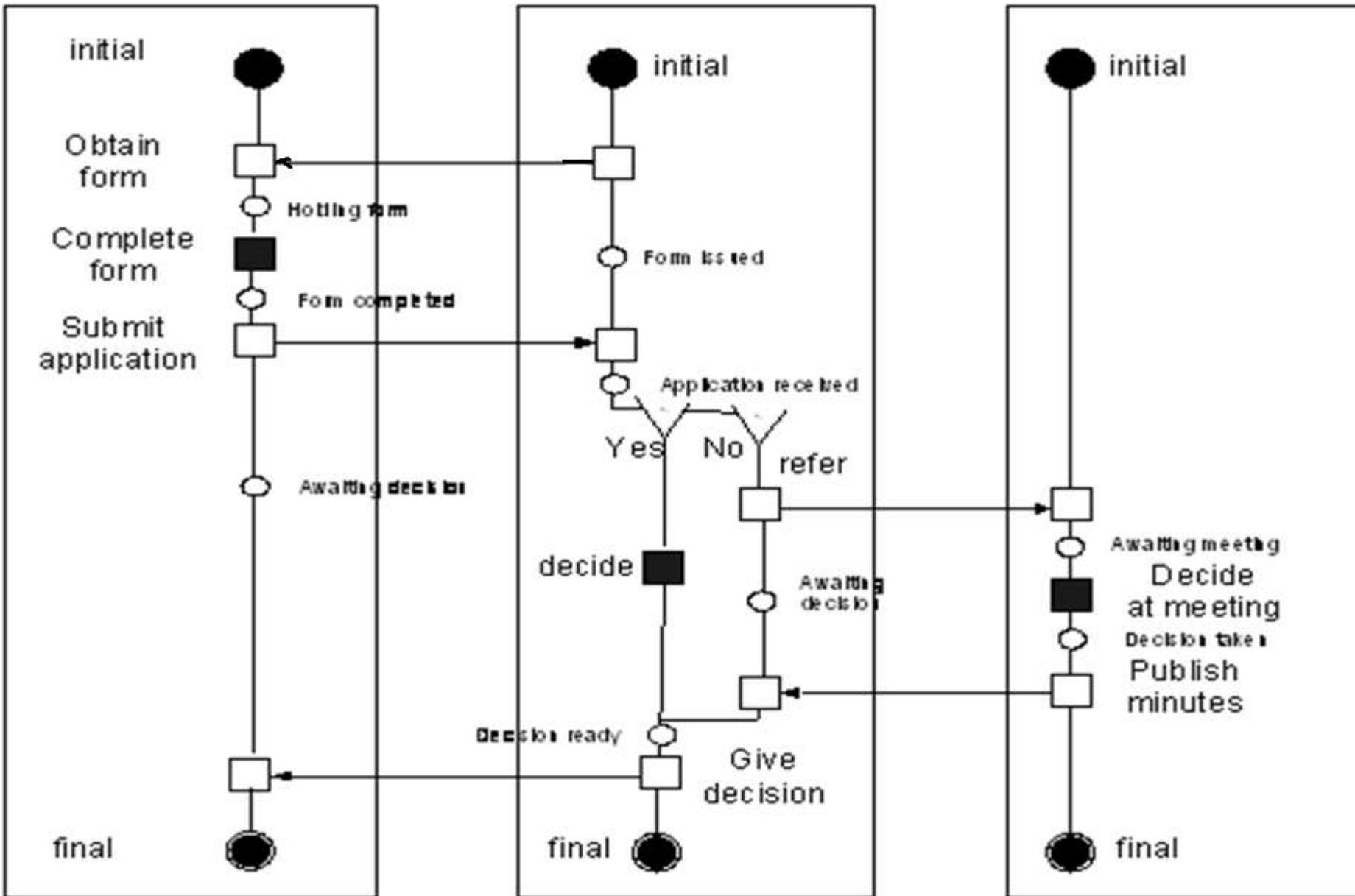
(Interaction between roles)



Client

Clerk

Committee

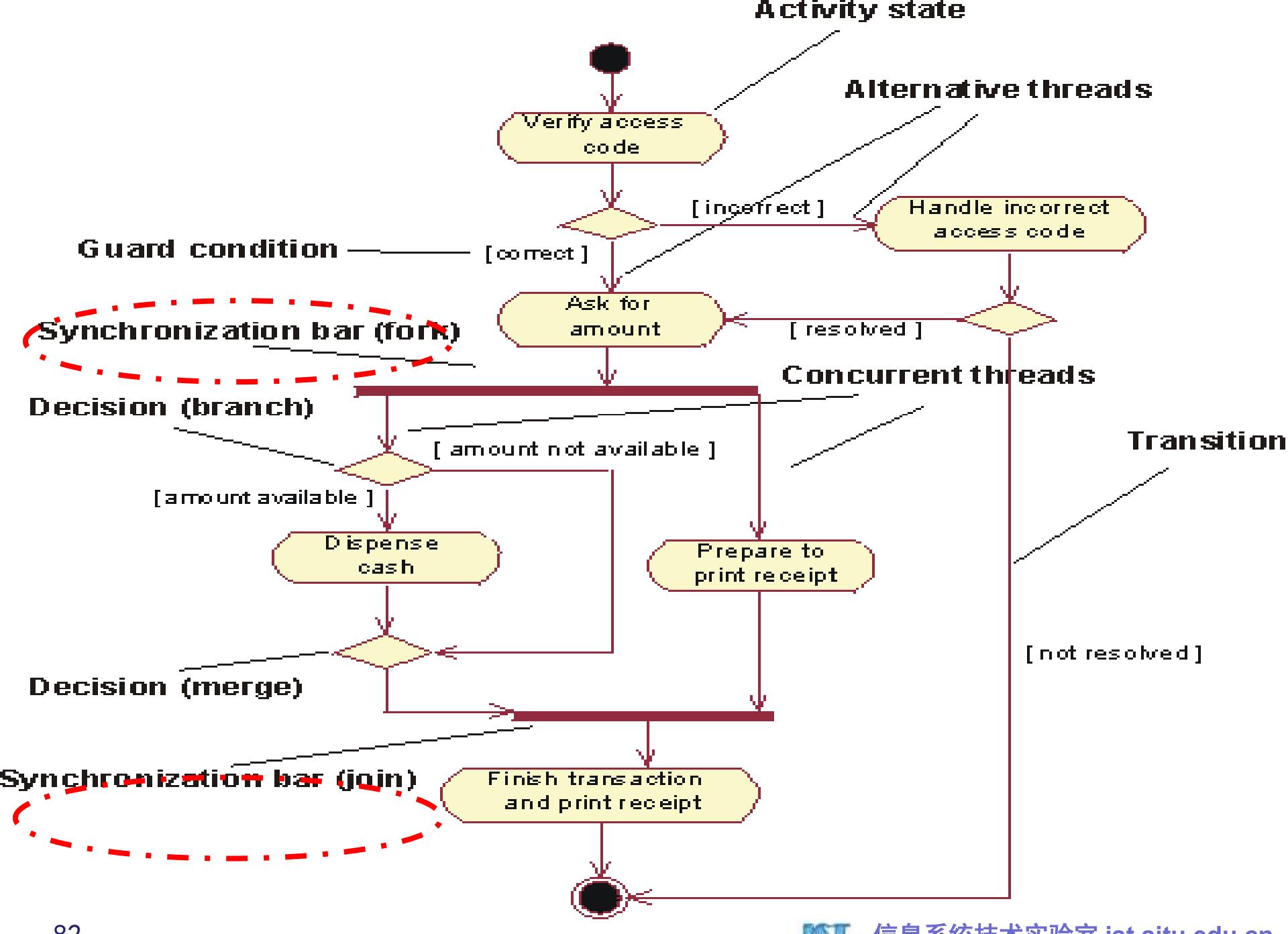


贷款审批流程的RAD图

- ④ 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。
- ⑤ RAD表示法扩展了基于活动的流程表示，虽然它使角色的交互明晰化，但也增加表示的复杂性，因此不便于表达复杂的流程。
- ⑥ RAD方法主要不足之处在于不具有模型分解的能力，这使其除了用于模型流程总览外，无法支持深入的流程描述。
- ⑦ 复杂性问题的解决：非层次化划分，活动可扩展为新的RAD图；

3. 2. 4 UML活动图

- 活动图描述的是对象的活动的顺序关系所遵循的规则，它着重表现的是**系统的行为**，而不是系统的处理过程。
- 在活动图中也没有通常的循环控制结构，活动图能够表现并发情形。
- 活动图着重表现活动的**控制流**，描述在对象之间传递的操作。（顺序图着重描述在对象之间传递的消息）

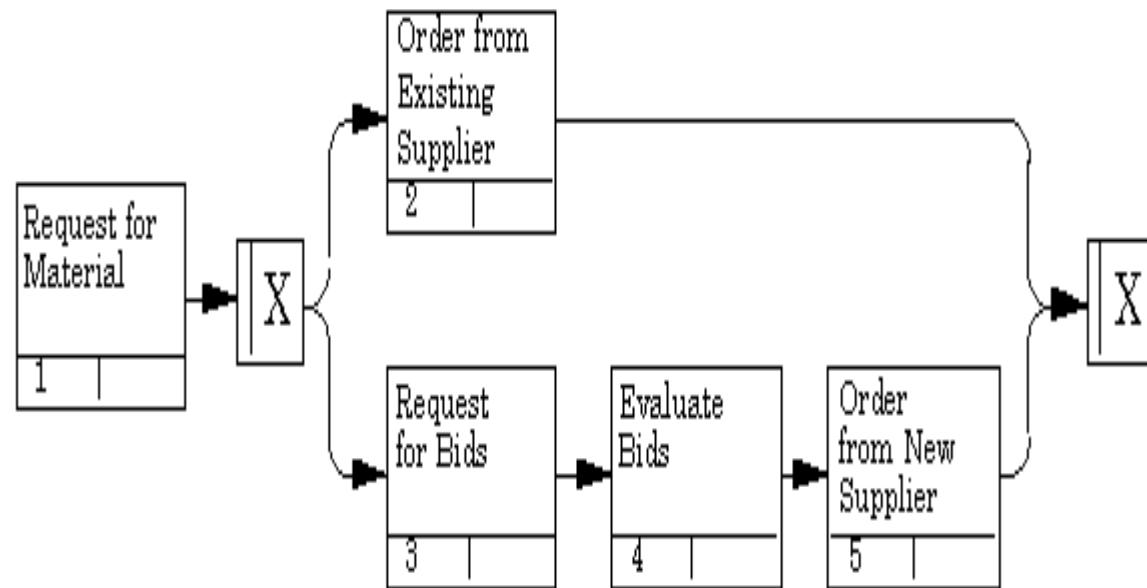


3. 2. 5 IDEF3方法

IDEF3是一种对过程和对象状态的描述方法。它提出了一种结构化的方法来表示一个系统的工作状态及状态向的变换。

IDEF3有两种描述方式：过程流和对象状态转变网络 (OSTN)。

- 过程 (Process)
- 条件分岔 (Junction)
- 连接 (Link)
- 参考指针 (Referent)



参考流程六大要素，分析比较各种描述方法：

1. 包含的要素及特色：

■ 组成要素

■ 特色

2. 指标：

■ 流程特点

■ 可理解性

■ 对流程改造的支持能力

■ 是否引入组织因素

■ 是否动态

3. 3 流程建模规范

Seven Process Modeling Guidelines 7PMG

- G1: Use as few elements in the model as possible
- G2: Minimize the routing paths per element.
- G3: Use one start and one end event.
- G4: Model as structured as possible.
- G5: Avoid OR routing elements.
- G6: Use verb-object activity labels.
- G7: Decompose the model if it has more than 50 elements.

■ **G1: Use as few elements in the model as possible.**

- Larger models tend to be more difficult to understand and have a higher error probability than small models.

■ **G2: Minimize the routing paths per element.**

- higher the degree of an element, harder to understand, higher the modeling errors.

■ G3: Use one start and one end event.

- Low error probability.
- Workflow engines.

■ G4: Model as structured as possible.

- as formulas with balanced brackets

■ G5: Avoid OR routing elements.



■ **G6: Use verb-object activity labels.**

- Verb-object style (“Inform complainant”) VS. Action-noun labels (e.g. “Complaint analysis”).

■ **G7: Decompose the model if it has more than 50 elements.**

- For models with more than 50 elements the error probability tends to be higher than 50%.



Agenda

1. 业务模型
2. 企业业务建模体系
3. 业务流程描述方法
4. 小结



4 小结

- 企业业务模型具有**层次性、多视角、复杂相关性**等特点；业务建模就是描述企业应用的标准规范手段，基于模型开发及实施企业信息系统是软件工程的基本要求，也是业务应用的要求。
- 目前存在许多建模方法和建模技术，很多方法与技术是互补的。应根据不同的信息系统特点，按照方法、体系、工具的层次选择模型。
- 业务流程描述时要强调与组织的关联，分析时要专注于核心增值业务的描述和改进。



信息系统分析与设计

Lecture2

基于状态的流程建模

蔡鸿明

hmcai@sjtu.edu.cn

软件学院信息系统技术实验室

lst.sjtu.edu.cn





agenda

1. Petri网概述
2. Petri网理论
 2. 1 经典Petri网
 2. 2 Petri网的形式化
 2. 3 高阶Petri网
3. 基于Petri网的过程建模
4. 建模实例及讨论
5. 小结



1 Petri网概述

- 经典的Petri网是由德国 Carl Adam Petri在 1962年的博士论文 < Kommunikation mit Automaten >中提出的。他发现当时的已有计算机理论如自动机和形式语言不适合描述物理系统，于是就提出了Petri网。
- Petri网是完全从过程的角度出发为复杂系统的描述与分析设计提供的一种有效的建模工具，能自然的描述并发、冲突、同步、资源争用等系统特性，
- Petri网发展也经历了三个阶段。最先集中在孤立网系统研究；1970年至1985年，开始通用网系统研究，但主要被用于理论界；自从80年中期后，实际的应用越来越多，这主要是由于引入高阶 Petri网和许多工具；



Carl Adam Petri

- Petri's work significantly advanced the fields of parallel computing and distributed computing, and it helped define the modern studies of complex systems and workflow management. His contributions have been in the broader area of network theory which includes coordination models and theories of interaction, and eventually led to the formal study of software connectors.
- In 2008 he got awards of IEEE Computer Pioneer Award (2008). He was a member of the Academia Europaea since 1989.



From Wikipedia



离散事件动态系统

- 离散事件动态系统（Discrete Event Dynamic System, DEDS）的**系统状态**通常只取有限个离散值，对应于系统忙闲等可能的状况。而这些状态的变化则由于诸如某些环境条件的出现或消失、系统操作的启动或完成等各种事件的发生而引起。**是由异步、突发的事件驱动状态演化的动态系统。**
- 对这种系统首先关心的是它的逻辑行为，可用其演化过程的状态序列和事件序列来刻画。由于其状态空间缺乏易操作的运算结构，难以用传统基于微分或差分方程的方法来研究。
- 常见于通信、交通等公共服务设施，机械、电子等各种离散型生产加工过程，多级控制系统，计算机信息处理等重要技术领域。



- 离散事件动态系统的研究自20世纪80年代后发展较快。针对各层次不同角度的问题提出了多种理论模型和分析技术。它们已开始在许多技术领域得到应用，并被认为是大型复杂信息处理和控制系统分析和设计的重要理论基础。
- Petri网具有严格的数学基础和规范化语义，作为DEDS的描述工具，可描述异步、同步、并行逻辑关系，是描述、分析和控制DEDS的最有效和应用最广泛的方法；
- Petri网，有限自动机、形式语言或等模型可以很好地描述这种逻辑层次的分析和综合问题。



Petri网主要用途

主要用途包括：

- 系统仿真：系统分析与评估的系统仿真。
- 数字分析：可通过结构变化描述系统的变化，支持DEDS形式的数学描述与分析；
- 系统性能分析：如制造系统设备使用率、生产率、可靠性等。
- 系统控制：直接从可视化模型中产生DEDS监控编码，进行系统实施控制。
- 还可以转化为其它的DEDS模型，如马可夫链等。



Petri网在计算机及软件工程中可以应用到：

- ① 工作流建模及管理
- ② 软件设计（尤其是模型驱动设计）
- ③ 数据分析
- ④ 并行程序设计
- ⑤ 协议验证
- ⑥ ...等。



从建模角度

- Petri网是完全从过程的角度出发为复杂系统的描述与分析设计提供的一种有效的建模工具；
- 能自然的描述并发、冲突、同步、资源争用等系统特性，并带有执行控制机制，同时还具备形式化步骤及数学图论相支持的理论严密性。
- Petri网的图形表达的直观性和便于编程实现的技术特点，使得它已经成为目前工作流建模的主要工具之一。
- **从建模角度——可视化图形描述却被形式化数学方法支持；**
- 研究领域趋向认为Petri网是所有流程定义语言之母；



agenda

1. Petri网概述
2. Petri网理论
 2. 1 经典Petri网
 2. 2 Petri网的形式化
 2. 3 高阶Petri网
3. 基于Petri网的过程建模
4. 建模实例及讨论
5. 小结



2 Petri网理论

- Petri网是由描述离散事件动态系统DEDS引出的。
- Petri网具有严格的数学基础和规范化语义，是描述、分析和控制DEDS的最有效和应用最广泛的方法；
- 基本Petri网包含库所、转移、以及它们的关系。
- 高阶Petri网 是对Petri网的扩展：
 - 颜色 (for the modelling of attributes)
 - 时间 (for performance analysis)
 - 层次 (for the structuring of models, DFD's)



2.1 经典Petri网

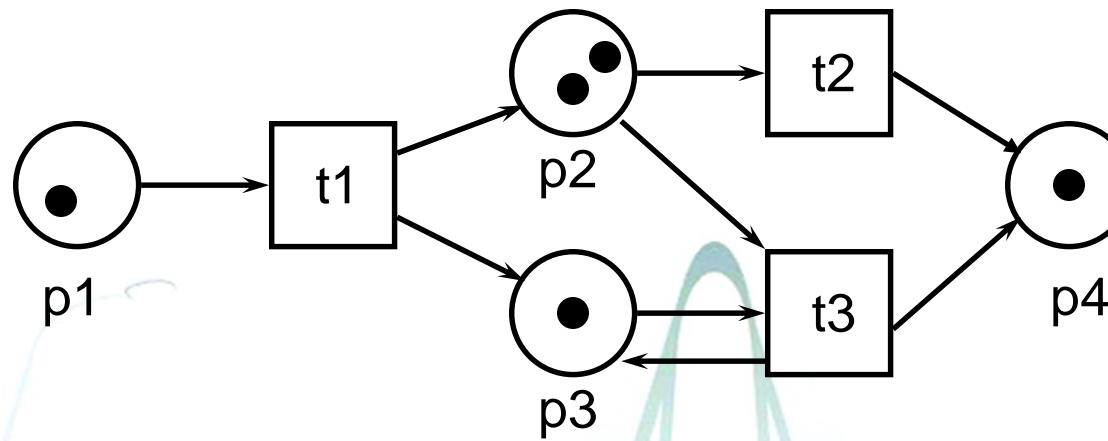
Petri网观点可简单的归纳到两个基本概念：

- 事件和条件, 许多系统均可从事件与条件的观点去建模;
- 系统**状态**可描述为一组条件, 条件就是系统状态的谓词或逻辑描述;
- 事件是系统**状态改变**的原因, 事件的出现是由系统状态控制的;
- **前条件**: 由于事件是动作, 所以它可以发生。为了使事件发生, 必须使某些条件成立, 这种条件称为事件的前条件;
- **后条件**: 事件的发生可能破坏前条件而使另外的条件成立, 这种条件称为事件的后条件。



Petri网的结构

经典的Petri网主要是由库所 (place) 以及转移 (transition) 构成的网络



连接 (Connection) 是库所和转移之间的有向弧，具有方向。
托肯 (Token) •是动态对象。
Petri网的状态由分布在库所中的托肯决定



Petri网的组成元素

Petri网简称PNG (Petri网 Graph), 它有库所和转移两种结点

- ① 库所 (Place) 小圆圈 P
- ② 转移 (Transition) 小方块 T
- ③ 连接 (Connection) 是库所和转移之间的有向边, 流关系 F,K
- ④ 托肯 (Token) 是库所中的动态对象, 可以从一个库所移动到另一个库所 •

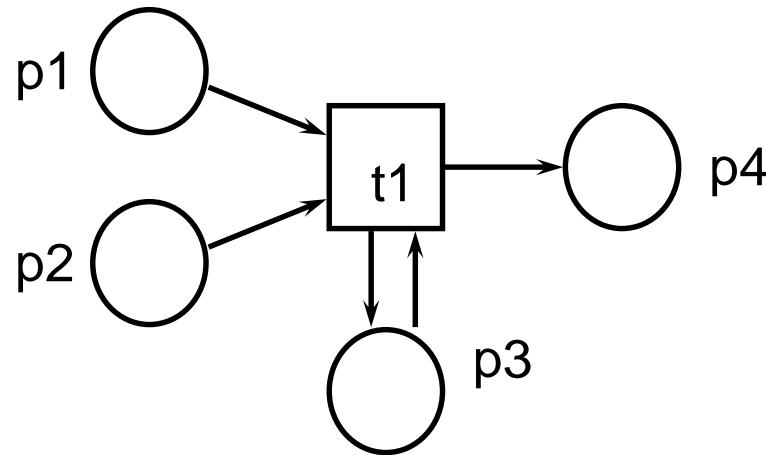


Petri网的规则

- 连接是有方向的，其上可以标出权重
- 两个库所或转移之间不允许有边，且不应该有孤立节点
- 库所可以拥有任意数量的托肯



输入库所/输出库所

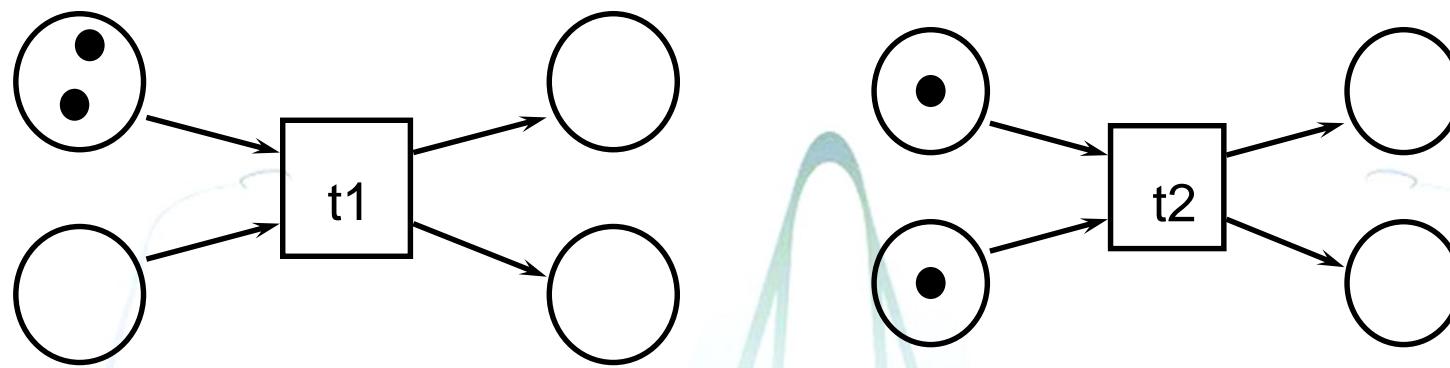


- 转移 t_1 具有三个输入库所 (p_1 , p_2 and p_3) 和两个输出库所 (p_3 and p_4).
- 库所 p_3 既是 t_1 的输入库所又是它的输出库所.



使能条件

- 转移是主动元素，而库所和托肯是被动元素
- 如果输入库所都包含了托肯，那么转移就被激活



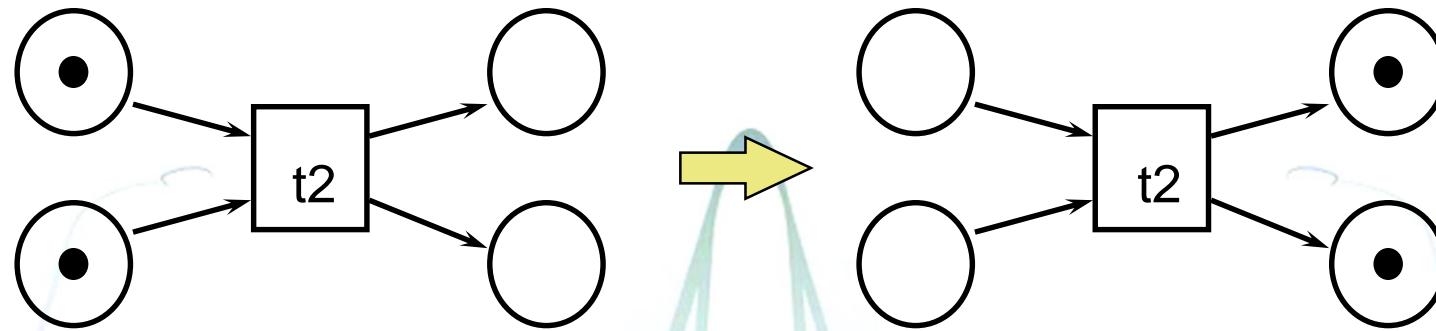
Transition t_1 is not enabled,

transition t_2 is enabled.



点火

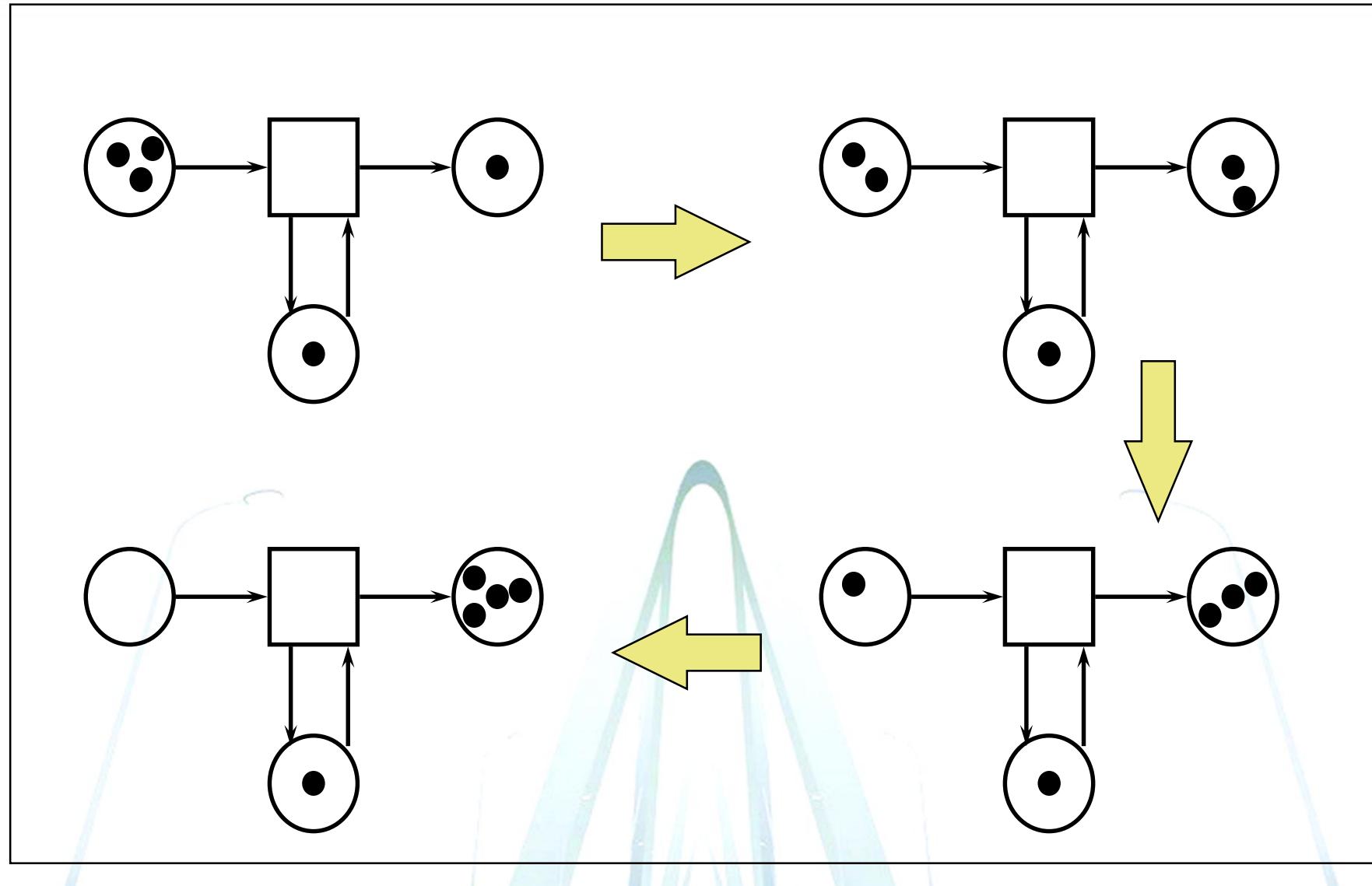
- 激活的转移可以被点火
- 点火将消耗输入库所的托肯，并为输出库所产生托肯



Firing is atomic.

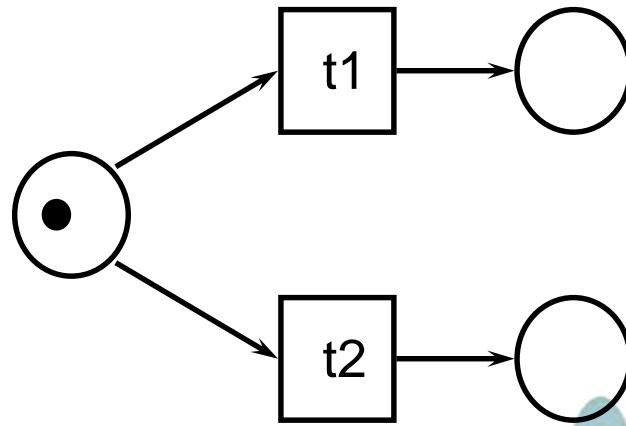


托肯迁移的例子





不确定性



- ◎ 两个转移竞争同一个托肯：冲突
- ◎ 即使有两个托肯，依然存在冲突

Petri网的行为特性小结

- ① 一个转移点火的规则是，如果一个转移的每个输入库所（input place）都拥有托肯，该转移即为被允许(enable)。点火时，输入库所(input place)的托肯被消耗，同时为输出库所(output place)产生托肯。
- ② 转移的发生是完整的，也就是说，没有一个转移只发生了一半的可能性。
- ③ 如网络中有两个或多个转移都被允许的可能，这种情况下转移发生的顺序没有定义，可认为是并发的。
- ④ 一个转移出现其输入库所的个数与输出库所的个数不相等时，托肯的个数将发生变化，也就是说，**托肯数目不守恒**。
- ⑤ Petri网络是静态的，也就是说，网络拓补结构是静态的，不存在发生了一个转移之后忽然冒出另一个转移或者库所，从而改变Petri网结构的可能。
- ⑥ 两个变迁争夺一个托肯的情形被称之为冲突。当发生冲突的时候，由于Petri网的时序是不确定的，因此具体哪个变迁得以发生也是不确定的。
- ⑦ **Petri网的状态由托肯在库所的分布决定。**也就是说，转移发生完毕、下一个转移等待发生的时候才有确定的状态，正在发生转移的时候是没有一个确定的状态的。

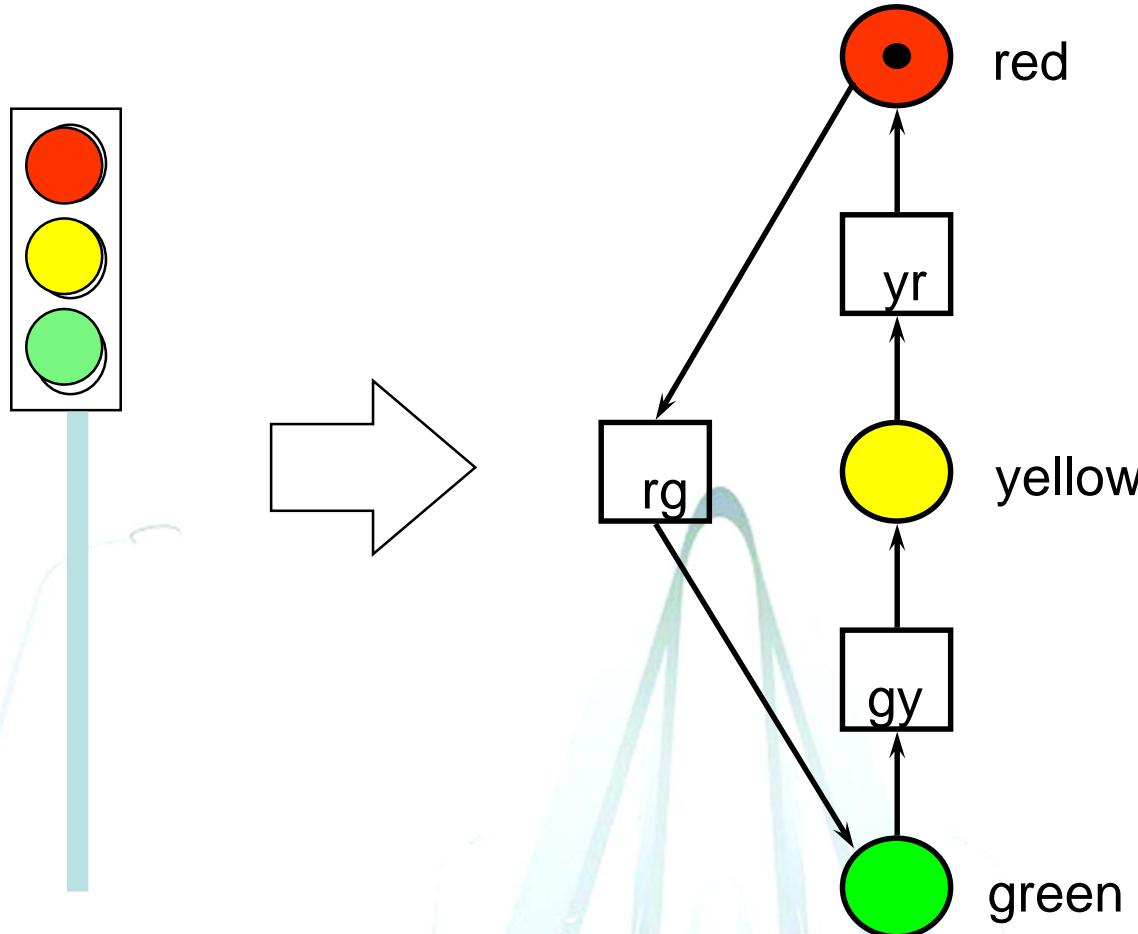


基于Petr i网的流程建模

- ① 库所代表缓存，渠道，地理位置，条件或者状态
- ② 转移代表时间，传输或者转换
- ③ 托肯表示对象 (humans, goods, machines), 信息或者对象的状态
- ④ 过程的状态用位于库所的托肯来表示，状态之间的变换用转移来表示
- ⑤ 讨论：一个航空公司的运营系统如何建模？飞机，航线，机场（位置），如何表示？

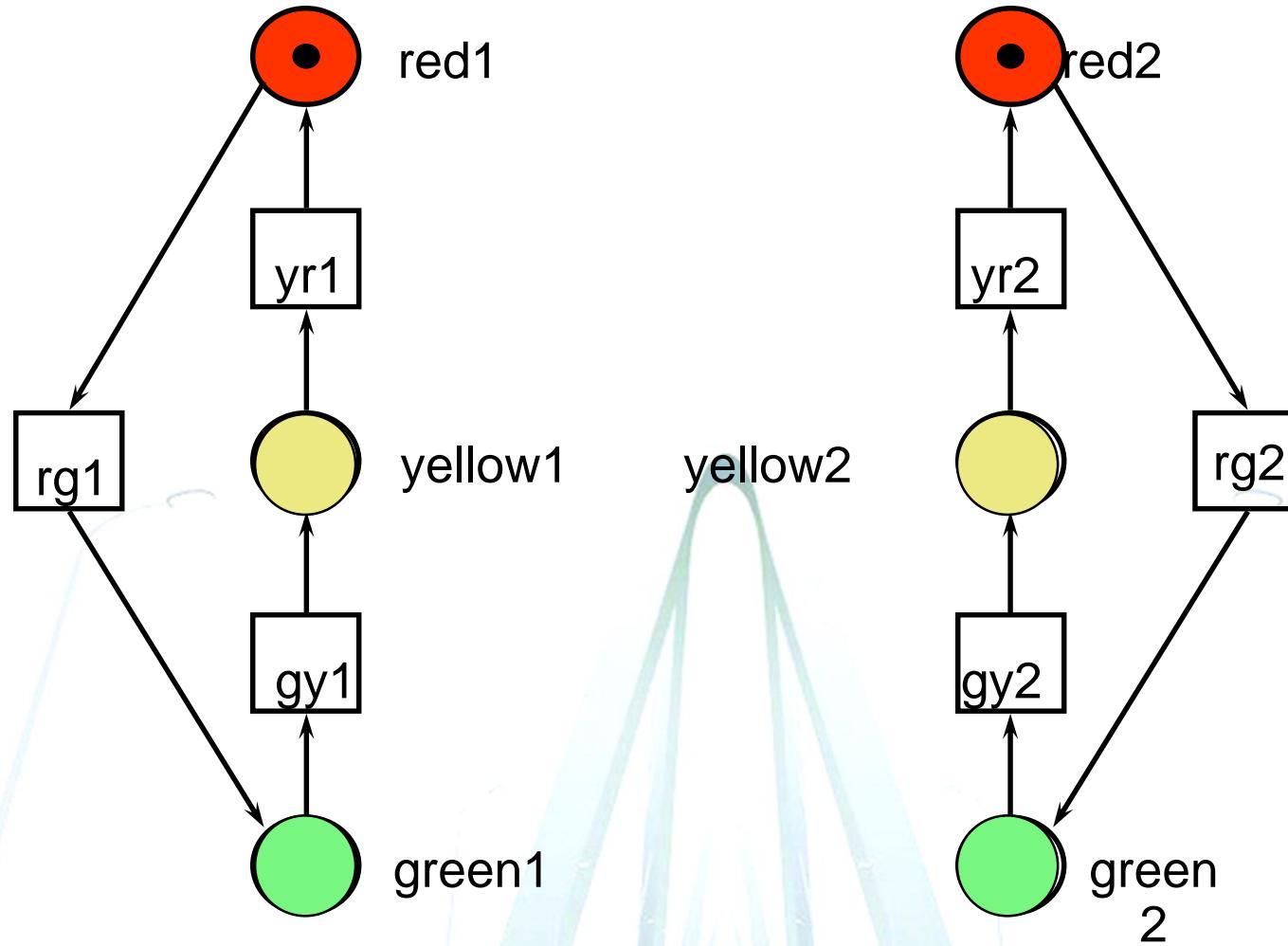


例子：交通灯



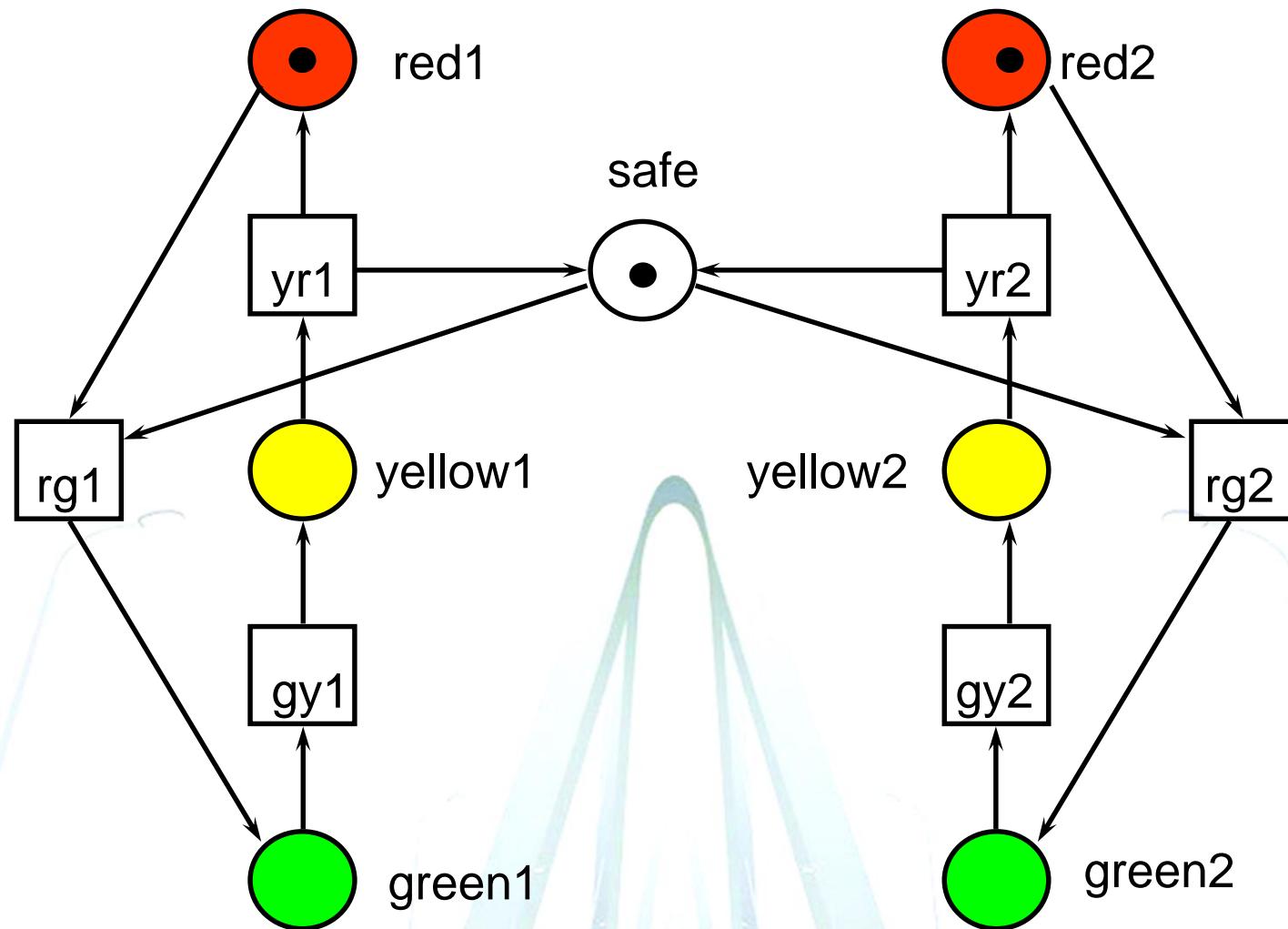


两个交通灯



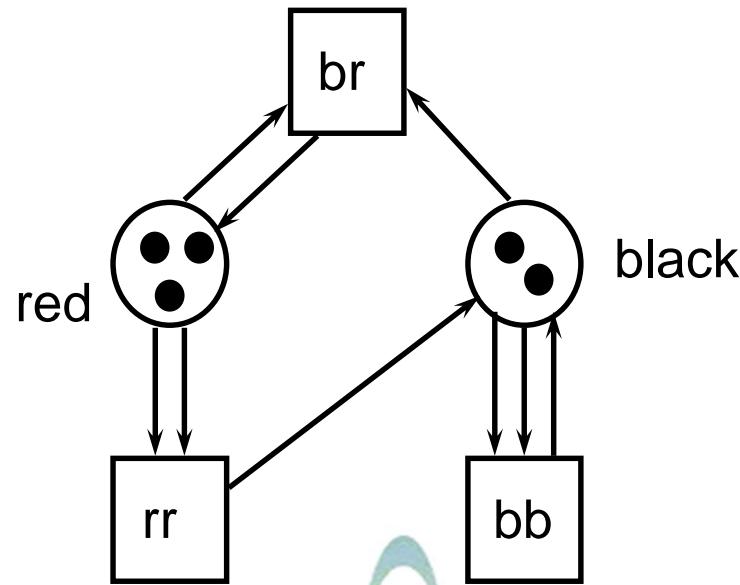


两个安全的交通灯





Ball game



- 拿到2黑或2红放回1黑；
- 拿到黑红各1放回1红；



某些定义

流程有当前状态，可达状态，不可达状态，死状态等划分

- **当前状态**

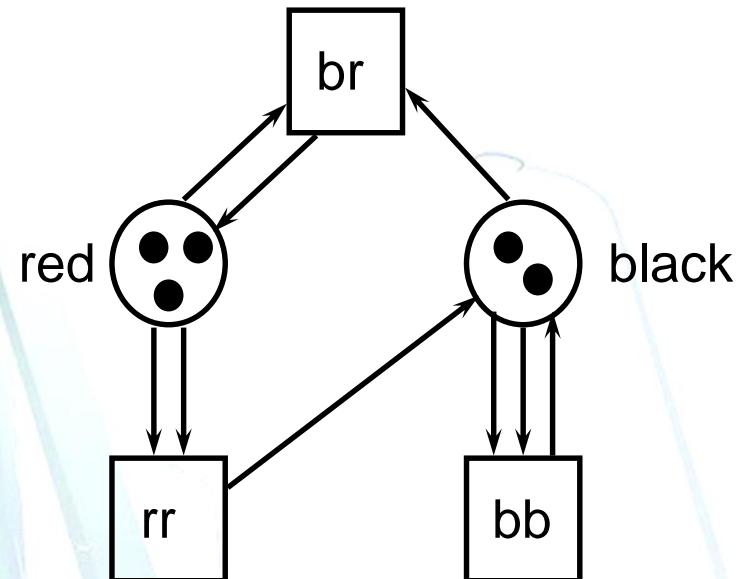
库所中托肯的分布情况.

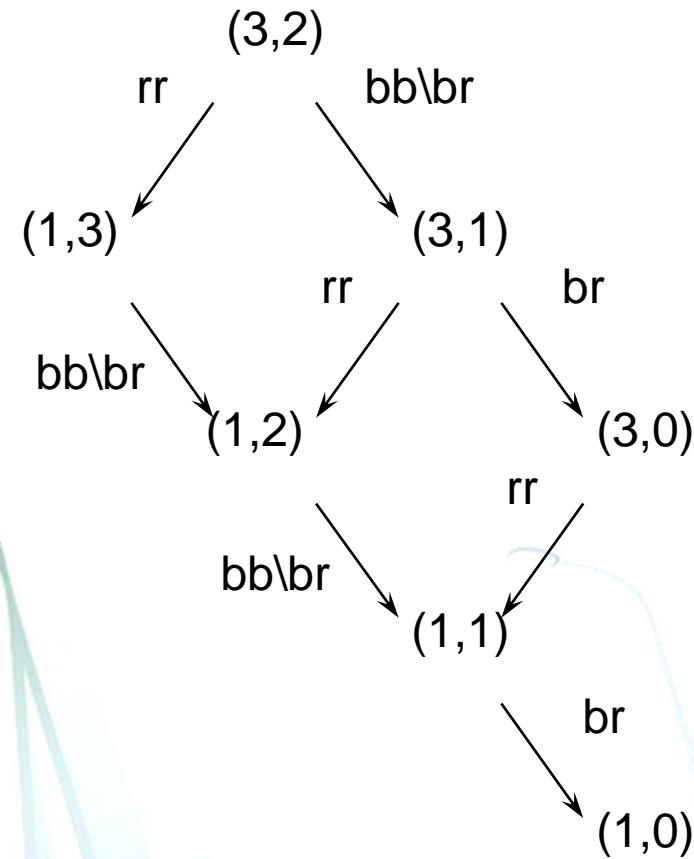
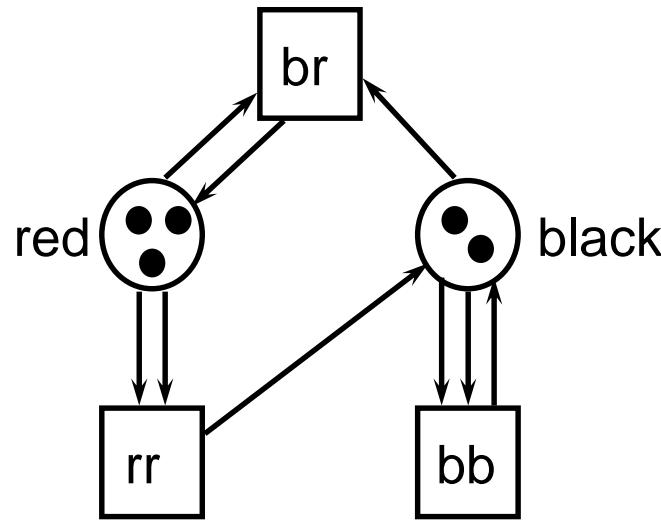
- **可达状态**

通过一系列激活的转移的点火，从当前状态可以达到的状态.

- **死状态 (dead state)**

没有转移能够激活的状态

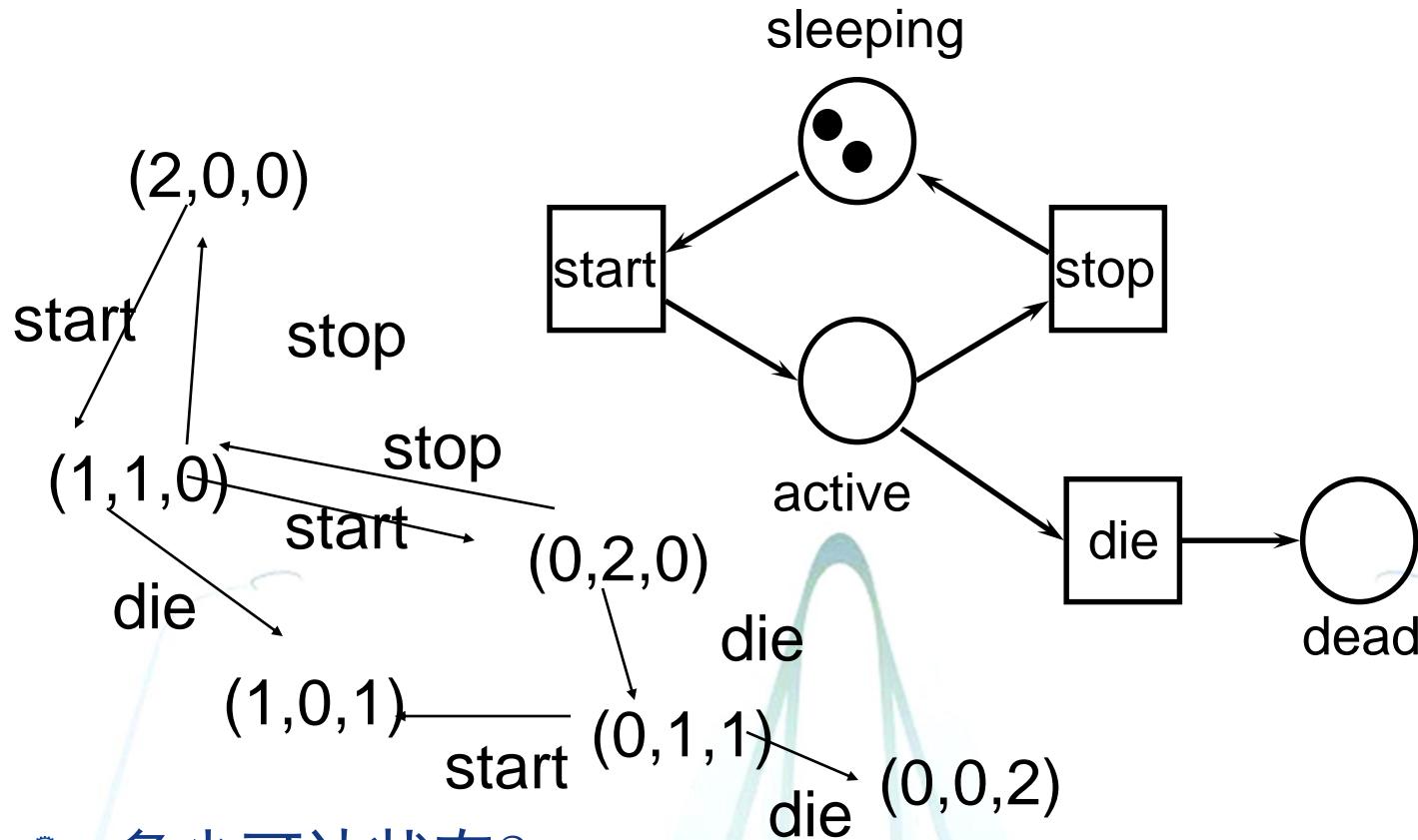




- 拿到2黑或2红放回1黑；
- 拿到黑红各1放回1红；
- 7 可达状态, 1 死状态.



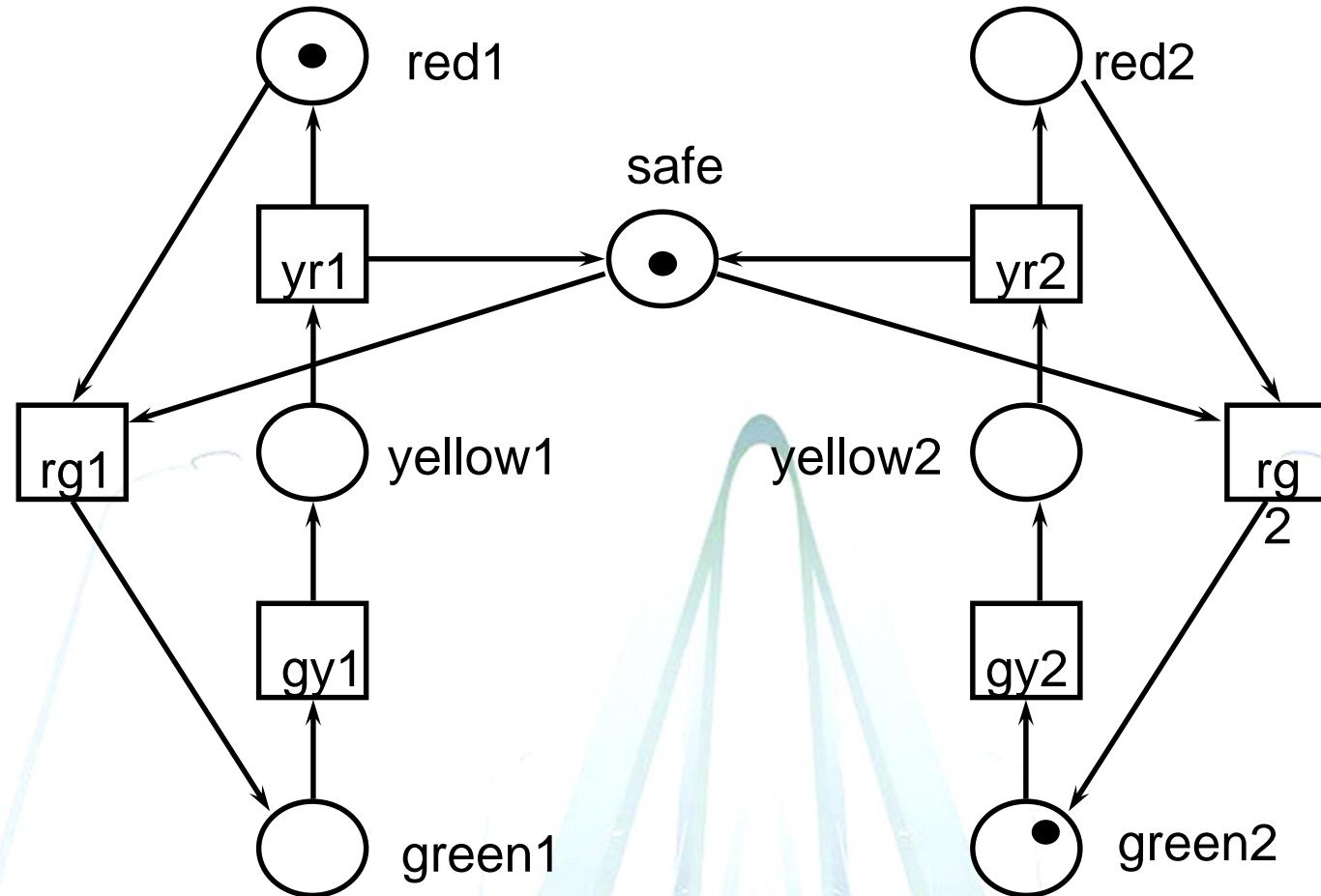
练习：2个人的一生



- 多少可达状态？
- 是否有死状态？

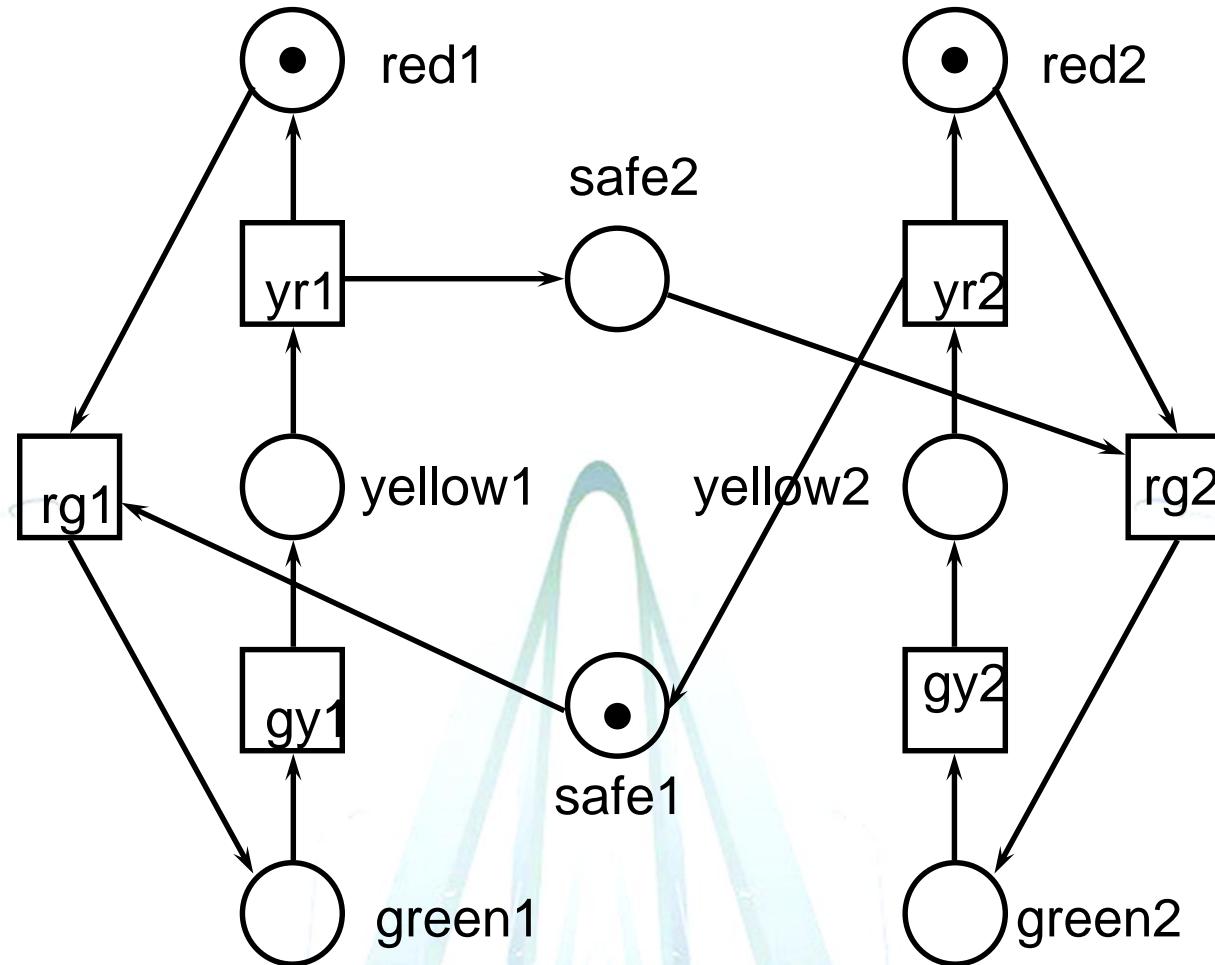


交通灯的可达图



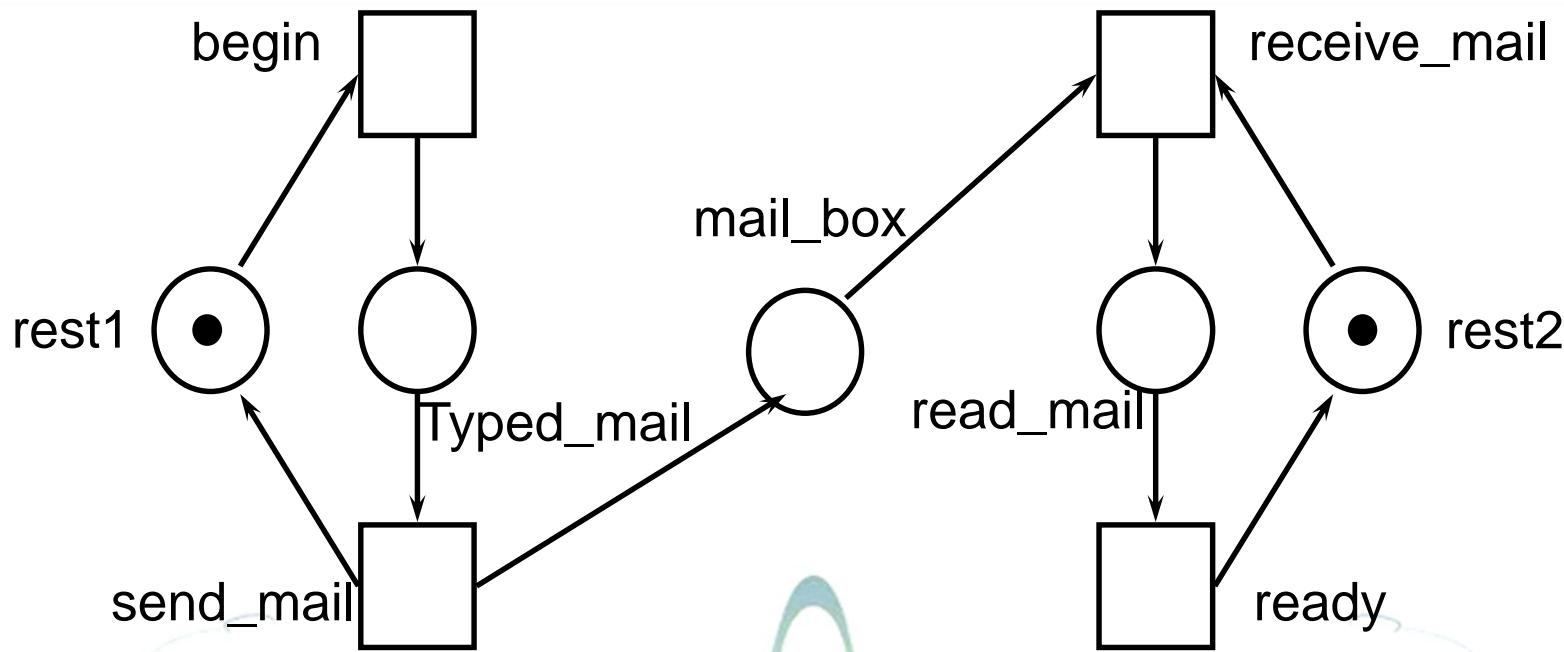


安全而公平的交通灯





练习



- ④ 画出可达图
- ④ 多少个可达状态?
- ④ 有无死状态?
- ④ 两个作者和三个读者的情况是怎样的?



2.2 Petri网的形式化表达

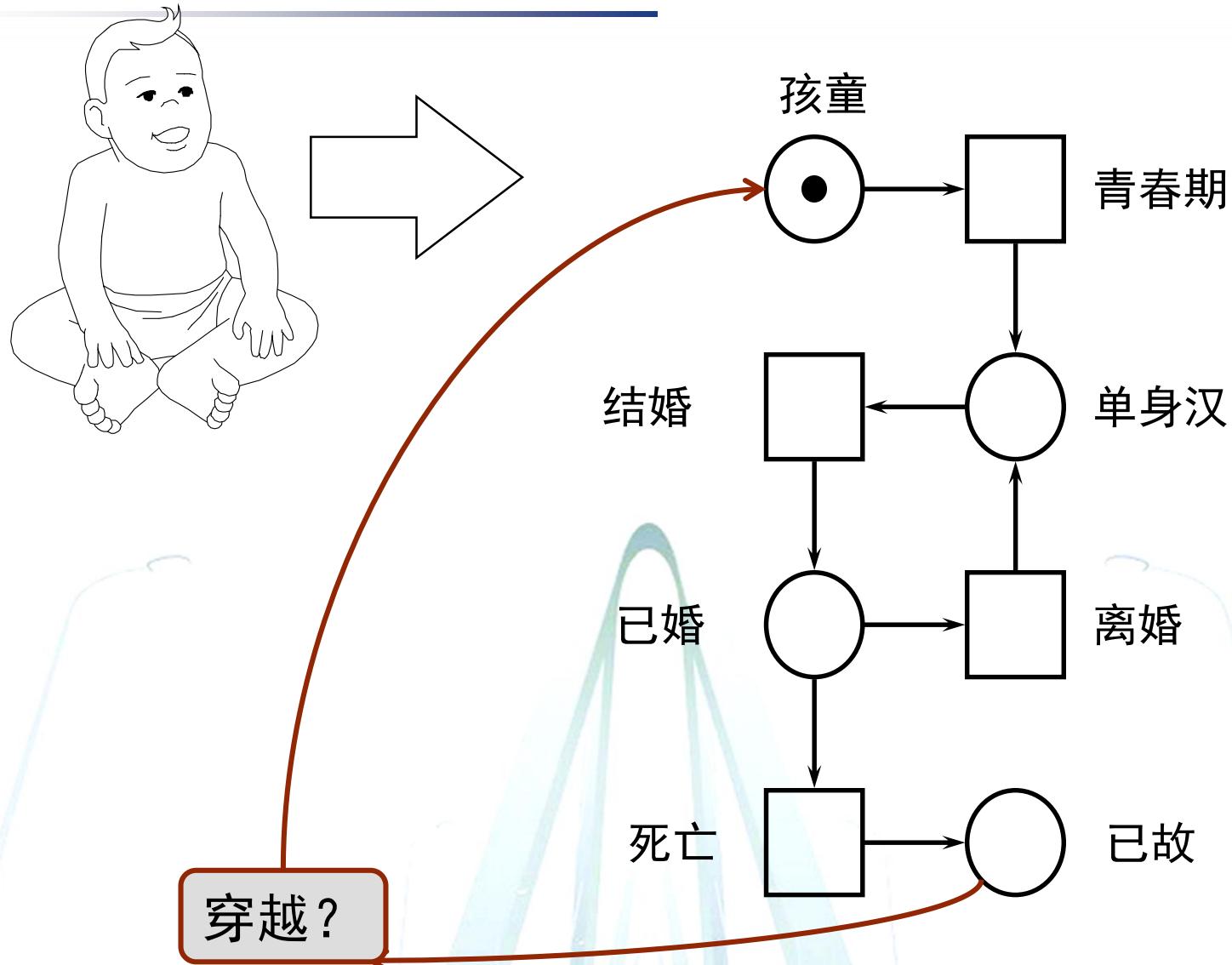
Petri网定义的一般可表达为五元组

$$\Sigma = (P, T, F, K, M_0)$$

- ① 其中， P 为位置的集合，用圆圈代表，表示系统的状态；
- ② T 为转移的集合，用空心矩形代表，表示系统中的事件；
- ③ F 称为 $P \rightarrow T$ 的流关系，其规定资源的输出流；
- ④ K 称为 $T \rightarrow P$ 的流关系，其规定资源的输入流；
- ⑤ M 称为Petri网 Σ 的状态， M_0 为初始状态的标识。
- ⑥ Token表示工作对象，转移是网络中的控制点。Petri网进行算法扩展，可以使它具有处理模型求解系统运行的能力。



人的一生





$$PN = (P, T, F, W, M)$$

- ④ (P, T, F) 是基网，此时 F 为不区分 F , K 的统一流关系的表述；
- ⑤ W 是有向弧的权重函数，
- ⑥ M 为状态，是一个多维向量集合， $M(p)$ 表示对应库所 P 中的token数
- ⑦ 对于 $t \in T$ ，用 $\bullet t = \{p \in P | (p, t) \in F\}$ 来表示转移 t 所有输入库所的集合；
- ⑧ 对于 $t \in T$ ，用 $t \bullet = \{p \in P | (t, p) \in F\}$ 来表示转移 t 所有输出库所的集合。
- ⑨ 在 PN 终止库所 o 和开始库所 i 之间插入一个转移 t^* 并添加相应连接弧，构成 PN 的扩展网络。



- 构成了一个工作流网，用 \overline{PN} 表示，其形式化表述为：

$$\overline{PN} = (P, T \cup \{t^*\}, F \cup \{(o, t^*), (t^*, i)\}), W, M$$

辅助转移

终止库所

初始库所

- PN 有两种特殊的库所： i 和 o ， i 是初始库所： $\bullet i = \emptyset$ ，
 o 是终止库所： $\bullet o = \emptyset$ ；
- \overline{PN} 是强连通的；

PN 是自由选择网当且仅当对于每个转移 t_1 和 t_2 ，当
满足 $\bullet t_1 \cap \bullet t_2 \neq \emptyset$ ，则 $\bullet t_1 = \bullet t_2$ 。



- MES系统从上游ERP系统接收到订单，提出加工请求申请，
- 在加工请求申请得到批准之后生成一个生产计划，
- 根据生产计划进行相应的工艺流程设计，之后对工艺流程进行审核，如果审核不通过则退回进行重新设计。
- 如果审核通过，流程将分两步走，一方面进行工艺工时定额，另一方面进行物料需求申请、备料。待两方面都准备好之后才能进行生产排产，
- 然后进行生产，生产结束之后将生产出的产品进行相应的质检，如质检通过则进行入库登记管理，否则将进入废品处理阶段，
- 最后将各阶段的生产资料数据进行归档统计生成报表。

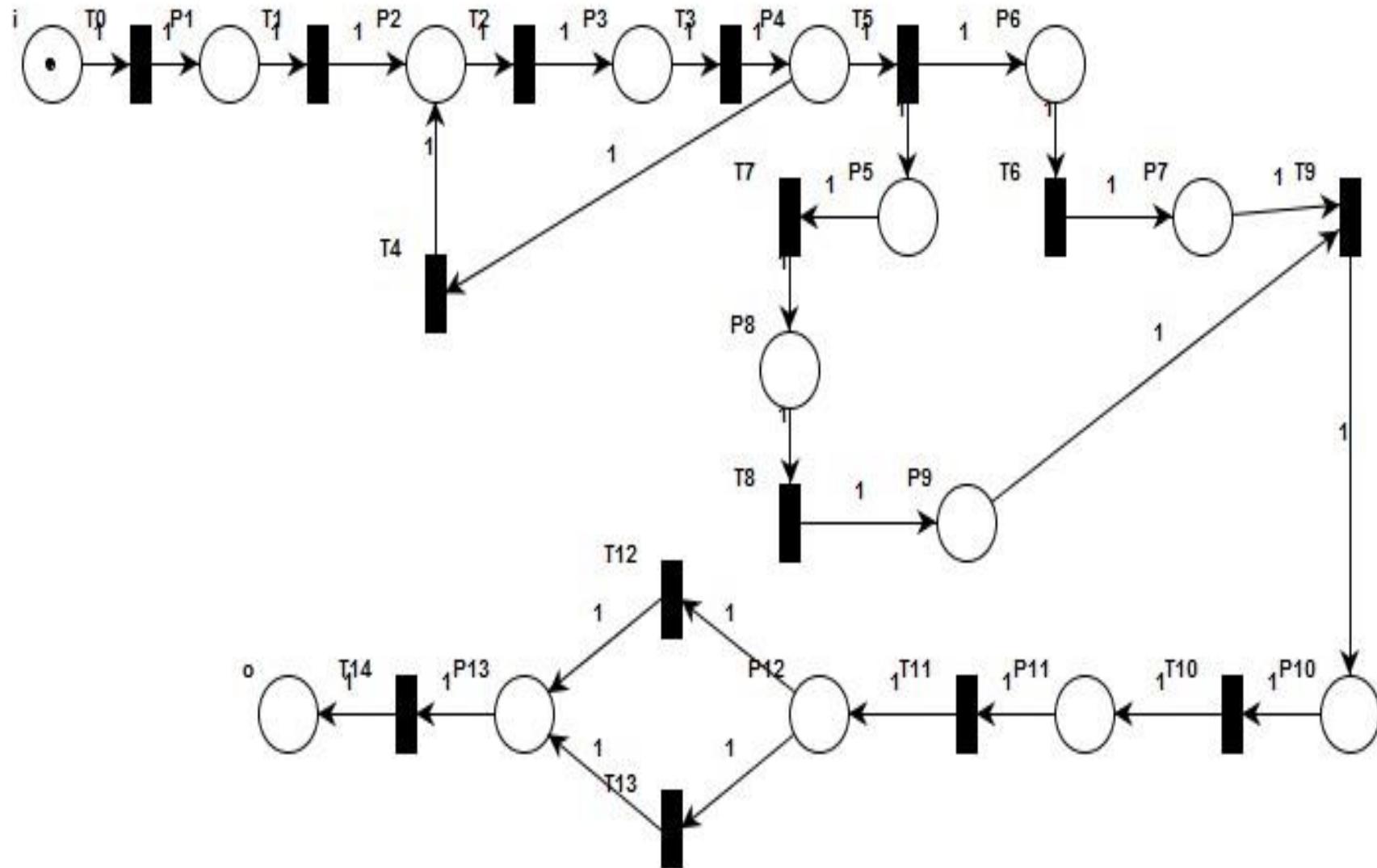


要素定义

| 库所P | 内容 | 转移T | 业务环节定义 | 权重w |
|-----|--------|-----|--------|-----|
| i | 开始 | T0 | 订单加工请求 | 10 |
| P1 | 请求执行 | T1 | 生成生产计划 | 5 |
| P2 | 开始工艺设计 | T2 | 工艺流程设计 | 1 |
| P3 | 工艺设计结束 | T3 | 工艺流程审核 | 2 |
| P4 | 审核开始 | T4 | 发回重新设计 | 1 |
| P5 | 文件处理完毕 | T5 | 审核文件处理 | 2 |
| P6 | 文件处理完毕 | T6 | 计算工时定额 | 2 |
| P7 | 工时定额完成 | T7 | 物料需求申请 | 10 |
| P8 | 需求申请结束 | T8 | 备料 | 2 |
| P9 | 备料结束 | T9 | 生产排产 | 5 |
| P10 | 生产排产结束 | T10 | 生产 | 1 |
| P11 | 生产完成 | T11 | 质检 | 2 |
| P12 | 质检完成 | T12 | 废品处理 | 10 |
| P13 | 库房 | T13 | 成品入库 | 5 |
| o | 结束 | T14 | 统计归档报表 | 2 |



生产流程的工作流网模型

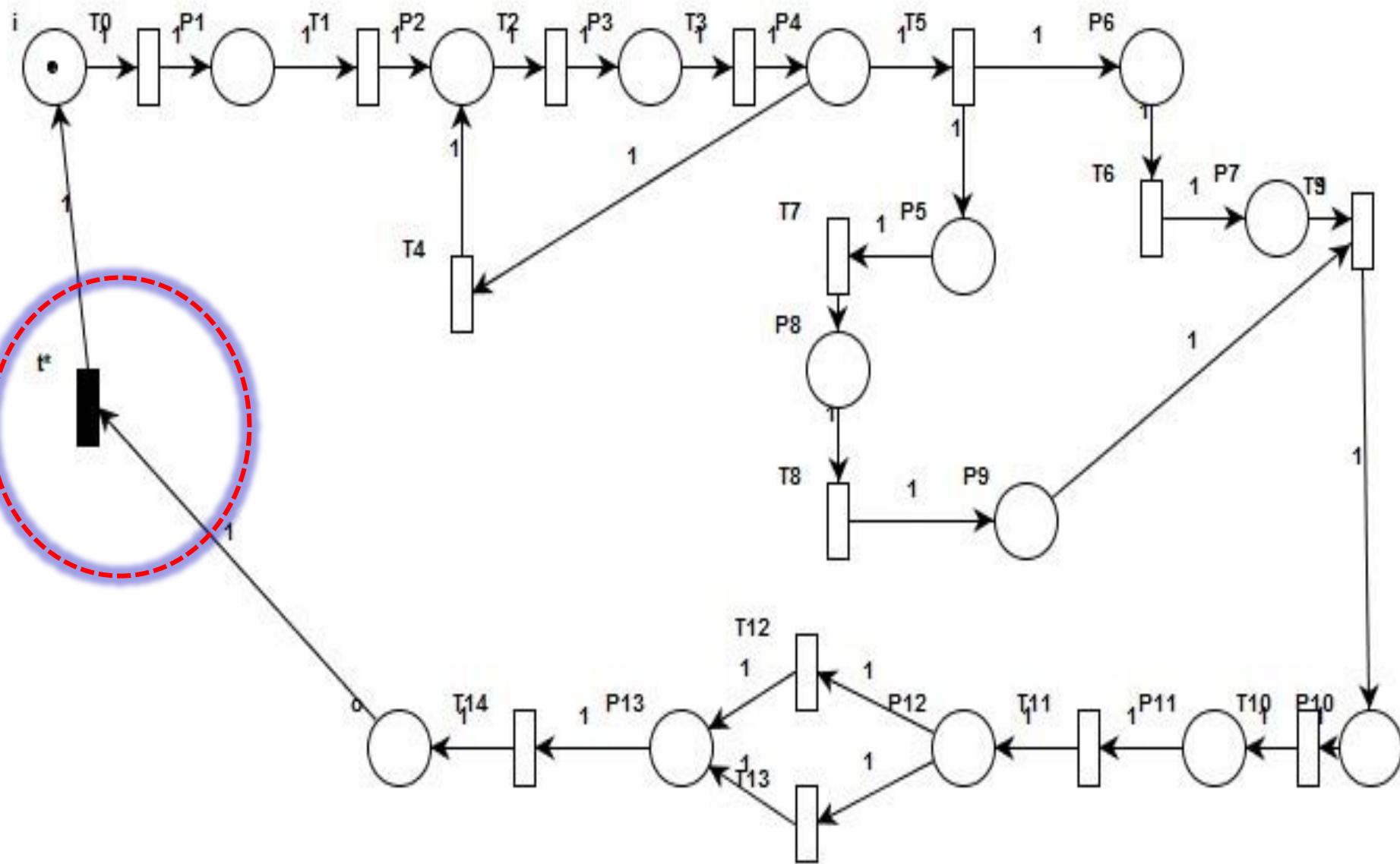




关联矩阵（流关系）

| | T0 | T1 | T10 | T11 | T12 | T13 | T14 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 | t* |
|-----|----|----|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|
| i | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| o | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | -1 |
| P1 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P10 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| P11 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P12 | 0 | 0 | 0 | 1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P13 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | -1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| P3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | -1 | 0 | 0 | 0 | 0 | 0 |
| P5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | 0 | 0 | 0 |
| P6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 |
| P7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | -1 | 0 |
| P8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 |
| P9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -1 | 0 |

1代表p为t的输出库所， -1代表p为t的输入库所， 0代表无联系





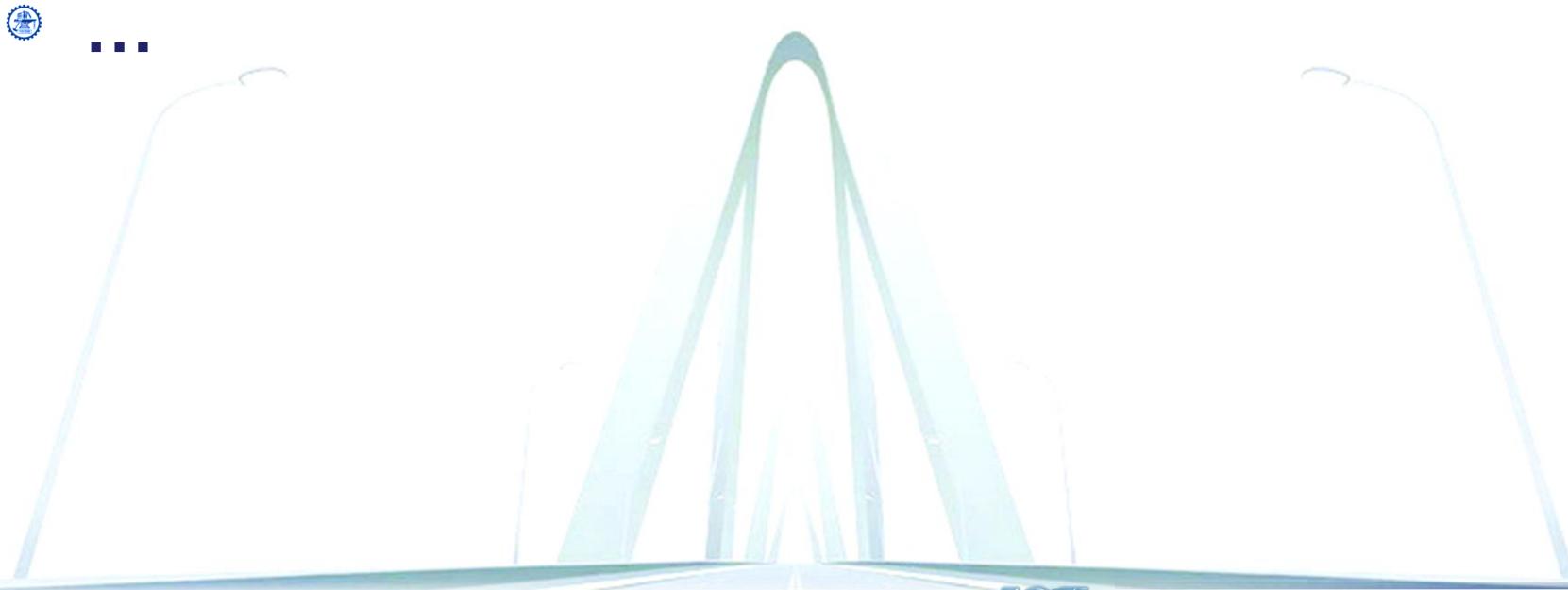
转移矩阵（各状态的转移）

| M_0 | M_1 | M_2 | M_3 | M_4 | M_5 | M_6 | M_7 | M_8 | M_9 | M_{10} | M_{11} | M_{12} | M_{13} | M_{14} |
|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-----------------|-----------------|-----------------|-----------------|----------------------------------|
| $-\lambda_0$ | λ_0 | | | | | | | | | | | | | |
| | $-\lambda_1$ | λ_1 | | | | | | | | | | | | $SPN = (P, T, F, W, M, \lambda)$ |
| | | $-\lambda_2$ | λ_2 | | | | | | | | | | | |
| | | | $-\lambda_3$ | λ_3 | | | | | | | | | | |
| | | | | $-\lambda_4$ | λ_4 | | | | | | | | | |
| | | | | | $-\lambda_5$ | λ_5 | | | | | | | | |
| | | | | | | $-\lambda_6$ | λ_6 | λ_6 | | | | | | |
| | | | | | | | $-\lambda_7$ | λ_7 | λ_7 | | | | | |
| | | | | | | | | $-\lambda_8$ | λ_8 | λ_8 | | | | |
| | | | | | | | | | $-\lambda_9$ | λ_9 | λ_9 | | | |
| | | | | | | | | | | $-\lambda_{10}$ | λ_{10} | λ_{10} | | |
| | | | | | | | | | | | $-\lambda_{11}$ | λ_{11} | λ_{13} | |
| | | | | | | | | | | | | $-\lambda_{13}$ | λ_{14} | |
| | | | | | | | | | | | | | $-\lambda_{14}$ | |

维度 λ (其中转移 t^* 是瞬时转移), 代表转移在单位时间内可实施的次数,



- ④ 结构特性分析：可达状态分析，状态方程方法；
- ⑤ 性能分析：加入延迟时间，随机数模拟；
- ⑥ 仿真：标记概率密度函数，位置中的平均标记数，转移利用率，转移的标记流速
- ⑦ ...





Petri网特点

利用Petri网建模具有以下优点。

- ① (1) Petri网建立在严格的数学基础上，精确描述系统中事件的依赖关系和不依赖关系，这是事件之间不依赖于观察的关系，已有了许多成熟的分析方法和工具。
- ② (2) 兼顾了严格语义与图形表示两方面，具有统一的语言描述系统结构和行为，方便建模仿真，从而起到沟通不同子系统间桥梁的作用；
- ③ (3) Petri网是一种基于状态的建模方法，与基于事件的过程建模方法不同，Petri网系统比其他图形建模工具更适于确定触发方式、描述同步并发系统，并具有更多的柔性。



2.3 高阶Petri网

● Petri网建模的一些缺点阻碍了其在复杂问题的使用：

- 没有测试库所中零托肯的能力
- 模型容易变得很庞大，建模可能要花太多的时间
- 模型不能反映时间方面的内容，不能处理时间和数据信息
- 不支持构造大规模模型，如自顶向下或自底向上

● 在实际应用中，经典的Petri网并不是非常有用：

- Petri网组成模型的元素数量过多；
- Petri网不如基于活动的图形容易理解；
- Petri网的建模中不能在网中体现数据流；

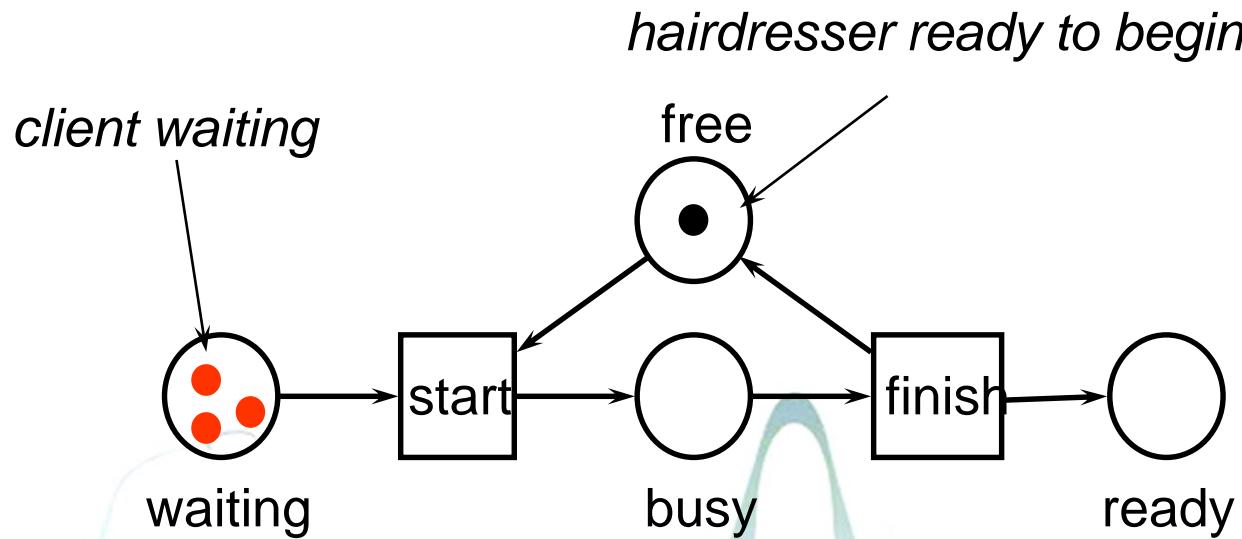


- 因此我们需要使用高阶Petri网，也就是说采用以下方式来扩展Petri网：
 - 颜色——属性描述
 - 时间——性能分析
 - 层次——结构分解
 - 时序——增加时序逻辑的定义，更好的描述行为过程。



例子

◎ 理发厅的例子

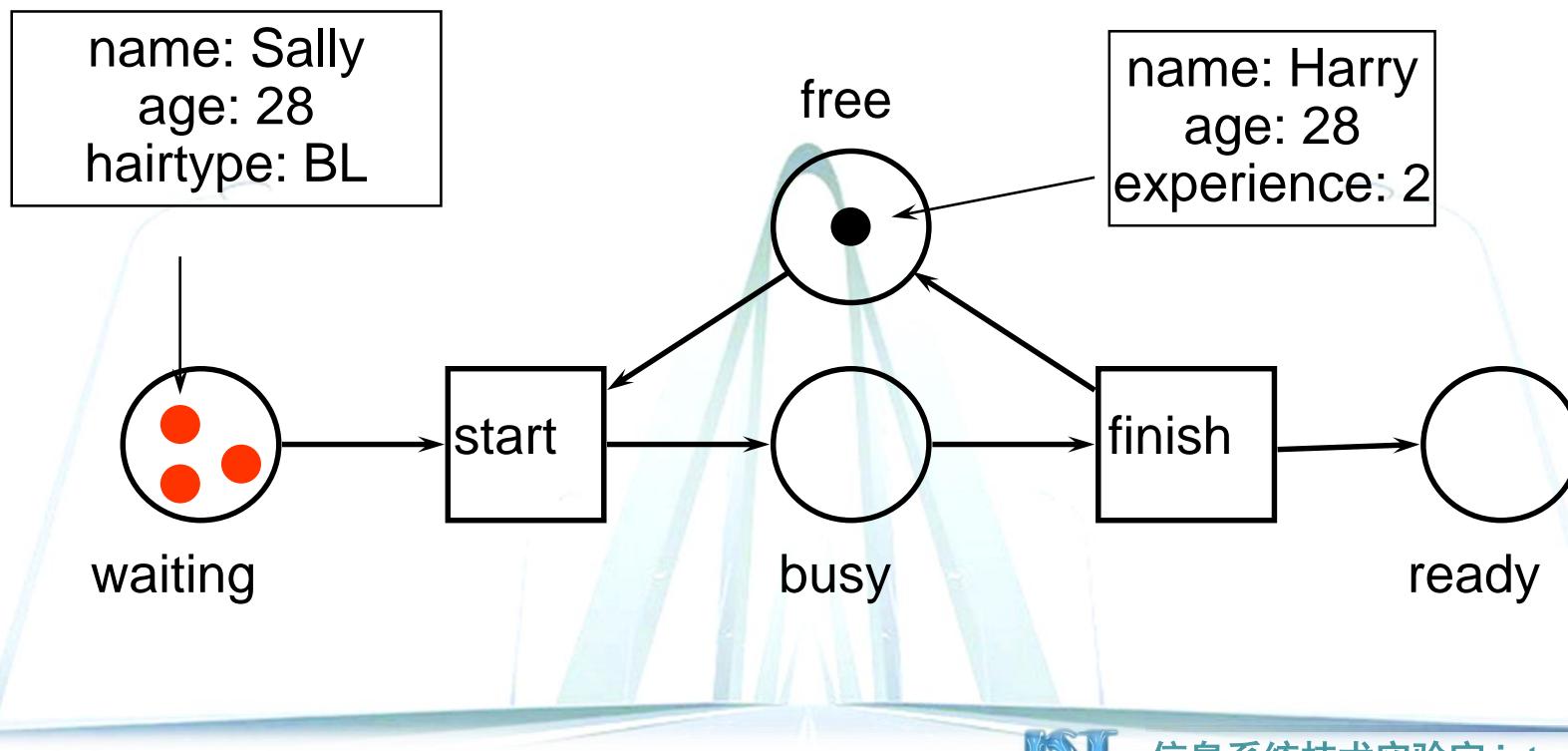


注意：如何简化了对多个理发师的建模



采用颜色进行扩展

- 某一托肯经常代表了具有某种属性的对象
- 因此，每一托肯具有值（颜色）以表示由托肯建模的对象的特定属性



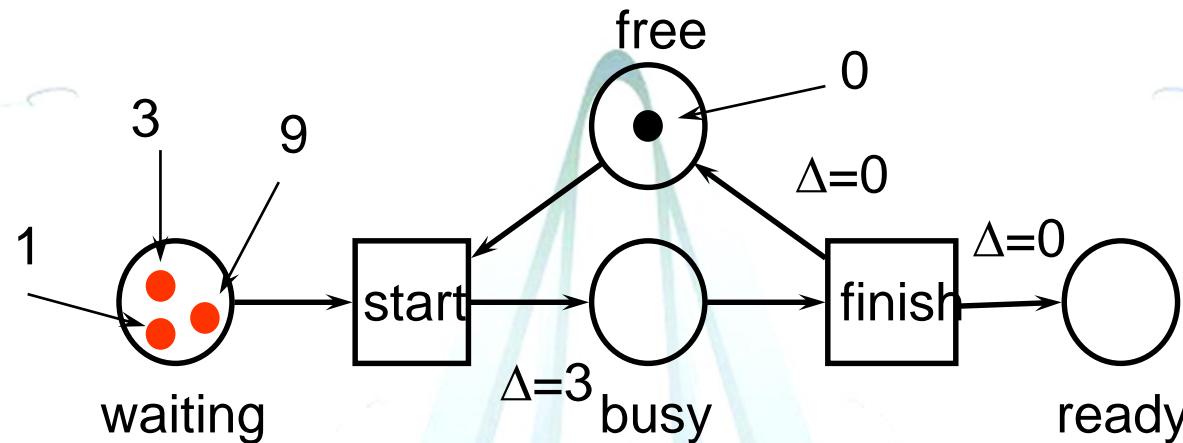


- ◎ 每一个转移可以有正式的（非正式的）描述：
 - 产生的托肯数目
 - 这些托肯的值
 - 和（可选）的一个前提条件
- ◎ 复杂性被分解到网络和托肯的值上
- ◎ 这种处理产生了紧凑、可管理和自然的过程描述。
- ◎ 例子：汽车装配过程



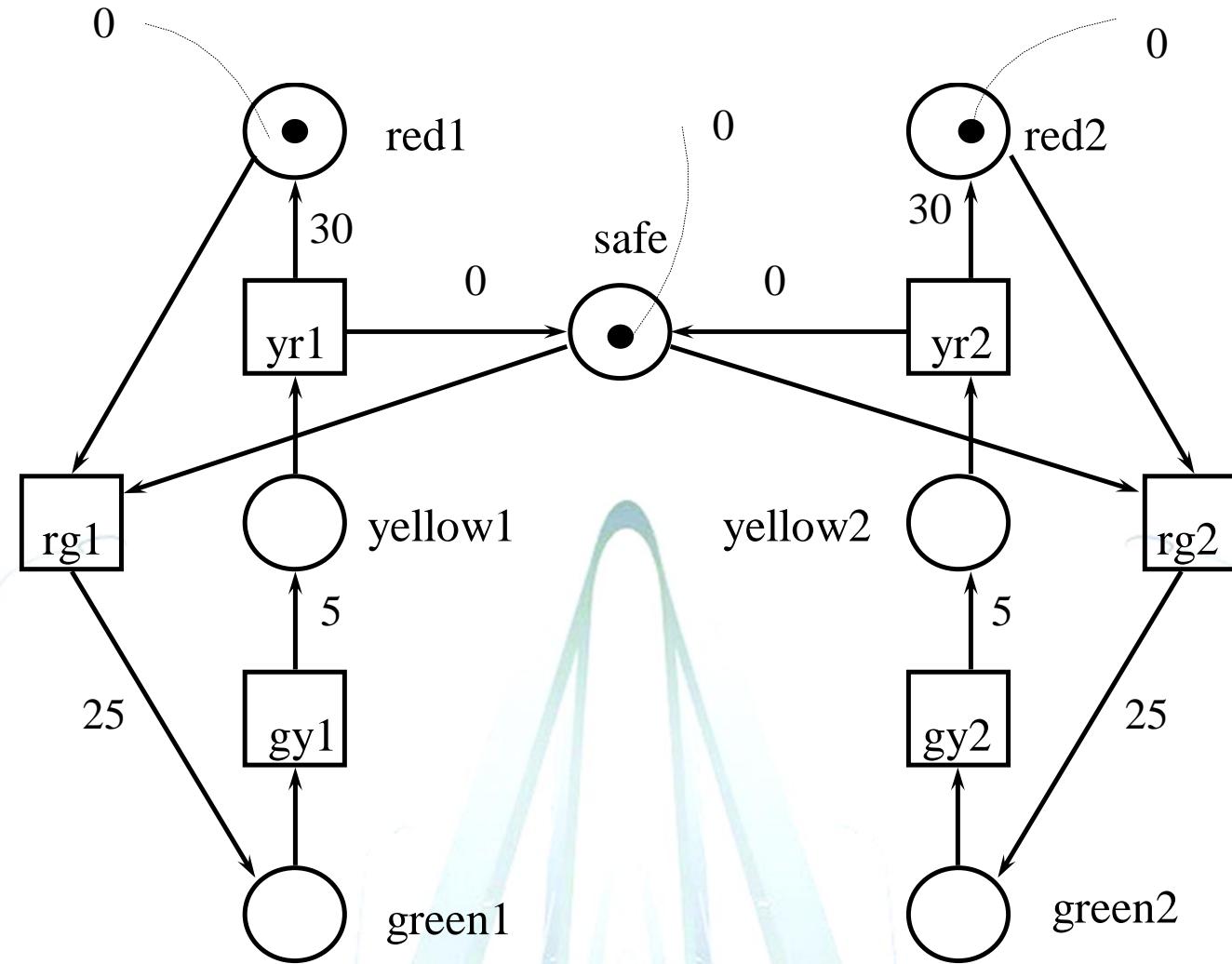
时间的扩展

- 为了进行**性能分析**，需要对**持续时间**，**延迟**等的时间概念进行建模
- 因此，每一个**托肯**都有一个**时间戳**，而**转移**确定了产生一个**托肯**的**延迟**





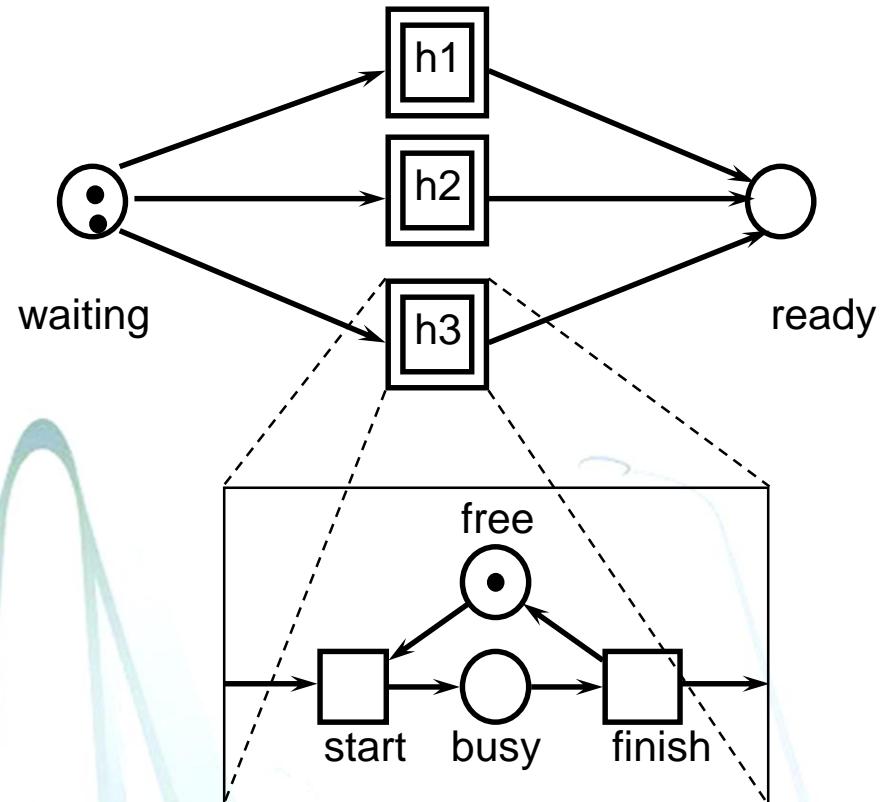
包含时间属性的交通灯





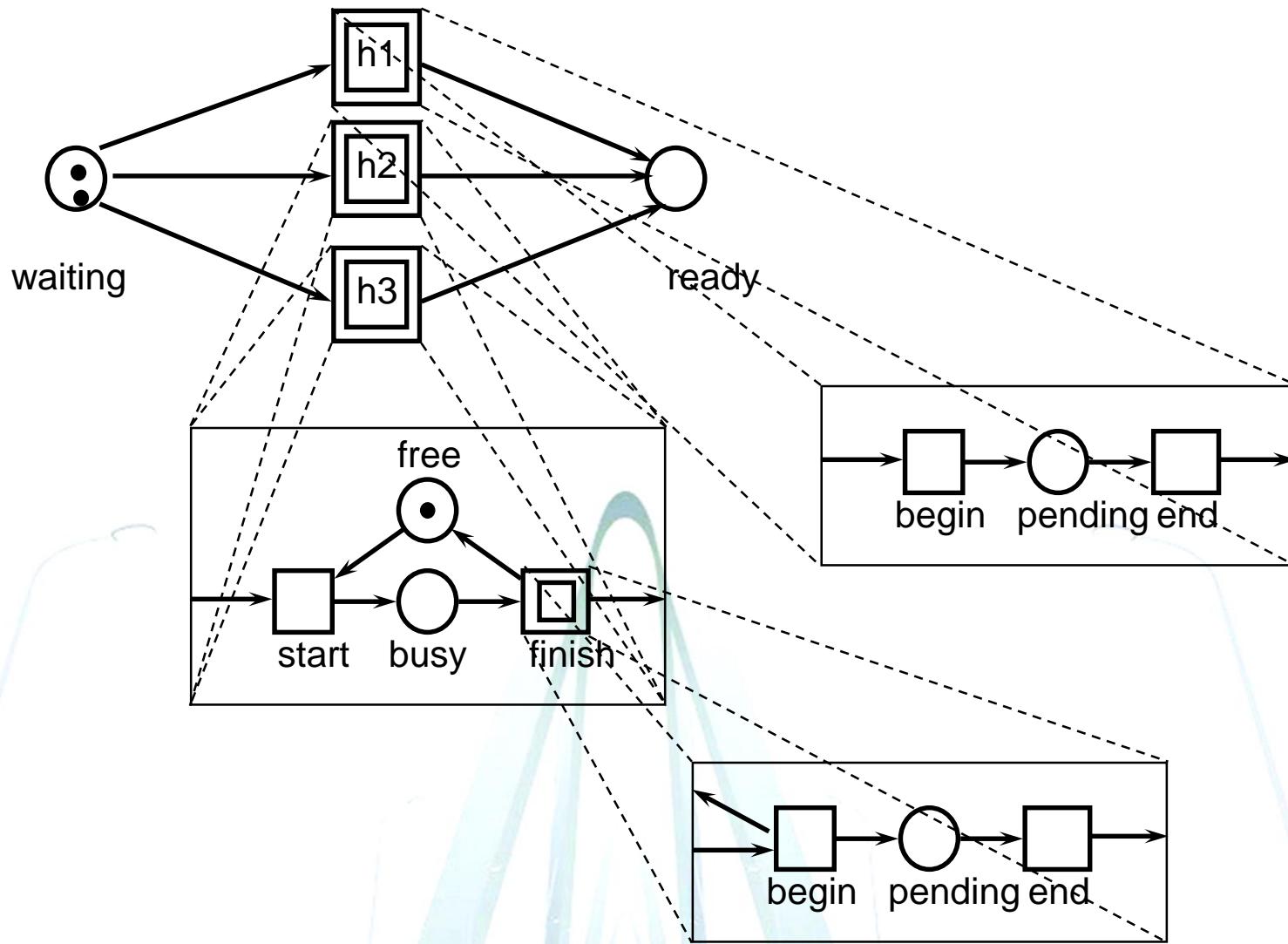
层次的扩展

- 对复杂的Petri网添加结构信息的方法，与DFD类似
- 一个子网是对库所，转移和子网的扩展





练习：把层次去掉





agenda

1. Petri网概述
2. Petri网理论
 2. 1 经典Petri网
 2. 2 Petri网的形式化
 2. 3 高阶Petri网
3. 基于Petri网的过程建模
4. 建模实例及讨论
5. 小结



3 基于Petri网的过程建模

Petri网建立步骤：

- ① 根据状态与事件的定义，确定系统的状态集和转移事件集。
- ② 确定系统中状态与转移的关系。
- ③ 将库所和转移对应起来，建立Petri网模型图。
- ④ 根据系统情况，决定Petri网模型图的初始状态，确定初始状态的下个状态的Token数。
- ⑤ 基于初始状态判断那些事件可被激发，当模型激活后，模型状态图将发生变化，又引起一些事件被触发。

A simple example

电子商务的交易流程可用语言描述为：

- ① 客户通过浏览信息向商家提交订单意向,商家接到提交的订单意向后, 通过查看库存信息形成可供订单;
- ② 对可供订单确认后, 客户输入用于电子支付的信用卡号和密码;
- ③ 得到银行的支付确认后, 商家将可供订单转为有效订单, 同时产生库存信息变更;
- ④ 商家按有效订单配送货物, 并修改库存信息。接着商家进行下次交易的处理。

Petri网建立的模型

转移（事件）可概括为：

T_1 ：支付确认； T_2 ：查看库存； T_3 ：配送货物。

状态（库所）可概括为：

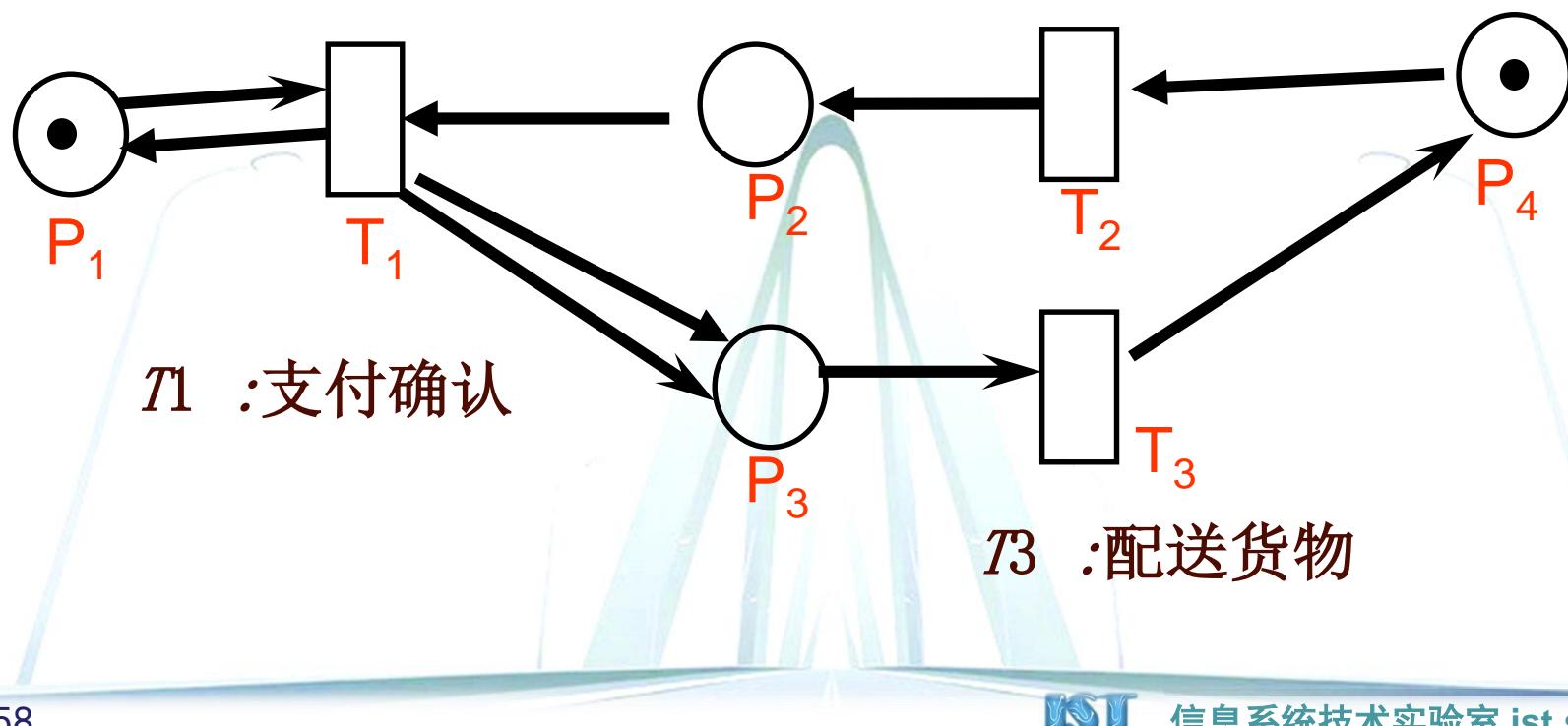
P_1 ：（客户）订单； P_2 ：可供订单； P_3 ：有效订单和库存信息；
 P_4 （商家）库存

| 事件 | 前条件 | 后条件 |
|-------|------------|------------|
| T_1 | P_1, P_2 | P_1, P_3 |
| T_2 | P_4 | P_2 |
| T_3 | P_3 | P_4 |



- 不妨认为这个场景有2个Token，通过这两个Token流转完成流程运作：
 - ① (商品) 库存
 - ② (客户) 订单

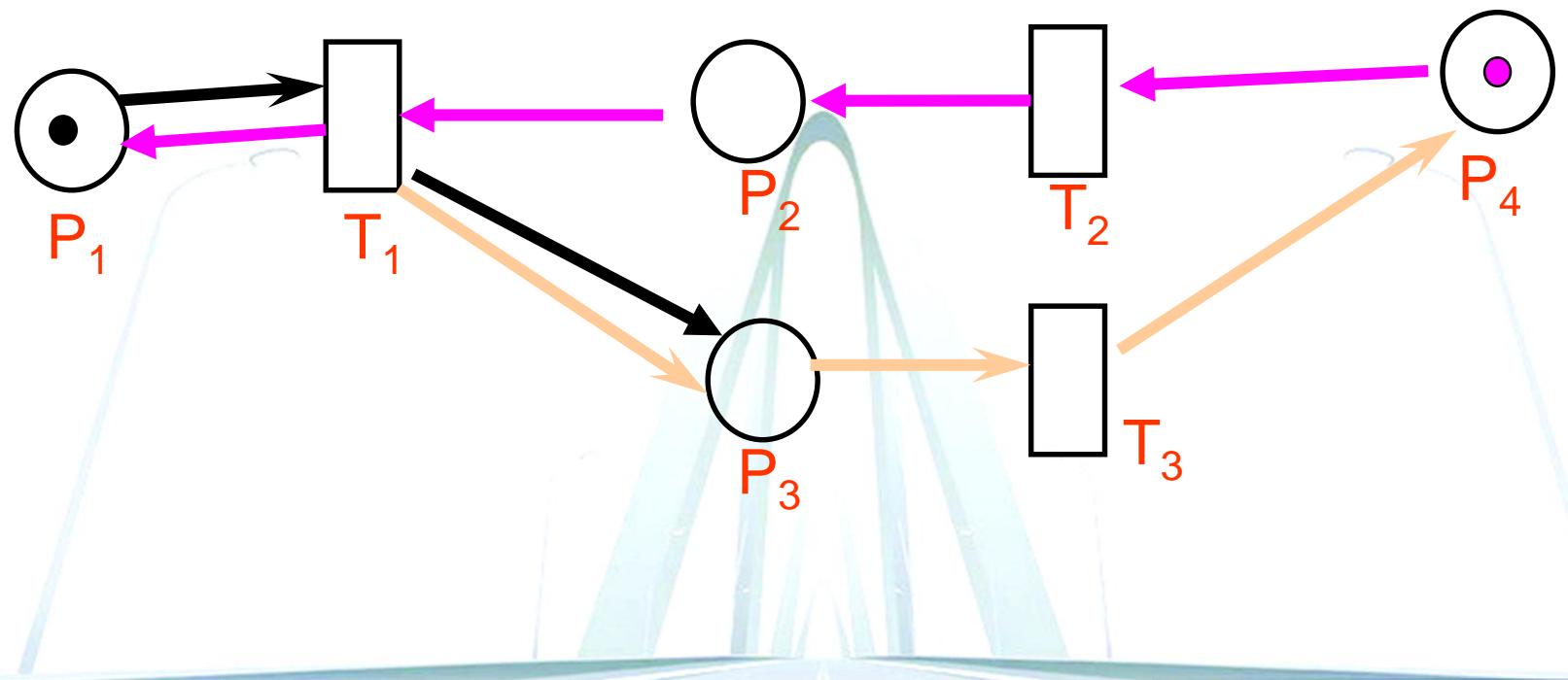
T_2 : 查看库存





Token ① (商品) 库存的流动过程;

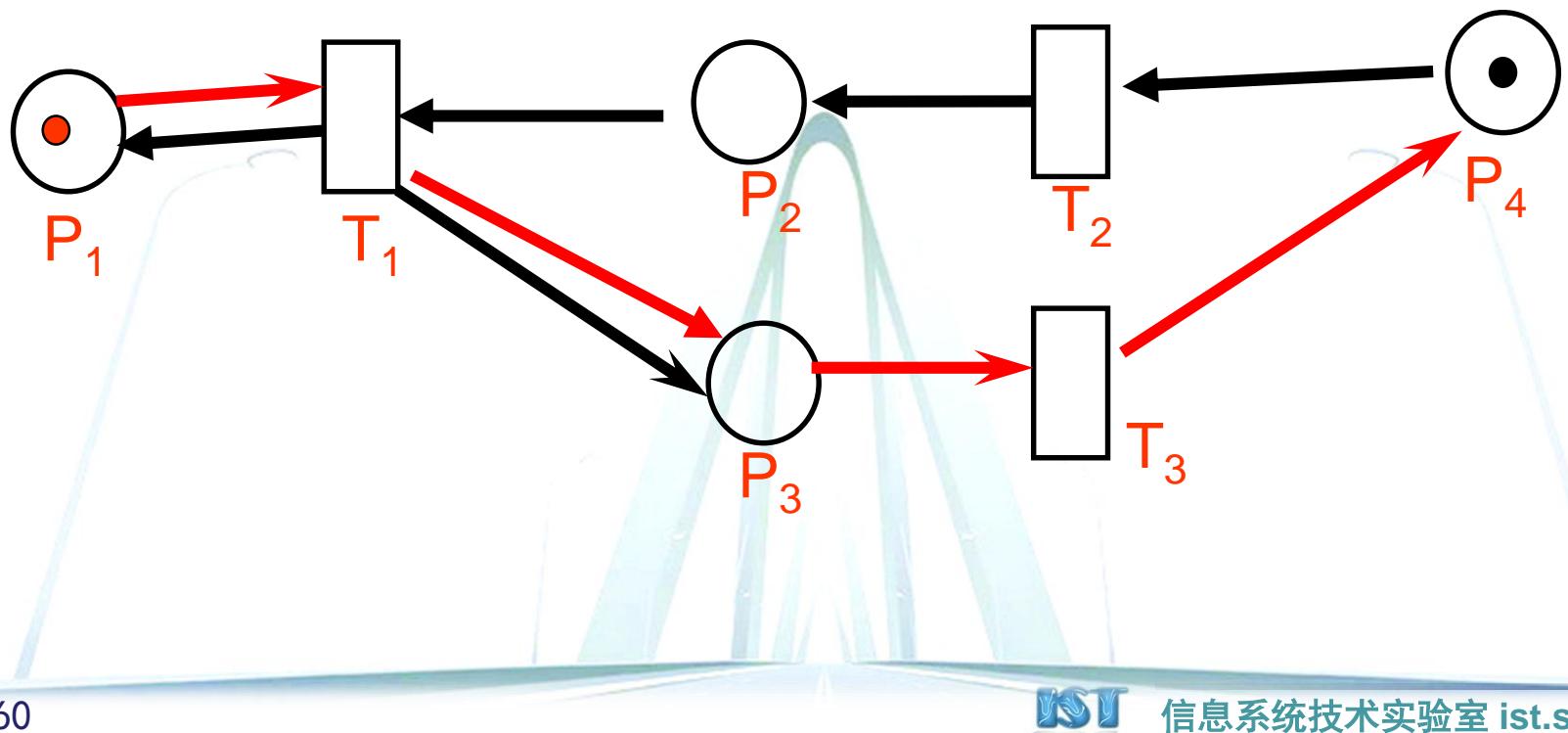
- $P_4 \rightarrow P_2 \rightarrow P_1 \rightarrow P_3 \rightarrow P_4$
- 转移: T_2, T_1, T_3





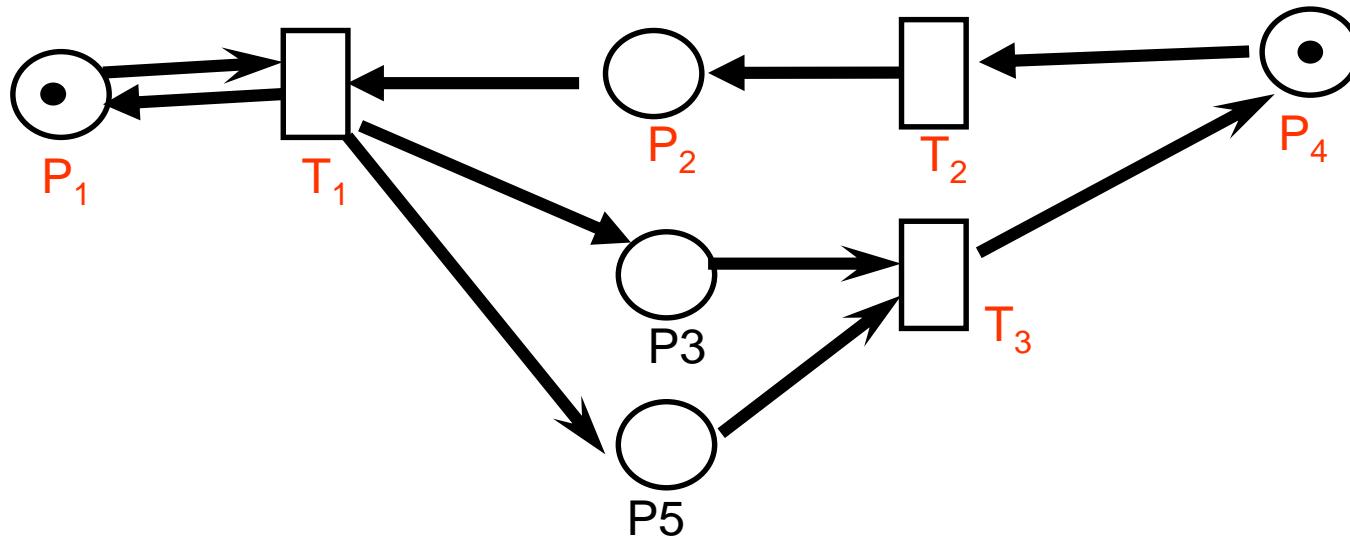
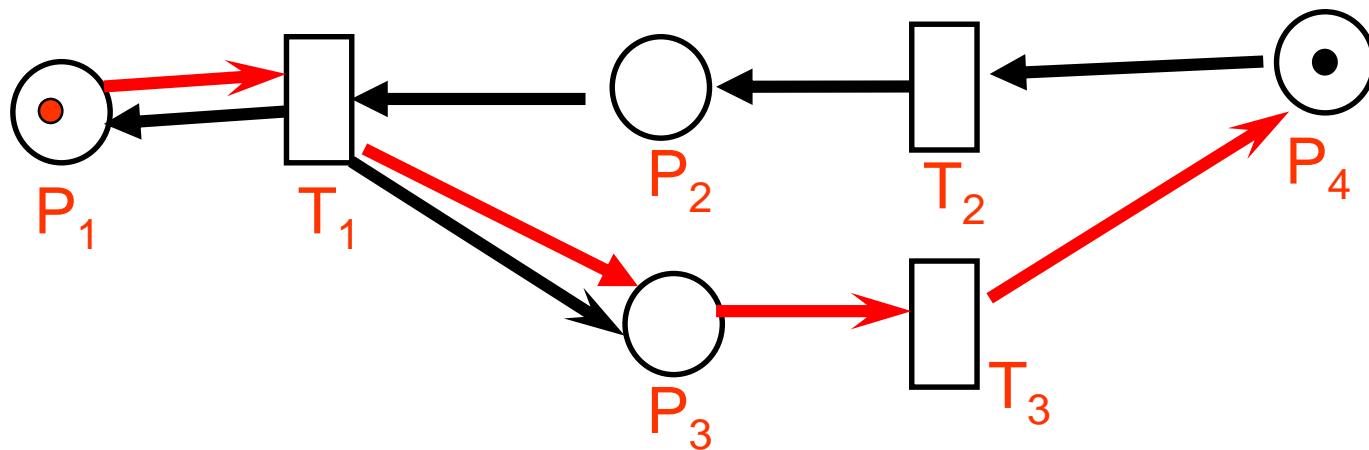
Token ② (客户) 订单的流动过程;

- $P_1 \longrightarrow P_3 \longrightarrow P_4$
- 转移: T_1, T_3





讨论





agenda

1. Petri网概述
2. Petri网理论
 2. 1 经典Petri网
 2. 2 Petri网的形式化
 2. 3 高阶Petri网
3. 基于Petri网的过程建模
4. 建模实例及讨论
5. 小结



5 建模实例及讨论

基于Petri网的仓储物流系统建模与仿真

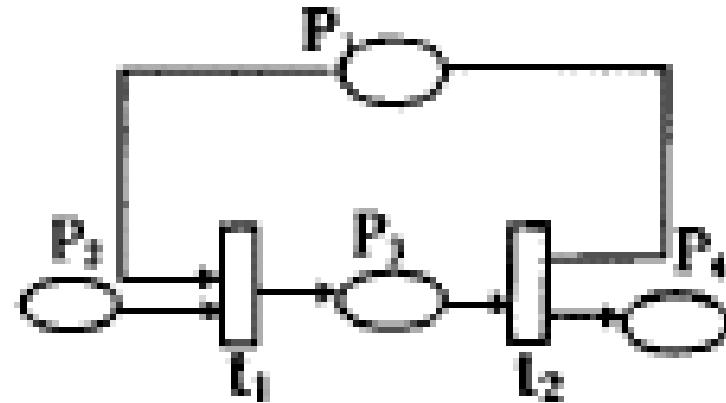
- 一个典型的现代仓储物流系统包含入库台、输送带、AGV 自动导引小车、堆垛机、托盘、货架、空托盘堆放区、分拣区、出库台等。
- 这个物流系统的运作流程主要有出库及入库流程：



- ① 入库流程: 仓库收到入库消息→货物进入库台→到达输送带→在输送的过程中向等待的AGV发出请求→AGV取货→取货的过程中向堆垛机发出请求→堆垛机送货到各个货位.
- ② 出库流程: 按照出库单查询出库货物的具体库位→堆垛机取货→货物进入分拣系统输送带→分拣系统拣货→出库→(空托盘清理).



基本单元符号



P_1 :空闲AGV, 待命出发

P_2 : 缓冲站上货物等待运送

P_3 :AGV取货送往仓库

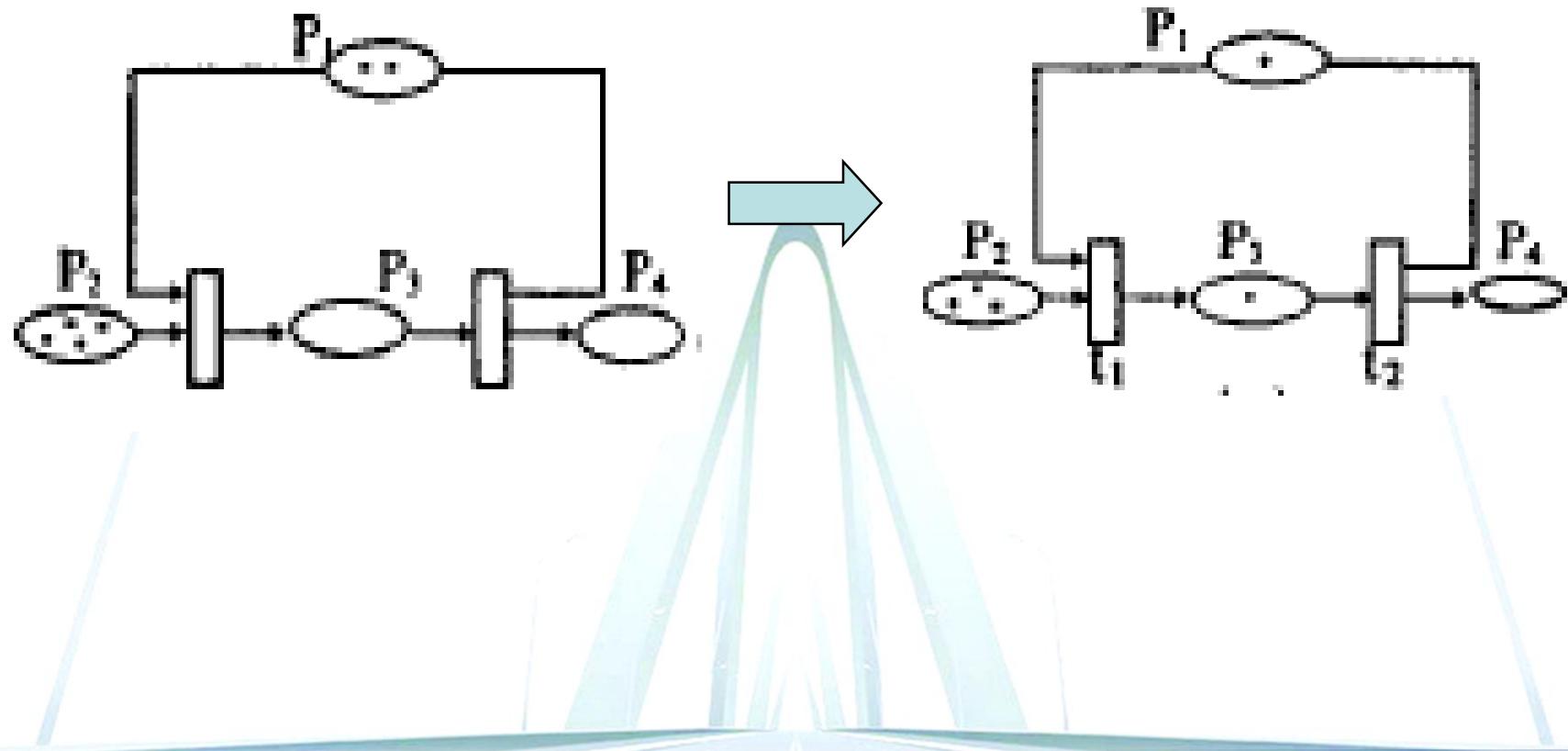
P_4 :货物一卸载在入库台

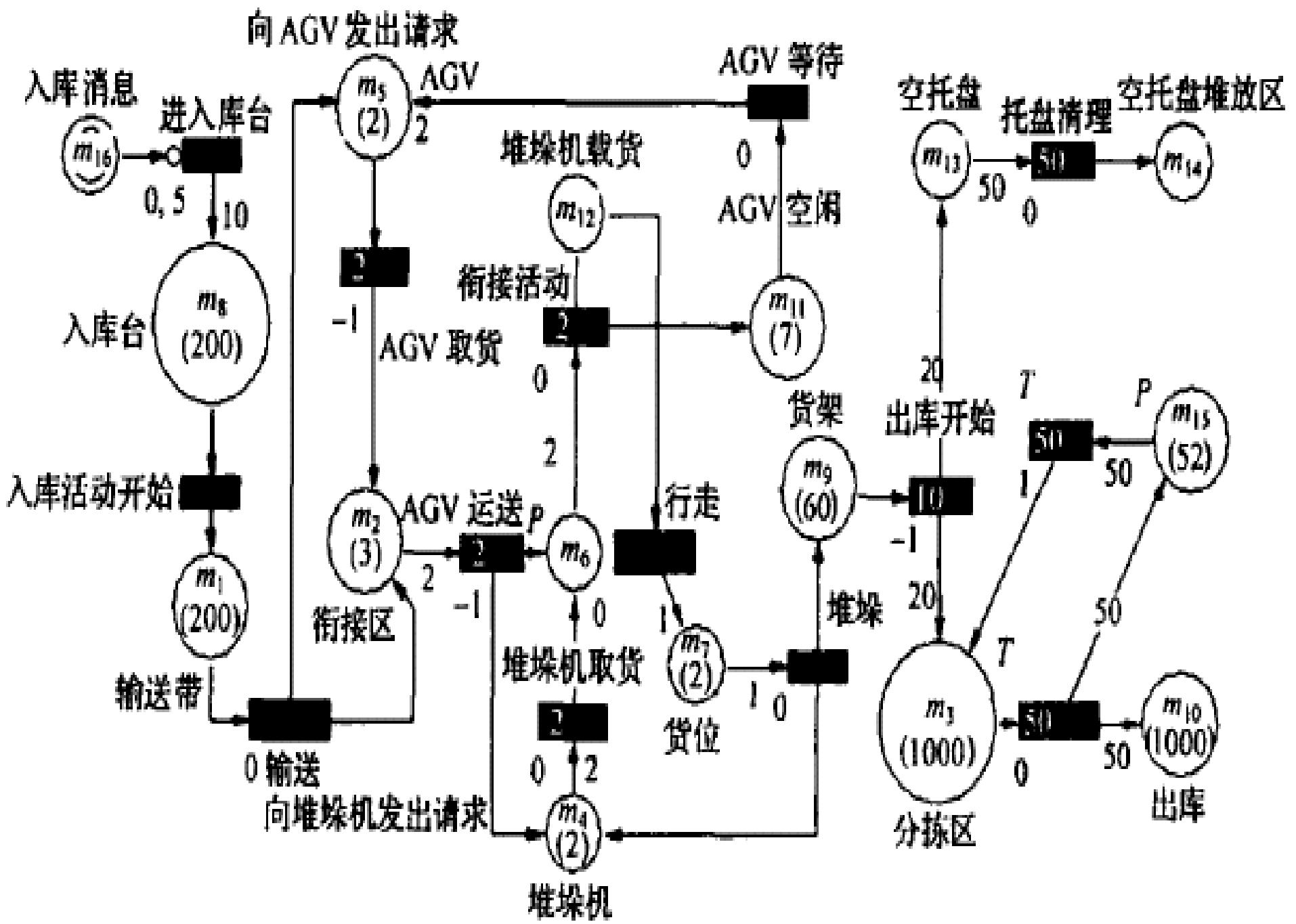
t_1 :AGV取缓冲站上货物事件

t_2 :AGV上货物卸载在入库台事件



- 2辆AGV,4箱成品货物,t1触发，1箱货物转移。





参数

- 其中入库台的容量设为200,AGV 的容量设为2, 衔接区的容量为3, 货架的容量为60, 货架上每个货位的容量为2, 分拣区和出库台的容量均为1000, 输送带为可积放式, 其容量为20。
- 位置的容量可以代表标识数(或token 数)。每条边上的数据均为权数, 未加权数的边其权数都默认为1, 权数为- 1 的边的权数为任意。
- 当每个位置的标识数达到了其输出边的权数, 则T 即事件发生。
- 设货物进入库台的延迟时间为30 s。



- ④ 仿真开始后, 当位置“入库消息”收到了10个token量的货物时, 事件“进入库台”便启动发生, 后续位置“入库台”接收到了10个token量的货物。
- ⑤ 接着后续事件启动, 货物到达“输送带”, token不断传递, 各个后续事件逐一被启动。



- 其中AGV 小车在收到输送请求及空闲AGV 的token 为1 时才触发, 并在衔接区取货, 且同时向空闲堆垛机输出一个token, 在堆垛机取货后, AGV 空车返回空闲处等待。
- 同样, 堆垛机在载货行走至货位、堆垛后返回原处。
- 在分拣区, 可依所需的包装数量进行设置其token、转移的容量。此时设置的转移的容量均为50。
- 空托盘的清理按定量方式进行, 例中, 其容量定为50 个token, 即50 个托盘作一次处理, 处理后在托盘堆放区增加一个token。



- 在仿真运行到20 m in 左右时, 入库台的容量达到饱和。入库台不再按设置的延迟时间每隔30s 收10 个token 的货物, 而是在相差大约15 个token的货物时, 自动传递10 个token。
- 当仿真运行到25 m in 12 s 时, 入库台的token数为192 个, 散放空托盘积累了30 个, 空托盘堆放区已有9 堆(每堆是50 个), 出库货物为450 个token 的量;
- 当仿真运行到50 m in36 s 的时候, 入库台的token数为187 个, 没有散放的空托盘, 空托盘堆放区已有20 堆, 出库货物为950 个托肯的量。



仿真结果

- ④ 基于Petri网的物流系统仿真能够清楚地展现在逻辑时序下整个仓储系统的工作流程及系统特性。
- ④ 从仿真结果数据来看, 入库量与出库量持平, 系统运行良好。
- ④ 然而, 用Petri网仿真, 只能模拟得出流程关系, 而难以模拟出其平面关系, 因此, 它的直接表现性还很不够。



仿真和随机过程

- ④ 仿真给出了流程状态的变化情况；
- ④ 实际过程是任意一个受概率支配的**随机过程**。
- ④ 但状态随着时间而改变，相关研究有一些特定变化的数学模型，如马尔可夫过程、高斯过程等。



agenda

1. Petri网概述
2. Petri网理论
 2. 1 经典Petri网
 2. 2 Petri网的形式化
 2. 3 高阶Petri网
3. 基于Petri网的过程建模
4. 建模实例及讨论
5. 小结



5 小结

- 事件和条件是Petri网的基础，能精确且严格地描述；
- 高阶Petri网扩展了颜色、时间、层次以描述复杂过程；
- Petri网建模是基于**状态**的过程建模方法，对描述并发自动系统非常有用；
- Petri网最大的优点是可视化图形描述却被形式化数学方法支持。

Petri网建模讨论

Petri网建模的缺点：

- ① Petri网组成模型的元素数量过多；
- ② Petri网不如基于活动的图形容易理解；
- ③ Petri网的建模中不能在网中体现数据流，尽管基于状态建模的Petri网能够精确、方便地对过程的控制逻辑进行定义。
 - 大部分情况下，数据流与控制流完全混合；
 - 当两者不一样的时候，Petri网就无法显式地表示这种独立于控制流之外的控制流；



信息系统分析与设计

Lecture 3

工作流建模及分析

蔡鸿明

hmcai@sjtu.edu.cn

软件学院信息系统技术实验室

lst.sjtu.edu.cn





agenda

- ① 1. 基于Petri网的工作流建模
- ② 2. 过程定义
- ③ 3. 资源分类
- ④ 4. 资源分配及管理规则
- ⑤ 5. 流程的分析方法
- ⑥ 6. 小结

1 基于Petri网的工作流建模

工作流从单个案例的动态行为出发，对流程进行控制，工作流即流程的自动化执行，工作流控制是流程自动化运行的关键。

基于Petri网的工作流建模由以下三部分构成：

- **过程定义：**

对过程的描述，建立包括活动执行次序的过程模板。

- **资源分类：**

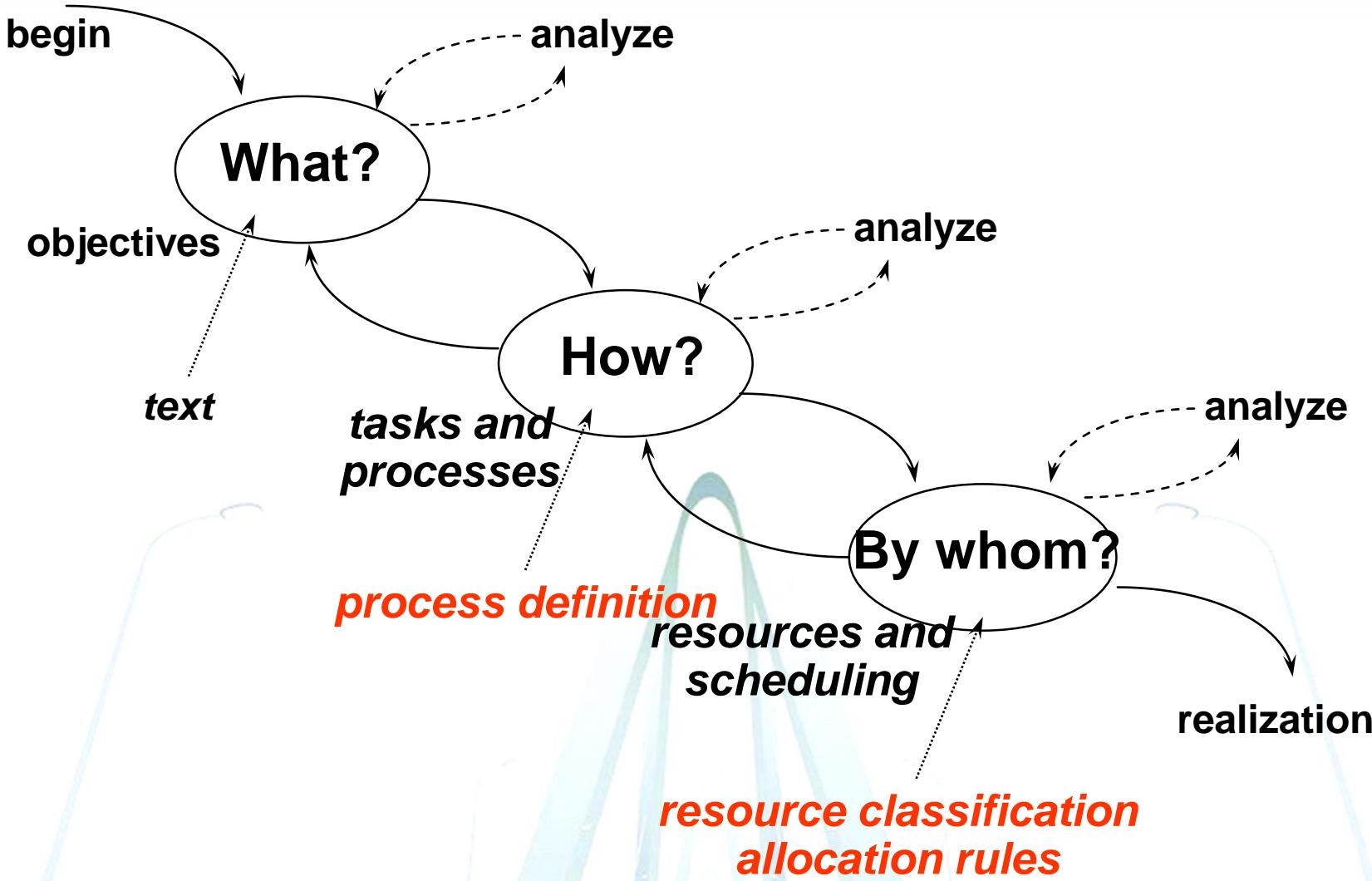
对所使用的资源进行分类，建立信息及人员的结构模型

- **资源分配及管理规则：**

实现资源到工作的映射，构造实际工作流



面向自动化执行的工作流建模





agenda

- ① 1. 基于Petri网的工作流建模
- ② 2. 过程定义
- ③ 3. 资源分类
- ④ 4. 资源分配及管理规则
- ⑤ 5. 流程的分析方法
- ⑥ 6. 小结



2 过程定义

工作流基本概念主要有：

- **过程 (Process)**
描述了案例的执行方式
- **案例 (Case)**
工作流处理的核心内容；
- **任务 (Task)**
工作的逻辑单元
- **启动 (Trigger)**
工作项执行的触发条件
- **路由 (Router)**
根据过程决定的任务执行的分支及次序



2.1 过程 (Process)

- 过程定义描述了所需的步骤和它们执行的顺序。
(路由定义, 步骤, 工作流脚本)
(销售订单, 报税, 保险赔偿申报)

- 过程定义有以下组成：
 - **任务** (步骤, 活动, 过程元素)
任务是原子的：提交或者回滚.
 - **条件** (状态, 阶段, 需求)
条件被用于确定某一任务是否可以被激活.
 - **子过程**
过程的分解



- ⑤ 过程可能有多个子过程构成，每个子过程又由其他的任务、条件以及可能更细致的子过程组成。
- ⑥ 案例的执行方式由相关过程来描述，可以把过程看作是某种案例类型的程序；
- ⑦ 过程定义了案例的生命周期，因此每个过程都有开始和结束，用来标识案例的出现和完成；



2. 2 案例

- ◎ 案例是依据过程定义所需要完成的具体事情
(被称为：过程实例，工作，项目)
(例如：销售订单，报税，保险赔偿申报)

- ◎ 案例的状态由下列方面决定：
 - 实例变量（实例的参数）
实例的逻辑参数，被用于实例过程的路由.
 - 条件
被满足的需求
 - 内容(应用数据)
可能由其它系统管理

案例 (Case)

- 有限的生命周期，对应一个工作流的开始和结束；
- 案例在某生命周期内的某个状态，有三个元素：
 - 相关属性的值；可随案例进展而发生变化；
 - 已经满足的条件；说明案例进展；
 - 案例的内容；工作流系统通常不保存案例内容和细节；



2.3 任务

一般的逻辑工作单元，而不是过程实例的一次执行；

工作的逻辑单元，不可分割且必须完整执行；

任务被分为：

- 手动的任务 (manual)；
- 自动的任务 (automatic)；
- 半自动的任务 (semi-automatic)

注意下面的概念的区别：

• 任务 A logical step which may be executed for many cases.

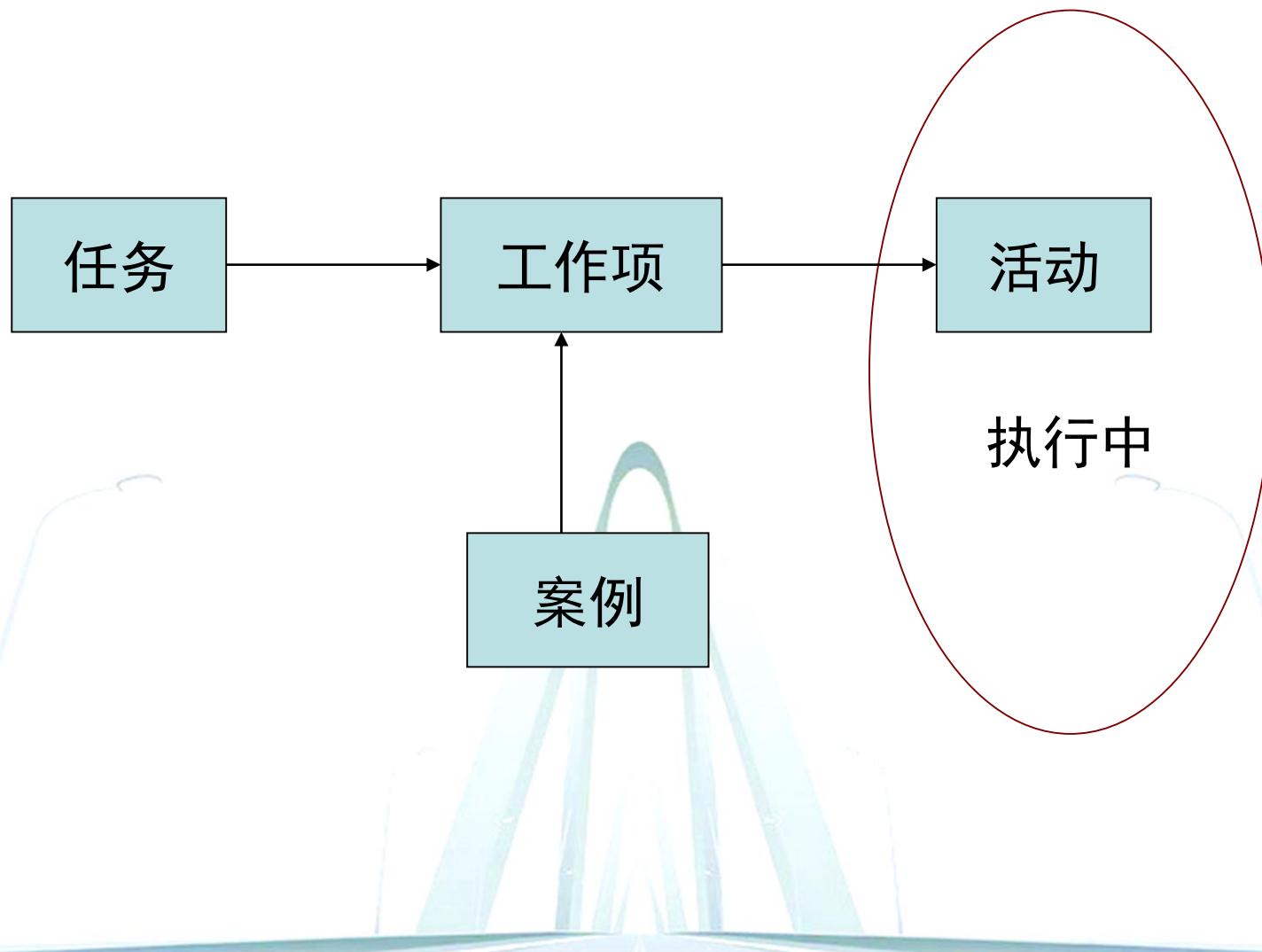
• 工作项 = task + case

A logical step which may be executed for a specific case.

• 活动 = task + case + (resource) + (trigger)

The actual execution of a task for a specific case.

工作项和活动是任务的实例





2. 4 触发条件 (Trigger)

- ◎ 在现实中，系统不能强迫事情发生：
 - 为了执行任务，需要EDI消息的到达.
 - 执行某一实例的任务所需要的资源.
 - 纸面文件的到达.
 - 确认购买订单的电话
- ◎ 是一个响应式系统，也就是说它被环境触发
- ◎ 某些任务需要触发条件
- ◎ Trigger是工作项执行的触发条件；



四种类型的任务及其表示：

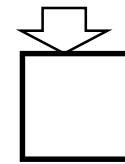
- **自动**

不需要任务触发



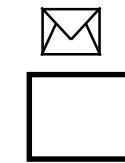
- **人工：** 用户操作

需要资源作为启动



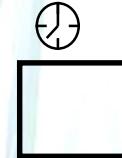
- **消息：** 外部触发

需要外部的事件（如消息，电话）



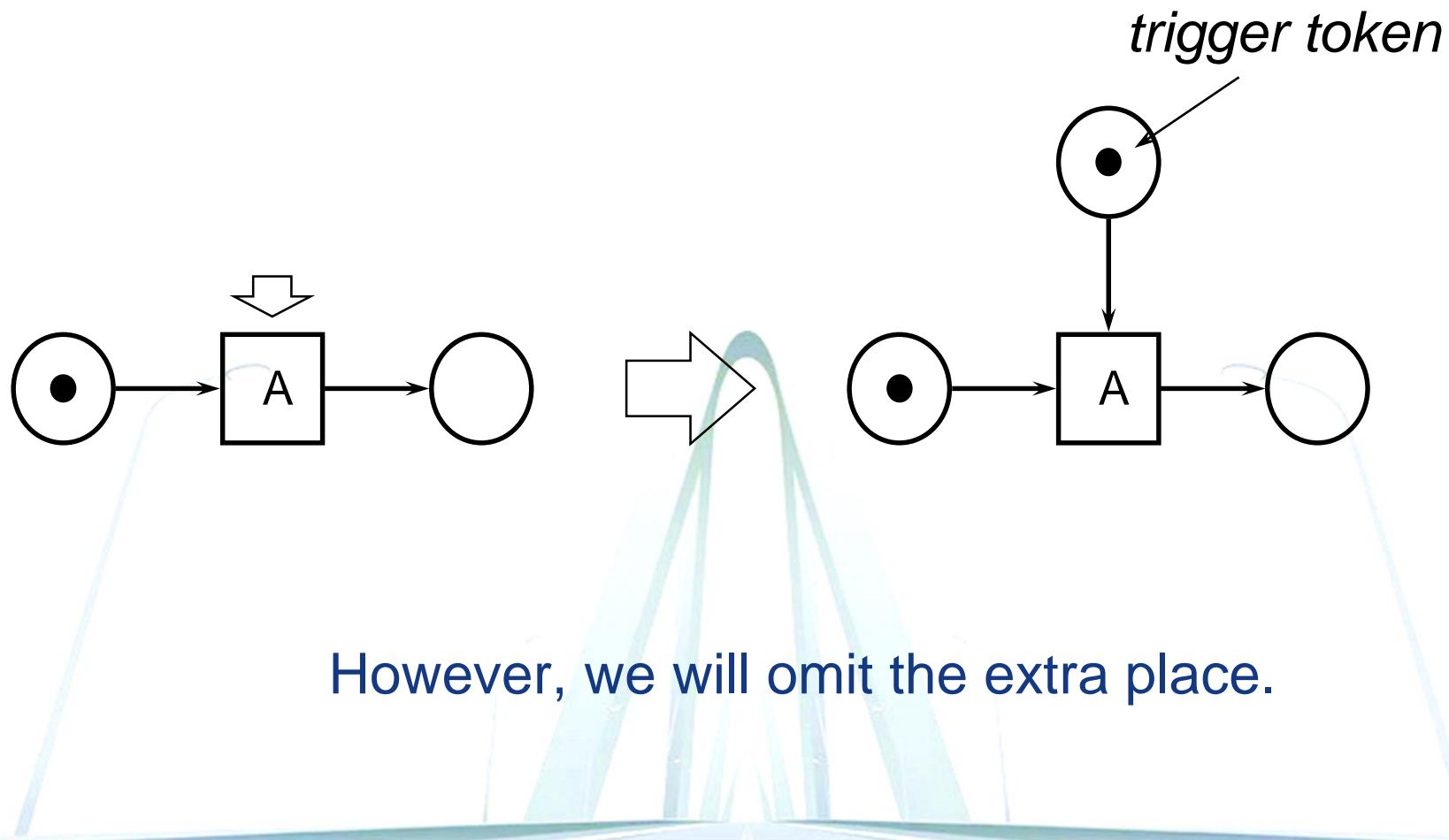
- **时间**

任务需要时间触发





触发的Petri网表示





2. 5 路由 (Router)

- 路由决定了哪些任务需要被执行或以何种次序执行；
- 有四种路由的基本结构：
 - 顺序执行；
 - 并行路由：两个任务能同时或以次序持续执行。AND-split, AND-join
 - 选择路由:OR-split, OR-join；
 - 循环路由：



顺序

"first A then B"

并行

"A and B at the same time or in any order"

- AND-split
- AND-join

选择

"A or B"

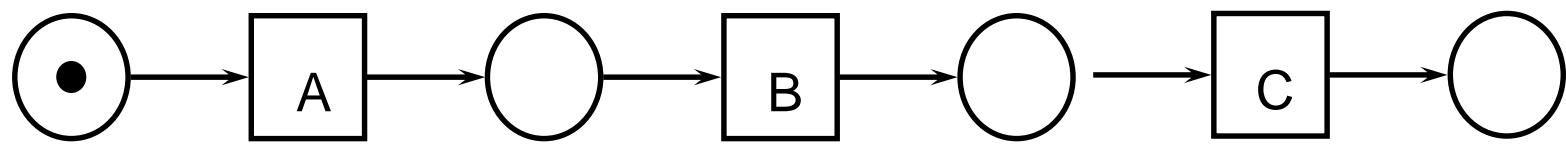
- OR-split
- OR-join

迭代

"multiple A's"



(1) 顺序路由

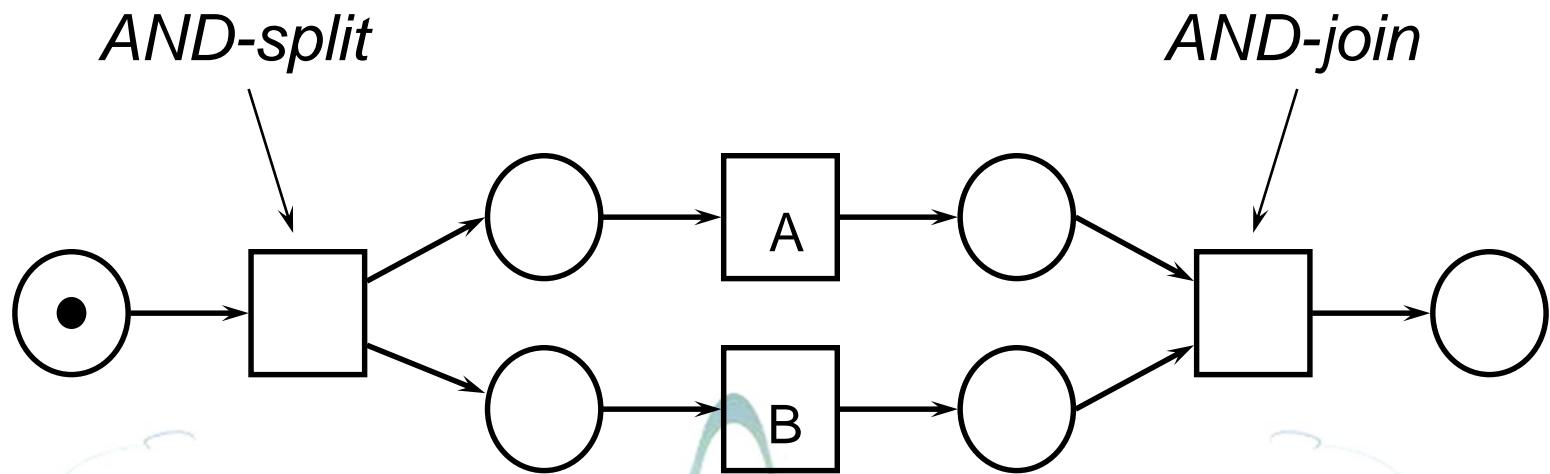


用来定义一系列按固定顺序串行执行的活动。

”先 A， 接着B， 然后C”



(2) 并行路由



定义没有严格执行顺序的，可同时进行的分支活动；

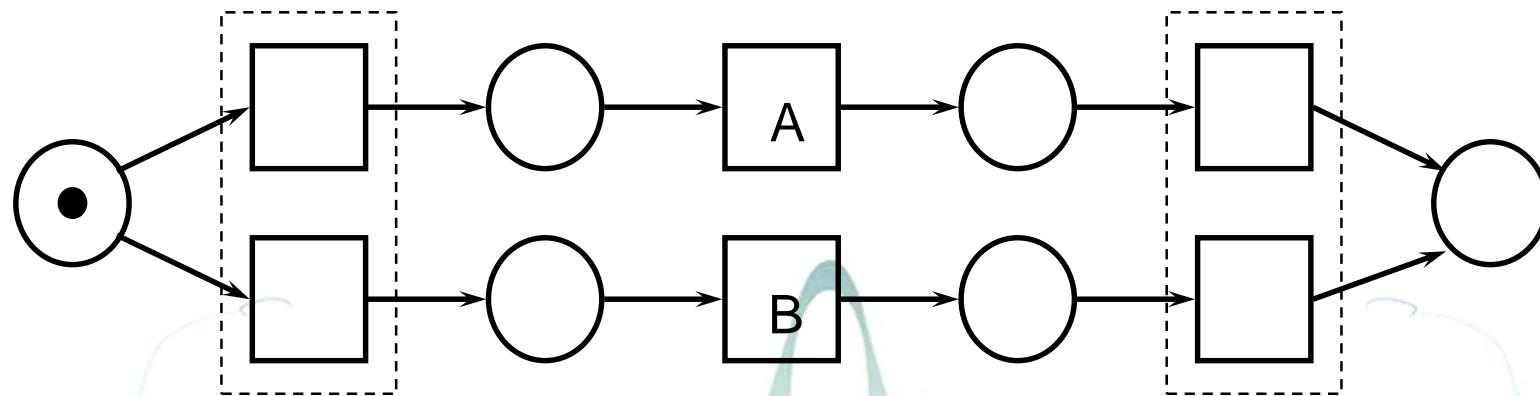
"A and B at the same time or in any order"



(3) 选择

OR-split

OR-join



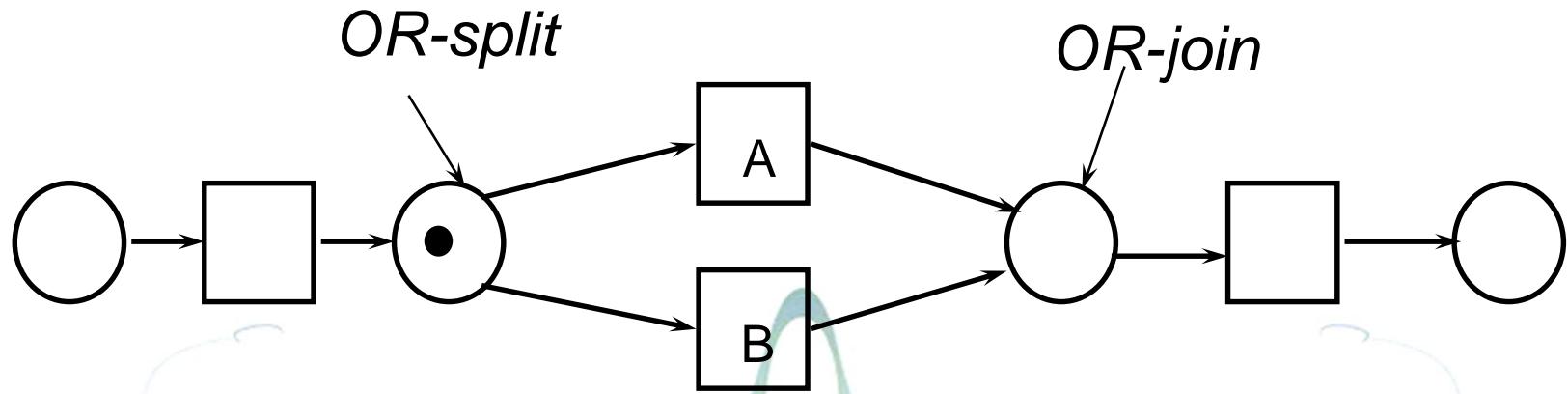
定义彼此之间具有相互制约与排斥关系的分支活动，往往根据具体执行情况来从中进行“多选一”或“多选多”。

"A or B"



隐性或分支

资源竞争方式达到选择的实现

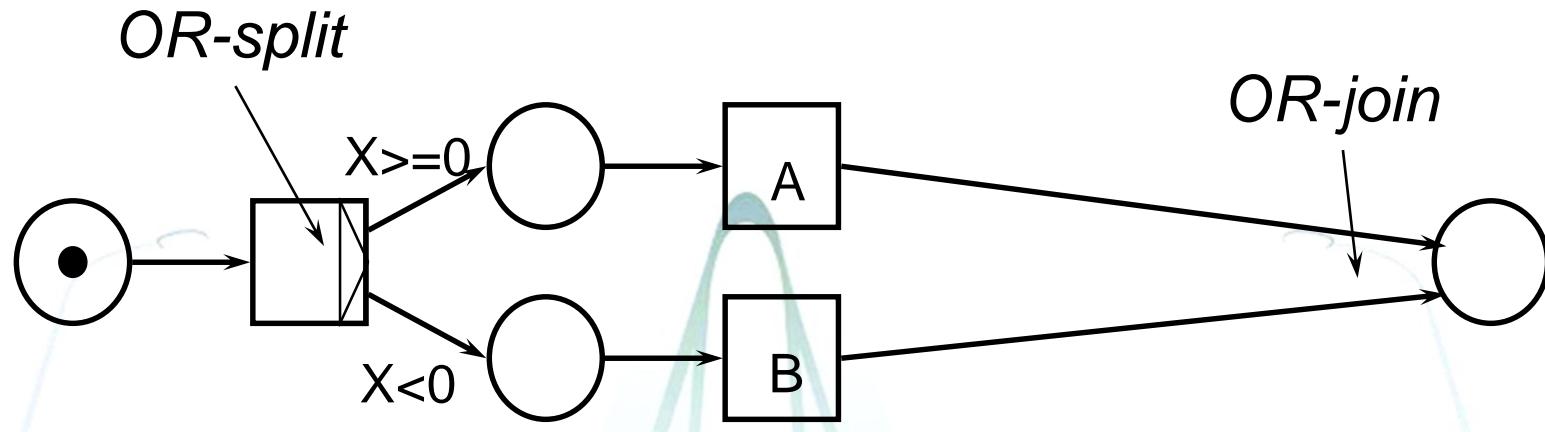


Implicit choice: it depends on the
“eagerness” of A and B!



显性或分支

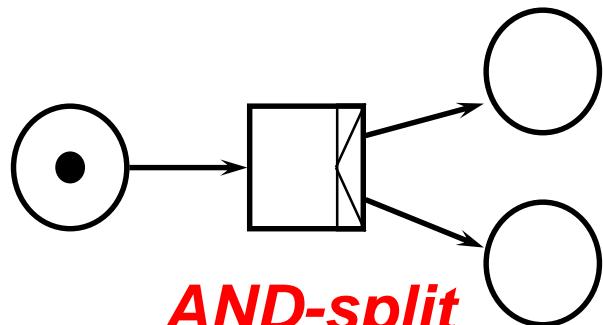
explicit OR-split



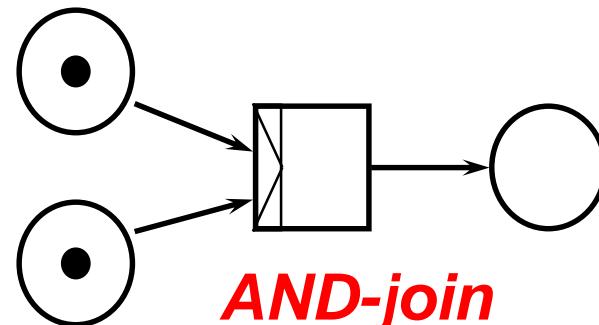
选择是显式的，可以基于逻辑值



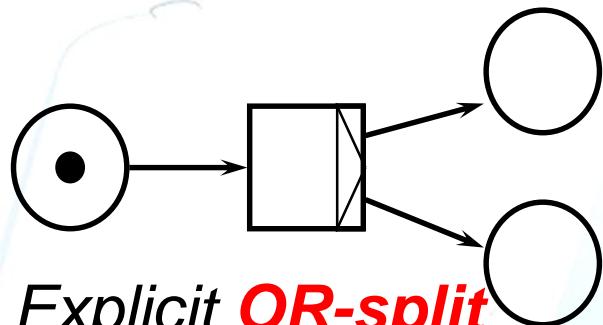
基于Petri网的工作流模型基本语法



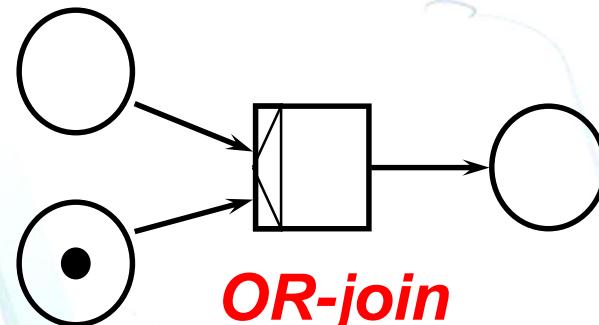
AND-split



AND-join



Explicit OR-split

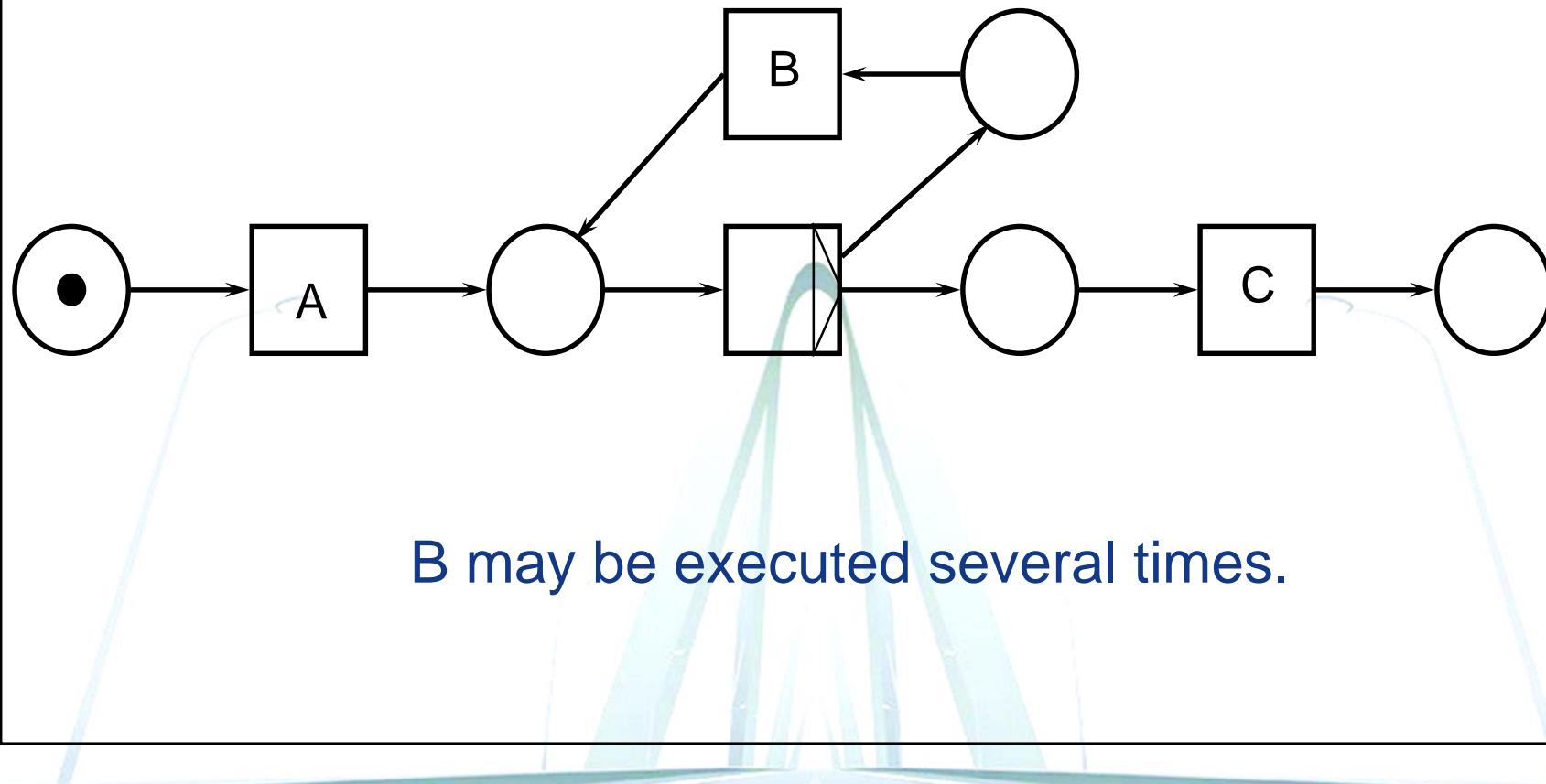


OR-join



(4) 迭代

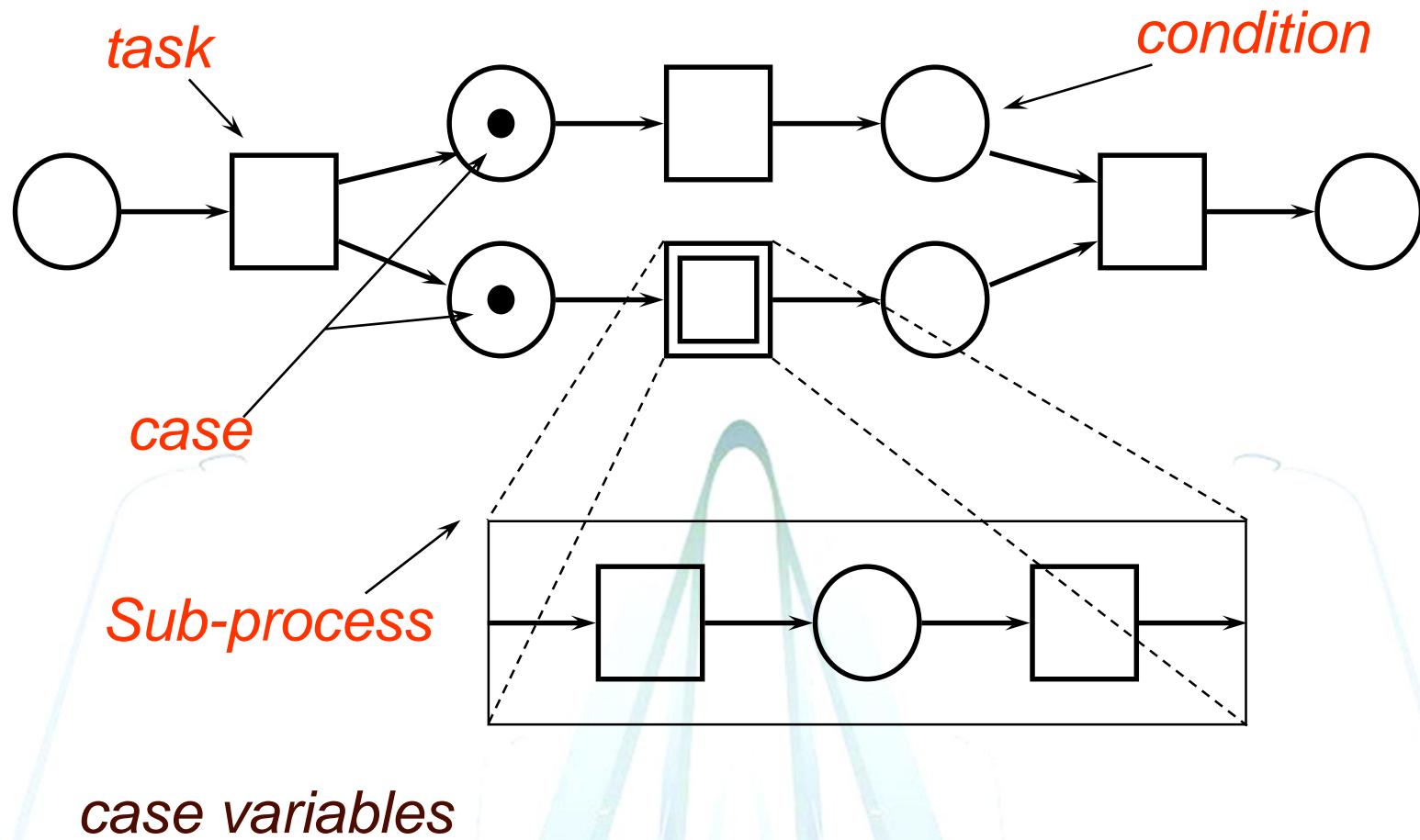
定义需要迭代或重复执行多次的活动。



B may be executed several times.

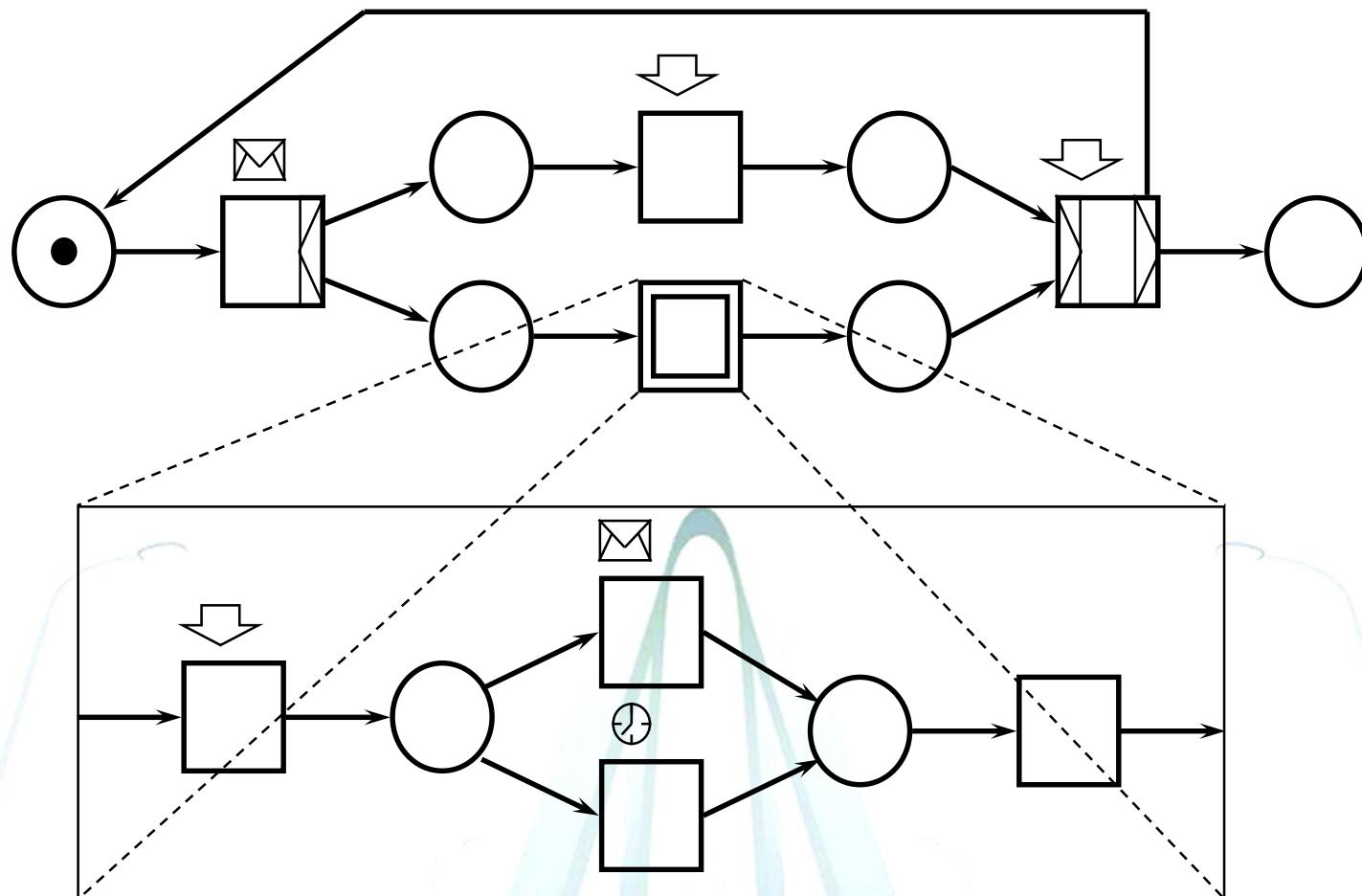


过程定义映射到Petr i网





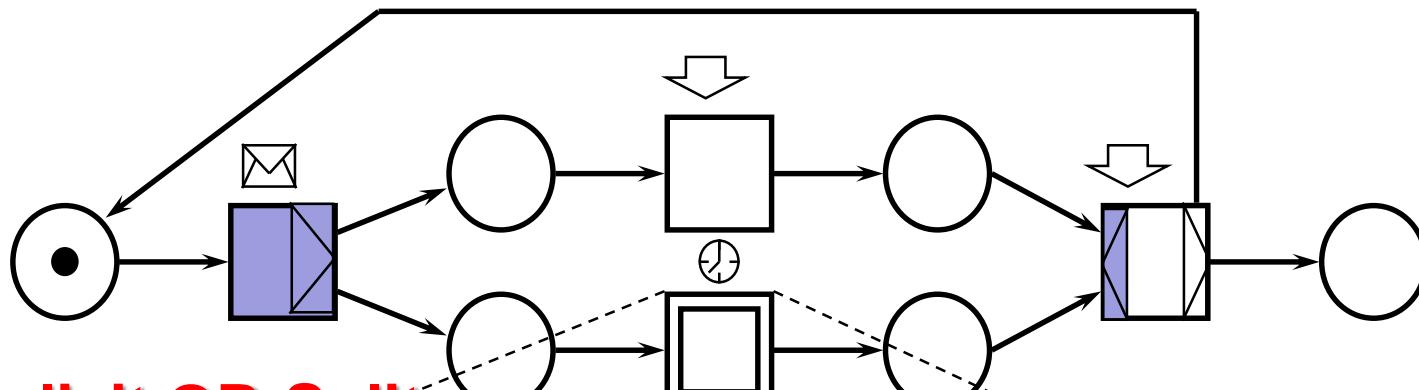
某一过程的定义



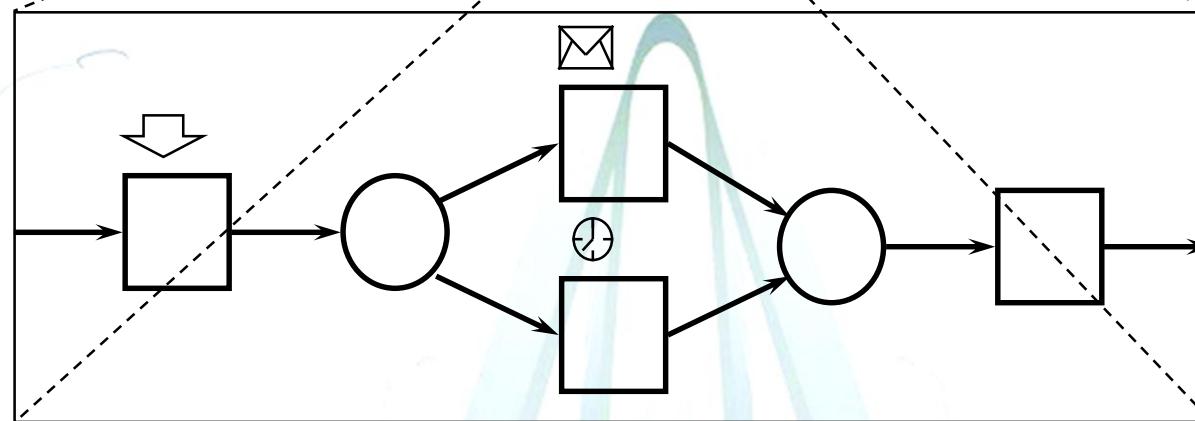
A (sub) process has one input and one output place.



OR-split 比较



Explicit OR-Split



Implicit OR-Split



agenda

- ① 1. 基于Petri网的工作流建模
- ② 2. 过程定义
- ③ 3. 资源分类
- ④ 4. 资源分配及管理规则
- ⑤ 5. 流程的分析方法
- ⑥ 6. 小结



3 资源分类

- 业务定义确定了业务执行的顺序，但没有确定每个任务由谁来做。
 - 资源（人/机器）是工作项的执行的基础。

- 资源(participant, actor, user, agent)

资源是可以被唯一的确认并具备一定的工作能力；

Human and/or non-human (printer, modem):
limited capacity.

- 资源类

具有共同特征的资源集合，可以完成某项特定的任务。



资源分类

- 资源分类过程即是资源类的概念和属性建模过程；
- 涉及到业务过程中的静态要素建模：
 - 人员建模：
 - 信息对象建模：
 - 基础设施或设备建模：
 - ...
- 资源与Petri网的各类托肯对应：



3. 1 人力资源分类

实际工作流中的资源主要是人力资源；
资源类的划分可以基于职能特征以及组织结构：

◎ 角色

(skill, competence, qualification)

基于能够做什么.

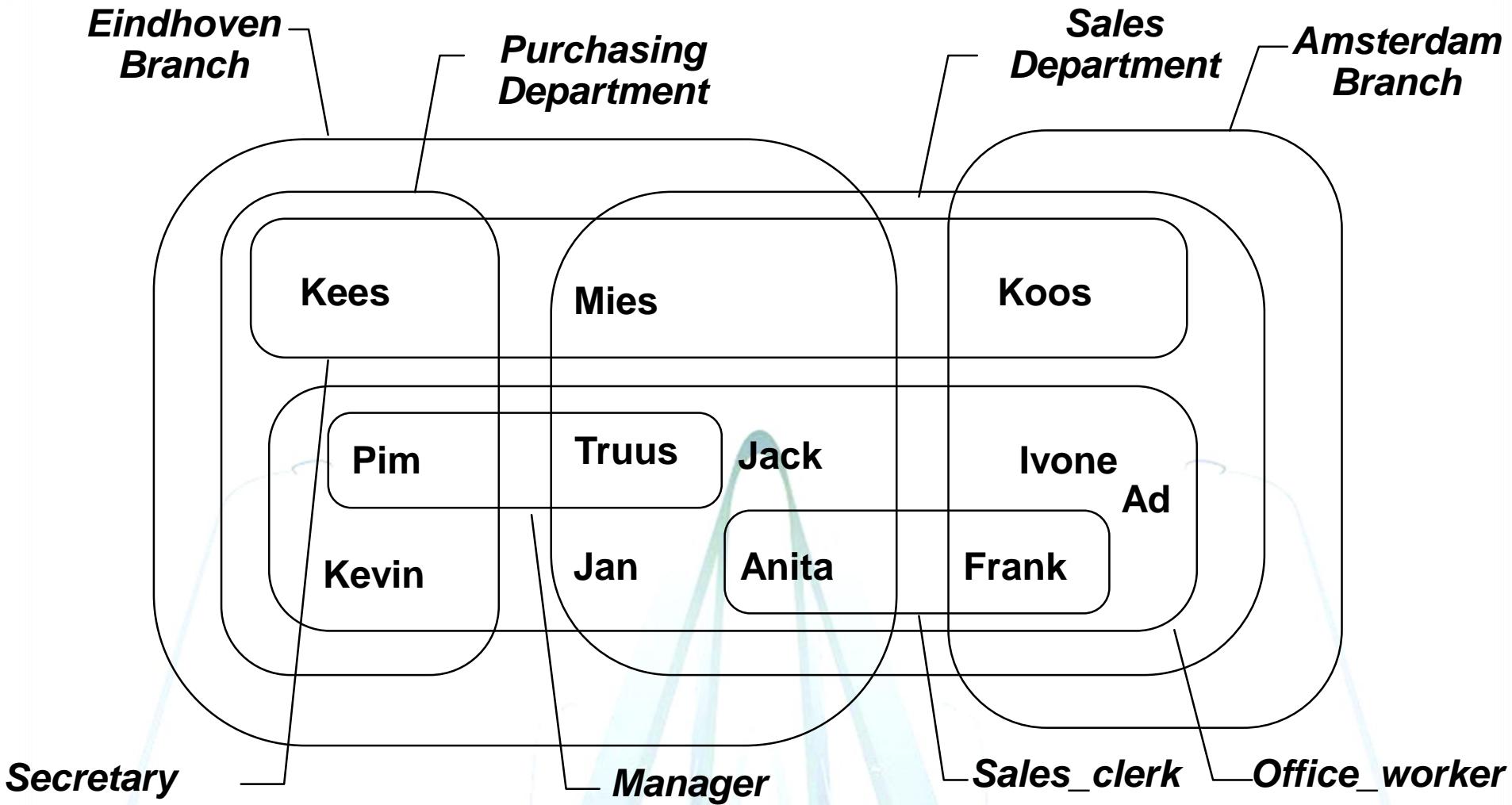
◎ 小组

(department, team, office, organizational unit)

基于组织信息.

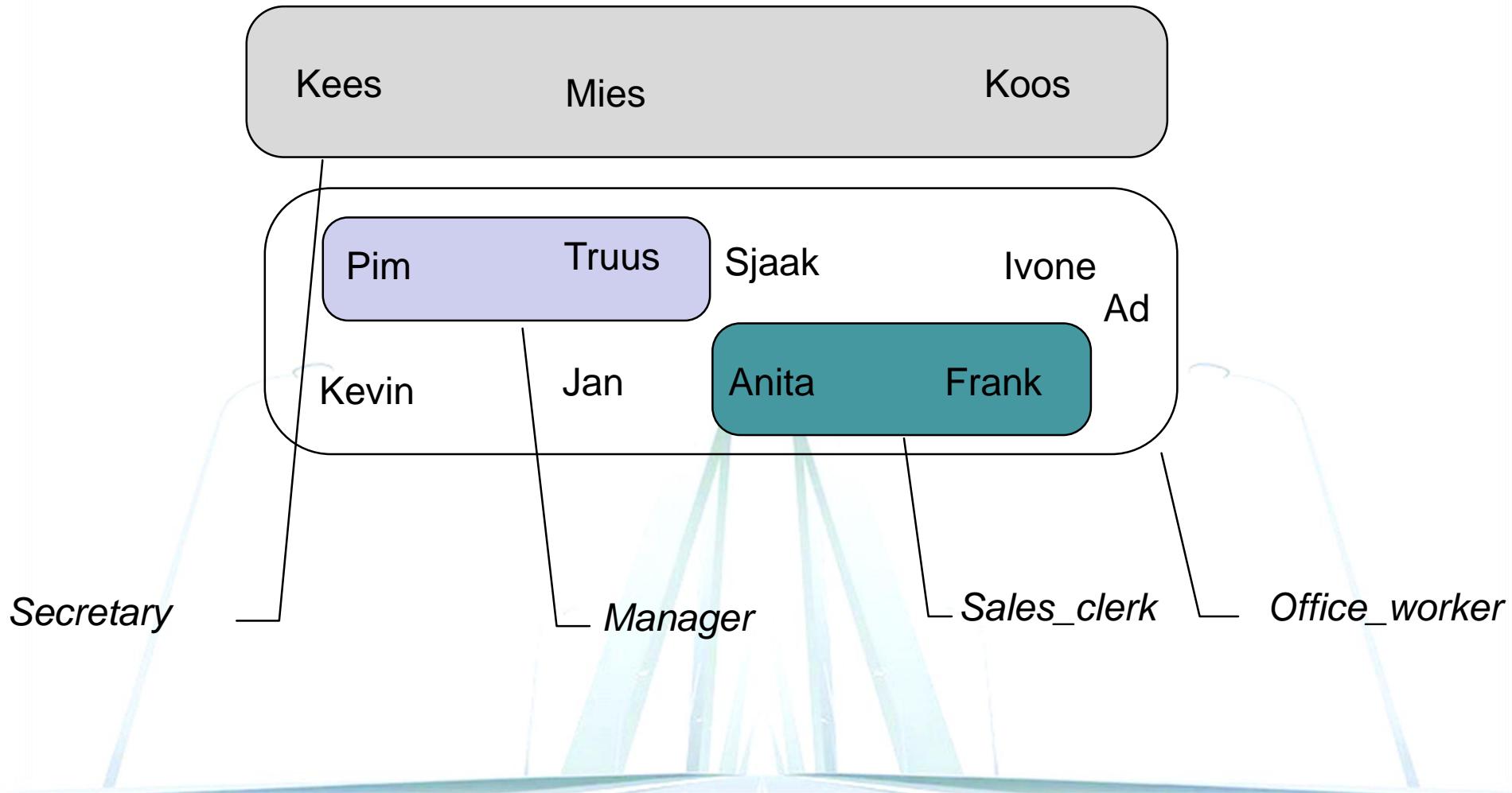


例：资源类别



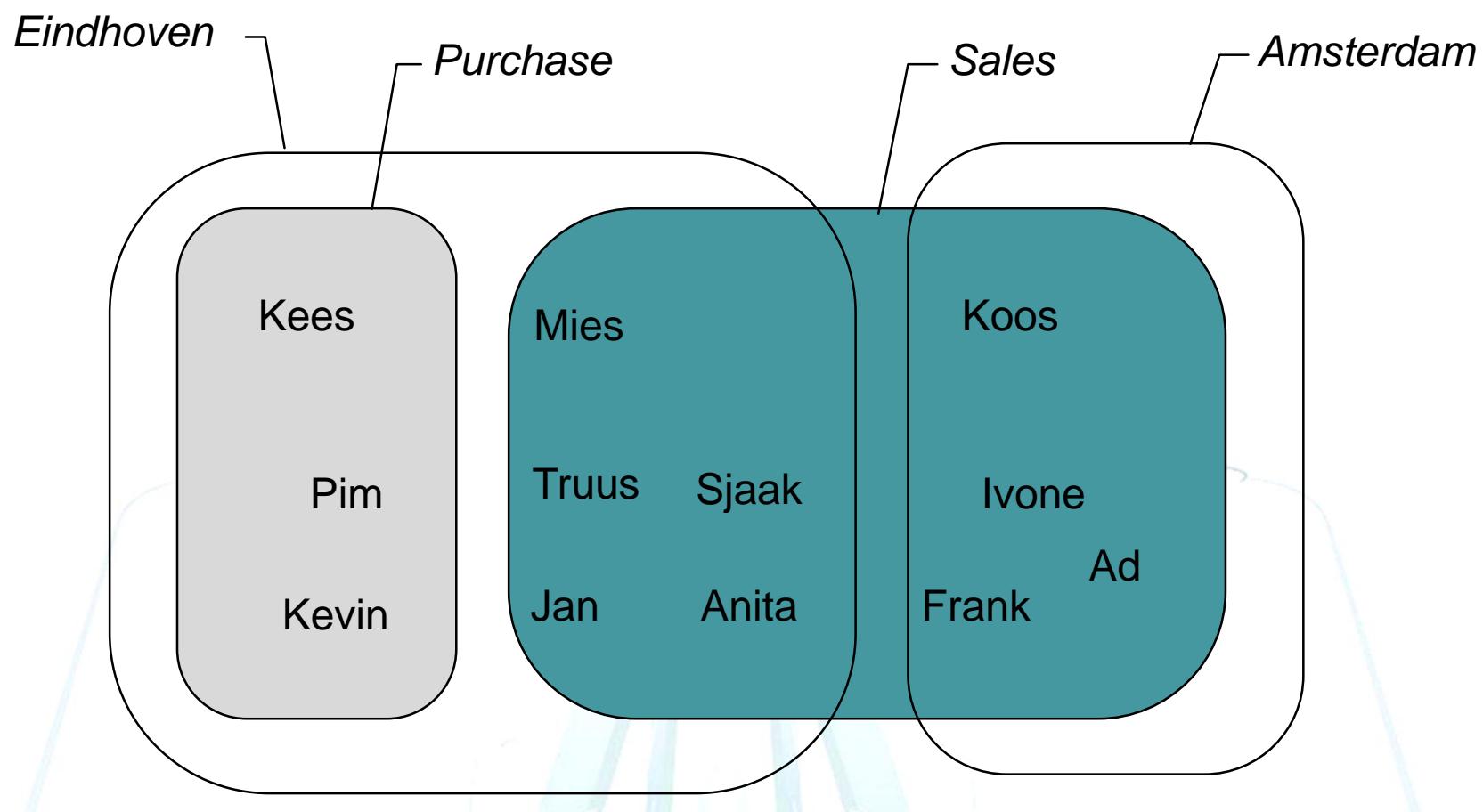


角色



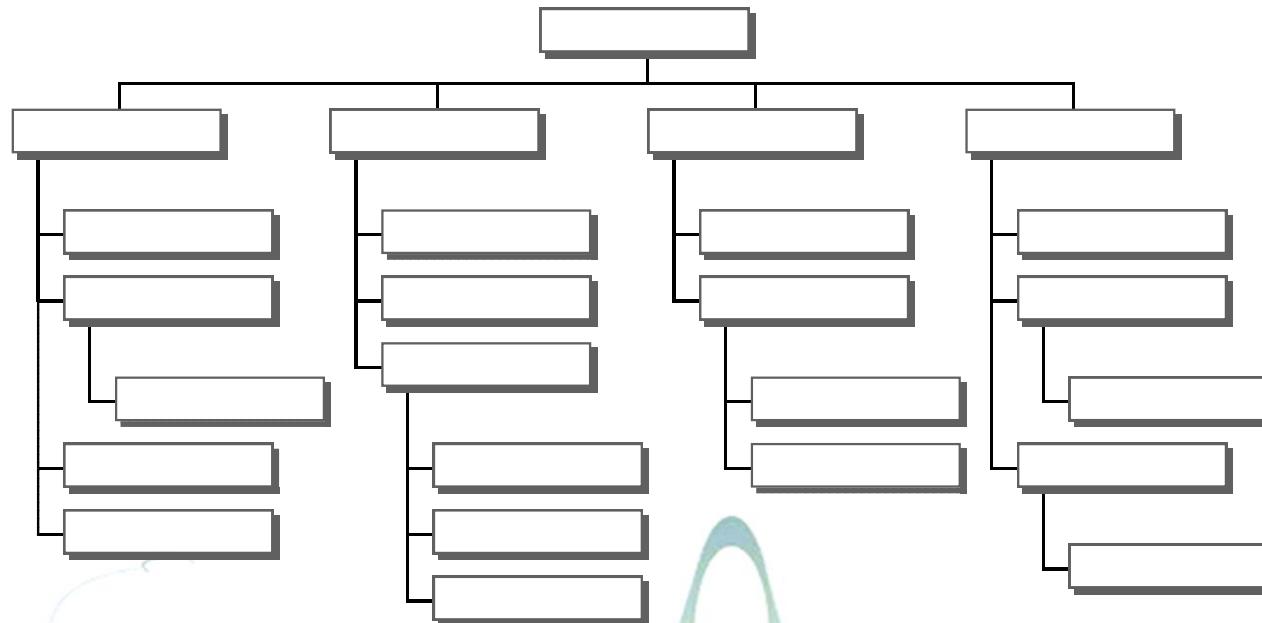


小组

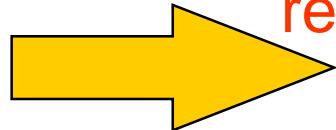




组织信息可以被用于资源分类

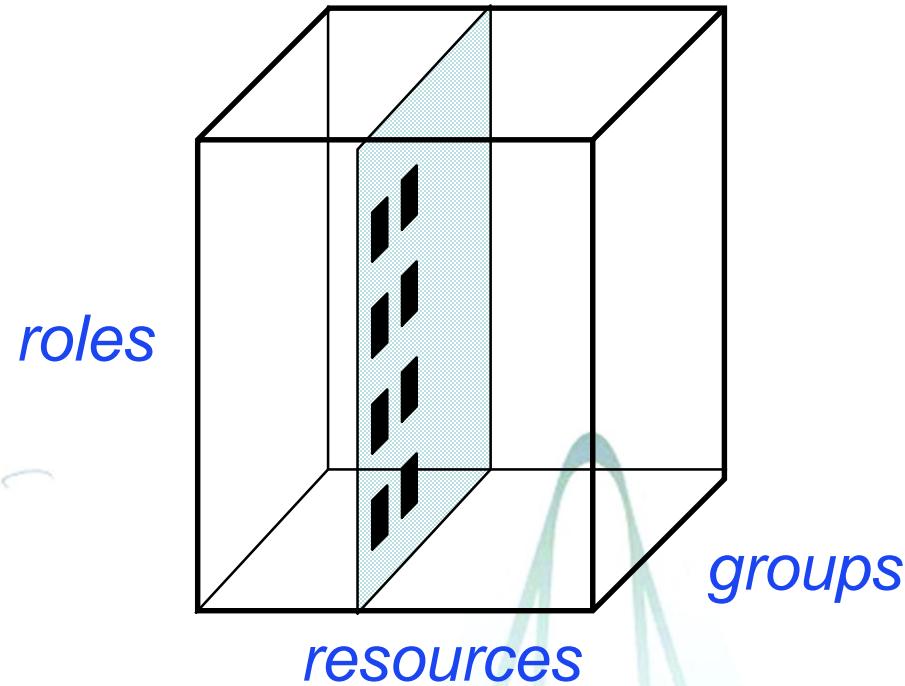


resources distributed over roles and groups



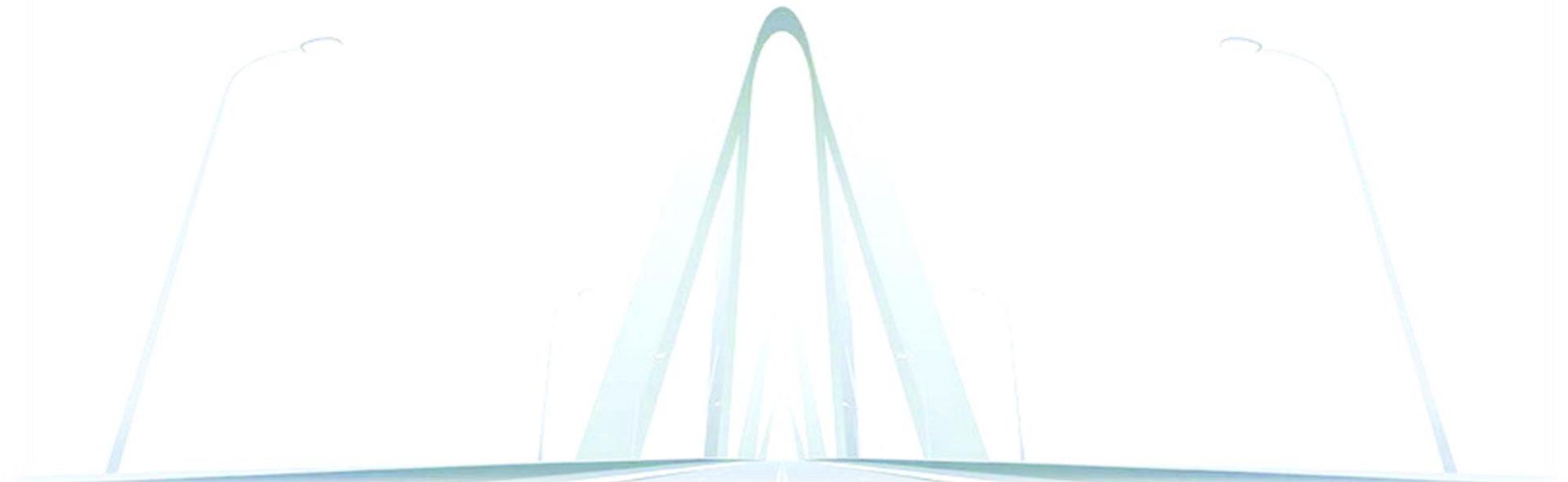
- 角色模型
- 组织模型
- 权限模型

资源定义:一个资源可能属于多个类别



资源分类是为资源的区分和选择：

- 指明资源应该归属的资源类，分配原则同时指明角色和组织单元，符合要求的资源必须属于这两个资源类的交集
- 职能分离的考虑
- 资源分配还可能依赖于一些特殊案例的属性(Attribute)





3.2 企业数据资源分类

从数据使用者不同，企业数据可分为：

主题域模型、概念模型、逻辑模型、物理模型

按业务应用及时间稳定性来分，企业数据可分为：

- 主数据
- 事务数据
- 状态数据（面向流程类应用的信息控制）



- SAP认为：
- ERP等系统中的数据主要分为主数据和业务数据两大类。
 - 所谓主数据，就是指客户、产品、供应商等在相当长一段时间内不会被频繁创建或修改的“静态”信息；
 - 而业务数据指的是如订单、报修单等随时会被创建或处理的“动态”凭证。



① Oracle认为：

- 主数据是描述核心业务实体的关键事实(例如客户、产品、员工、地区等)和这些事实之间的数据关系，这些关键事实在多个业务系统中被用到反复用到，在多个IT系统交互时，相同的实体可以有所区别。

② Microsoft认为：

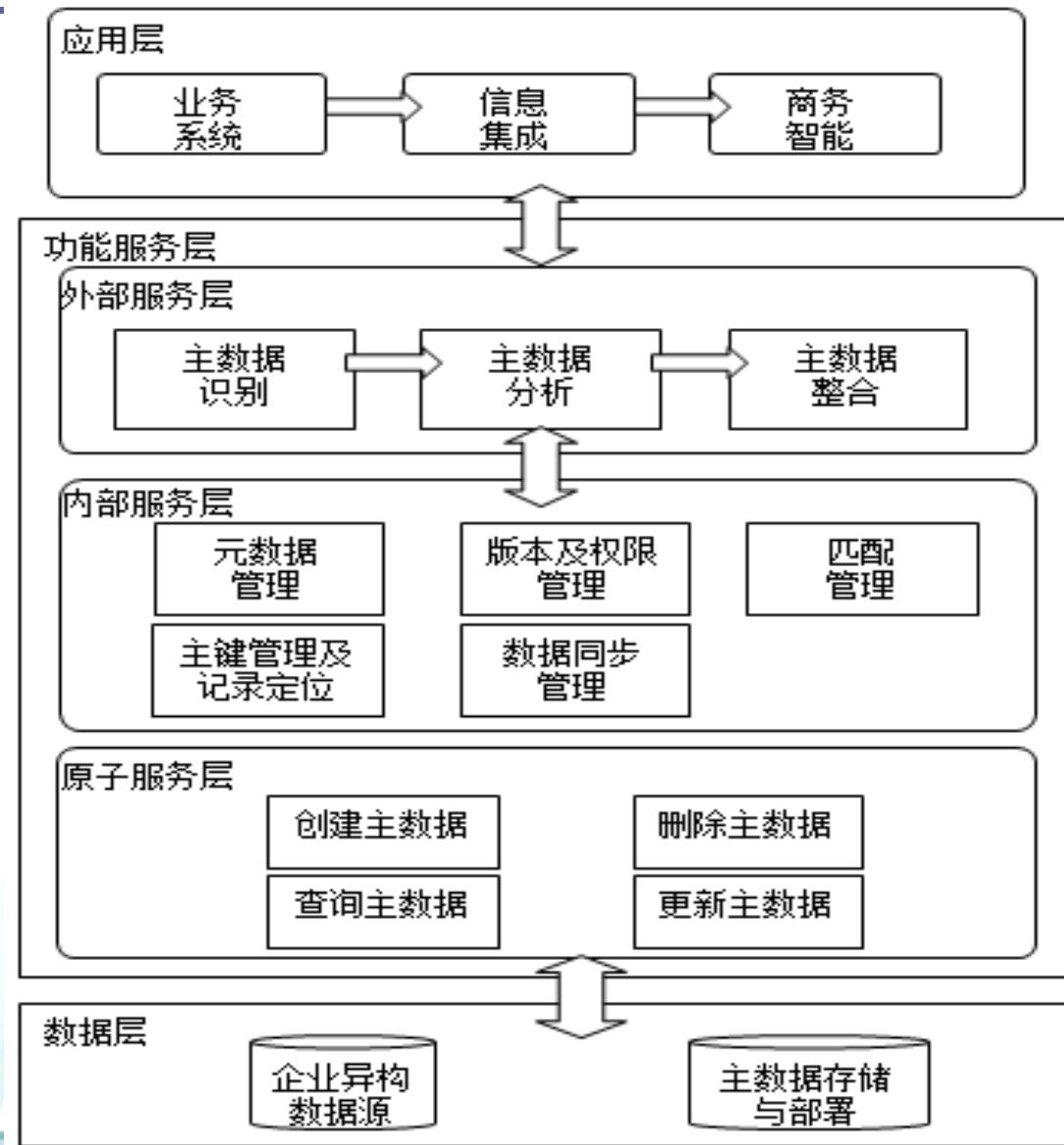
- 主数据是改变缓慢，并可以跨系统共享的业务关键词，通常包括人、物、地点和抽象对象等业务实体的信息，企业的数据分为五类：非结构化数据、元数据、交易数据、分级数据和主数据。

● IBM认为：

- 主数据是用来描述企业核心业务实体的数据，比如客户、合作伙伴、员工、产品、物料单、账户等；它是具有高业务价值的、可以在企业内跨越各个业务部门被重复使用的数据，并且存在于多个异构的应用系统中。
- 主数据定义企业核心业务对象，与交易流水信息不同，主数据一旦被记录到数据库中，需要经常对其进行维护，从而确保其时效性和准确性；
- 主数据还包括关系数据，用以描述主数据之间的关系，如客户与产品的关系、产品与地域的关系、客户与客户的关系、产品与产品的关系等。
- 主数据的特点包括：它是准确的、集成的，其次它是跨业务部门的，再有就是它是在各个业务部门被重复使用的。数据管理的内容及范畴通常包括交易数据、主数据以及元数据。



业界主数据管理主流框架主要是基于 SOA 架构特点来搭建系统架构。





数据存储方式的简要比较

| Features | RDBS | NOSQL DBS | DBS integrated with HDFS | Main-Memory DBS | Bigtable DBS | Graph DBS |
|--|----------|-----------------------|--------------------------|-----------------------|--------------|--|
| Support for ACID | Not well | Yes | Yes | Common | Common | Common |
| Support for semi-structured data and unstructured data | No well | Yes | Yes | Yes | Yes | Use graph structures with nodes, edges |
| Support for structured data | Yes | Not well | Yes | Yes | Yes | Use graph structures with nodes, edges |
| Support for scalability | Not well | Yes | Yes | Not well | Yes | Yes |
| Support for massive and distributed processing | Not well | Yes, but not flexible | Yes | Yes, but not flexible | Yes | Yes |



3. 3资源分类管理技术

- ④ **Ontology:**
 - ④ 描述领域(domain)内的概念(Classes)、概念间的相互关系(Object properties)、概念所包含的实体 (Individuals) 。
 - ④ **RDF (Resource Description Framework)**
 - RDF Used to describe resources on the Web
 - RDF can be read and understood in computer
 - RDF is not designed to display to people
 - RDF is writing in XML
 - ④ **OWL(web ontology language)**
 - web standard, upon RDF, for computer reading
 - Sub language of OWL
- OWL-Lite OWL-DL OWL-Full**



- Ontology
- Wikipedia: An formally represents knowledge as a set of concepts within a domain, and the relationships between those concepts. It can be used to model a domain and support reasoning about concepts.





OWL Elements

- Class (概念)

- simply a name and collection of properties that describe a set of individuals

- Individual (实例)

- members of those sets

- Property

- Object Property, Datatype Property, Annotation Property

- Axiom

- 公理 (axiom) 描述(如：继承，必要条件，充分必要条件，所包含的成员，非相遇类等)。



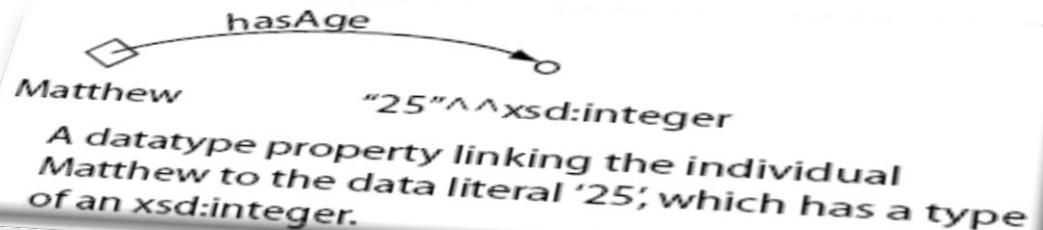
Property用于描述类间的关系

Object Property



An object property linking the individual Matthew to the individual Gemma

Data Property



A datatype property linking the individual Matthew to the data literal '25'; which has a type of an xsd:integer.

Annotation property



An annotation property, linking the class 'JetEngine' to the data literal (string) "Matthew Horridge".

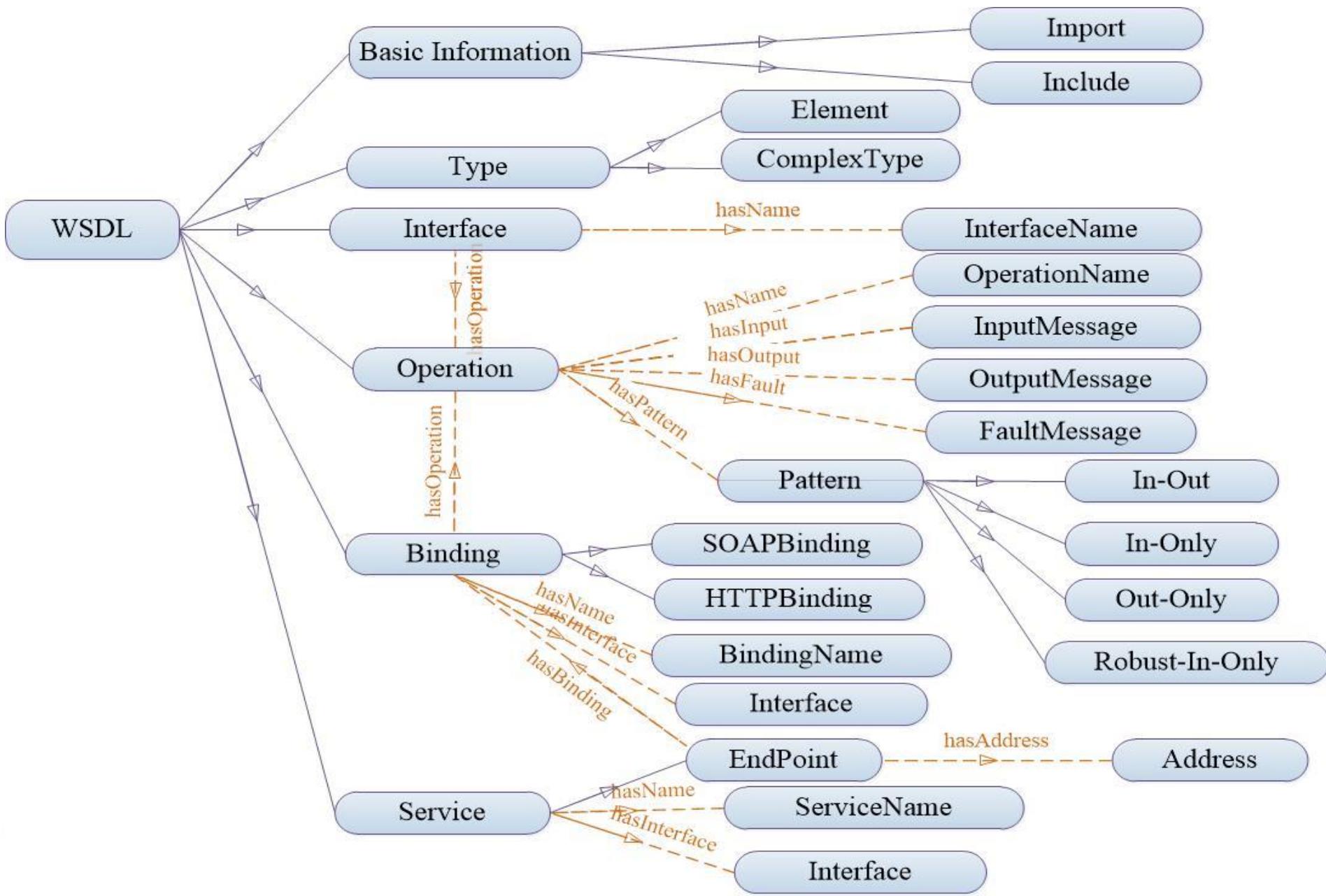


关系主要有四种：

- ① 1) 部分--整体(part-of);
- ② 2) 继承(kind-of);
- ③ 3) 实体--概念 (instance-of) ;
- ④ 4) 属性特征



WSDL 2.0 ontology





使用 OWL管理资源的原因：

- 形式化定义了领域概念相关的类和属性；
- 使得在这些类和实体基础上的推理得以开展；
- OWL本体的优势是可以采用工具进行推理，以获得额外的信息。

总体而言，便于理解，便于共享，便于推理。



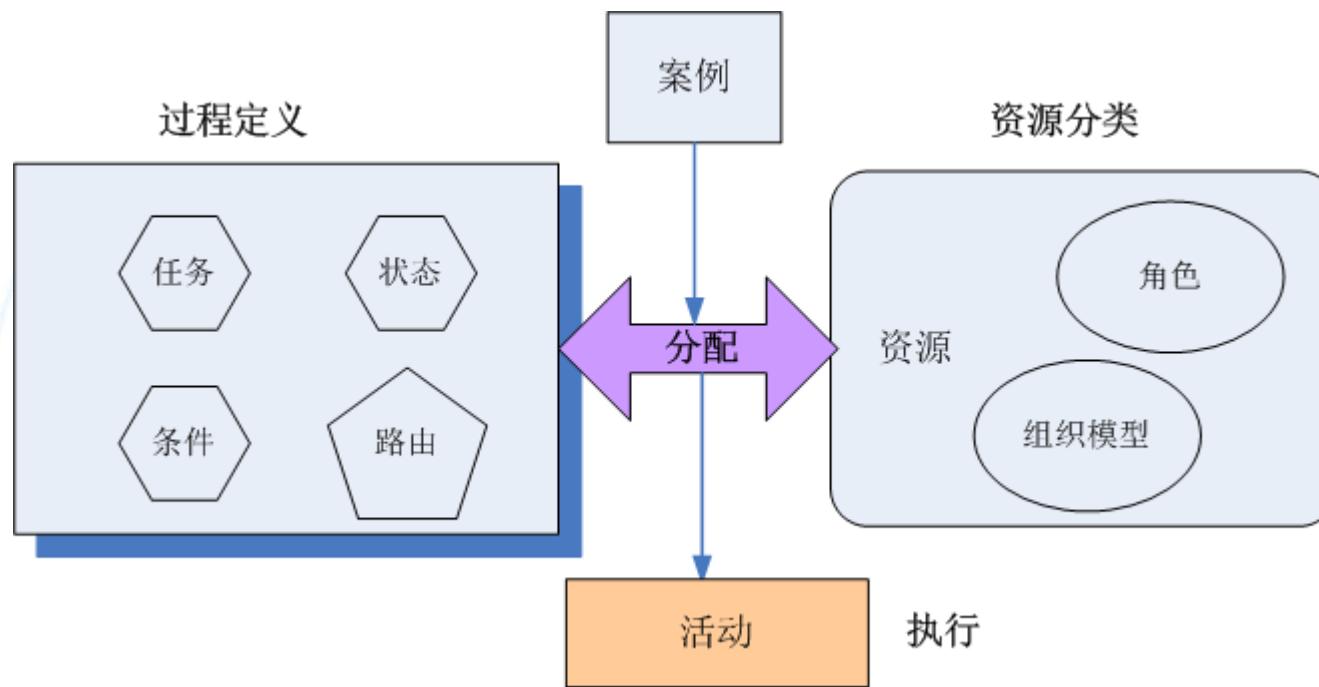
agenda

- ① 1. 基于Petri网的工作流建模
- ② 2. 过程定义
- ③ 3. 资源分类
- ④ 4. 资源分配及管理规则
- ⑤ 5. 流程的分析方法
- ⑥ 6. 小结



4 资源分配及管理规则

- 将工作项 (task + case) 分配到资源。
- 工作流系统的目标是为了尽可能快捷地完成工作项，为了实现案例到活动的转化；



④ 工作流服务器必须决定：

- 由谁做什么，或者说活动的执行采用哪个资源。
Push and pull control, load balancing, specialization, flexibility.
- 以什么样的顺序?
FIFO, LIFO, SPT, SRPT, EDD, priority control.



资源的选择

工作流中资源的选择过程

是一个资源与工作项的比对过程；



(1) 遵循工作流中的约束：

- 任务上一般标记有对资源类别的要求
- 资源被指定为若干类型
- 其它约束: separation issues, case management.

(2) 工作项分配资源的模式：

- ◎ **推模式：**工作流引擎将资源“推”倒工作项。
 - 引擎把工作项和资源进行匹配，资源自身不能选择，一旦资源执行完一个活动，他就被分给一个新的工作项。
- ◎ **拉模式：**资源“拉”动工作项。
 - 资源主动考察能够执行的工作项，并从中选择一个。
- ◎ 一般采取介于推动和拉动之间的方法，采用拉动原则，同时用工作流引擎生成的工作项次序作为辅助手段。



资源排队原则

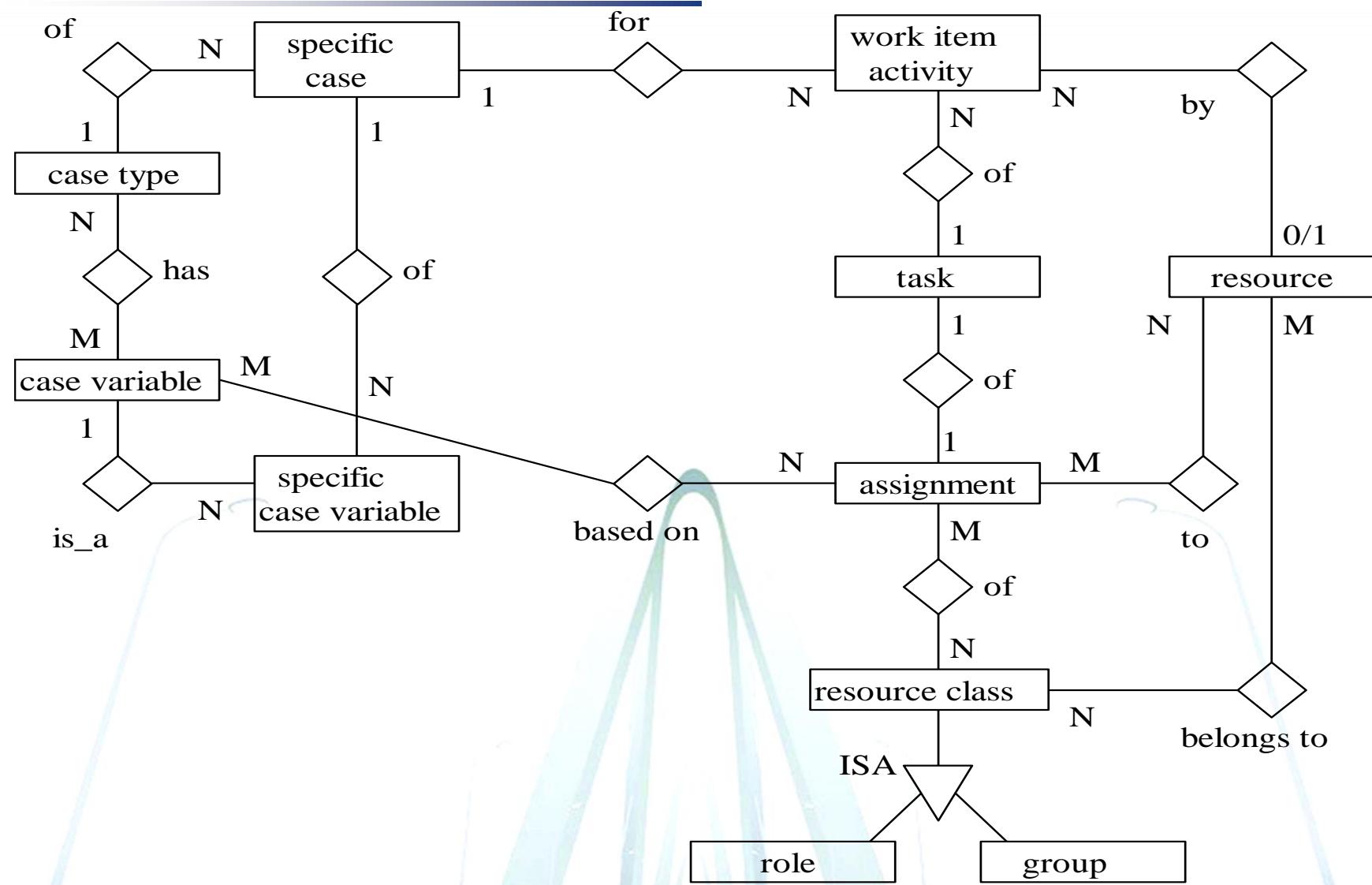
- ① 先进先出 (FIFO)
- ② 后进先出 (LIFO)
- ③ 最短处理时间 (SPT) : 时间和路由
- ④ 最短剩余处理时间 (SRPT) : 处理时间
- ⑤ 最早截至期限 (EDD) : 执行结束时间
- ⑥ 每一种排队规则所需的信息量存在很大不同。还有一些更高级的排队规则，甚至包含对未来状态的预测信息。

(3) 资源选择策略

- 让资源发挥专长，针对任务特性的划分专长；
- 让一个资源连续做类似的任务，缩短预热或准备时间；
- 预留弹性资源，让还有可选余地情况下，尽量让通用性好的资源空闲；



以ER图表示的工作流概念关系





- 一个分配和任务类，案例属性，资源类、资源等因素发生联系，非常复杂。
- ER图说明了资源管理的静态方面；
- 动态资源管理和分配和具体过程相关，依据实际流程而定；



工作流设计的规则（定性）

基于BPR经验，工作流设计有一些最佳实践。体现出一些基本规则，涉及到过程设计、资源分类和任务分配。

- 过程的目标；
- 定义过程是不要考虑现有资源；
- 尽可能让某一资源对案例的处理负责；
- 考虑任务的必要性；
- 考虑任务规模；
- 过程越简单越好；

- ① 通用过程和一个过程的多个版本的选择；
- ② 任务的粒度设定：精化和泛化；
- ③ 尽可能采用并行处理；
- ④ 让资源发挥专长；
- ⑤ 让资源连续执行相似任务；
- ⑥ 尽量给将来留有余地；
- ⑦ 尽量让一个资源在同一案例上工作；



agenda

- ① 1. 基于Petri网的工作流建模
- ② 2. 过程定义
- ③ 3. 资源分类
- ④ 4. 资源分配及管理规则
- ⑤ 5. 流程分析方法
- ⑥ 6. 小结



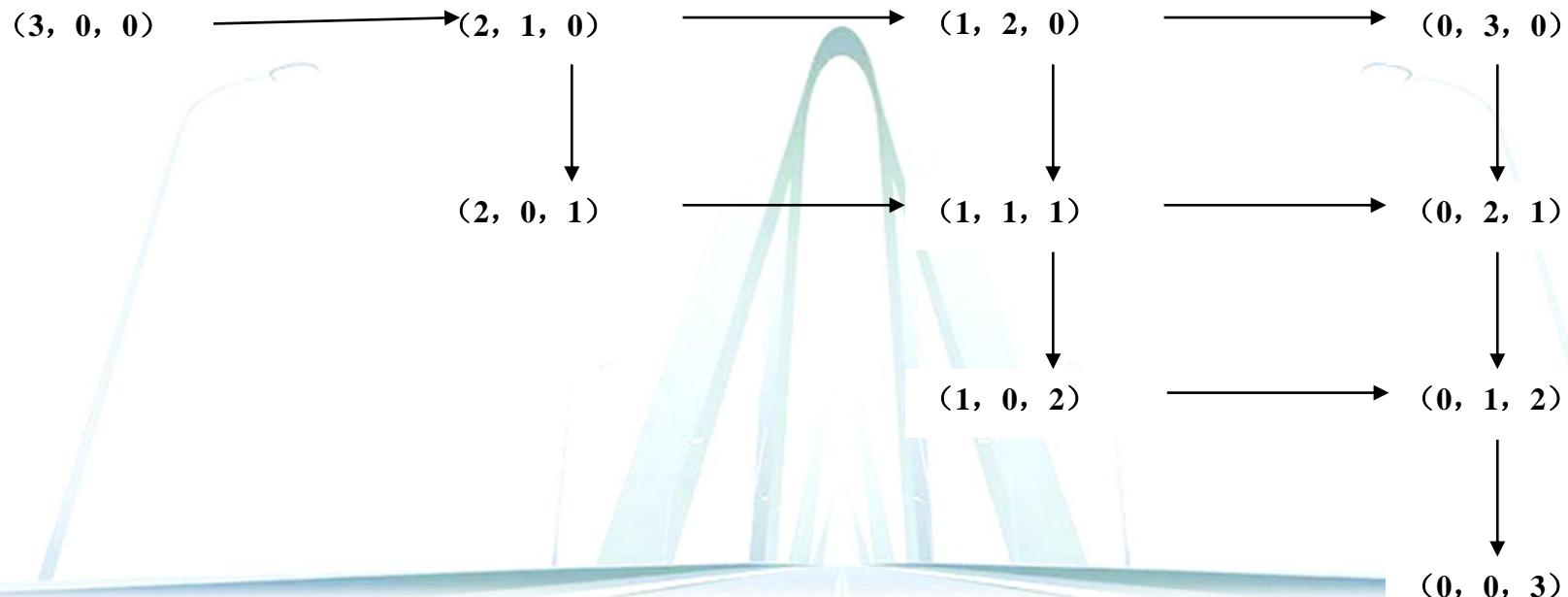
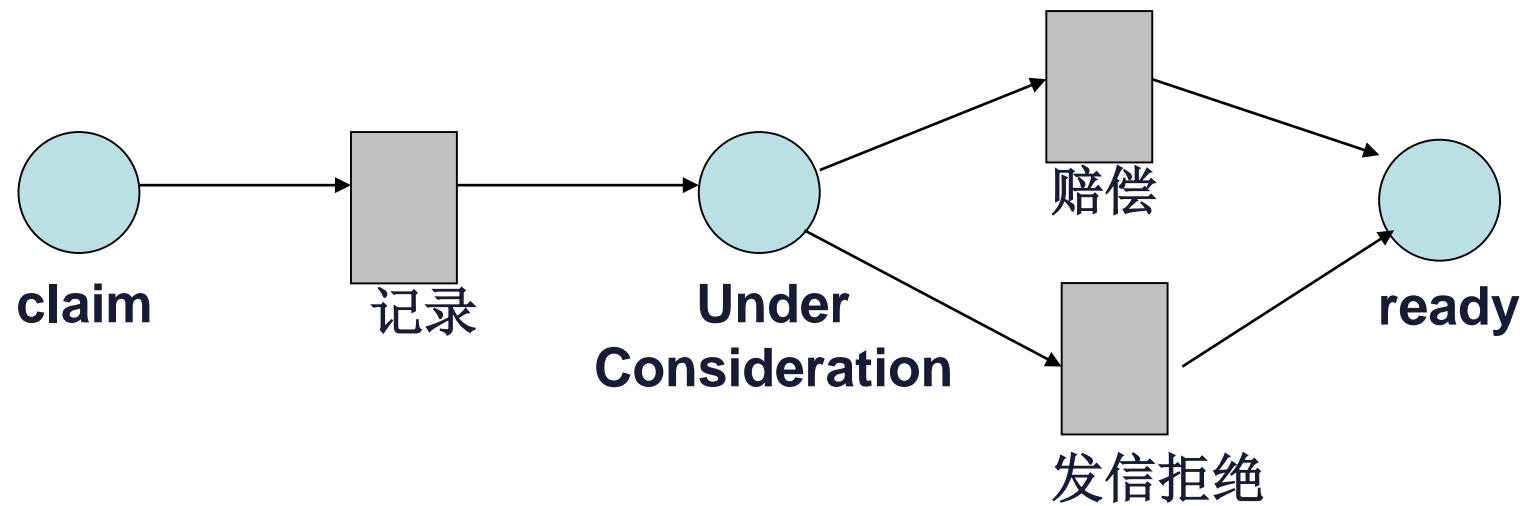
5 流程的分析方法

- 可达性分析
- 结构分析
- 性能分析
- 资源能力规划



5.1 过程的可达性分析

- ④ Petri网的初始状态决定了那些**状态可以达到**，以及他们的**次序**；
- ④ 可达图（reachability graph）被用来描述工作流的行为；其本质上是一种有向图，包含节点（可达状态）和箭头（转移或者说状态转变）。
- ④ 采用可达图可以判定工作流的正确性；





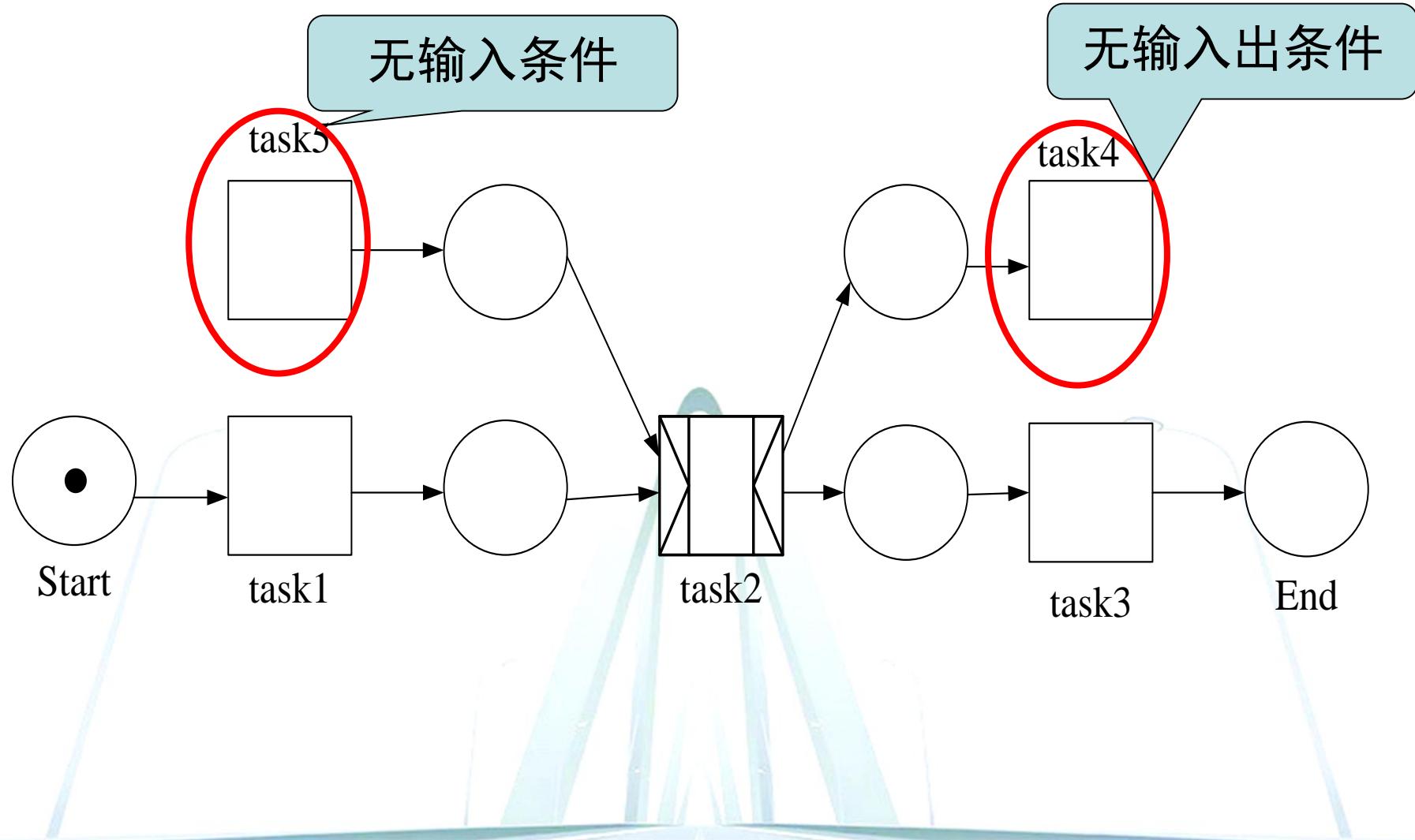
5.2 过程的结构分析

过程定义中常见的错误：

1. 任务没有输入或输出条件；
2. 存在死任务：任务永远不能被执行；
3. 存在死锁：任务在到达end前发生了堵塞；
4. 存在活锁：把案例带进无休止的循环；
5. 在到达end后，仍有活动执行；
6. 案例完成后，定义的过程中仍然存在标记；

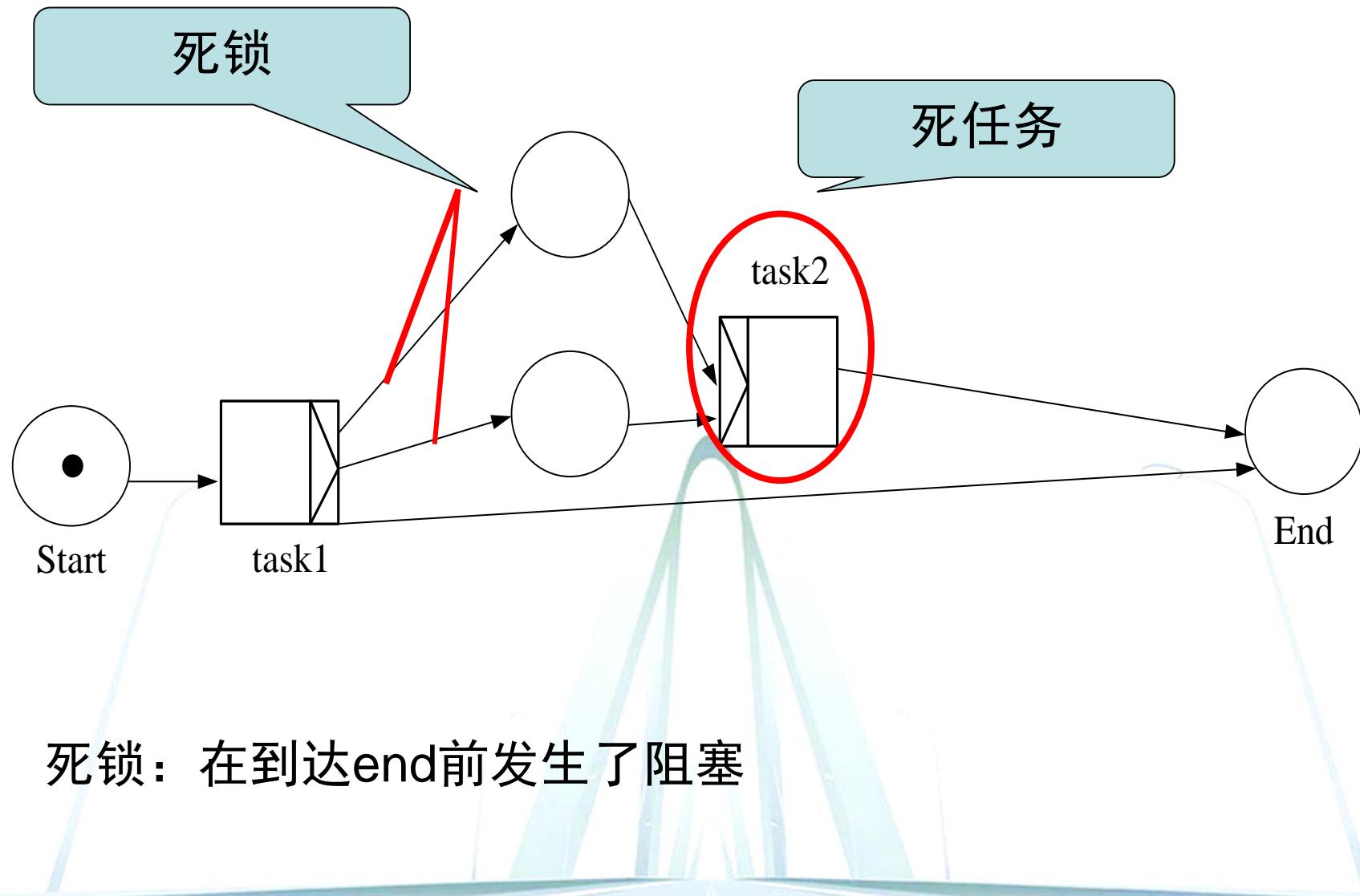


Situation A



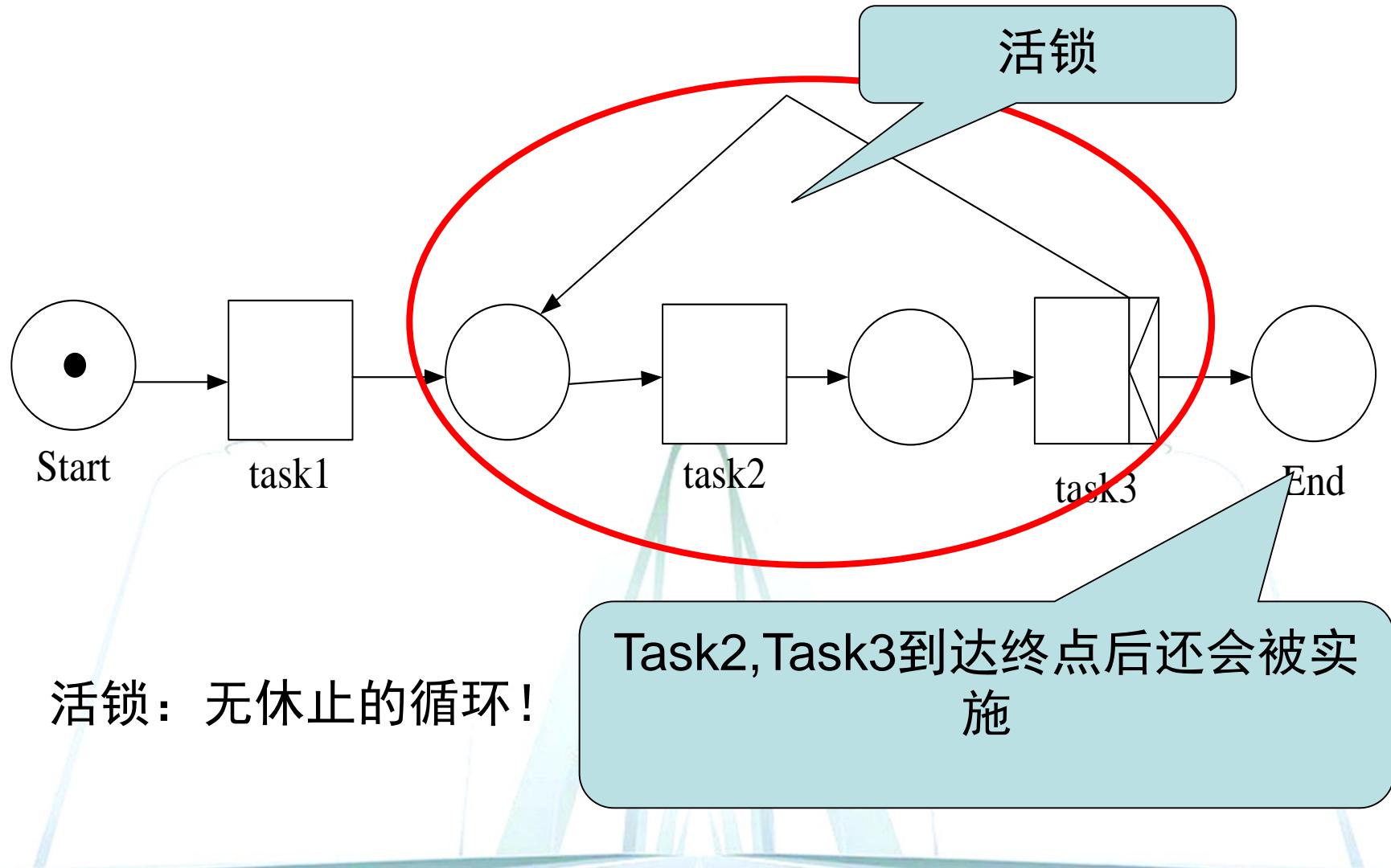


Situation B



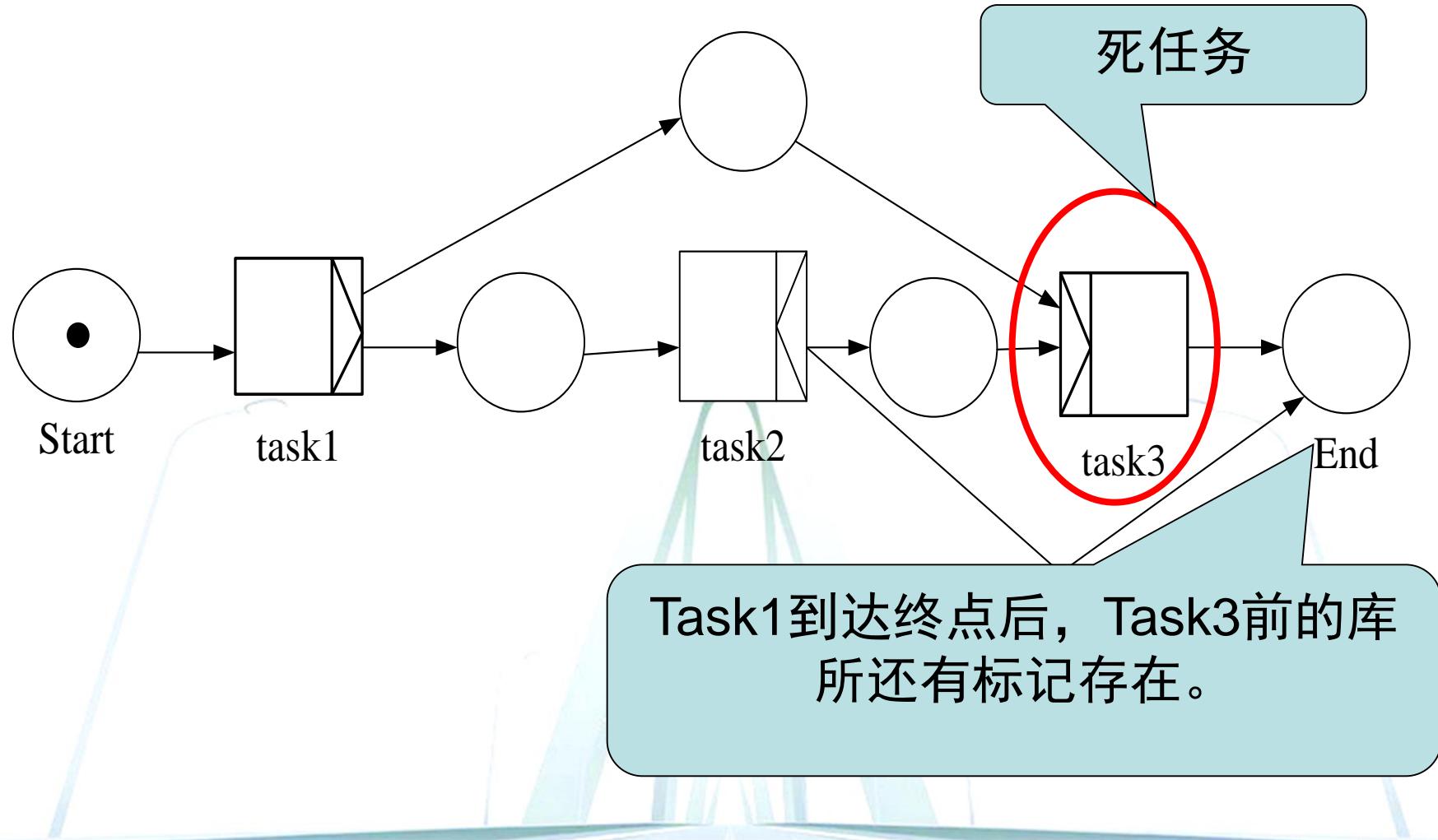


Situation C





Situation D

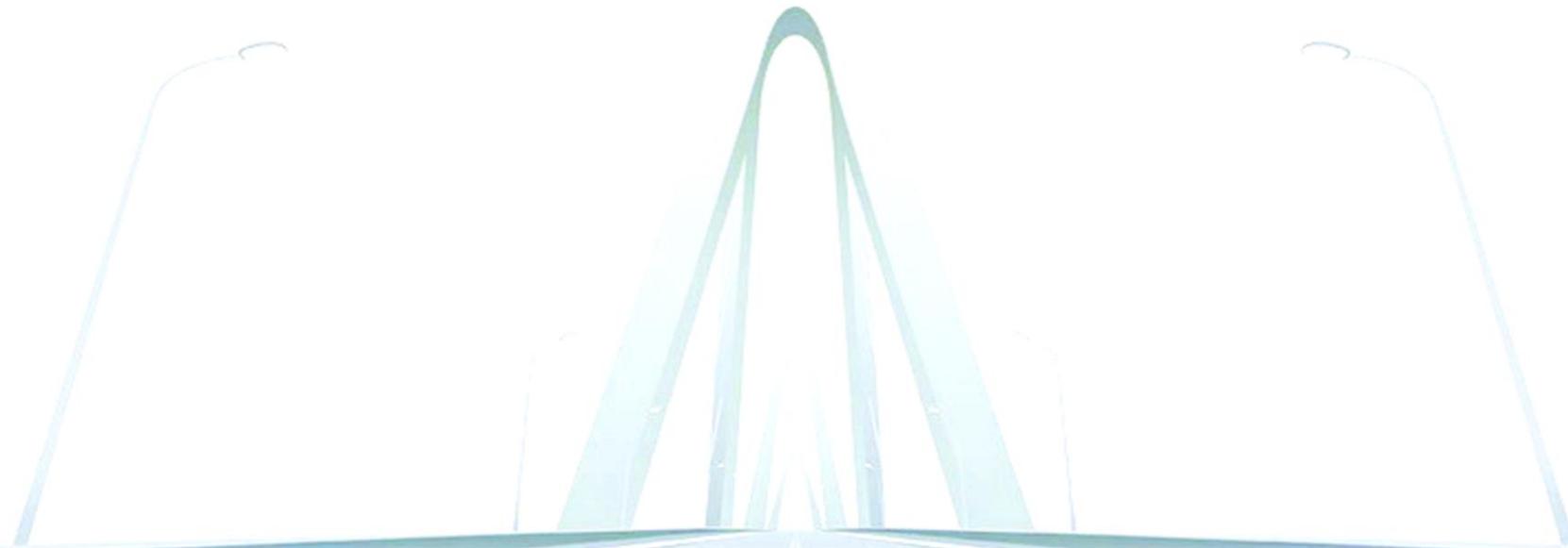




合理性分析

每个合理（Sound）过程都必须满足以下要求：

- ① 过程不包含不必要的任务；
- ② 每个提交的案例必须能够完全完成；
- ③ 完成后再没有对案例的引用；



工作流模型的一些常见检测，主要包括：

- ① 死锁检测（Deadlock）：通过深度/广度优先搜寻算法重复探索未到达的状态，直到发现死锁；
- ② 活锁检测（Live lock）：检验模型中是否存在一直在某些状态中循环转移而无法到达其他可用状态的情况。
- ③ 确定性检测（Deterministic）：检测模型最后是否会到达一个唯一的最终状态。若进程最终会指向一个唯一的状态，则称该进程是确定的；
- ④ 无终止检测（Nonterminating）：检测过程是否会终止。
- ⑤ 可达性检测（Reachability）：检测过程是否可以到达某个特定的状态。
- ⑥ 线性时序逻辑检测（Linear Temporal Logic : LTL）：检测过程是否符合给出的线性时序逻辑的公式。
- ⑦ 精化检测（Refinement/ Equivalence）：精化检测是两个过程之间的比较，可用于对两个过程模型的一致性检测。



5. 3 性能分析

工作流的性能反映了工作流定量方面的特性。

工作流不能正常工作的典型征兆：

- 过程中的案例过多；
- 实际处理时间过长；
- 服务水平低

流程的性能指标

性能指标 (performance indicators) 包括：

- ① 外部性能指标(面向案例): 工作流环境所关注的一些方面, 如平均完成时间和平均完成时间未定性指标。
- ② 内部性能指标(面向资源): 从内部资源出发, 如何提高外部性能, 如资源利用率水平, 在处理案例数量, 周转量和回退的数量。
- ③ 分配额外资源来提高外部性能的方法往往是最后的选择, 重构工作流结构和更好的分配策略来取得费用和性能的平衡更为重要。



流程主要性能指标：

- 案例完成时间（平均流动时间，等待时间）；
- 平均WIP（正在执行的工作）；
- 资源利用率（关键设备占用情况）；
- 单位时间处理案例数量；
- 预定标准时间完成案例的百分比；
- 周转量和回退的数量；
- ...

工作流性能分析主要有以下方法：

① 马尔科夫链：

- 实际上就是增加了转移概率的可达图；
- 通过成本和时间的引入extended Markov Chain, 可获得性能指标；
- 缺点在于：不是所有各方面的性能都能分析出来；另外应用Markov chain的分析过程非常耗时，即便是简单的情况下。

② 排队论：

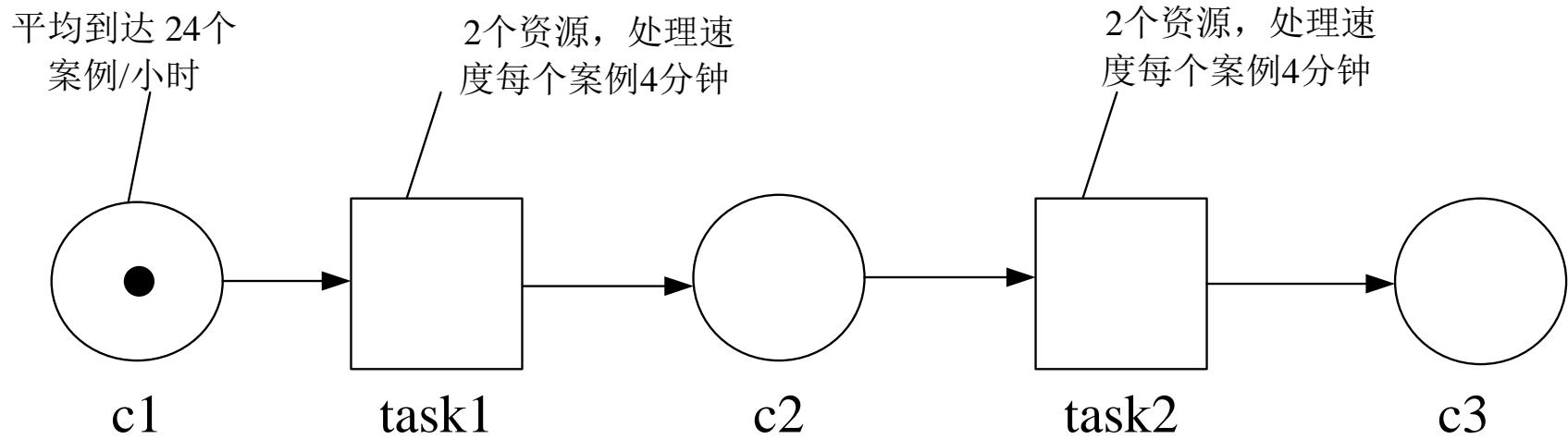
- 大量相同资源的队列时适用；
- 难适应有并发路由的情况；一些假设对工作流也不适用。
- 评估整个工作流，需研究队列网络。

③ 仿真：

- 图示仿真工具应用较为普遍；
- 容易被没有数学背景的人接受，也可以跟踪执行状态。



Situation1

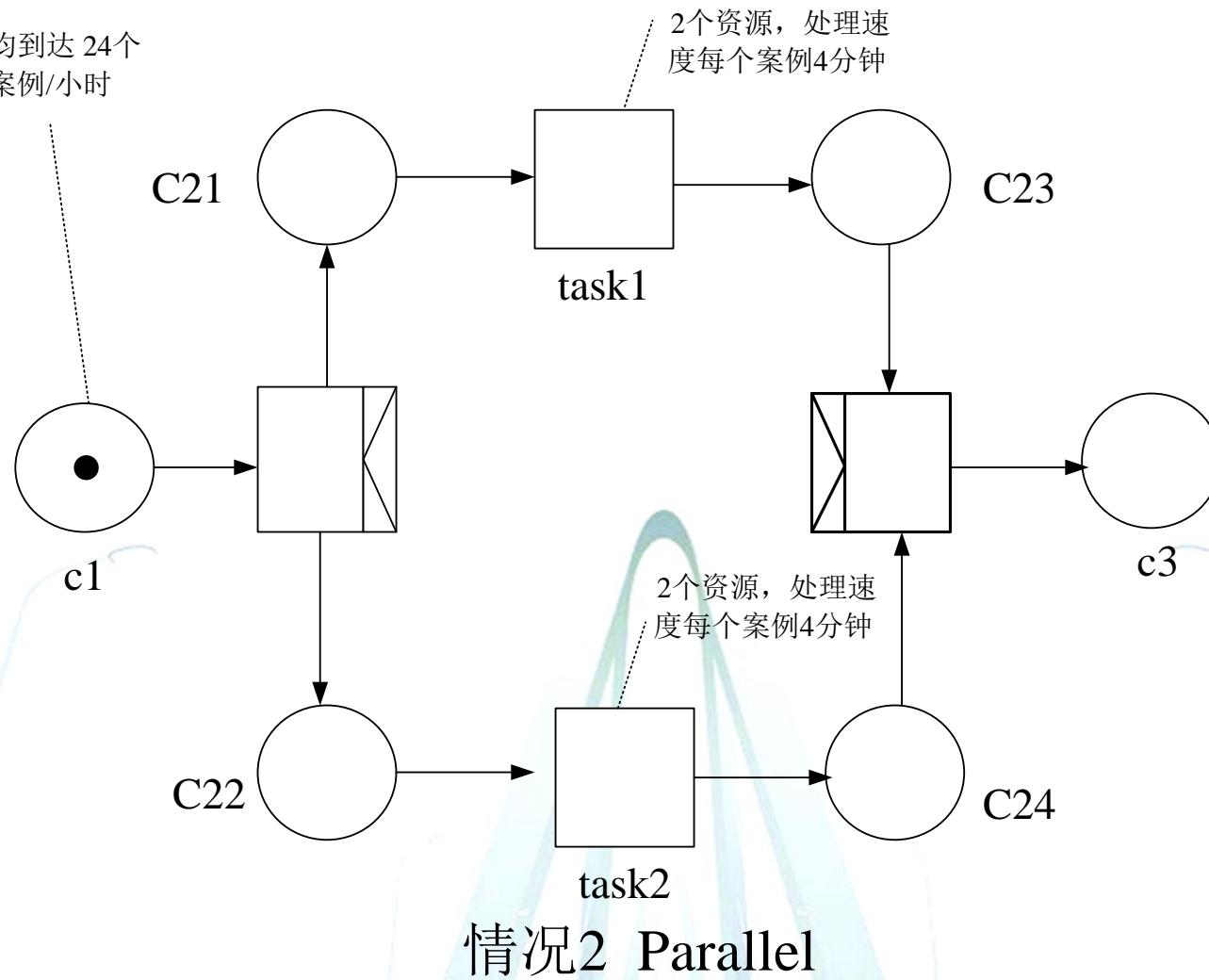


情况1 Sequential



Situation2

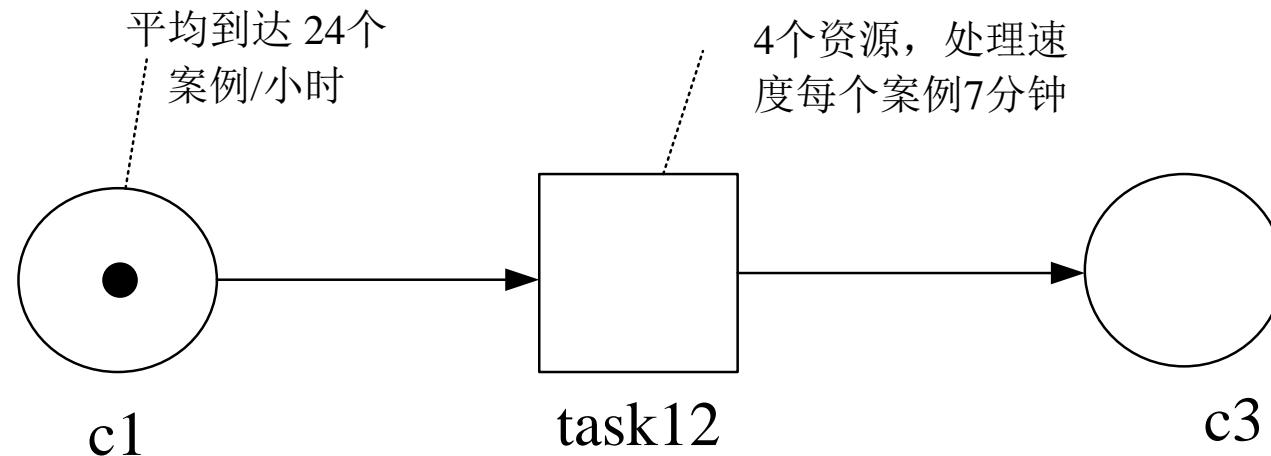
平均到达 24个
案例/小时



情况2 Parallel



Situation3



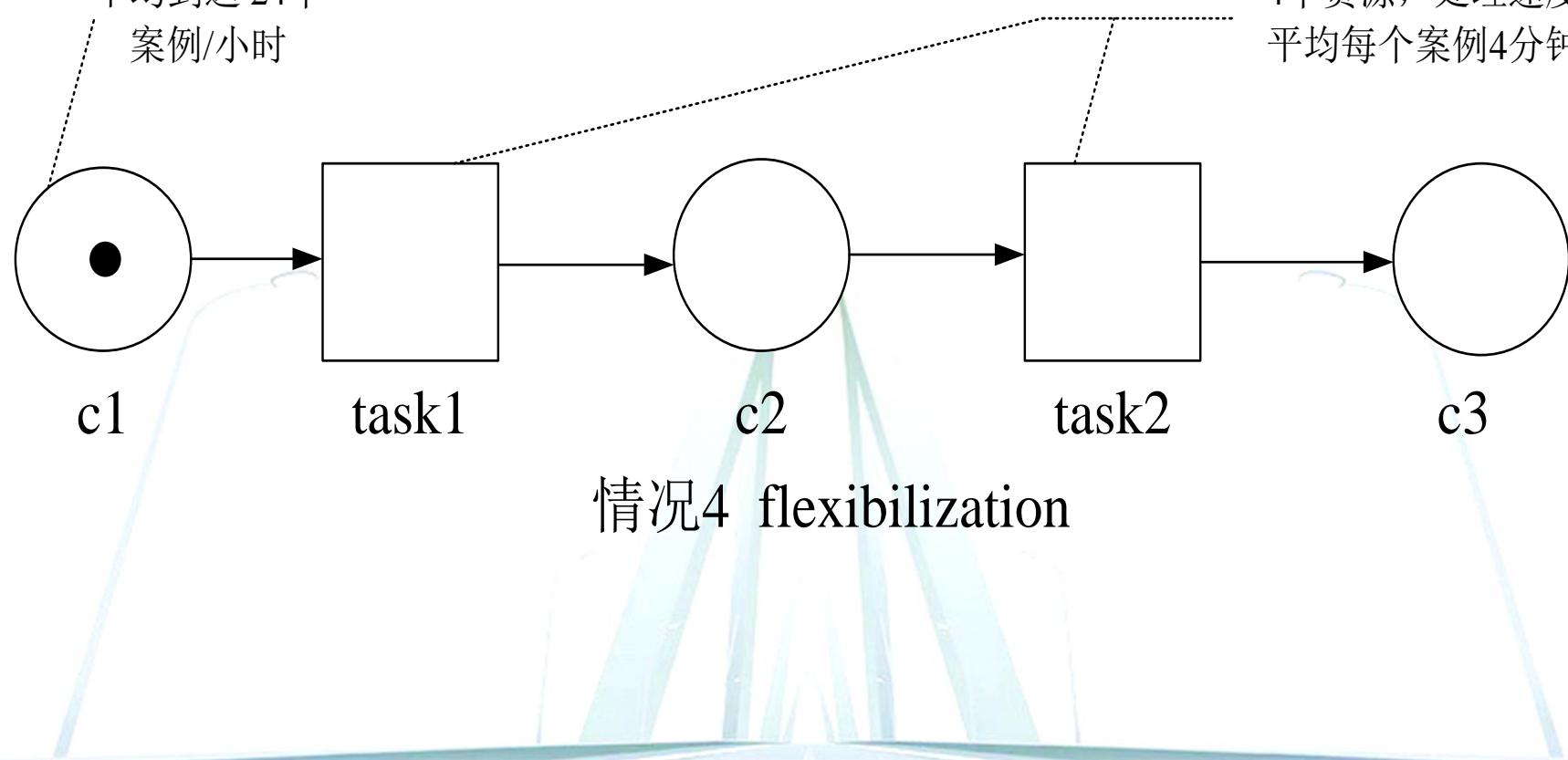
情况3 Composition



Situation4

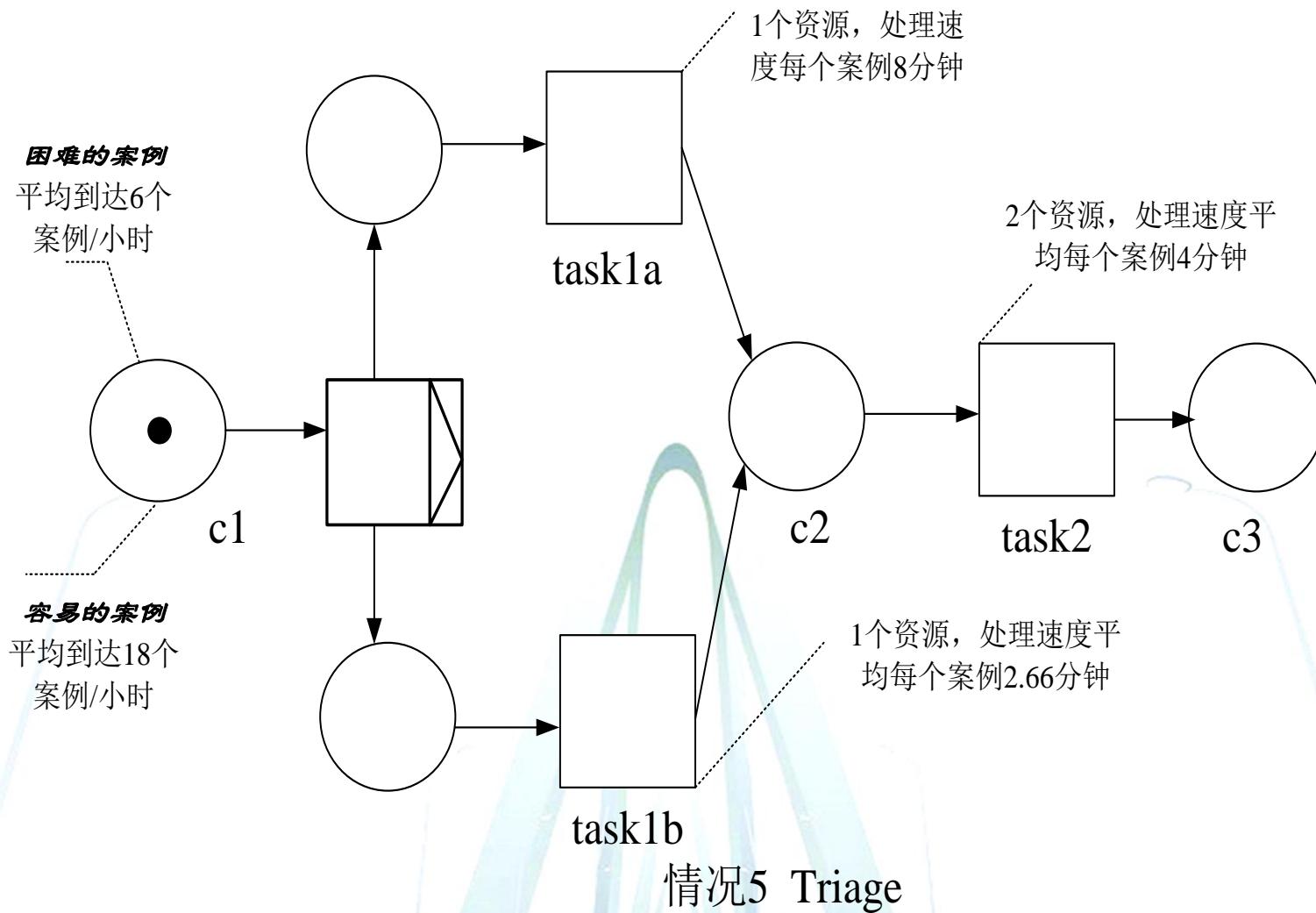
平均到达 24个
案例/小时

4个资源，处理速度
平均每个案例4分钟





Situation5

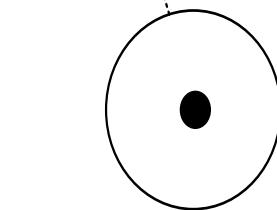




Situation6

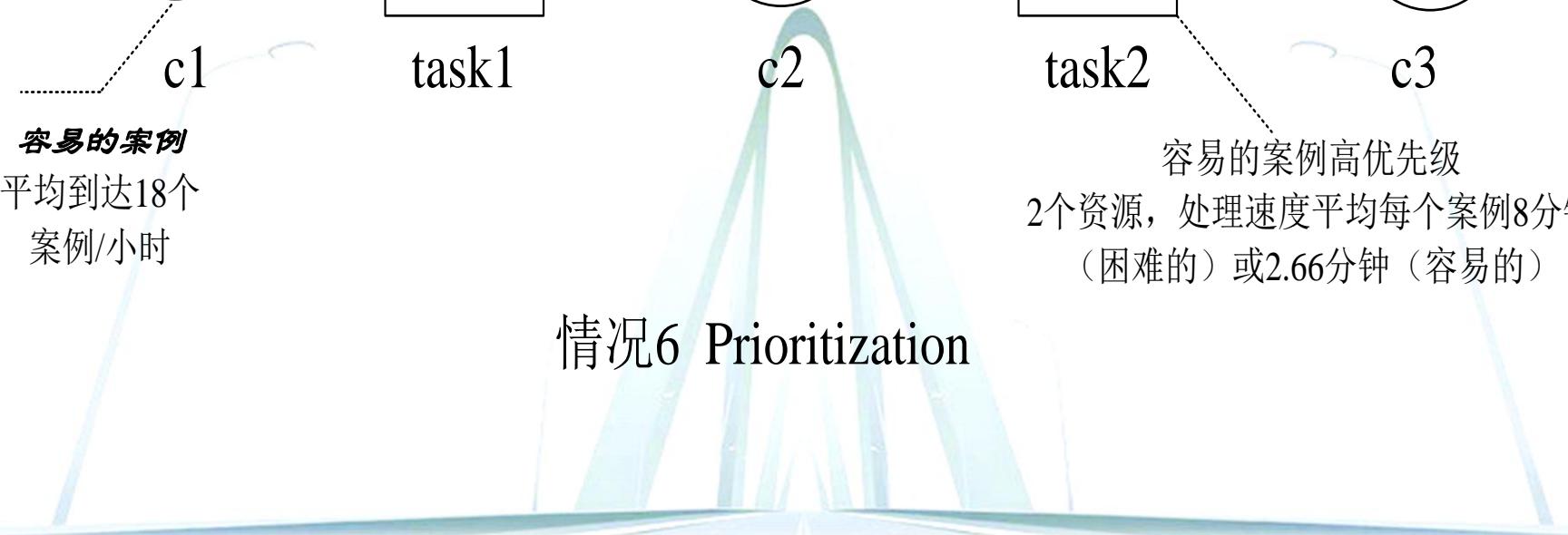
困难的案例

平均到达6个
案例/小时



容易的案例

平均到达18个
案例/小时



情况6 Prioritization



Result of 6 Situations

| 情况 | 描述 | 平均完成时间 | 平均处理时间 | 平均等待时间 |
|----|-----------------|--------|--------|--------|
| 1 | Sequential | 22.2 | 8.0 | 14.2 |
| 2 | Parallel | 15 | 4 | 11 |
| 3 | Composition | 9.5 | 7.0 | 2.5 |
| 4 | Flexibilization | 14.0 | 8.0 | 6.0 |
| 5 | Triage | 31.1 | 8.0 | 23.1 |
| 6 | Priorization | 14.0 | 8.0 | 6.0 |

工作流过程设计要遵循的三个方针：

- ① 尽可能并行执行任务；并行处理能缩短处理时间；
- ② 争取提高资源弹性。确保资源能胜任尽可能多的任务，弹性资源地使用能带来高水平的资源利用率和较短的完成时间；
- ③ 尽可能按照处理时间为顺序来处理案例，通常，最好给处理时间短的案例比处理时间长的案例更高优先级。



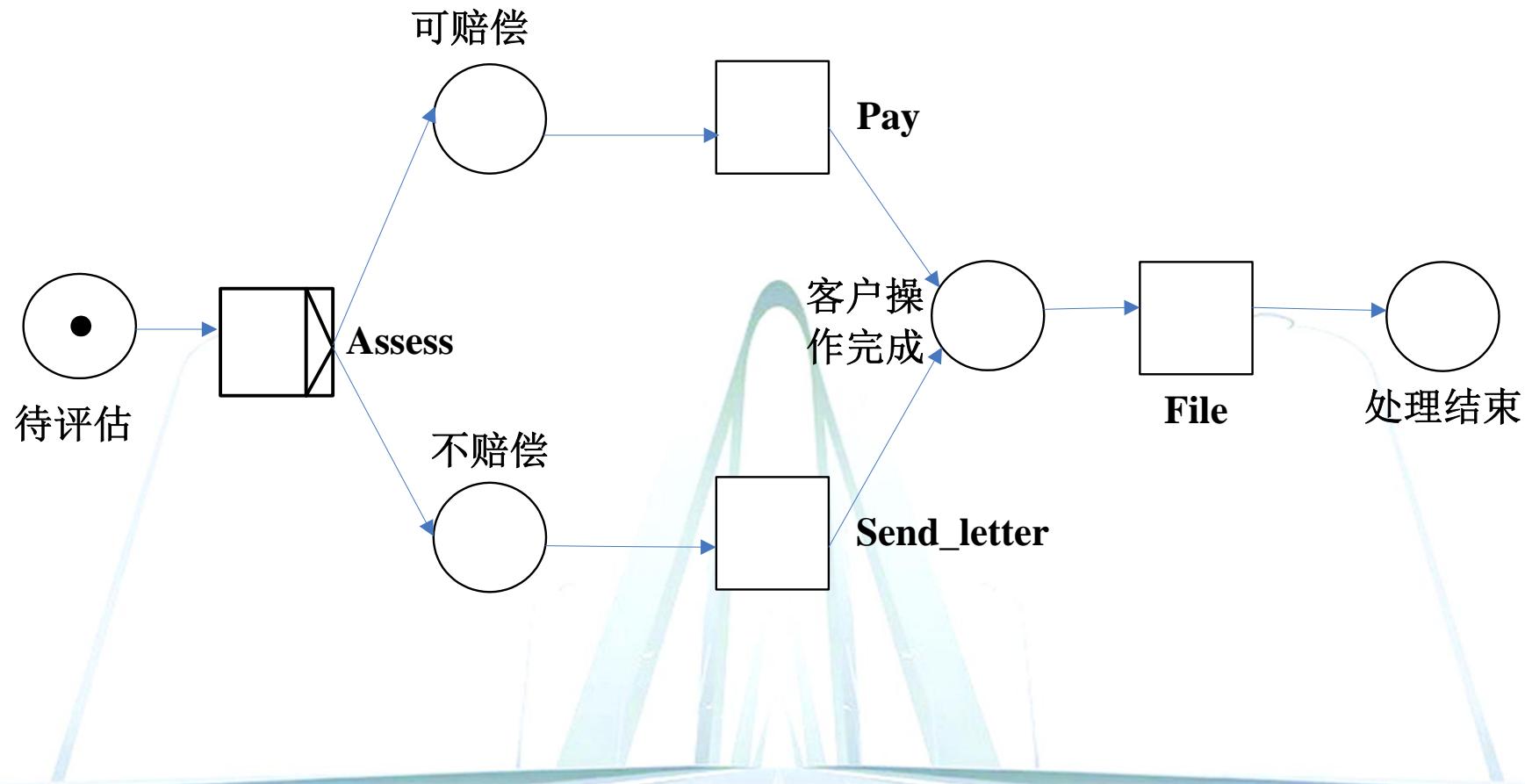
5. 4 能力规划

- ④ 能力规划说明每个时期都需要什么类型资源以及资源的数量；
- ④ 能力规划可分为短期规划和长期规划。
- ④ 在短期规划中，主要考虑的因素有工作量波动、病假、放假、加班、临时工的聘用等；
- ④ 长期规划中，主要考虑的因素有：需求预测、季节性影响、机器采购、招新策略等；



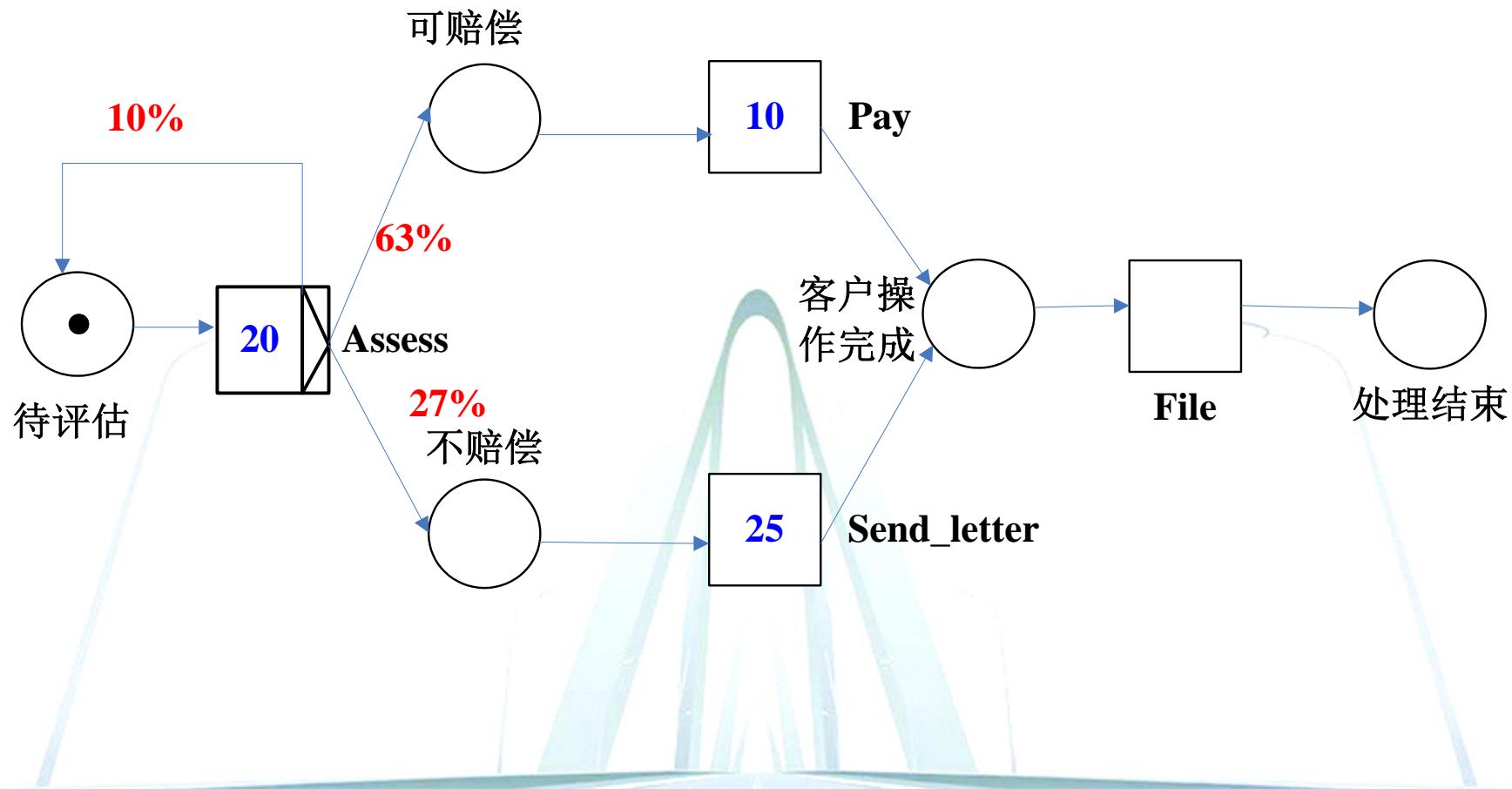
Situation5的具体化

一个简化的保险理赔处理流程





加入了处理时间和执行概率的流程





每个任务的能力需求

- 平均每天50个任务。
- 因Assess任务，10%需重复，63%可赔偿，27%不赔偿，实际执行次数为1.111。所以平均每天执行56个。

| 任务 | 每天的平均个数 | 平均处理时间 | 平均分钟数 |
|-------------|---------|--------|-------|
| Assess | 56 | 20 | 1111 |
| Pay | 35 | 10 | 350 |
| Send-letter | 15 | 25 | 375 |
| File | 50 | 0 | 0 |



◎ 任务和按角色以及部门分类后的资源关系

| 任务 | 角色 | 组织单元 |
|-------------|----------|-----------|
| Assess | Assessor | Compliant |
| Pay | Employee | Finances |
| Send-letter | Employee | Compliant |
| File | - | - |



每个资源类的能力需求

| 资源类 | 每天的平均分钟数 | 能力在80%时的资源个数 | 能力在60%时的资源个数 |
|------------|----------|--------------|--------------|
| Employee | 725 | 1.89 | 2.52 |
| Assessor | 1111 | 2.90 | 3.86 |
| Complaints | 1486 | 3.87 | 5.15 |
| Finances | 350 | 0.91 | 1.22 |

- ⑥ **能力规划：需要什么类型资源以及资源的数量。**
- ⑥ **结果分析：办公室至少需要4个人，其中三个是评估员；财务部需要1个人。**
- ⑥ **注意资源交迭的影响， Employee类属于办公室也属于财务部。**



agenda

- ① 1. 基于Petri网的工作流建模
- ② 2. 过程定义
- ③ 3. 资源分类
- ④ 4. 资源分配及管理规则
- ⑤ 5. 流程的分析方法
- ⑥ 6. 小结



6 小结

- ④ 基于Petri网工作流建模的三个过程是过程定义、资源分类与资源分配；
- ④ 工作流的主要概念有案例、任务、工作项、过程、活动、路由以及触发条件；
- ④ 基于Petri网的工作流基本路由主要包括AND-Split, OR-Split, AND-join OR-Join等；
- ④ 过程的分析是工作流建立及优化的基础，主要方法有马尔可夫链，排队论，仿真等；



- 一个任务构成的过程。单位时间内，有 λ 个新案例到达，需要被一个资源处理。这个资源单位时间内能够完成 μ 个案例。(M/M/1模型)
- 那么这个资源的能力利用率 ρ ：

$$\rho = \lambda / \mu :$$

- 假设处理时间和案例到达时间间隔都服从负指数分布，过程中的平均案例数量 L :

$$L = \rho / (1 - \rho)$$

- 平均等待时间 W : $W = L / \mu = \rho / (\mu - \lambda)$

- 平均系统时间 S : $S = W + 1 / \mu = 1 / (\mu - \lambda)$

- 对于某task，每小时平均有8个案例到达，每小时平均处理10个案例；

那么 $\lambda = 8, \upsilon = 10$

$$1. \cdot \rho = \lambda / \upsilon = 8/10 = 0.8$$

$$2. \cdot L = \rho / (1 - \rho) = 4$$

$$3. \cdot W = L / \upsilon = 4/10 = 0.4 \text{hour} = 24 \text{min}$$

$$4. \cdot S = W + 1 / \upsilon = 24 + 6 = 30 \text{min}$$



④ 给定资源利用率下，平均WIP

| 利用率 | WIP平均数 | 利用率 | WIP平均数 | 利用率 | WIP平均数 |
|------|--------|------|--------|-------|--------|
| 0.10 | 0.11 | 0.80 | 4 | 0.98 | 49 |
| 0.25 | 0.33 | 0.85 | 5.66 | 0.99 | 99 |
| 0.50 | 1.00 | 0.90 | 9 | 0.999 | 999 |



信息系统分析与设计

Lecture 4 基于事件的流程建模

蔡鸿明

hmcai@sjtu.edu.cn

软件学院信息系统技术实验室

lst.sjtu.edu.cn





Agenda

1. ARIS框架
 2. 基于EPC的过程建模方法
 3. ARIS视图
 4. ARIS建模过程实例
 5. ARIS软件工具
 6. 小结
- A1 BPMN建模

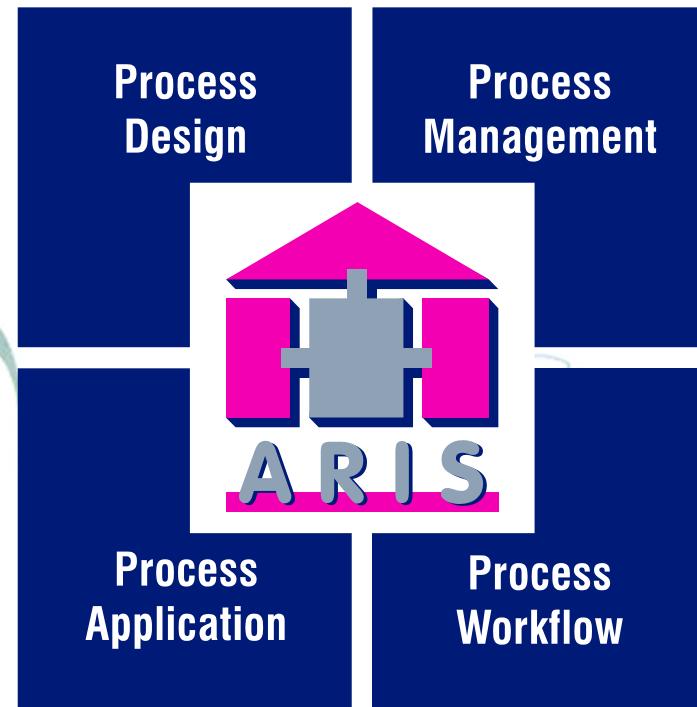


ARIS (architecture of integrated information system)是德国 Saarland大学的Scheer教授提出的一个集成化的信息系统模型框架，其本质是一种面向过程的模型结构，提出在这个框架中开发、优化、集成应用系统。

- 由于Scheer教授创办的IDS公司在此基础上开发了完整的ARIS工具（ARIS toolsets），将各个视图信息进行了有效关联，并提供了基本的仿真分析手段，在商业上获得了很大的成功。
- SAP公司注资IDS公司，使用ARIS作为其ERP软件实施的企业建模、分析、诊断的工具，指导SAP的实施，使得ARIS得到广泛的应用。
- 超过60%的ARIS产品是在SAP R/3基础上进行定制和应用的。



- Gartner Group连续多年将ARIS评为世界公认的业务流程建模（BPM）工具的领导者。



- 业界对ARIS的解释
- 用于企业流程设计和ERP架构的著名软件。

Scheer教授建立ARIS的目的是：

- 为了寻求一种能够用来缩小企业与信息技术之间差距的应用框架；
- 提供一种能够对业务流程概念进行准确描述和详细分析的方法；
- 为企业信息系统的开发实施提供了一个基础起点。

Scheer教授是这样定义ARIS的：

- ✓ 一种描述业务流程的体系结构；
- ✓ 一系列包含有各种元模型的建模方法；
- ✓ ARIS工具集软件系统的理论基础；
- ✓ 基于计算机的业务流程管理的概念。

- ◎ ARIS的成功主要在于
 - » 强大的工具支持和坚实的理论基础；



- ARIS有一套标准的软件工具集支持ARIS建模及仿真实施。自1990年以来，IDS Prof.Scheer公司一直致力于基于ARIS模型的软件实施及相应工具的发展。
 - IDS Scheer 是德国第二大软件公司，并位居前十名 IT 服务商之列。
 - 2004年初，IDS Scheer中国公司正式成立，注册名称爱迪斯（上海）软件有限公司，致力于业务流程管理和 SAP 系统咨询实施服务，直接为国内广大的客户提供支持。
 - 2007年ARIS7.0正式发布并投入使用。

ARIS 工具的产品包括：

- ARIS EasyDesign
- ARIS Toolset
- ARIS forR/3
- ABC (Activity Based Costing)
- Weblink和标准界面。
- 这些产品可互相集成

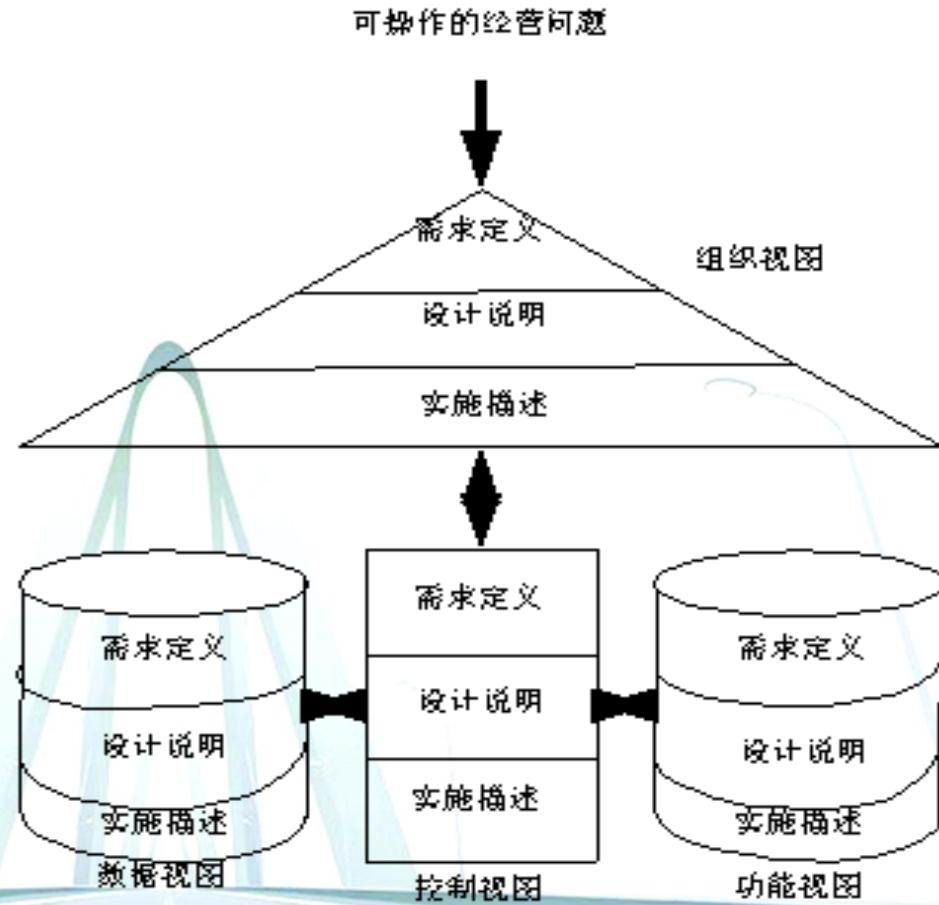


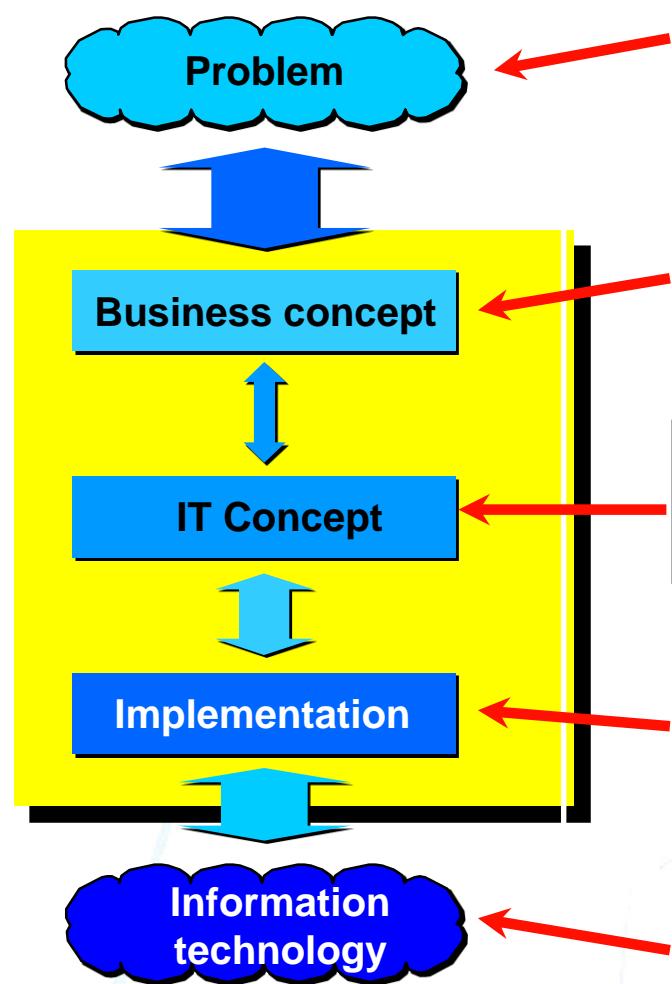
四个视图、五个层次、两个阶段

- ARIS共有四个视图，即功能视图、数据视图、组织视图和控制视图。其中控制视图起“粘合剂”的作用，其将系统功能、使用数据和参与组织联系在一起。
- ARIS采用了五个层次，各建模层次依次为现行系统分析，需求定义、设计说明、实施描述和运行维护。
- 其中现行系统分析，需求定义、设计说明、实施描述层称为系统建立阶段（Build time），运行维护层称为系统运行阶段(Run time)。
- 从技术层次上总结，其特性有三点：
 - (1) 覆盖业务问题解决的生命周期；
 - (2) 过程为核心的集成；
 - (3) 面向对象的建模和关联；



- 面向生命周期， ARIS定义的多个层次的概念：
 - 系统分析
 - 需求定义**
 - 设计说明**
 - 实施描述**
 - 运行维护





“Real World”:
The actual situation of your company

Define the requirements:
Analyze “as-is” and design “to-be”

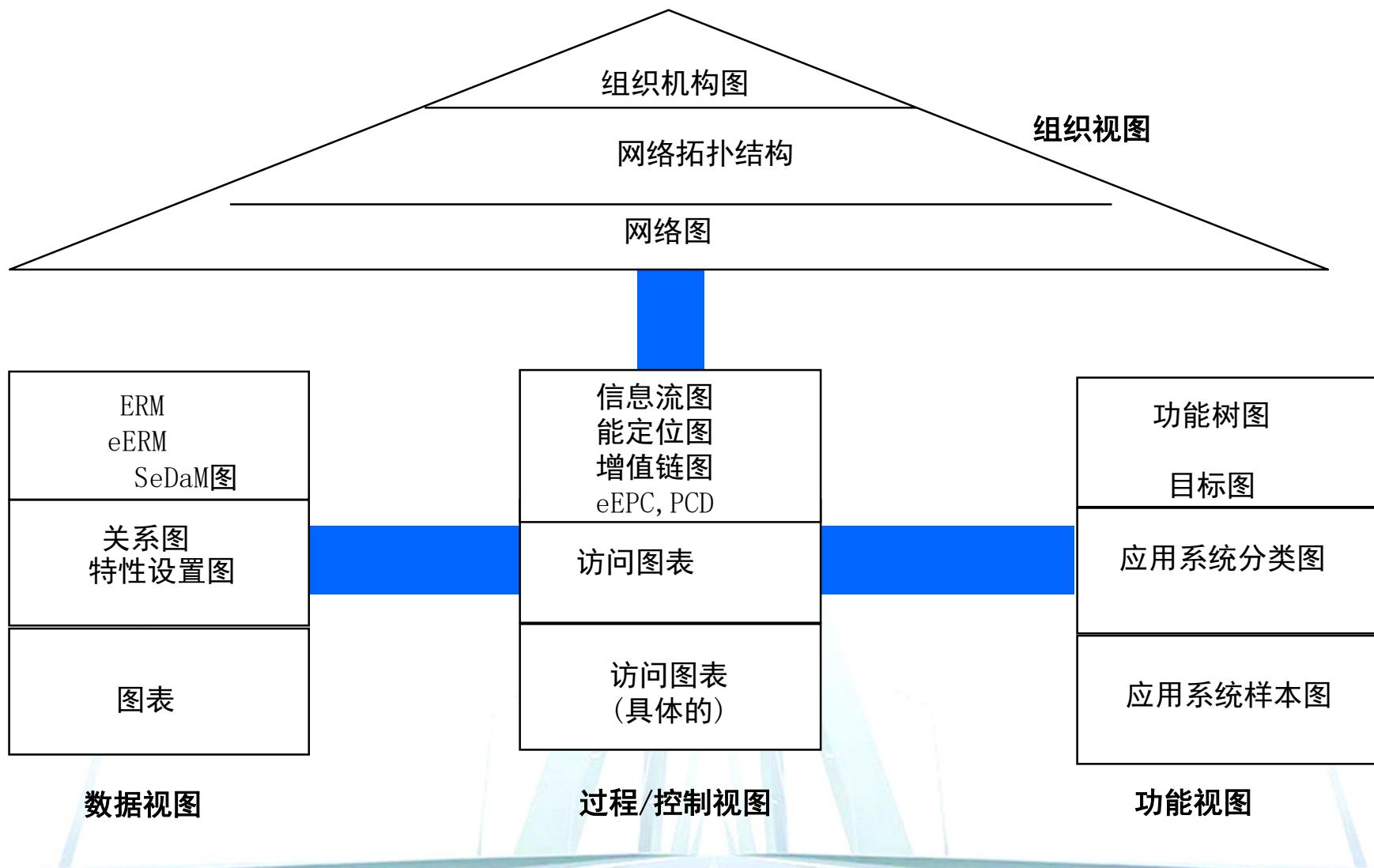
Design specification:
“Translate” into data structures, hardware, ...

Technical implementation:
Create physical hardware structure, ...

“Enablers”:
PC's, networks, standards, databases, ...



生命周期各阶段模型逐步推导





ARIS结构是从**过程链模型**抽取、发展而来的；

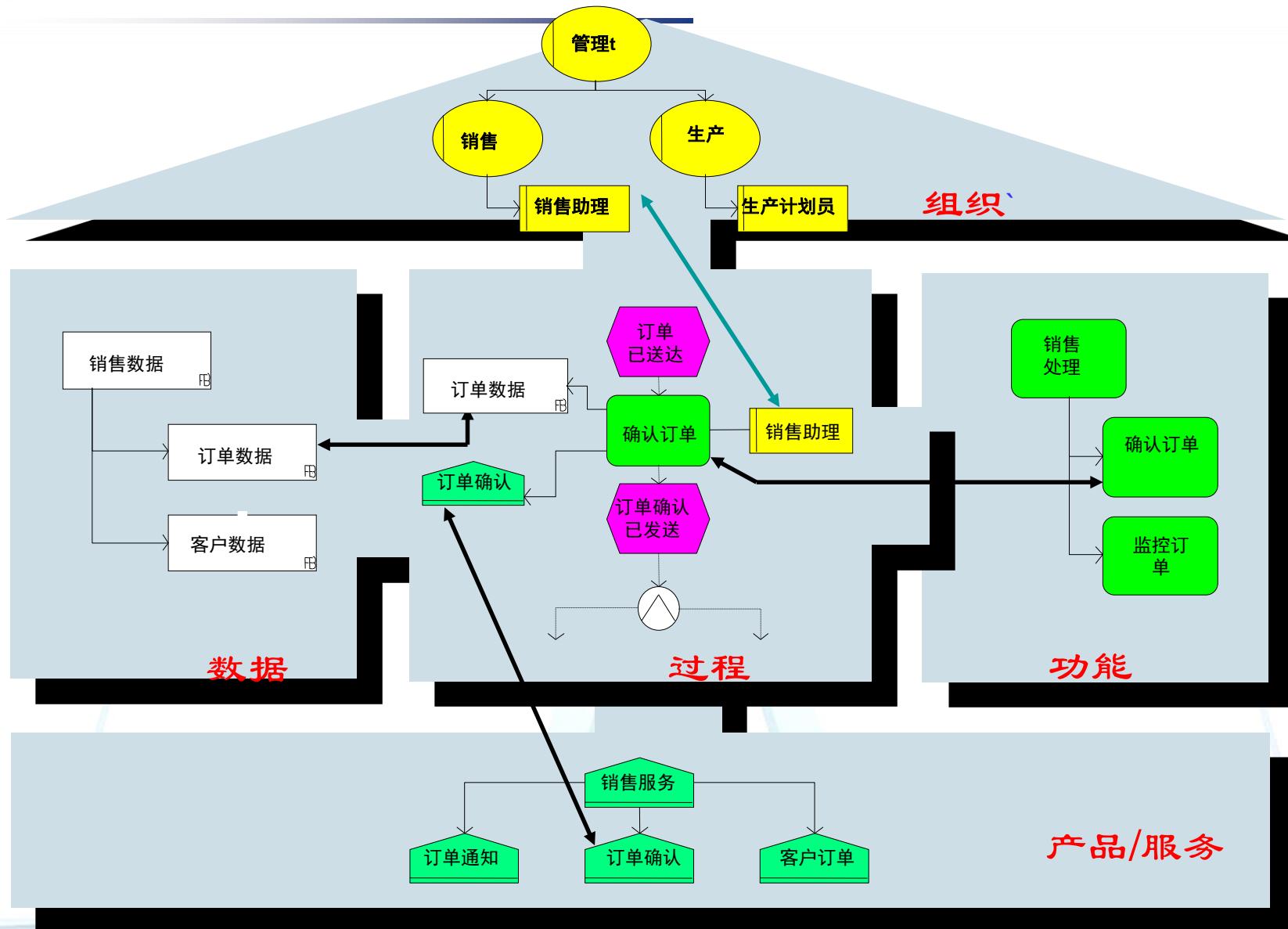
- 根据Gutenberg的生产理论，生产过程不再是物料、人力和生产设备的使用，而是生产过程中数据的交换。
- 从信息过程的角度来看，通过过程链就可以完整地描述企业过程。
- “过程”是由起始事件和终止事件定义的，过程的对象可以是**物质的转变**过程（如生产过程），也可以是**信息的转变**过程（如管理过程）。
- 以过程为核心的控制视图包含了多种过程模型，最主要的是扩展的事件过程链（eEPC）。
- ARIS体系结构实现了多视图、多层次、多关联、全生命周期地描述企业信息系统的各个方面，并提供各个建模信息的关联关系，为多种描述方法之间的自动转换和综合分析提供基础。

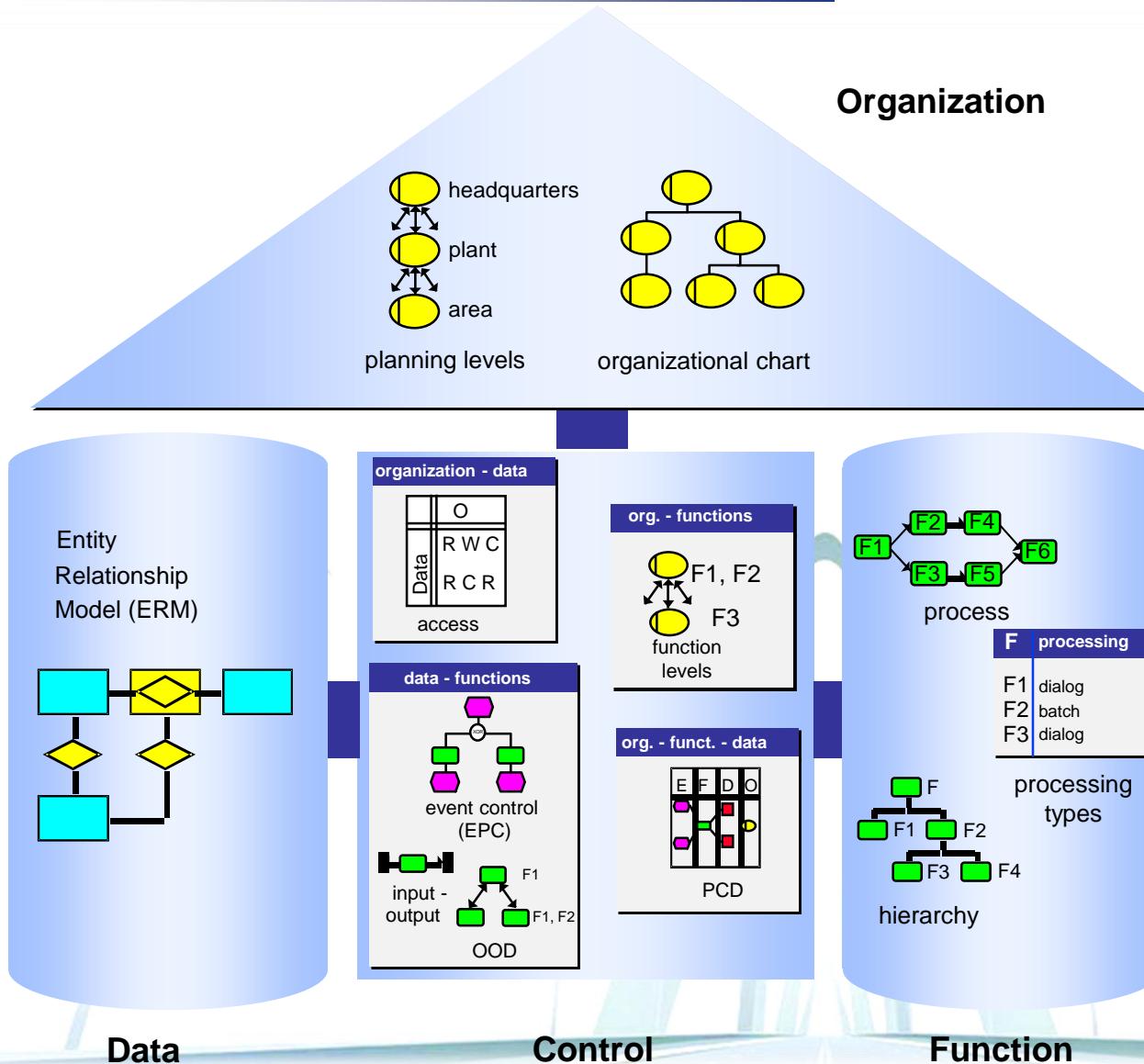


- 在ARIS结构中，组织、数据和功能视图的发展过程是相对独立的，它们间的关系由控制视图来描述。
- 由于功能视图、组织视图和数据视图是相对独立的，不仅减少描述问题的复杂性及冗余，还为将来实施基于 Web的分布式协同建模提供了基础；
- 建模人员可以依据视图描述规则对功能、数据和组织进行独立建模，通过从控制视图演化而来的控制规则来维护模型的一致性和完整性。控制视图是ARIS区别其他结构的重要特征，它用来记录和维护组织视图、数据视图和功能视图间的关系。
- 由于ARIS是面向过程建模的，所以在过程模型的基础上引入工作流管理系统成为可能。



过程为核心的集成方法





Legend:

| | |
|-----|------------------------------|
| ERM | = Entity Relationship Model |
| Fn | = Function n |
| E | = Event |
| D | = Data |
| O | = Organizational Unit |
| R | = Read |
| W | = Write |
| C | = Create |
| PCD | = Process Chain Diagram |
| EPC | = Event-driven Process Chain |
| OOD | = Object Oriented Design |



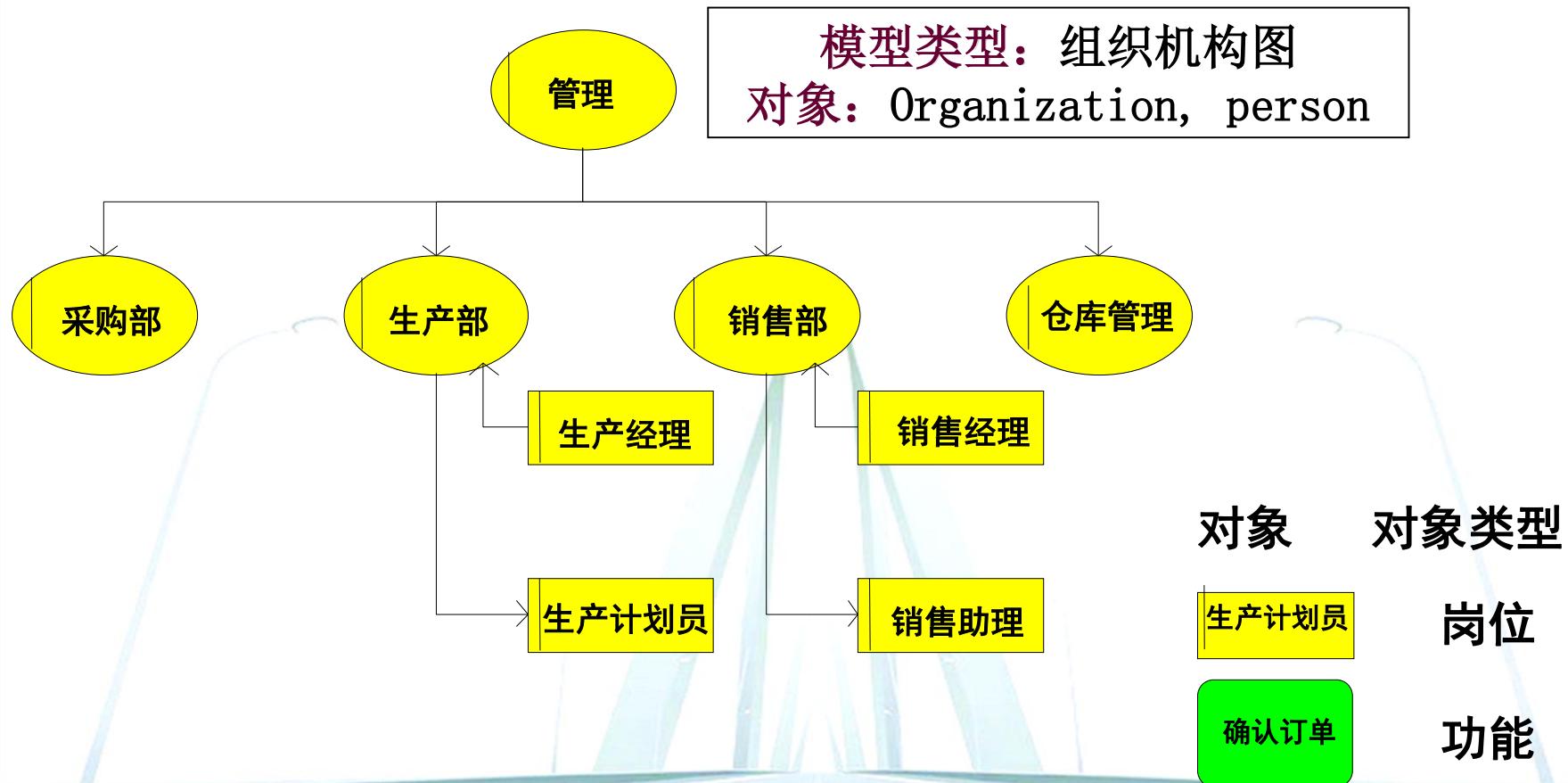
特性（3）面向对象的建模及关联

- ④ ARIS以面向对象的方法描述了企业的组织视图、数据视图、功能视图，以及控制视图中的各个模型。
- ④ 通过对企业这些侧面的分析，建立了各个视图的信息对象及对应方法。这些对象构成的过程作为模型、构成了ARIS应用实施的业务蓝图基础。

- ④ 各视图都有其不同的模型
- ④ 视图中的每个模型都有其具体的类型
- ④ 每个模型都都有其特定的对象
- ④ 对象通过不同类型的关系联系到一起.

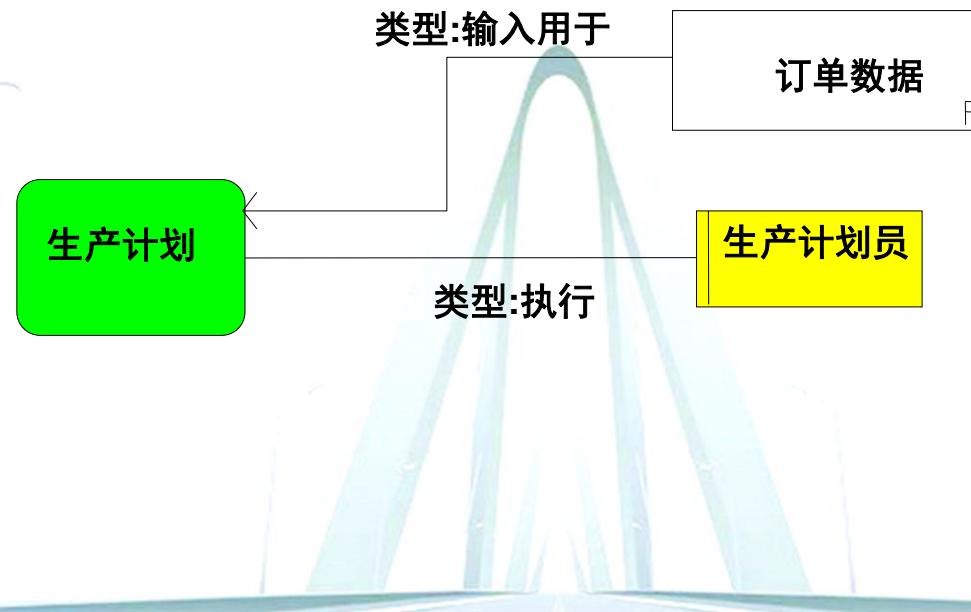


每个模型都有其具体的类型
每个模型都都有其特定的对象



通过不同类型的关系将对象联系在一起.

关系从源对象出发终止于目标对象，关系可能是直接或是非直接的。
关系的类型由源对象和目标对象符号以及模型类型所决定。





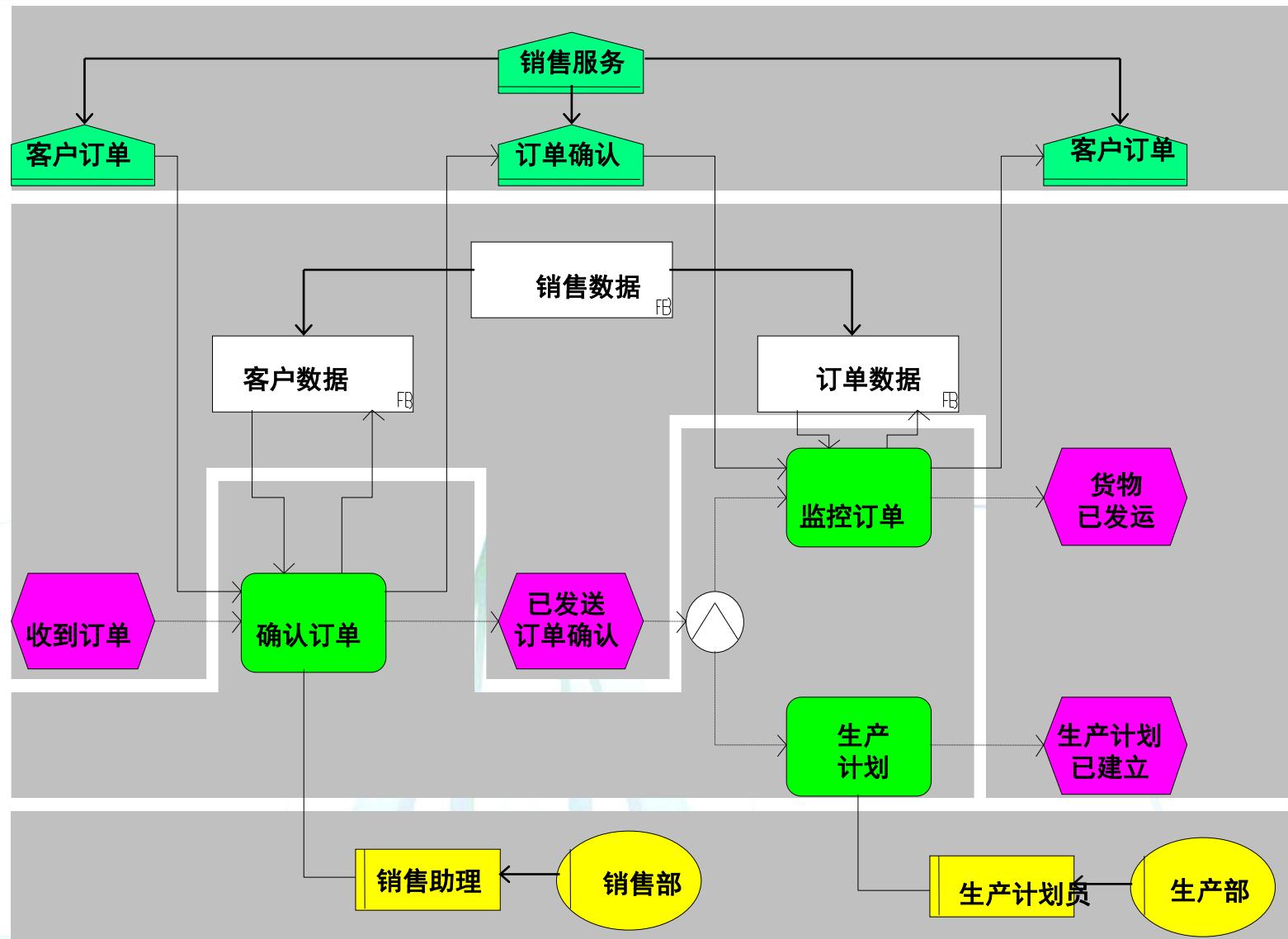
ARIS对象间的联系

产品/服务

数据

功能

组织





Data model variants

- technical term diagram
 - SAP-SERM
- IEF data model
- SeDaM method

Workflow modeling

- Workflow diagram

Core methods

- eEPC
- function tree
- Organizational chart
- eERM
- eERM attribute allocation
- information flow diagram
 - functions allocation diagram
- PCD

Analysis

- Analyzer rule diagram
- Analyzer typology diagram

Object-oriented modeling

- object model
- dynamic model
- functional model
- data value decomposition
 - class diagram

Extension of the core methods

- event diagram
- classification diagram
 - rule diagram
- value chain diagram
- goal diagram
 - Y diagram
- funct./org. level diagram
 - graphical description

Description of material flow

- Material diagram
- eEPC with material flow
- PCD with material flow

SAP-specific methods

- SAP-SERM
- SAP module diagram
- communication diagram
- Process selection matrix

Description of IT infrastructure

- relation diagram
- attribute allocation diagramm
- system attribute domain
 - system attribute
 - table diagram
 - access diagram
- access diagram (phys.)
- appl. system typ diagram
 - appl. system diagram
 - network topologies
 - network diagram
- technical resources



Agenda

1. ARIS框架
2. 基于EPC的过程建模方法
3. ARIS视图
4. ARIS建模过程实例
5. ARIS软件工具
6. 小结



- ① EPC 是一种过程建模方法，全称是Event-driven Process Chain。在九十年代初期才由Keller 等人提出，强调“过程”是由起始事件和终止事件定义的。
- ② 事件过程链（EPC）模型借鉴了ER模型和DFD建立，基本元素有**事件、过程，分支/规则等**。
- ③ 被广泛应用：
 - SAP R/3 Business Workflow (SAP AG),
 - ARIS (IDS Prof. Scheer),
 - LiveModel/Analyst (Intelicorp. Inc.)



- 在EPC出现之前，其他描述业务组织和过程的方法都太复杂了。
- 它们既不能提供并行过程的链接，也不会提供诸如组织或信息流视图这样的观察角度。
- 其他的方法也过于面向IT，或者使用了太复杂的符号。
- 另一方面，EPC方法从用户、管理人员和咨询顾问的角度出发描绘了业务信息系统，同时加入了其他一些重要特征，如组织结构、功能、数据和信息流。利用实际业务经验和易于理解的符号和语言。



eEPC的核心是几种类型的对象：

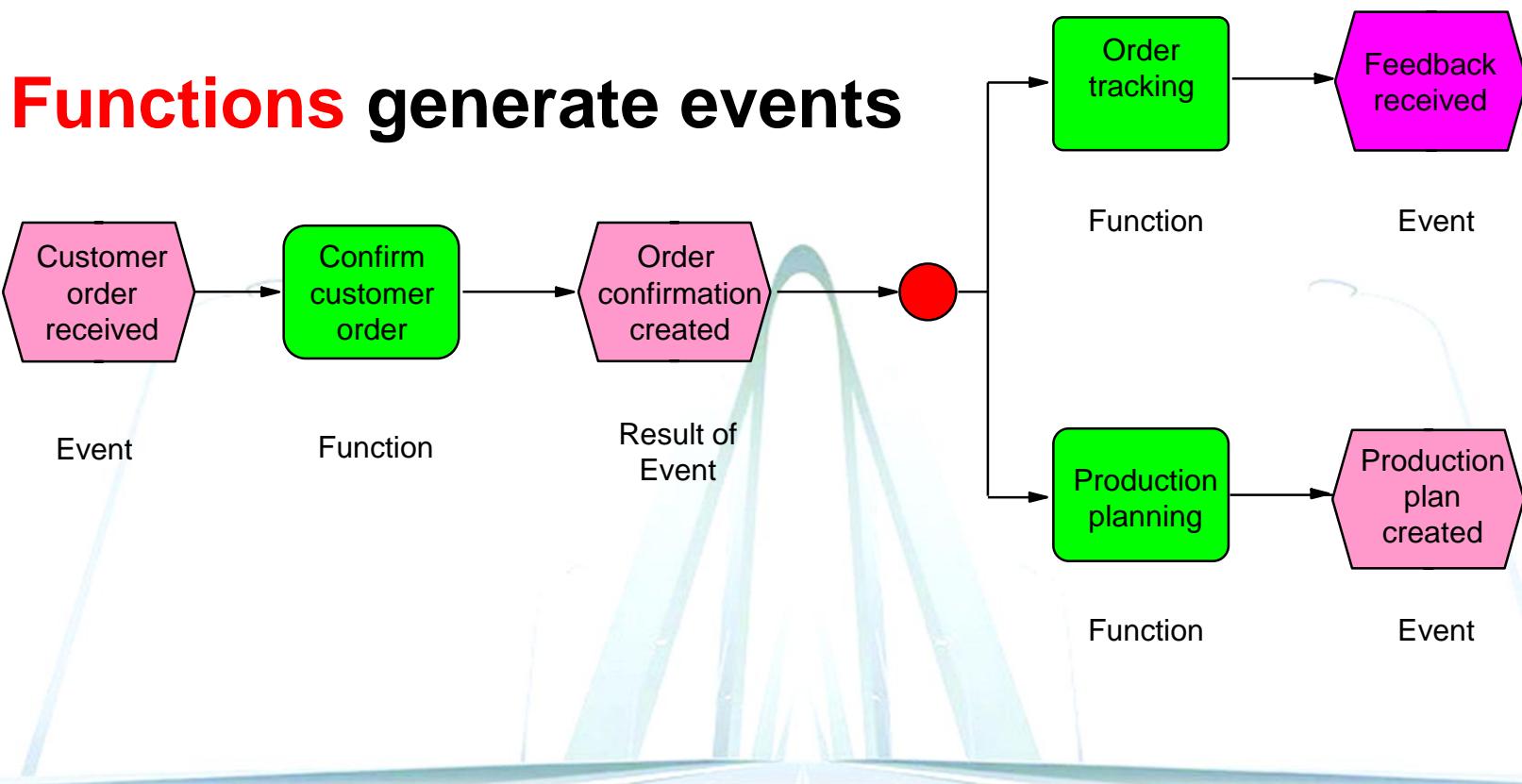
- 事件Event
- 功能Function
- 规则Rule
- 数据Data
- 人员Person



Concept

- 功能是为达到一个或多个企业目标而作用在（信息）对象上的一个任务，操作或活动。
- 一个功能可以由时间和成本来定义特性。

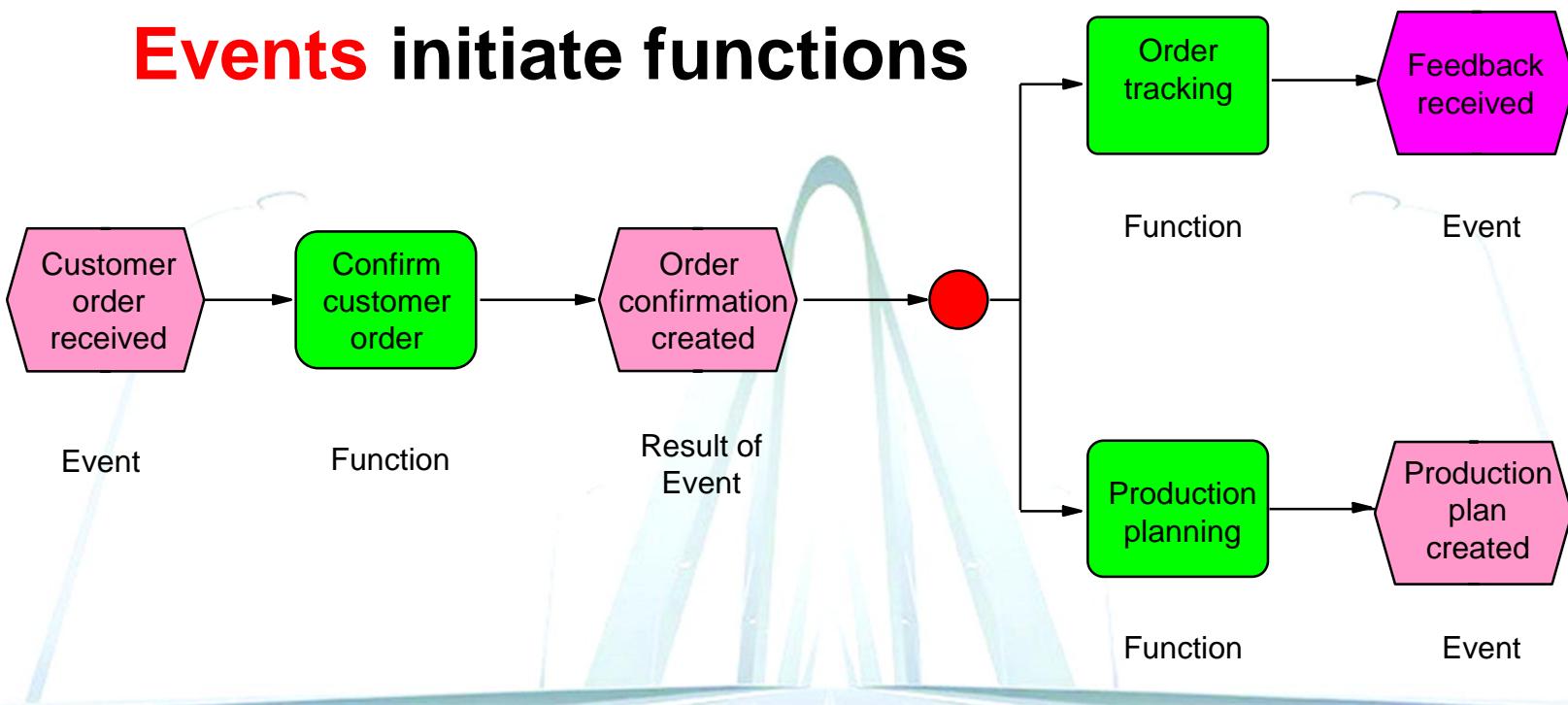
Functions generate events





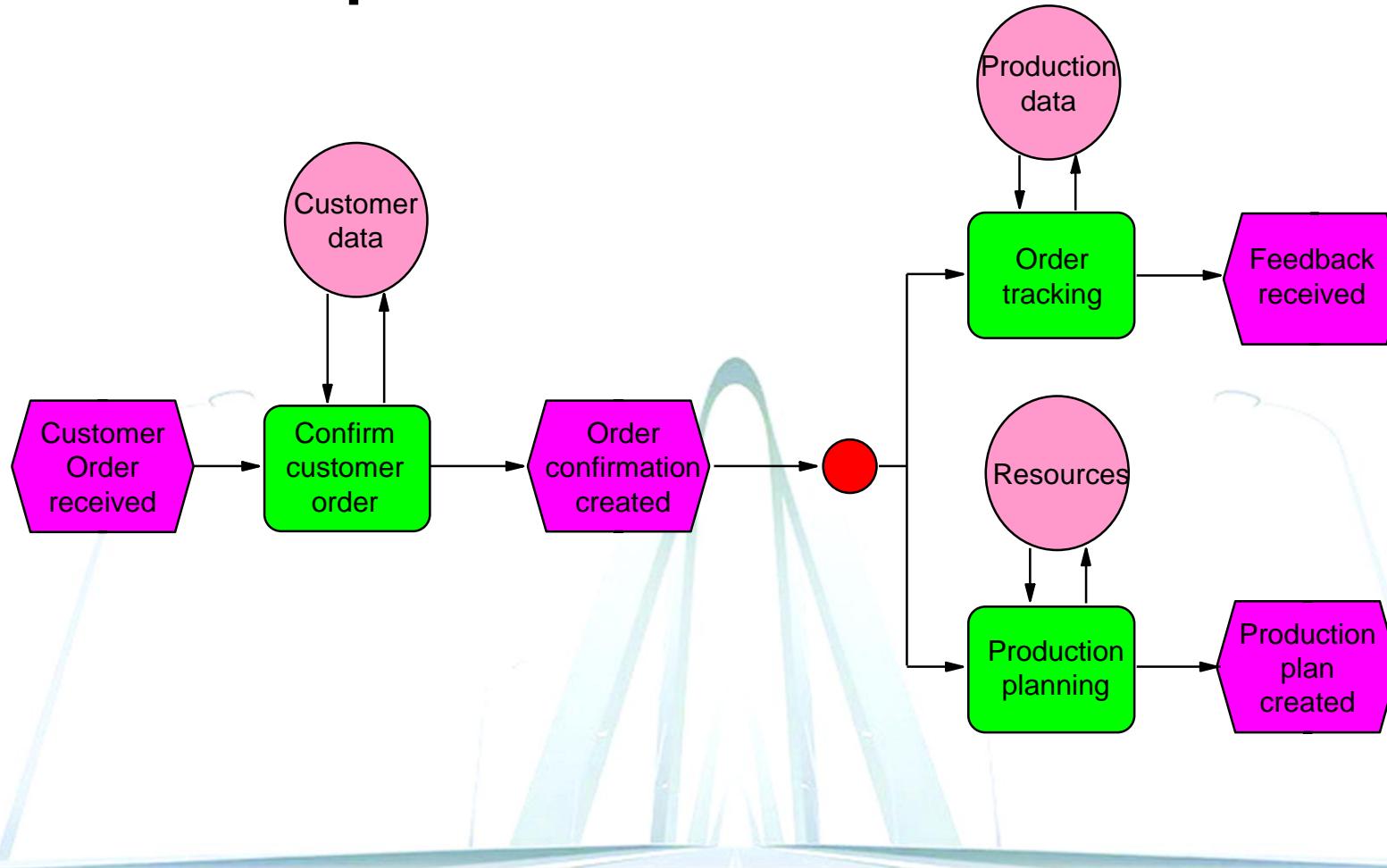
- 事件描述与业务相关的信息对象的状态。
- 某种行为的消息或请求，也可理解为某种状态的改变
- 这个状态可包括功能的pre-conditions及 post-conditions，可能控制或影响业务过程的运行。

Events initiate functions



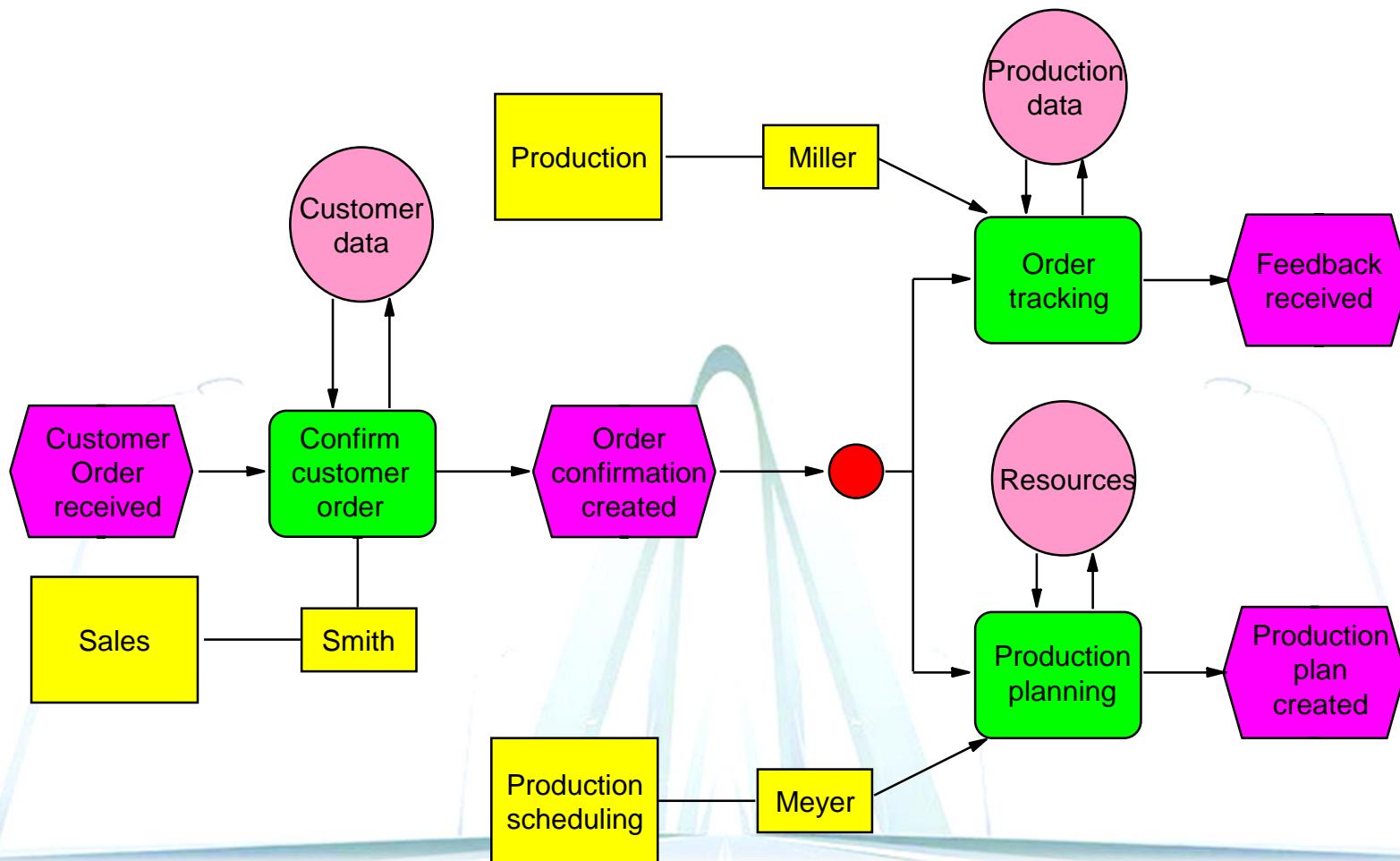


① Data are processed in functions

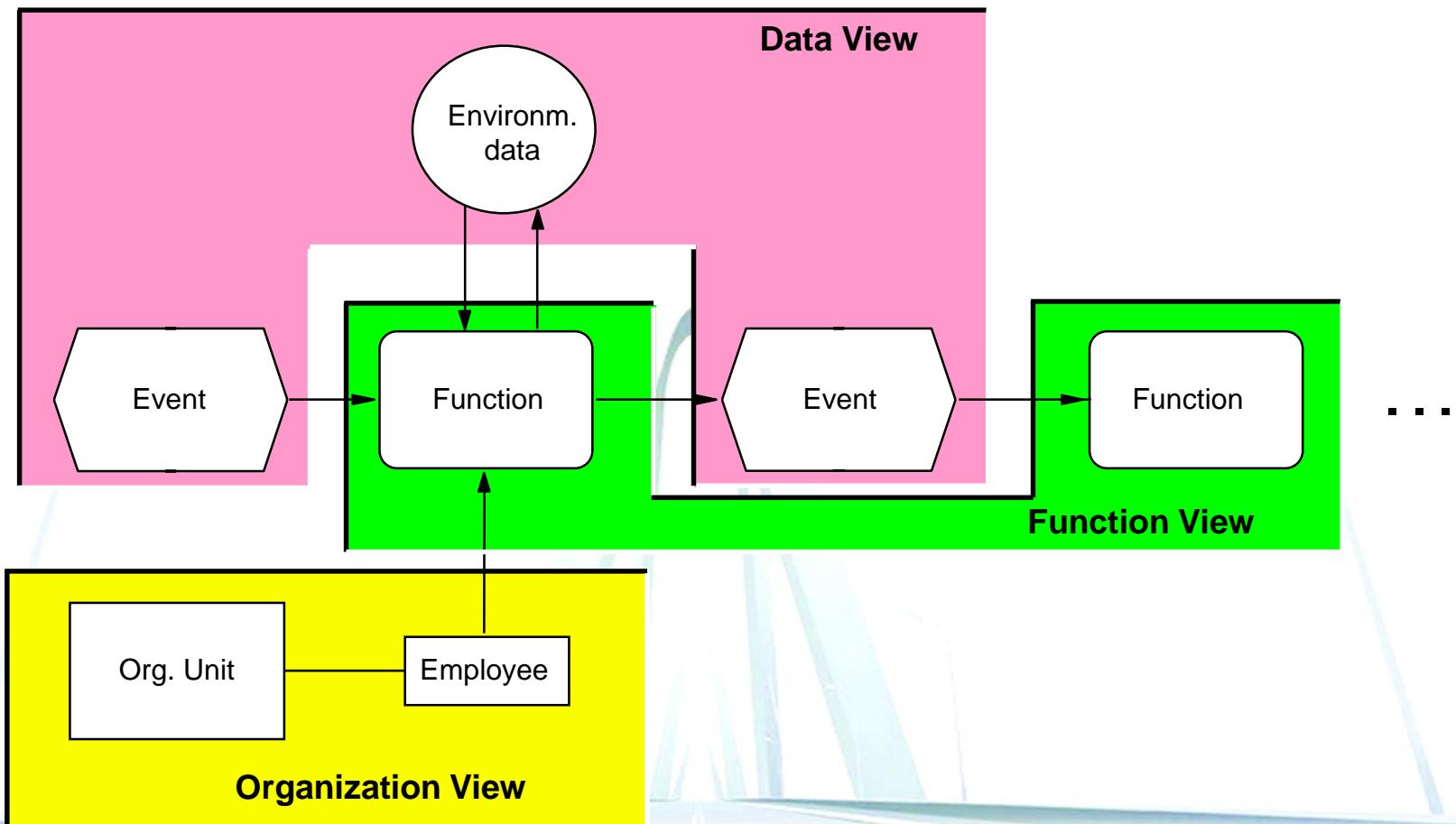




Employees/Persons are in charge of functions



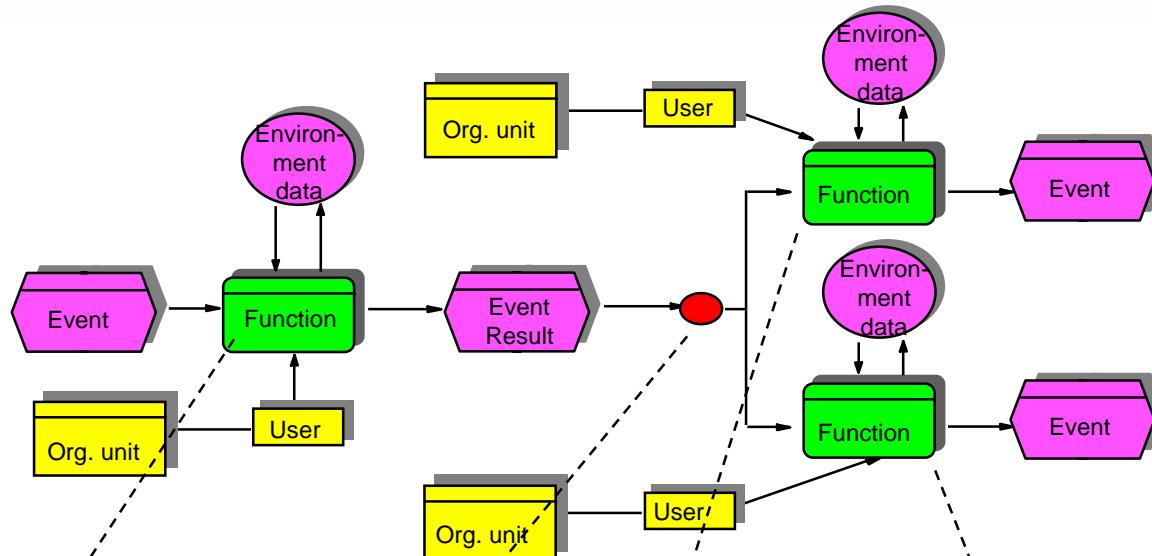
Reducing the **complexity** of business processes by creating different views



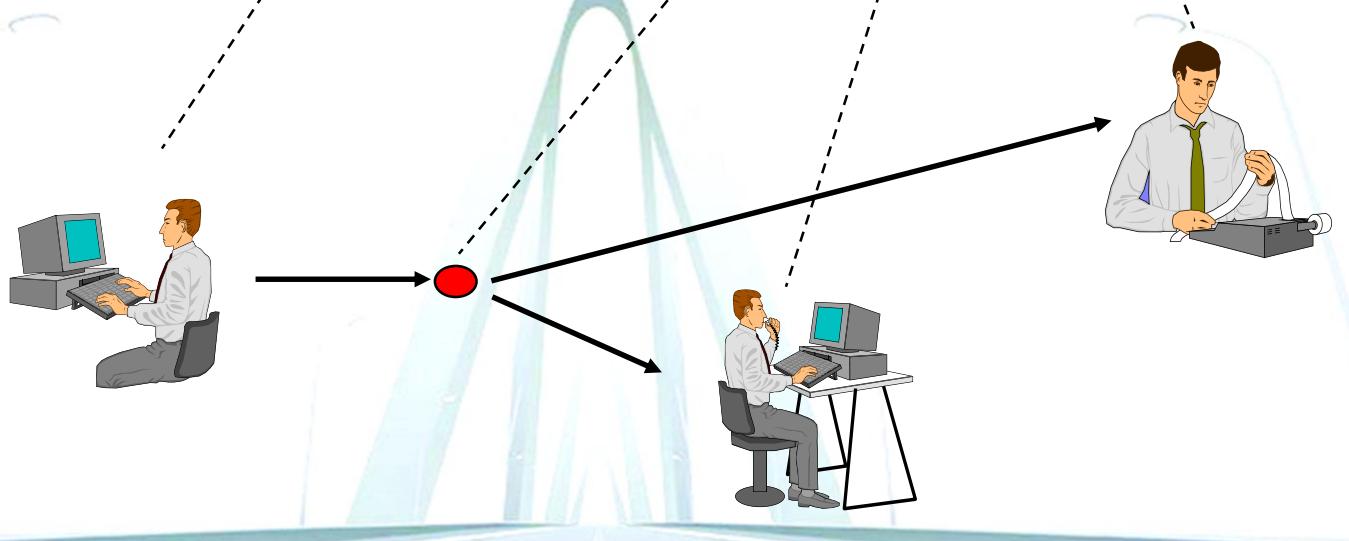


WORKFLOW

Buildtime

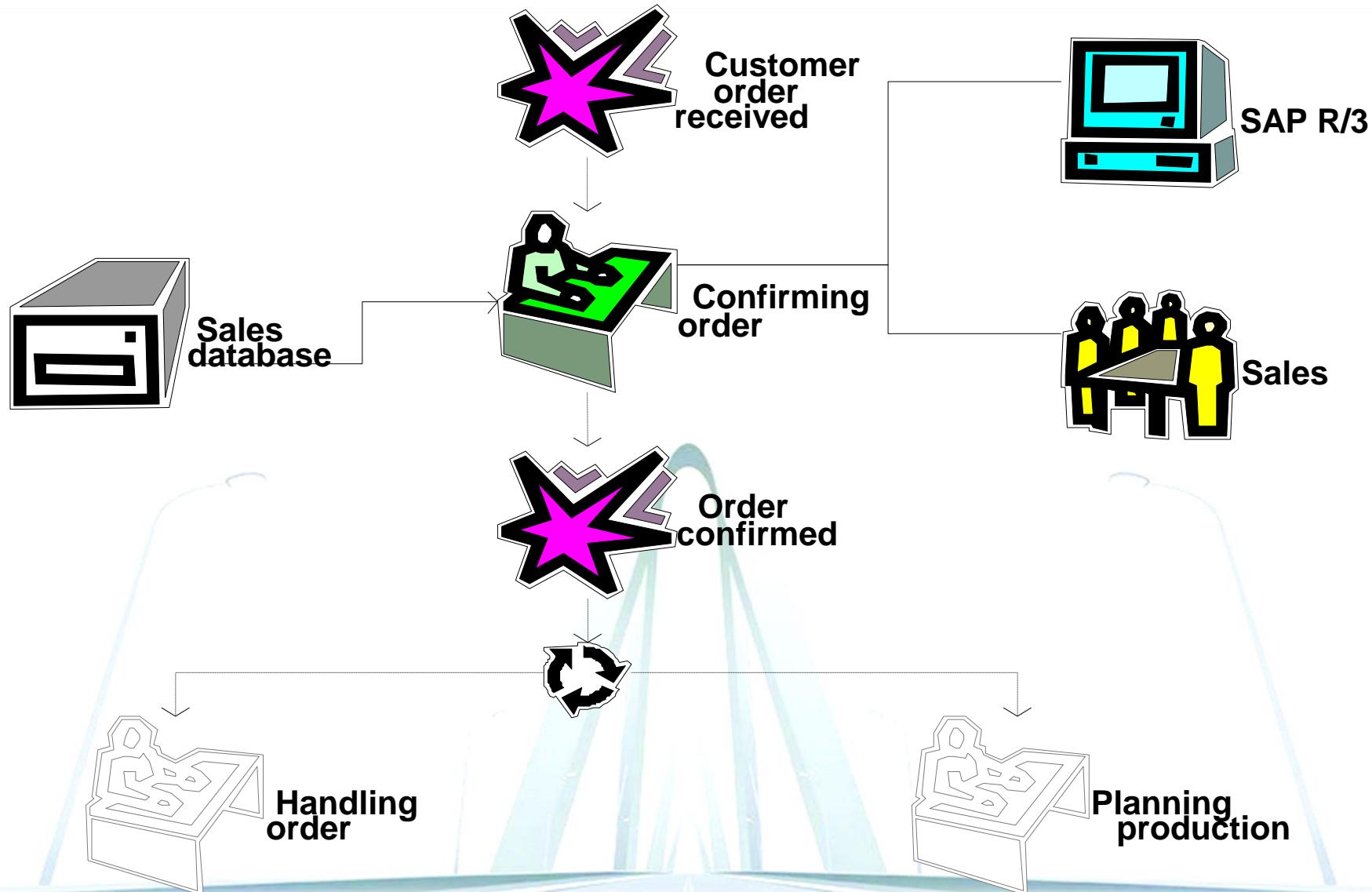


Runtime



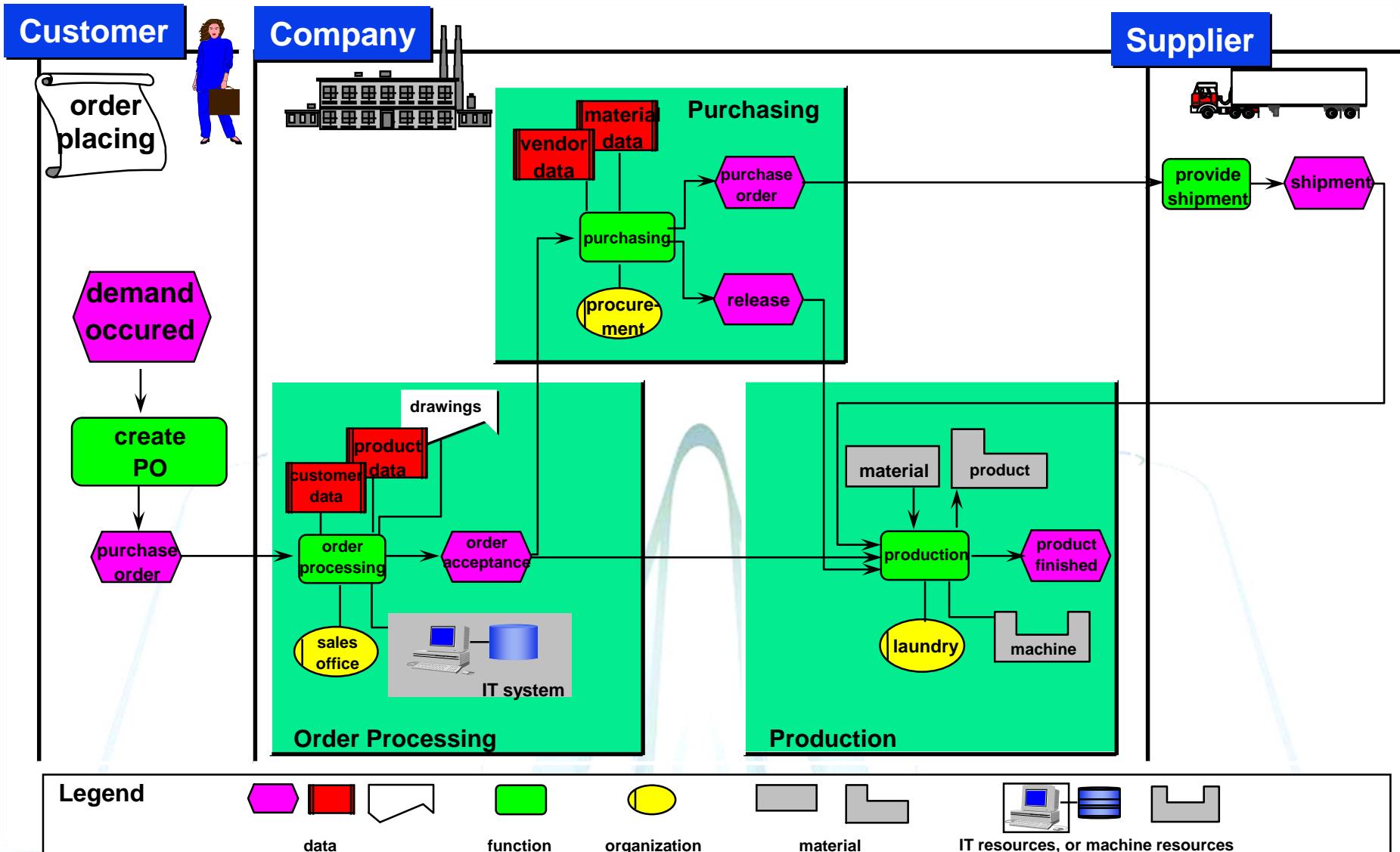


Example of an Office Process





Example: 以功能活动为中心的应用实现





2.2 EPC基本语义

EPC
Function

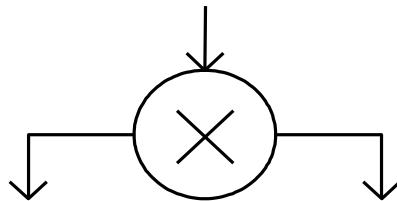
Functions:

activities of the process

EPC
Event

Events:

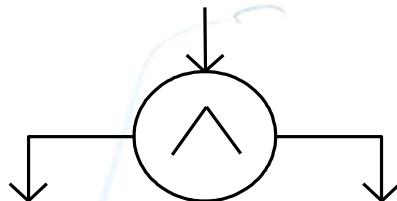
pre- and post-conditions
of functions



XOR
Split

XOR split:

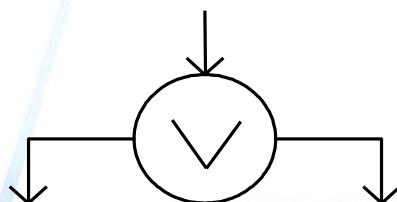
defines a choice to activate one of multiple subsequent branches.



AND
Split

AND split:

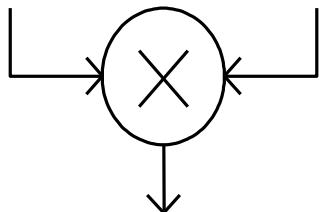
activates all subsequent branches in concurrency



OR
Split

OR split:

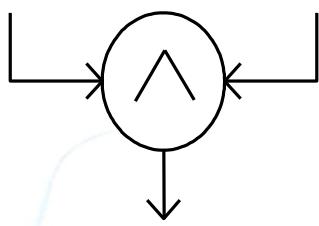
triggers one, two or up to all of multiple subsequent branches.



XOR
Join

④ XOR join:

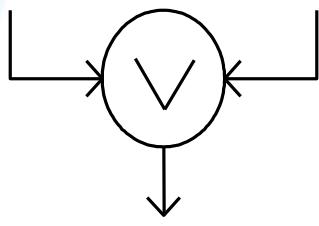
continue when one of alternative branches has completed.



AND
Join

⑤ AND join:

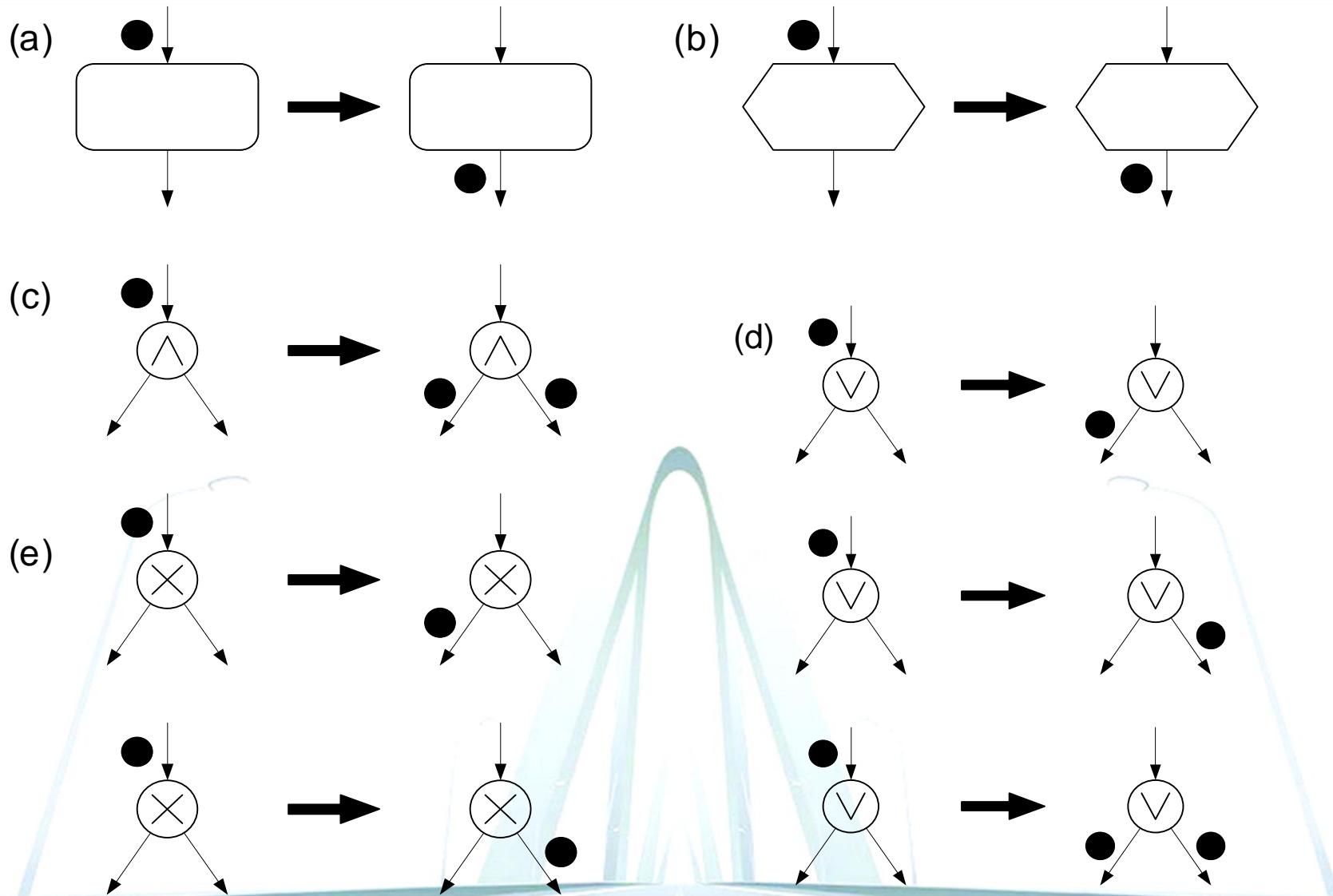
waits for all incoming branches to complete. 进入分支(incoming branch)



OR
Join

⑥ OR join :

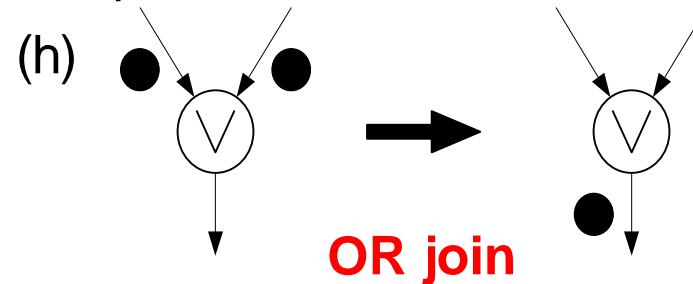
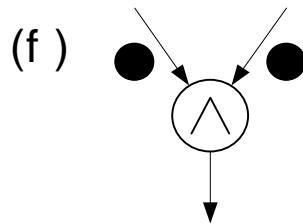
waits for all active branches to complete.





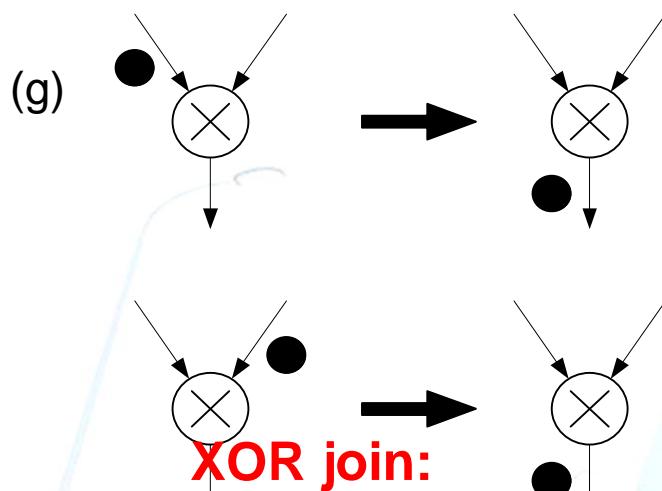
AND join:

waits for all *incoming branches* to complete.



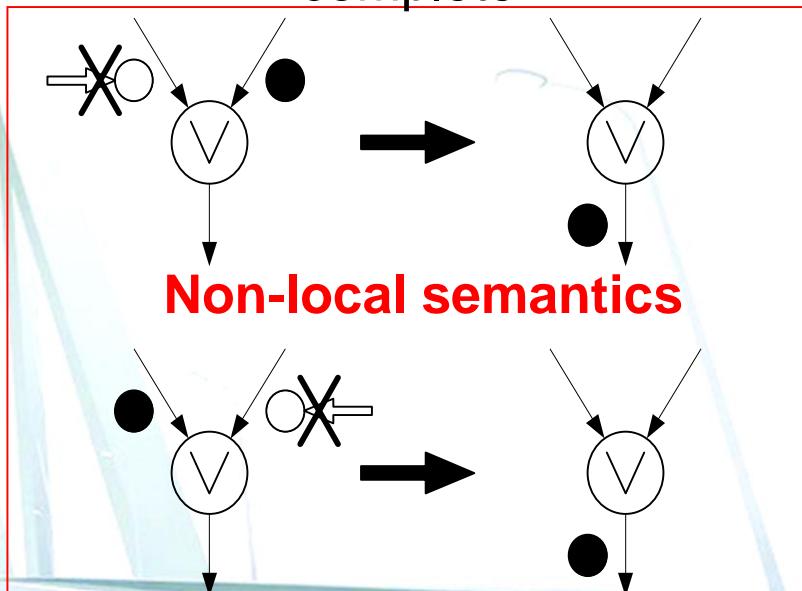
OR join

waits for all *active branches* to complete.



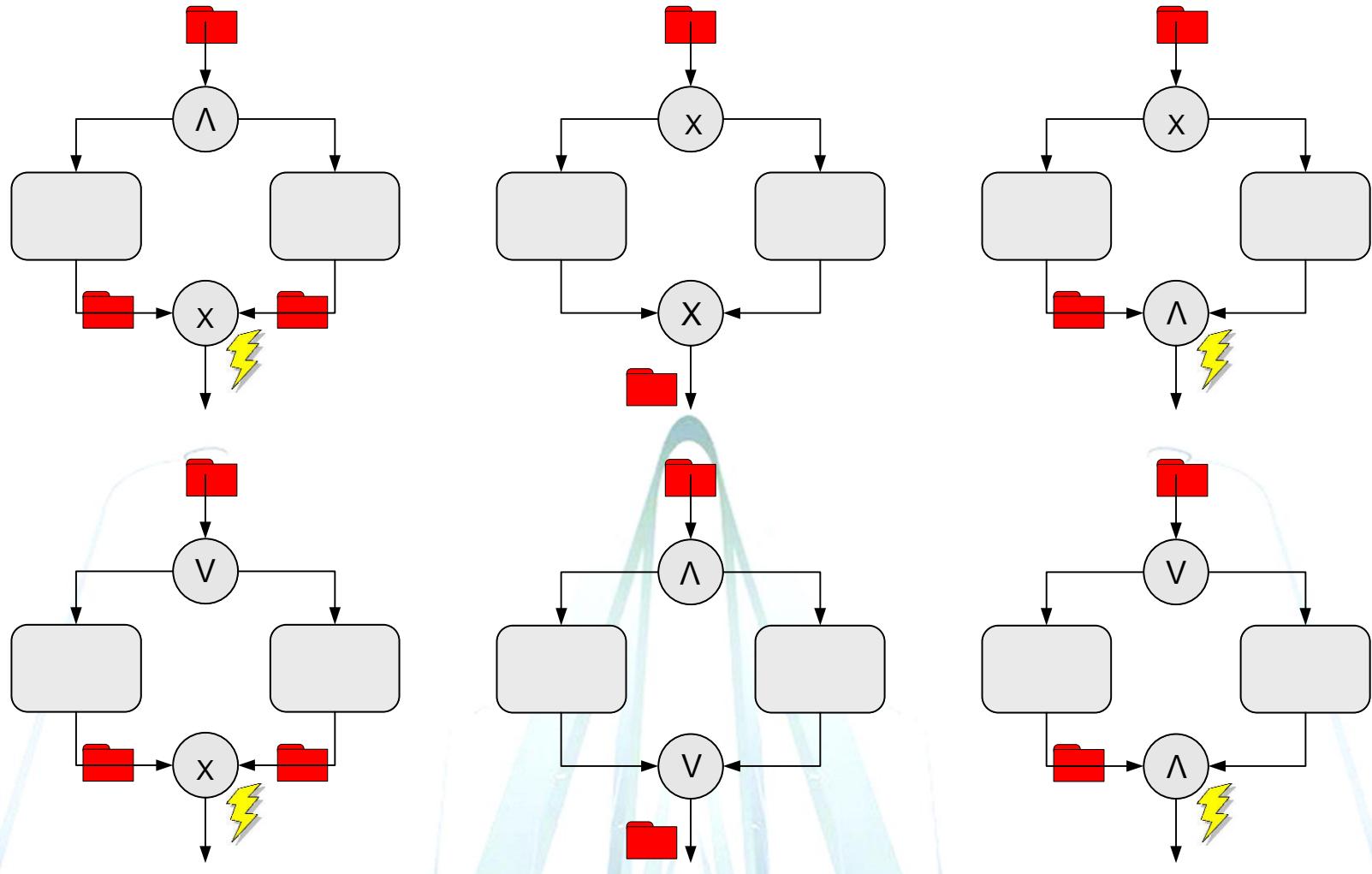
XOR join:

continue when one of alternative
branches has completed.





不正确的匹配



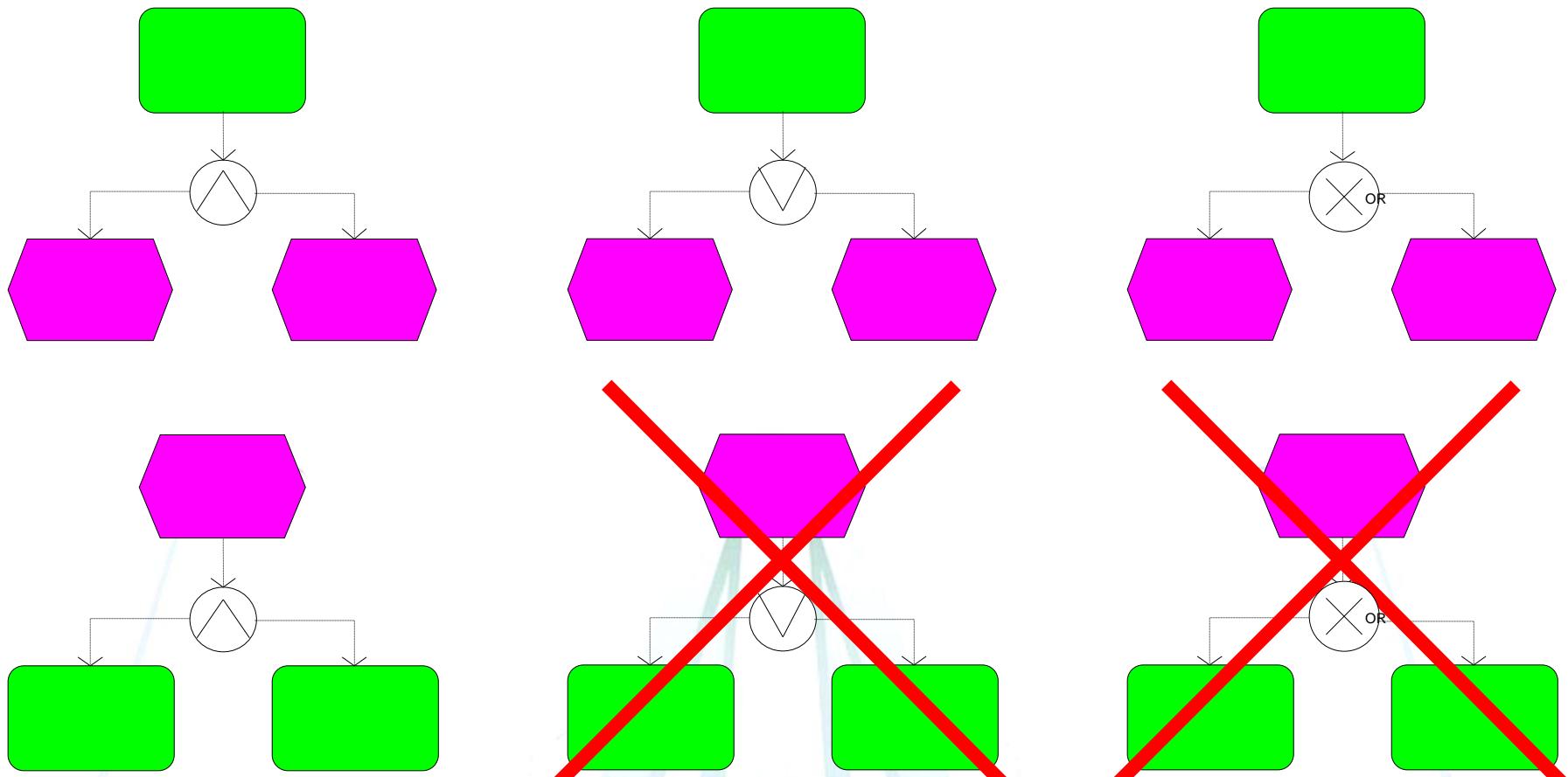


2.3 Rules

- Rule1: An EPC model has to **start with an event**;
- Rule2: An EPC model has to **end with an event**;
- Rule3: Functions and Events should be **alternated**;
- Rule4: When connecting functions and events, such nodes can only have **one input and one output**;
- Rule5: An Event is a **passive element** that does not have any decision competency!
 - This rule primarily affects the usage of **logical operators** and the connection of functions and events based on this connection.



Rules for Branches



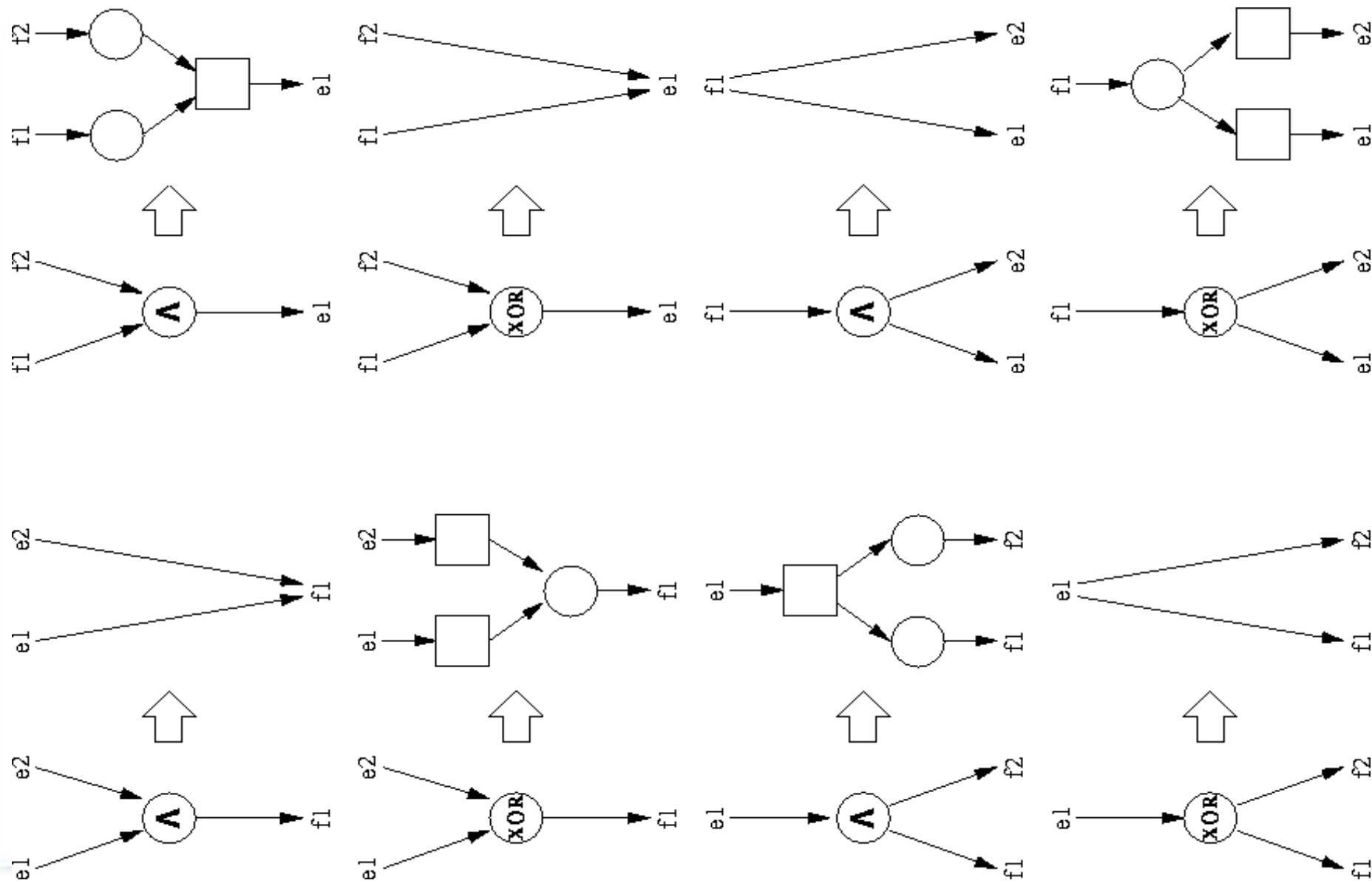
Do not use OR / XOR after events!!



- ④ Event 可以映射为PN 中的库所（Place）
- ④ Function 可以映射为PN 中的变迁（Transition）。
- ④ 至于逻辑连接符AND 和XOR 的映射在采用Place 的控制，很容易表示。
- ④ 因为OR 的语义有多种诠释，不是非常清晰明确，因此不涉及到对OR逻辑连接符的映射。
 - OR split:
 - OR join:



EPC 到 PetriNet 的直接映射





2.4 扩展EPC

- eEPC 的四个主要要素：事件(Event)、功能(Function)、组织单元(Organization Unit)、信息(Data).

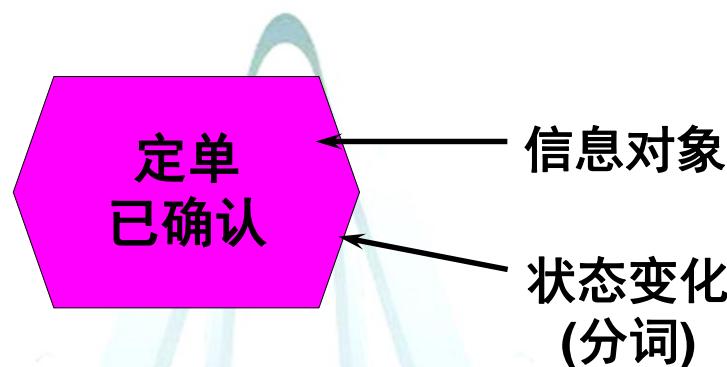
| 元素 | 图例 | 描述 |
|--------------------|----|--|
| Event | | 描述了状态的发生, 同时又充当了一个触发器 |
| Function | | 功能描述了一个任务的执行, 代表了一个 start event 和 end event 转换过程 |
| Organization Unit | | |
| Data | | |
| Process Path | | 流程之间的连接关系 |
| Logical Connectors | | 逻辑连符号: AND,OR,XOR |



扩展事件驱动过程链(eEPC)

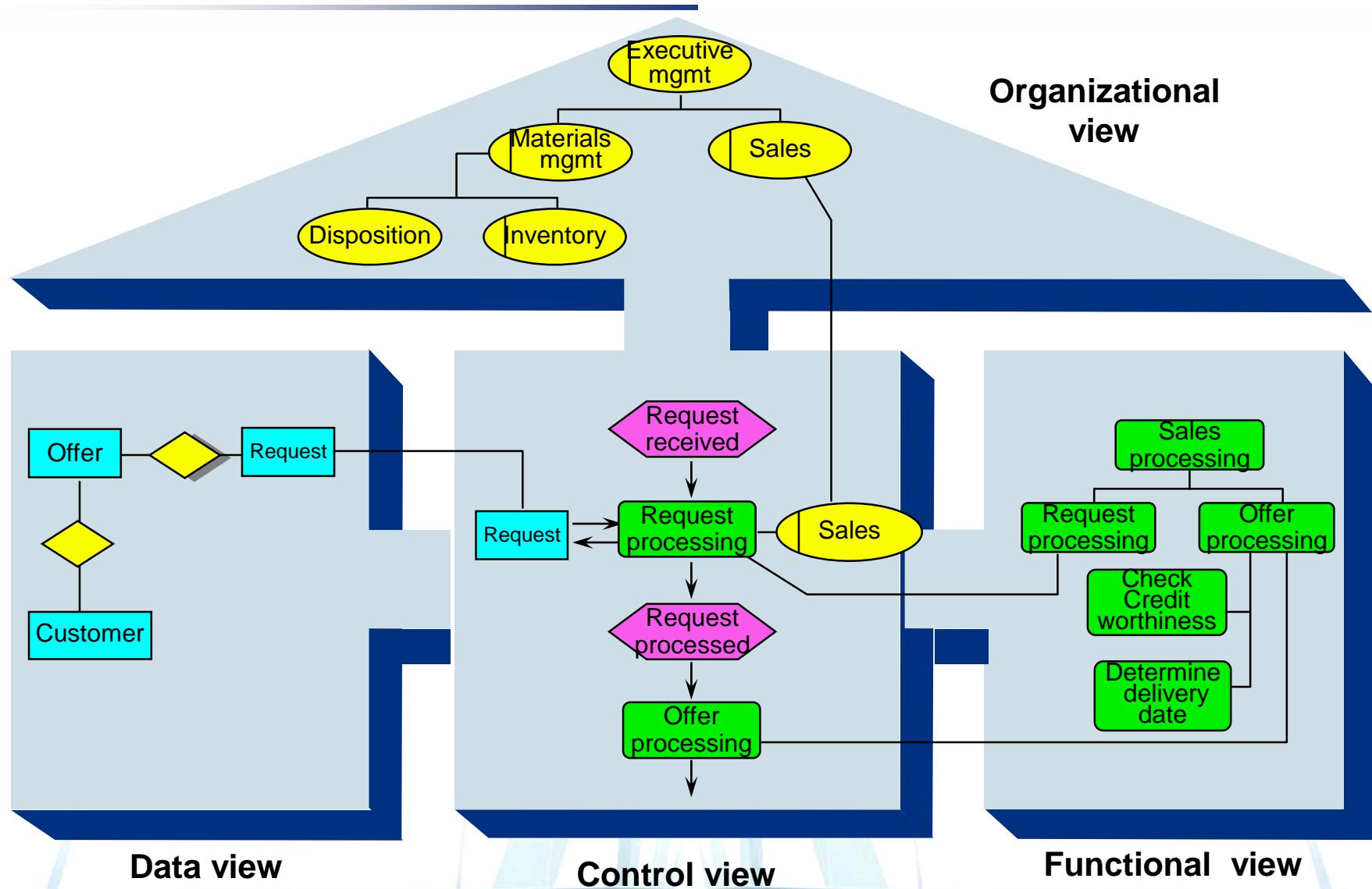
ARIS在事件过程链EPC之上引入组织和信息建模元素，可建立扩展事件驱动过程链(eEPC)。

- 其中，事件描述与业务相关的信息对象的状态。这个状态可能控制或影响业务过程的运行。
- 图形表示：



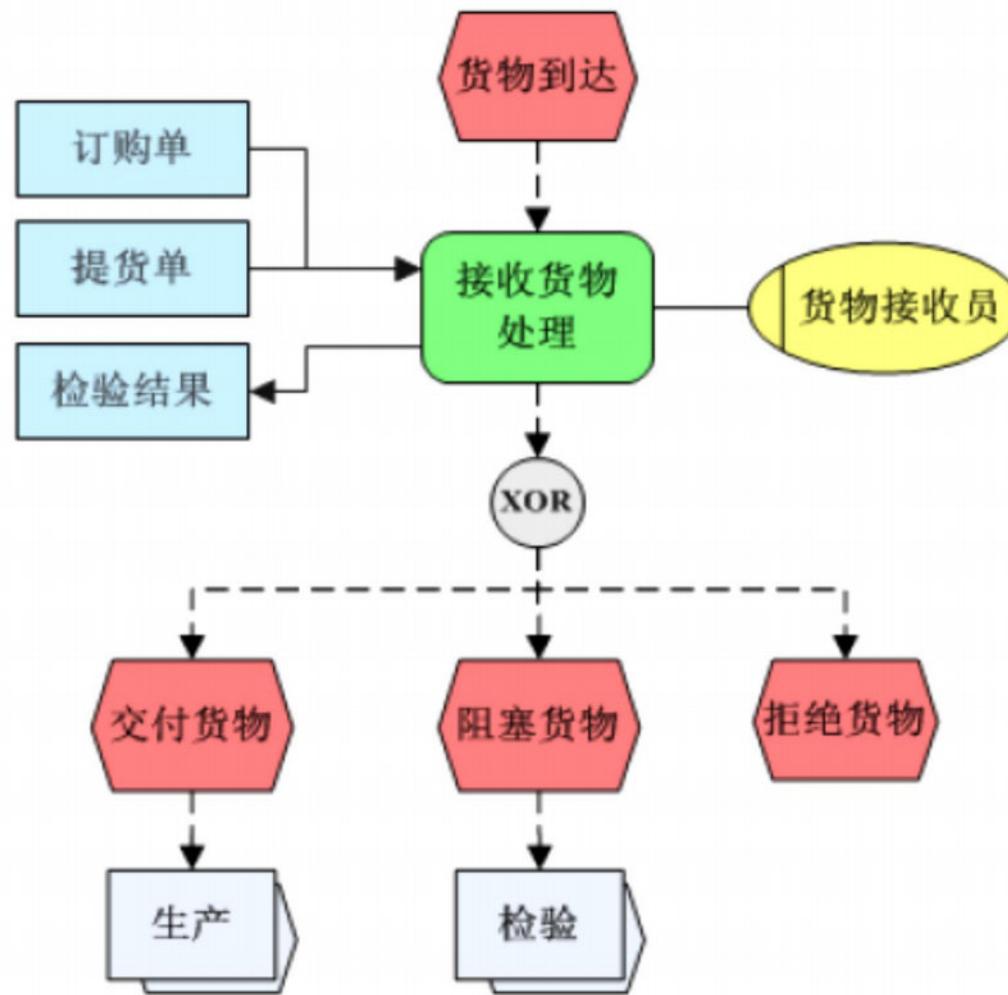


ARIS对EPC的扩展



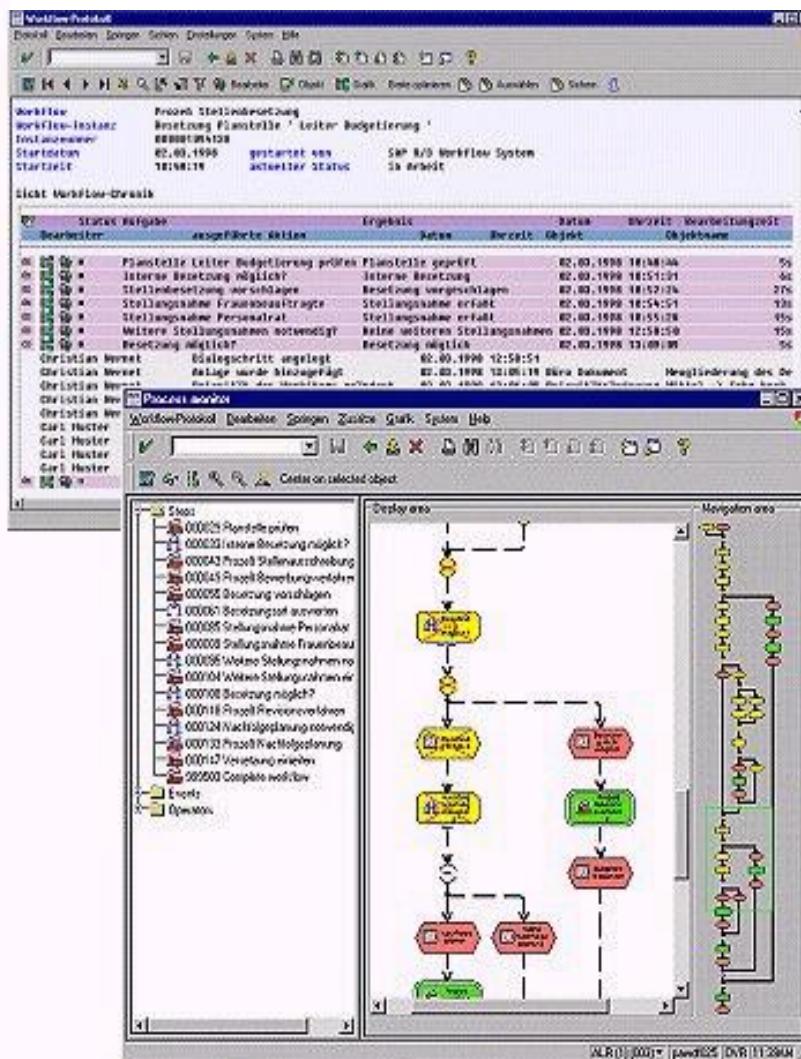
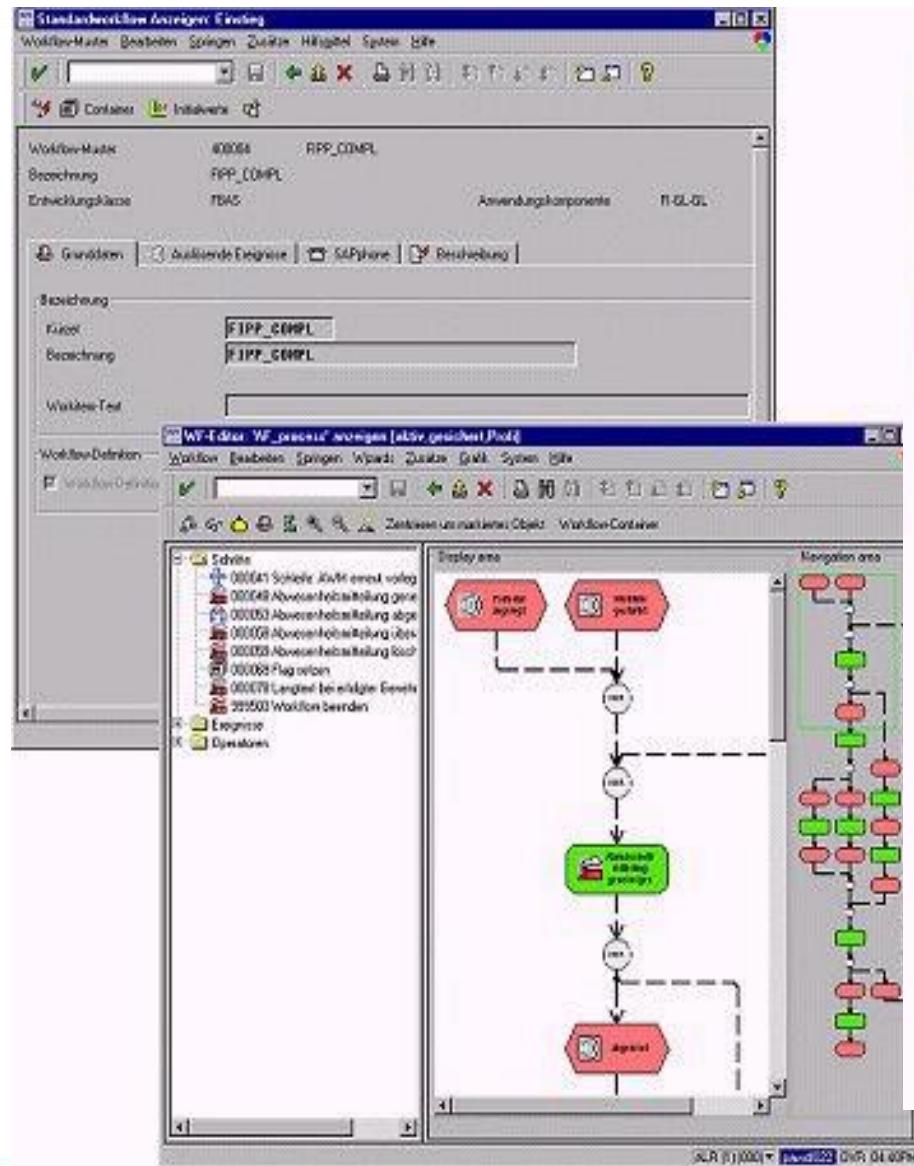


货物接收处理的流程例子





SAP 中的 EPC





Agenda

1. ARIS框架
2. 基于EPC的过程建模方法
3. ARIS视图
4. ARIS建模过程实例
5. ARIS软件工具
6. 小结

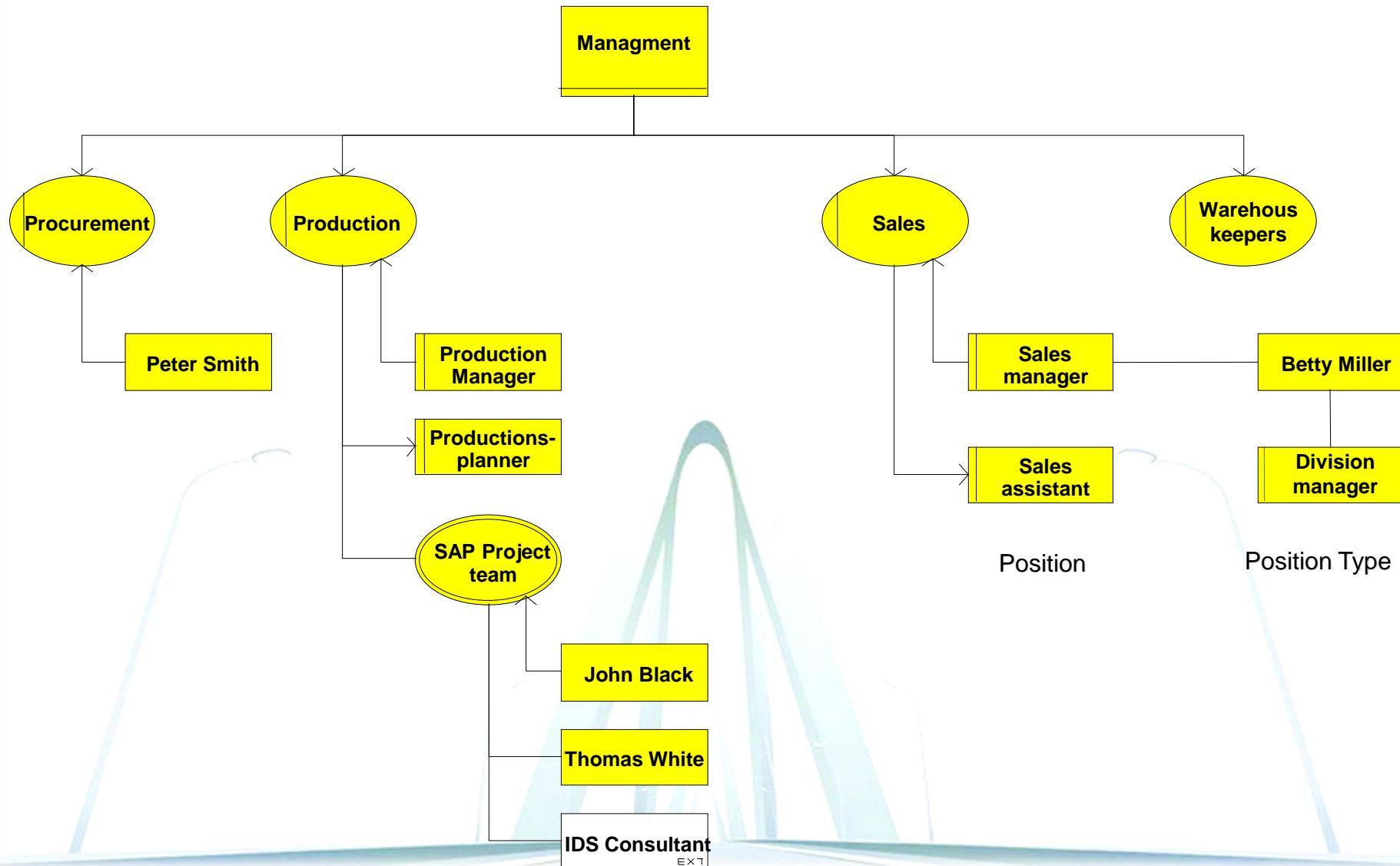


- ARIS体系被分解成4个视图：**数据视图、功能视图、组织视图和控制视图。**
 - 功能视图定义了控制模型中的所有功能以及功能之间的层次关系。
 - 数据视图定义了过程中所用到的数据及数据间的关系。
 - 组织视图对人员分工、关系、位置和作用以及权限、义务等进行了描述，组织视图包含企业内部所有相关的职责信息。
 - 控制视图定义了任务的实现，包含了系统中各功能之间逻辑关系、因果关系、事件关系等。



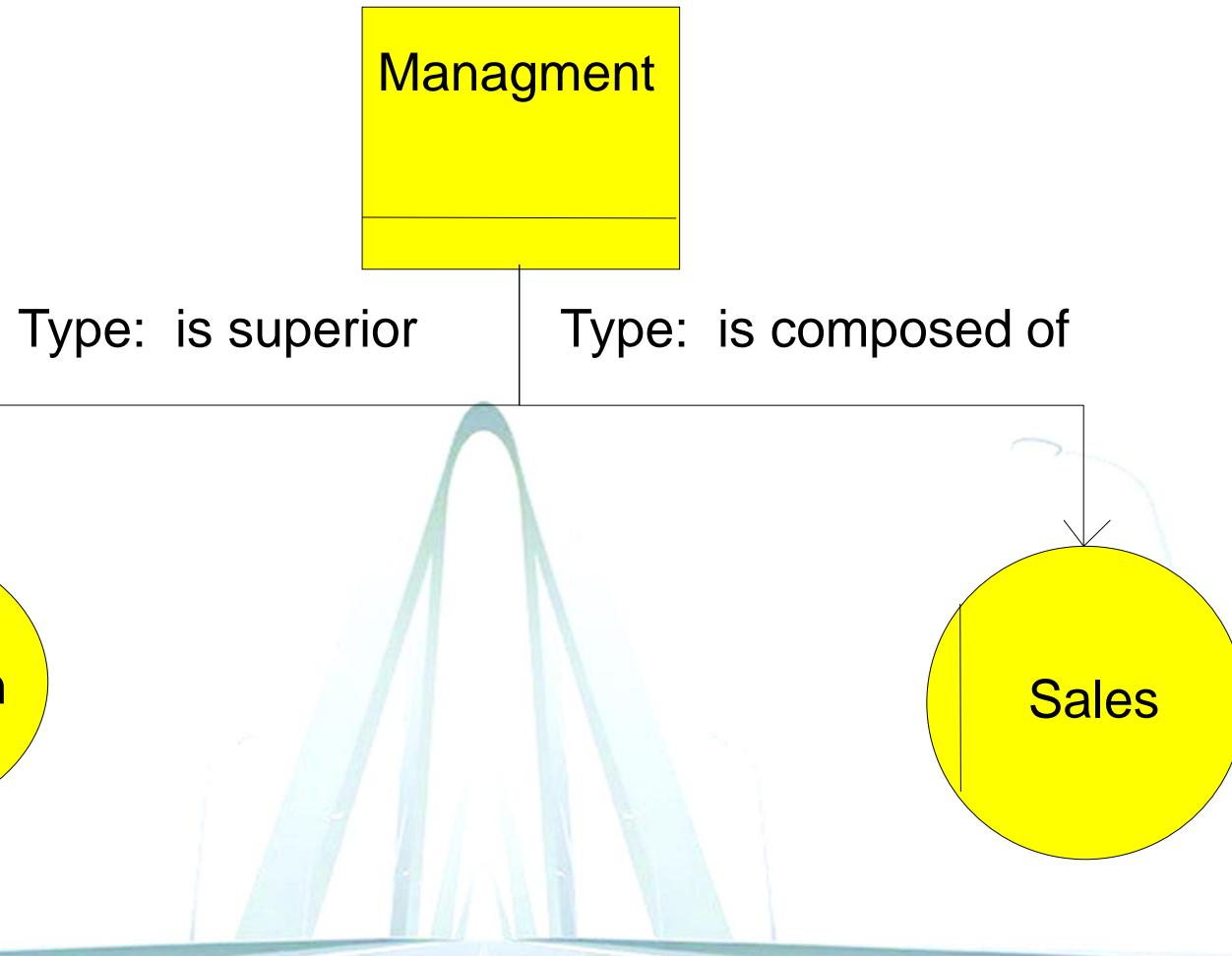
3.1 组织视图

- ① 组织视图描述了公司任务操作者之间的静态关系, 使用组织机构图(organizational chart)的方式描述组织单元、组织的分解关系、层次结构、隶属关系等。
- ② organizational chart基本元素除了组织单元外，还包括人员、职位、场所等图元，全面描述企业组织结构的各种信息。
 - **Organizational units** links the task performers with human performance of work.
- ③ 但是，ARIS的组织视图在实施描述阶段仅提供建立企业网络拓扑的信息，和组织建模是有一定差异的。





Organization unit

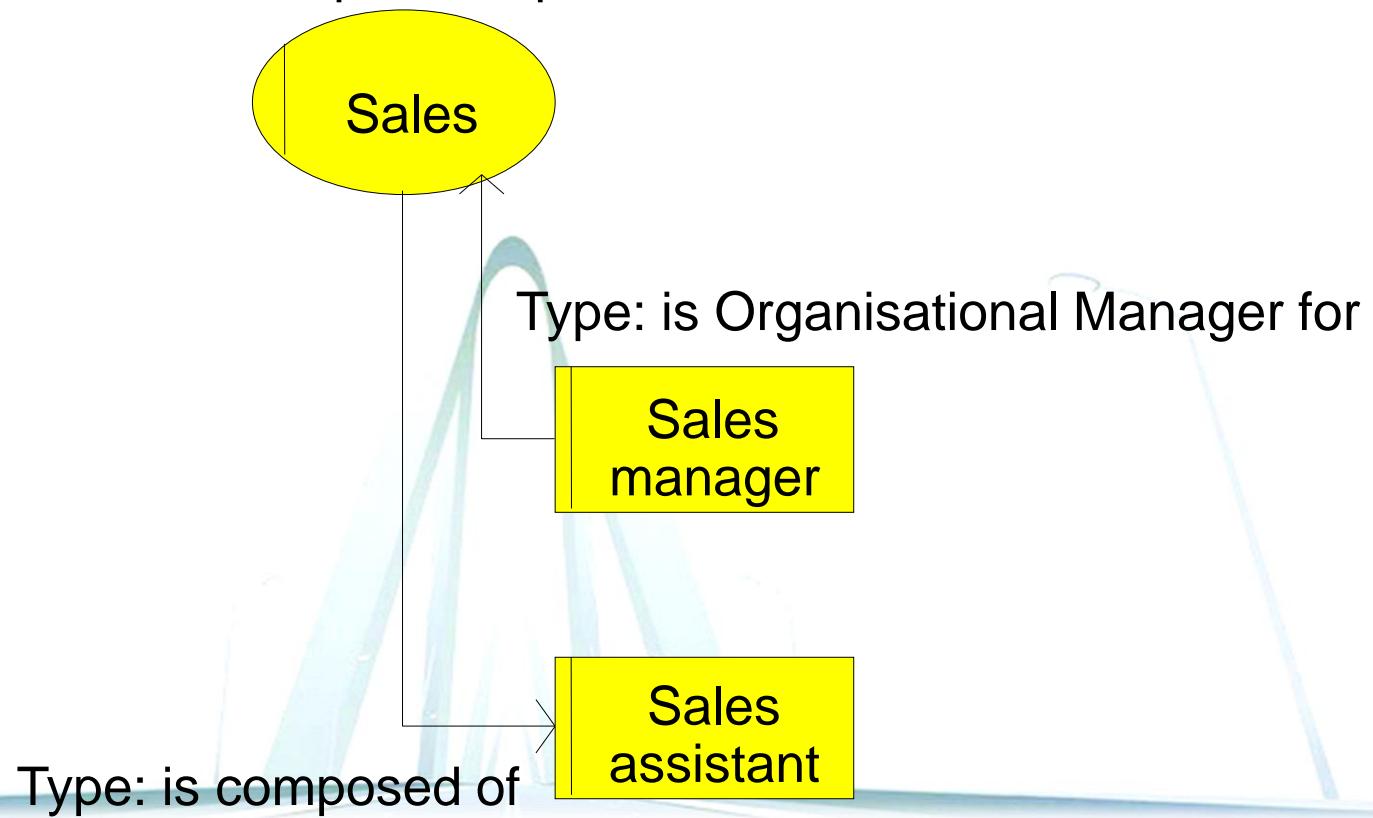




Positions

- A **position** is the smallest organizational element within a company. The responsibilities and the managerial authority are defined in the corresponding job description.

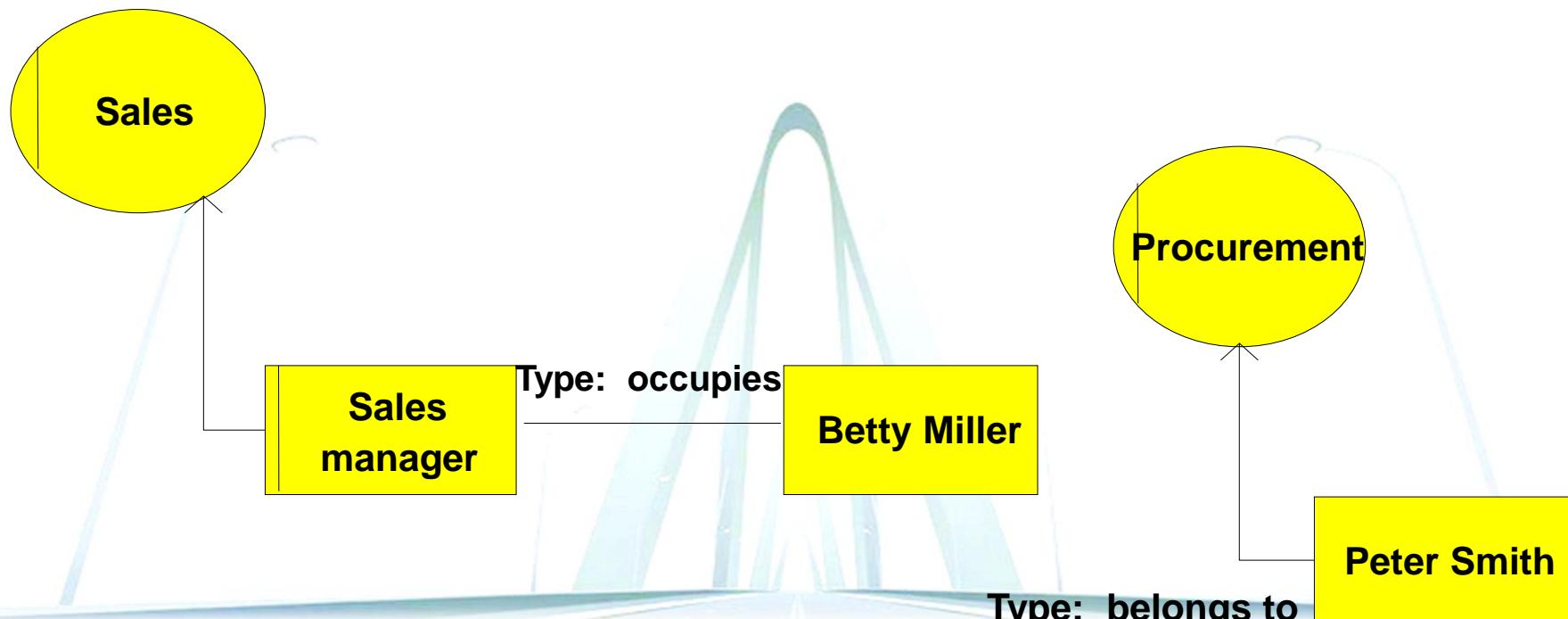
- Graphical representation:





Personnel

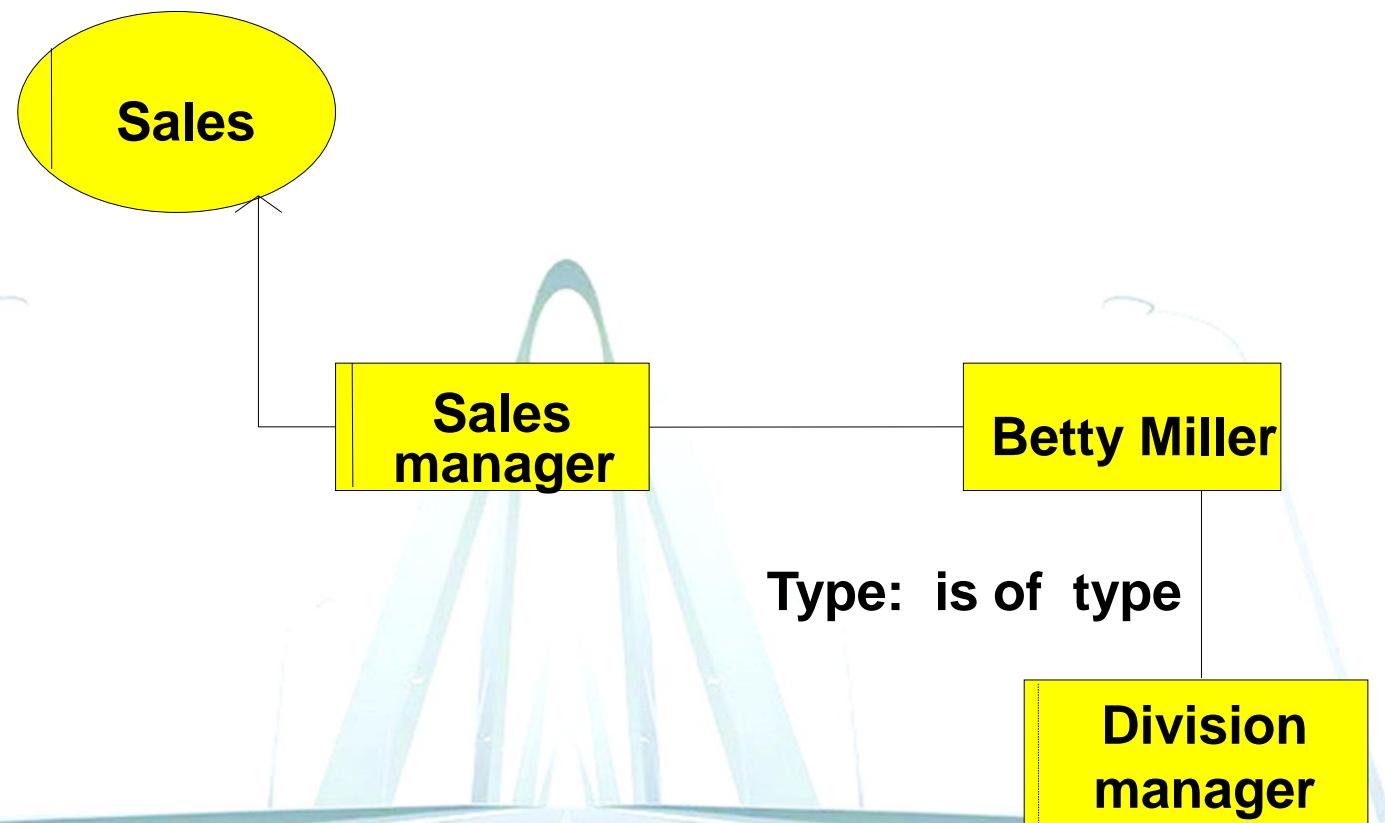
- **Persons** are the concrete employees of a company who have been assigned a unique personnel code.
Persons or employees either can be allocated to positions or directly to organizational units.
- Graphical representation:





Personnel Types

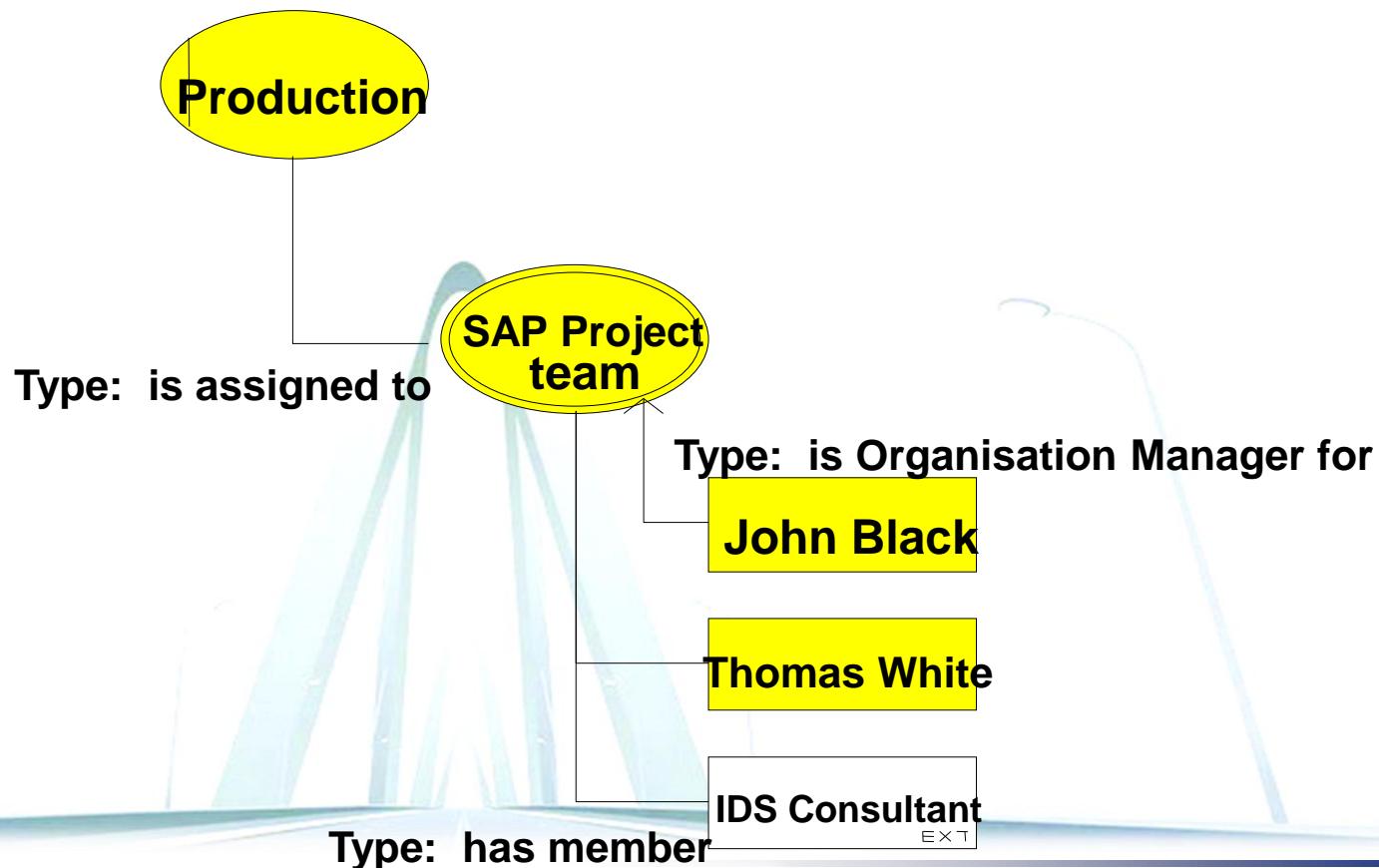
- A **person type** typifies single employees who have same properties (i.e. rights or responsibilities).
 - Graphical representation:





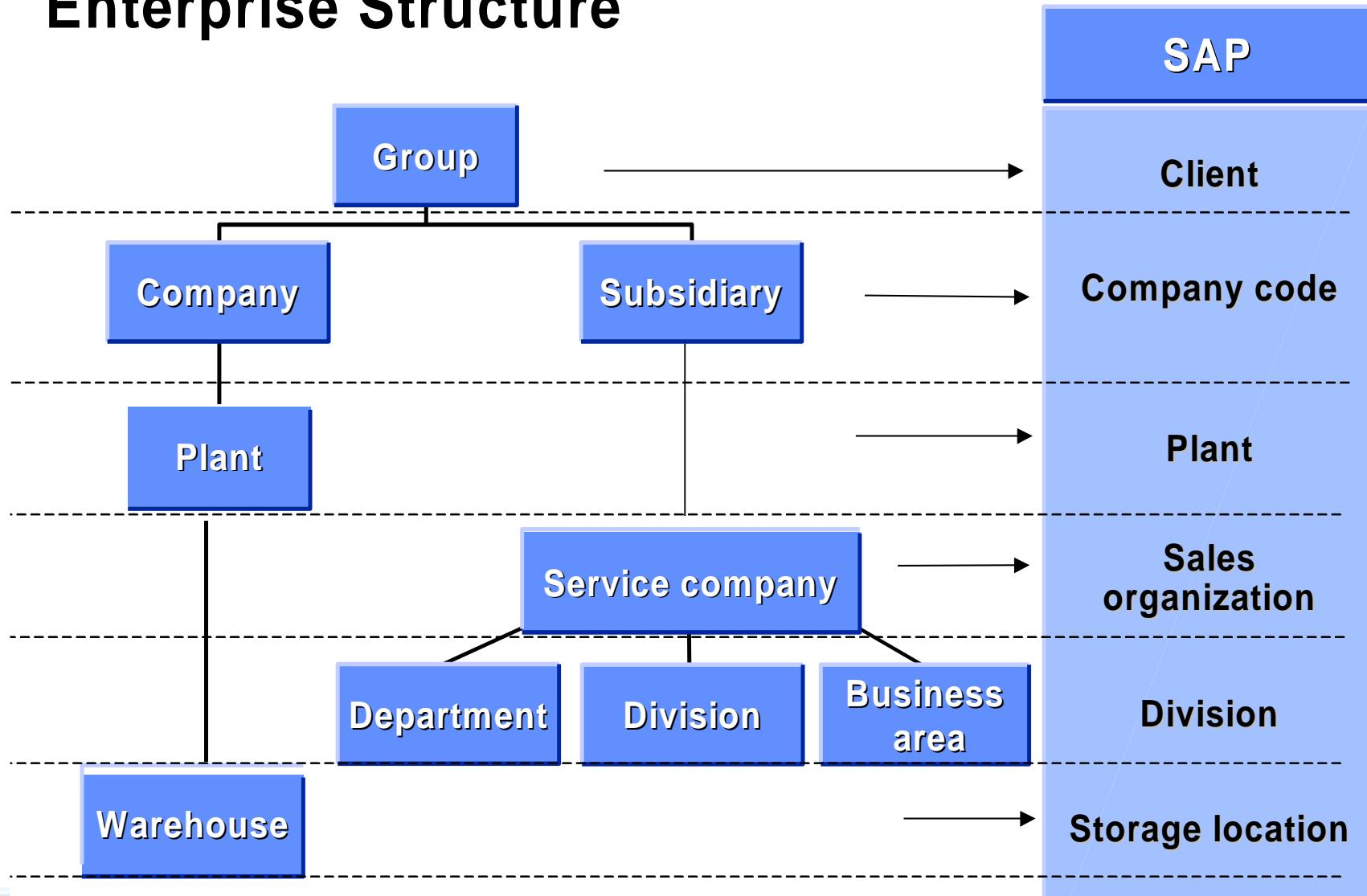
Groups

- A **group** represents a grouping of employees / persons who work together on a specific task for a limited period of time (e.g a project team).
- Graphical representation:





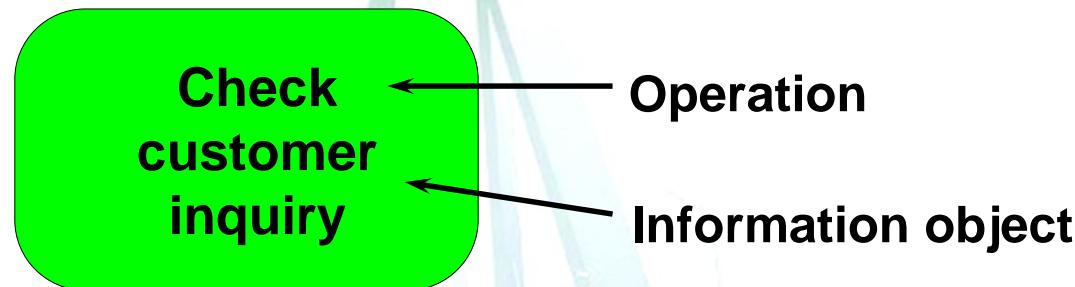
Enterprise Structure





3.2 功能视图

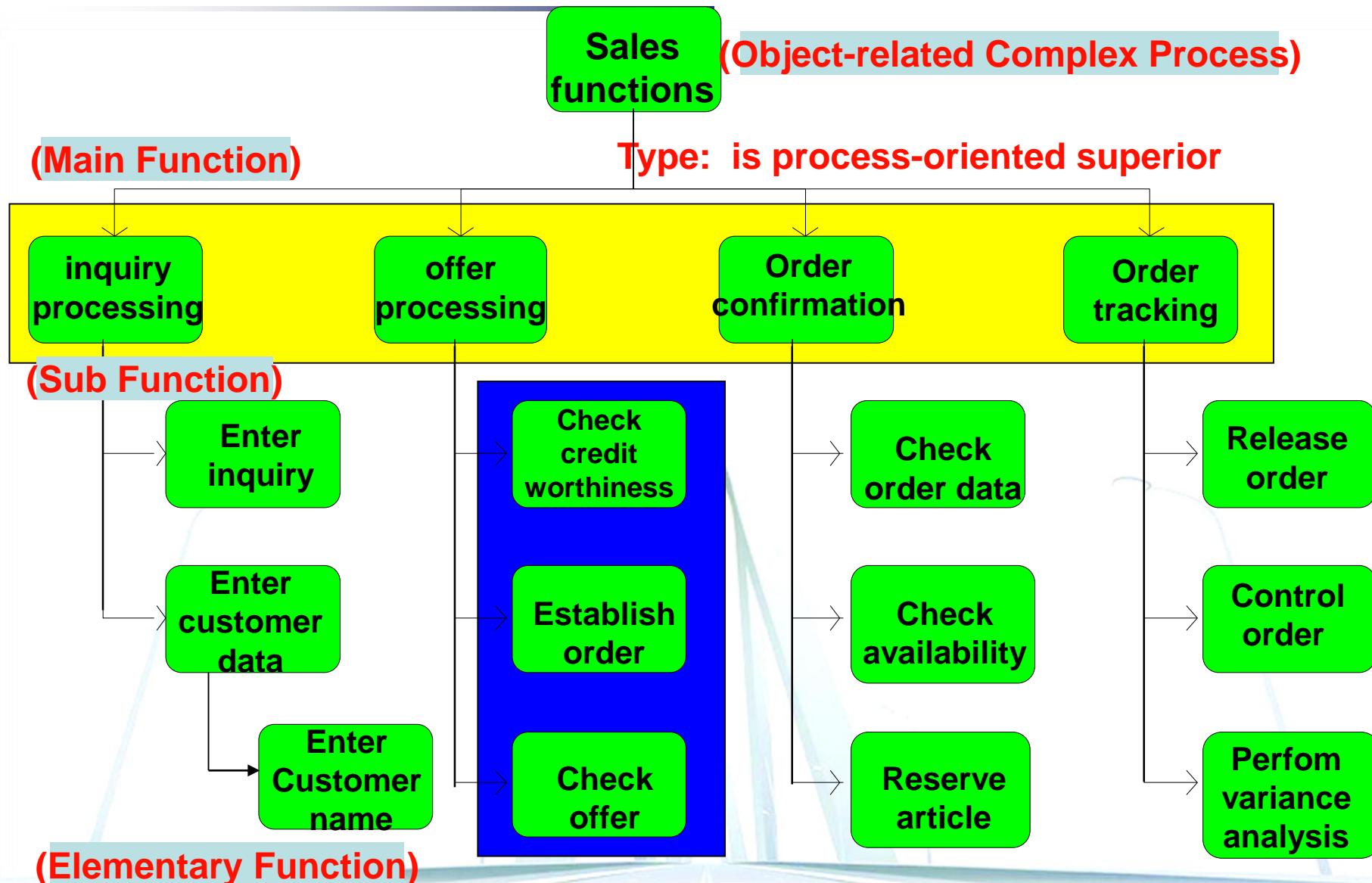
- ④ ARIS功能视图的主要形式是以功能树形式描述企业功能分解的层次关系的配置图。与IDEF0不同的是，ARIS功能视图不描述功能模块间的输入、输出、控制和支撑关系。
- ⑤ ARIS功能树可以从对应的eEPC图自动转化而来。
- ⑥ ARIS功能（function）：
 - 一个功能是为达到一个或多个企业目标而作用在（信息）对象上的一个任务，操作或活动。
 - 一个功能可以由时间和成本来定义特性。



类似，数据对象、组织对象等也可以有相类似属性，使得过程的量化分析成为可能。

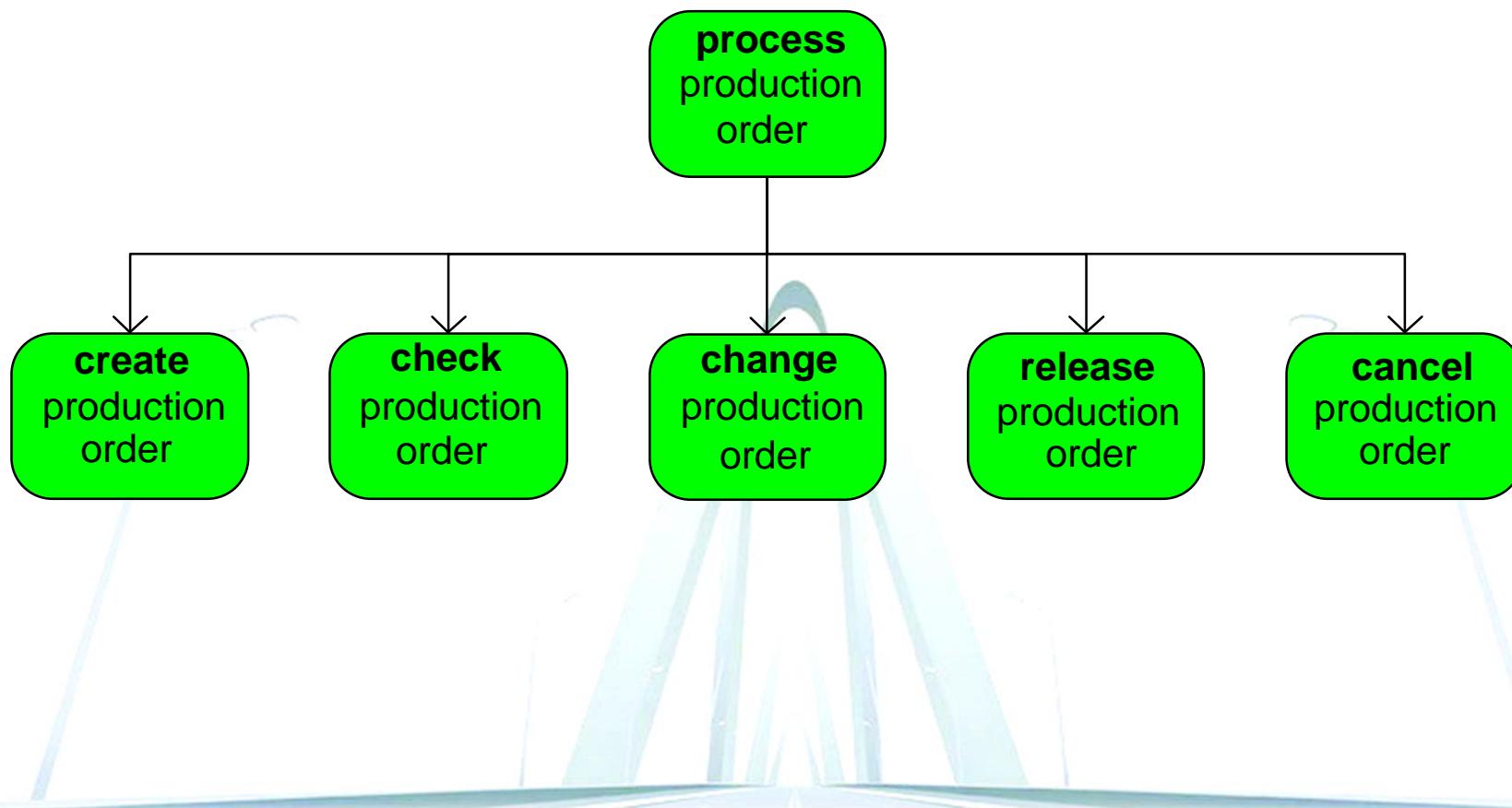


process-oriented function tree 示例



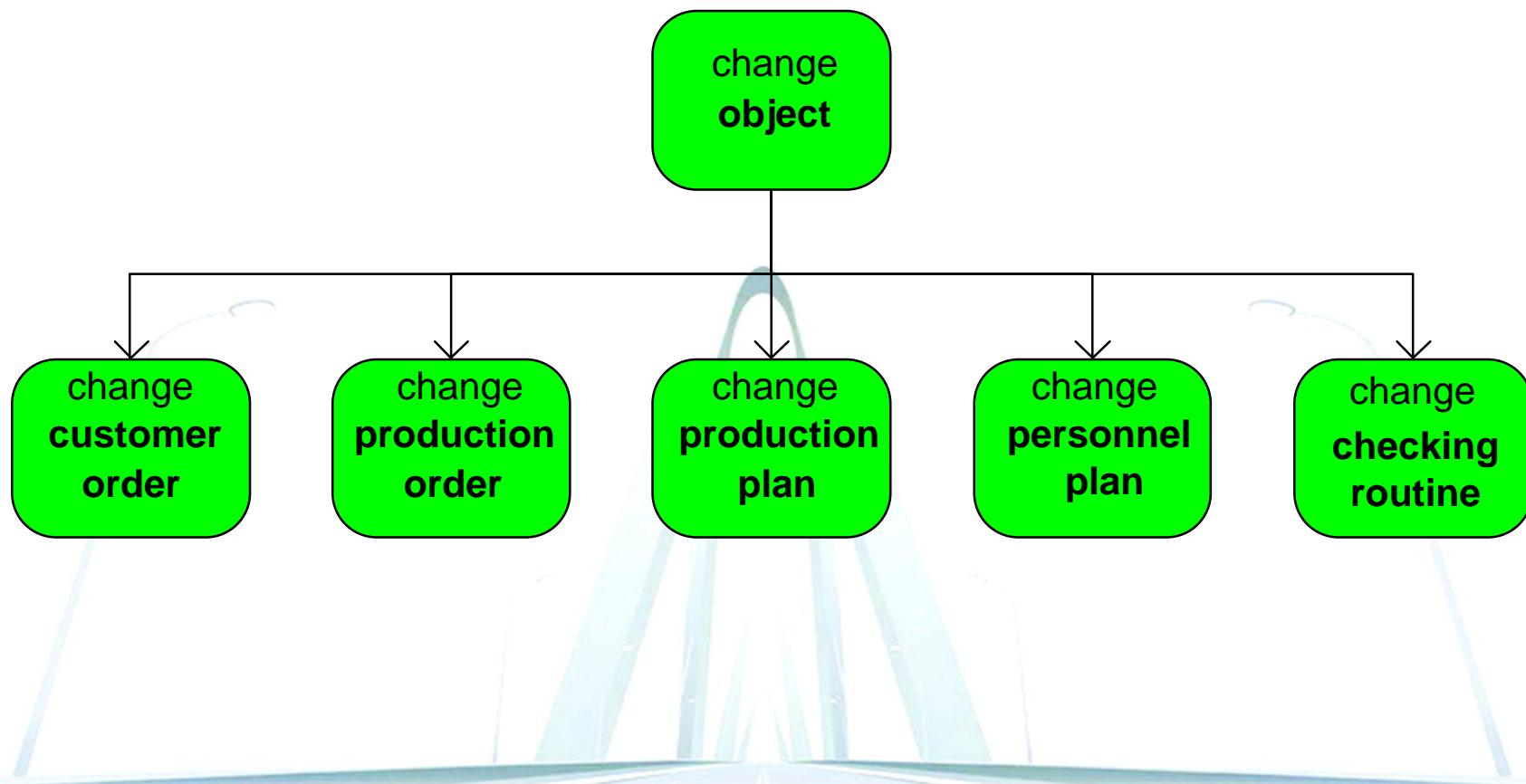


Example: Object-oriented Function Tree



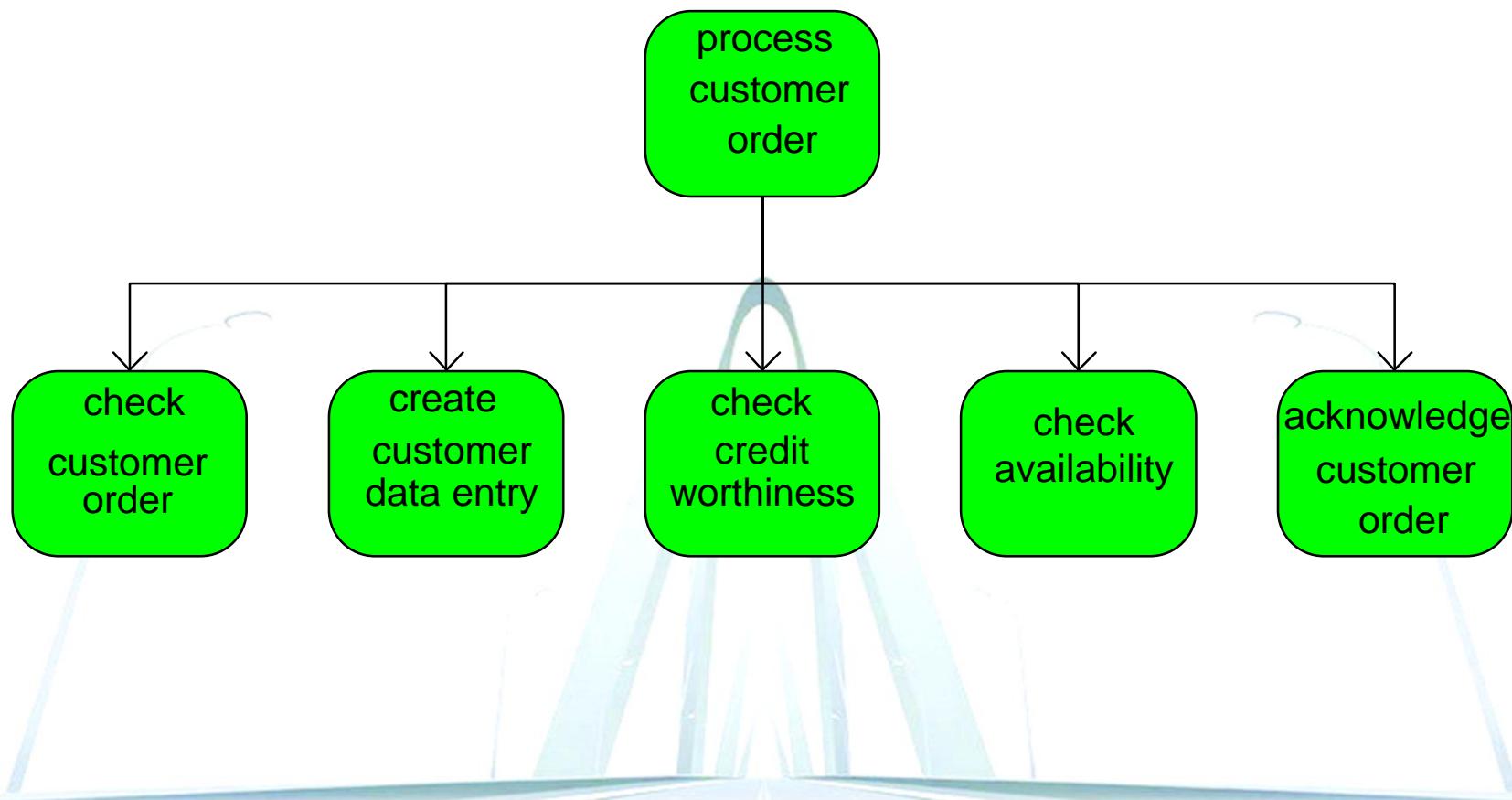


Example: Execution-oriented Function Tree





Example: Process-oriented Function Tree





3.3 过程视图

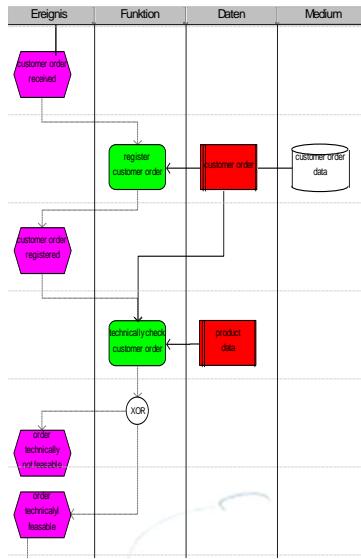
- 过程视图是ARIS集成思想的特色体现，起到粘合剂作用，把功能、数据、组织和输出（产品/服务）视图联系到一起。

ARIS过程视图有一系列建模方法组成，包括：

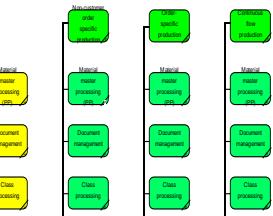
- 关联组织和功能视图：功能/组织图
- 关联数据和功能：功能分配图、信息流图、eEPC图
- 关联组织、功能和数据：eEPC、增值链图、规则图、通信图、分类图
- 面向对象建模：类图（Class diagram）
- 过程变量：过程选择矩阵（Process Selection Matrix）
- 物流建模：物流图、带物流的eEPC



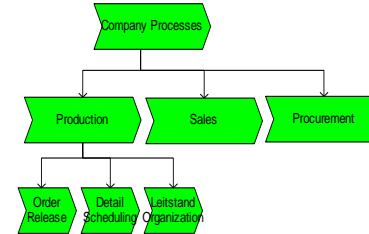
Process Chain Diagram



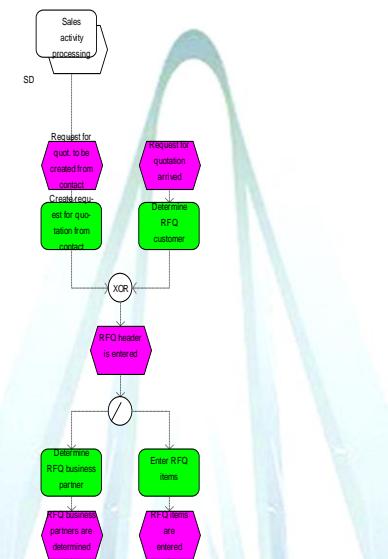
Process Selection Matrix



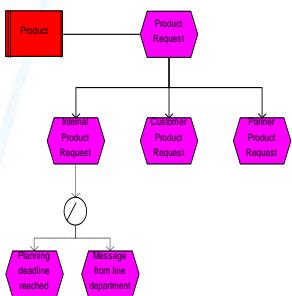
Value Added Chain Diagram



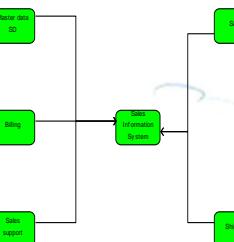
Ext Event Process Chain



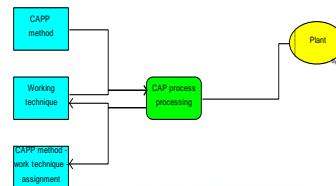
Event Diagram



Information Flow Diagram



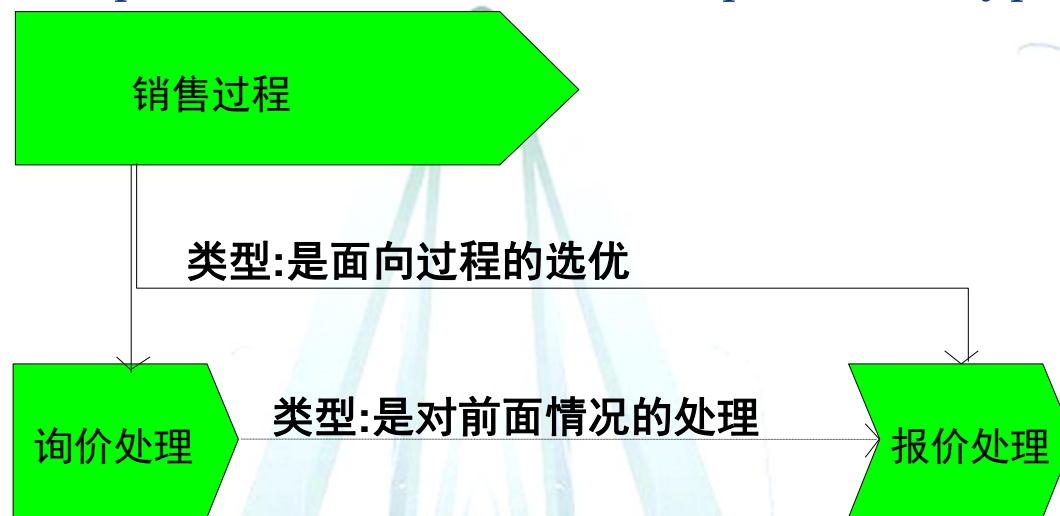
Function Allocation Diagram



Value added chain diagram

- ◎ 价值增值链图描述直接给企业带来价值增值的过程；
 - 价值增值链具有对象类型功能；
 - 在价值链中功能过程按照特别的序列（价值增值链）连接。
 - 价值增值链图中的功能可以分解为子过程；
 - 分解类型是process-oriented superior type.

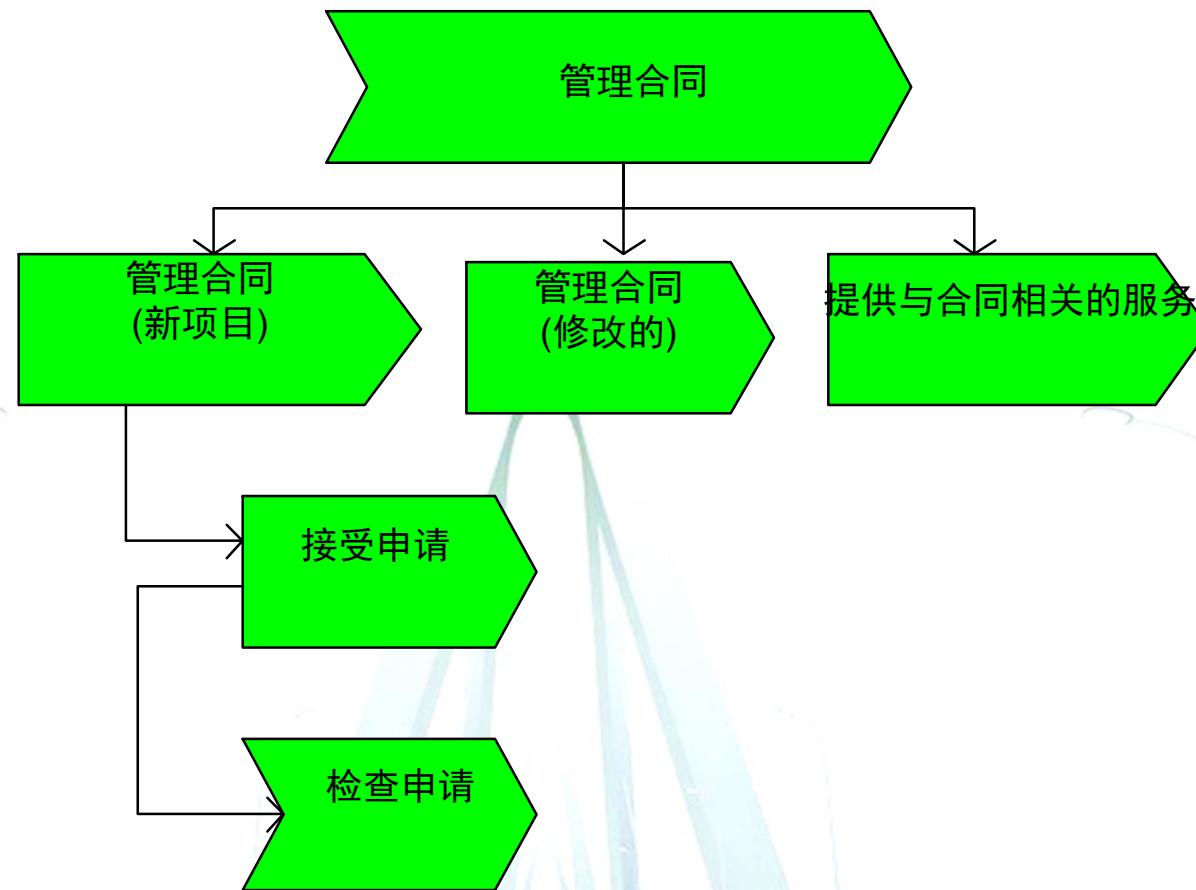
- ◎ 图形表示：

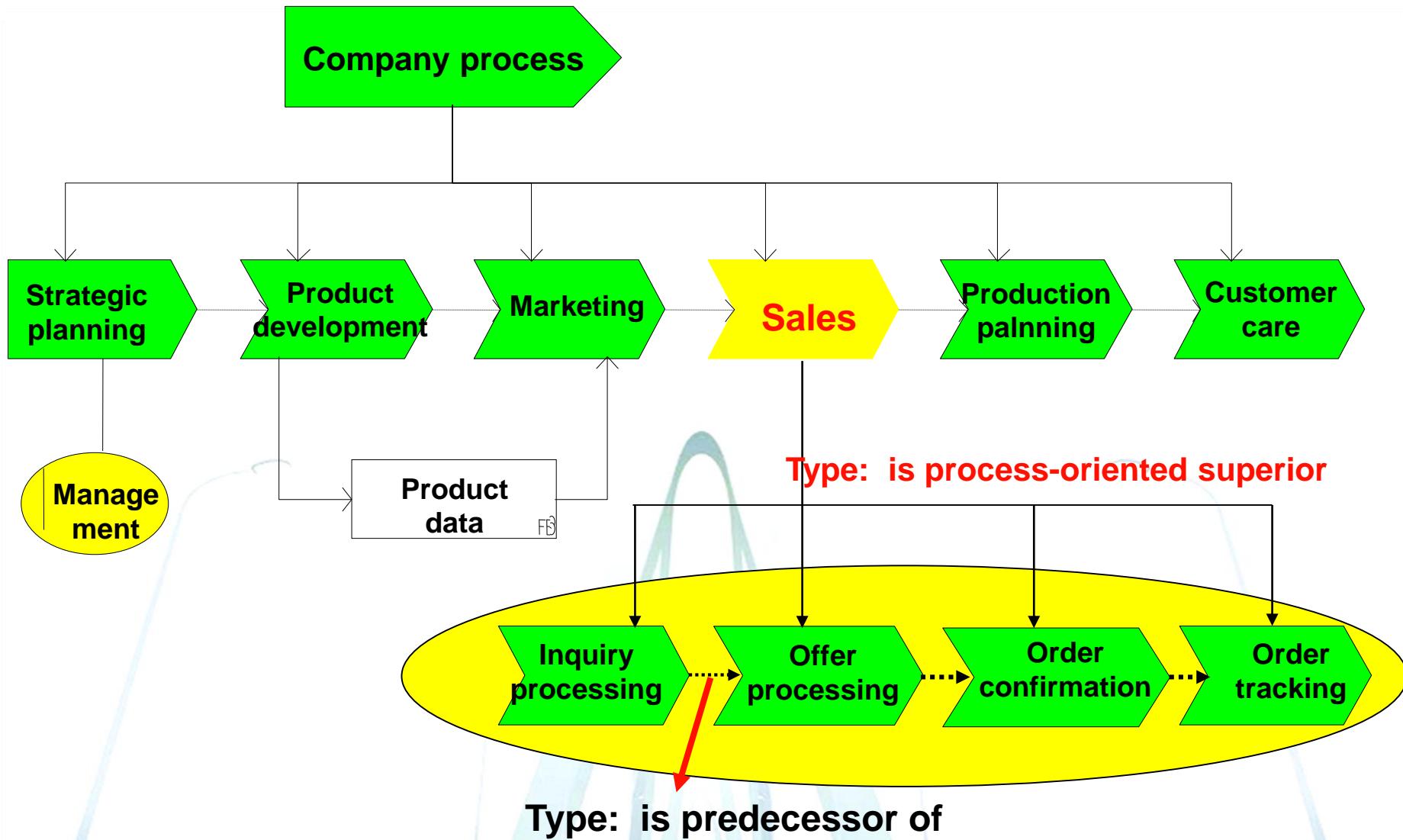




② 主要功能：

展示和描述直接给企业带来增值效益的功能

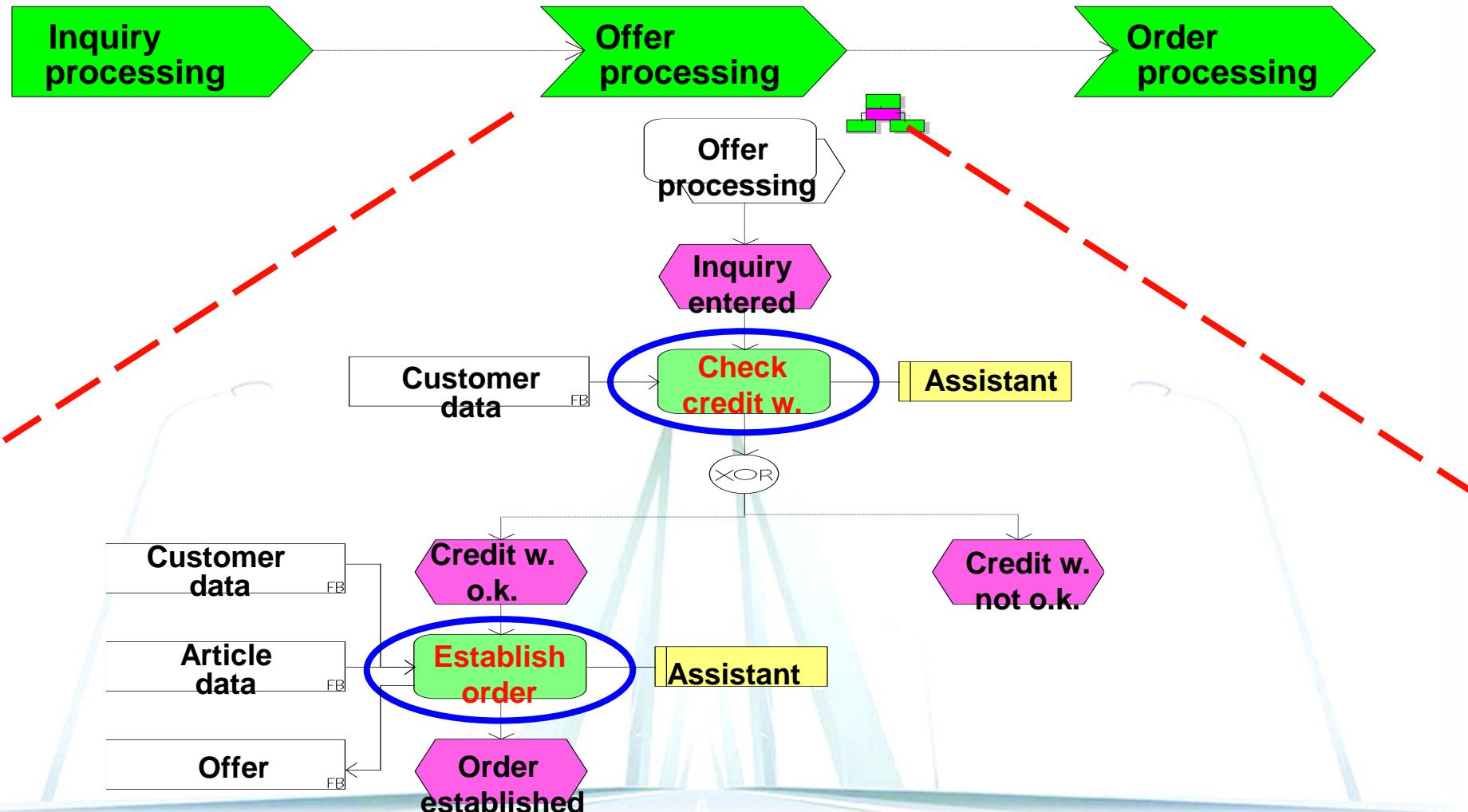






Assignment of an Object

功能分配图 (Function allocation diagram)





3.4 数据视图

- ① ARIS数据视图包括各种建模方法，其中最主要的是实体关系模型（ERM），并演化出
 - SAP的SERM;
 - IEF (information engineering Facility) 数据模型;
 - SeDam (Semantic Data Model) 数据模型等。
- ② 数据视图中的实体与组织单元、功能实体有紧密的联系。



- ◎ A **cluster** represents a logical view on (several) entity- and relationship types of a data model.
- ◎ It may describe a complex object.

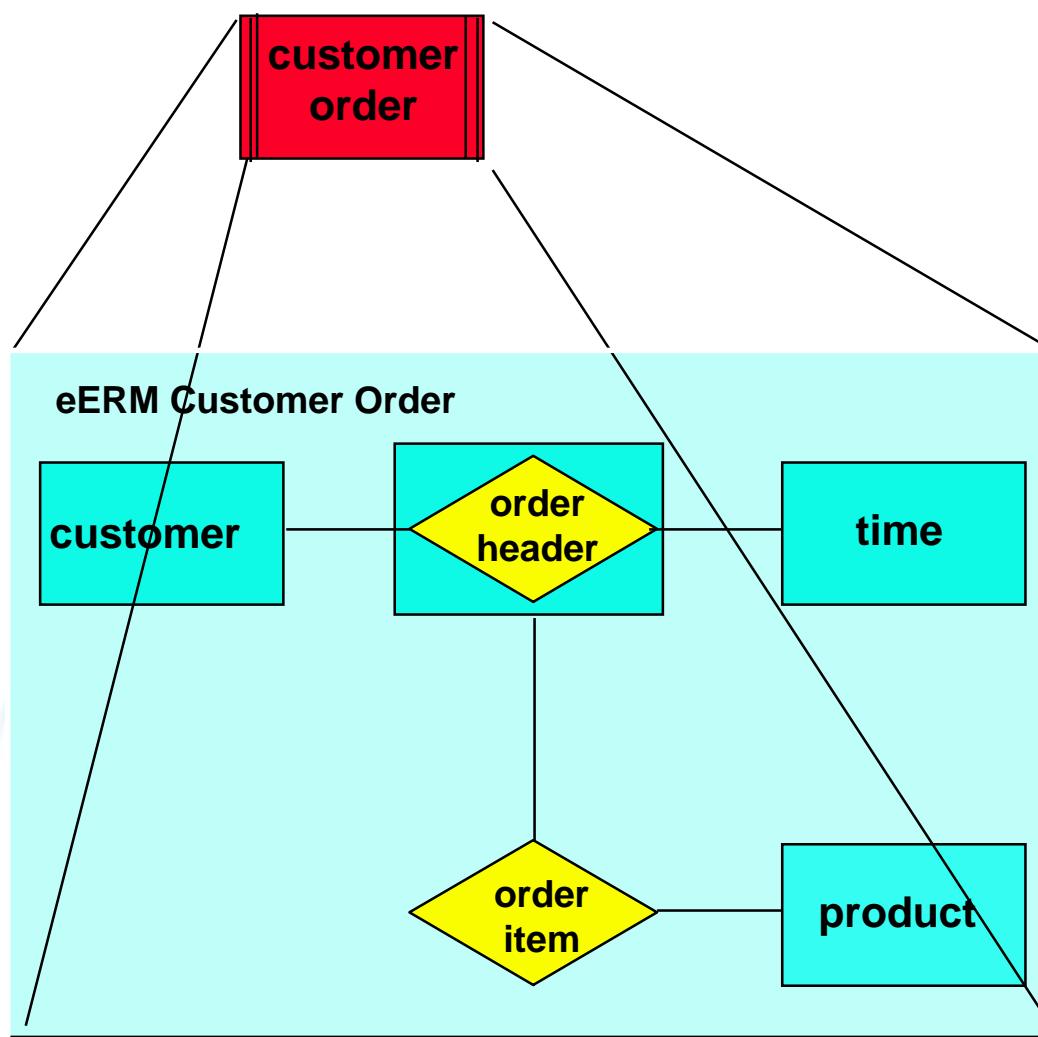
In the ARIS-Toolset:

Clusters may be described in more detail by entity- and relationship types or by (more detailed) clusters.





Example of a Cluster



Cluster

linked
eERM



Basic elements of the ERM

■ Entities

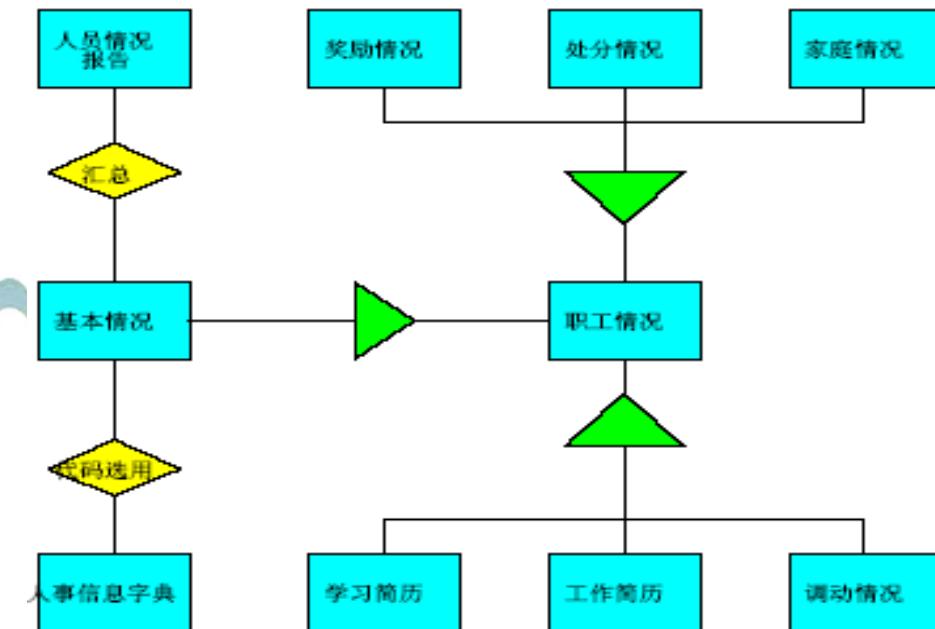
are real or abstract objects that are of interest for a certain subject within a particular segment of the enterprise.

■ Relationships

are logical links between entities.

■ Attributes/attribute types

represent properties which describe a particular entity/entity type.

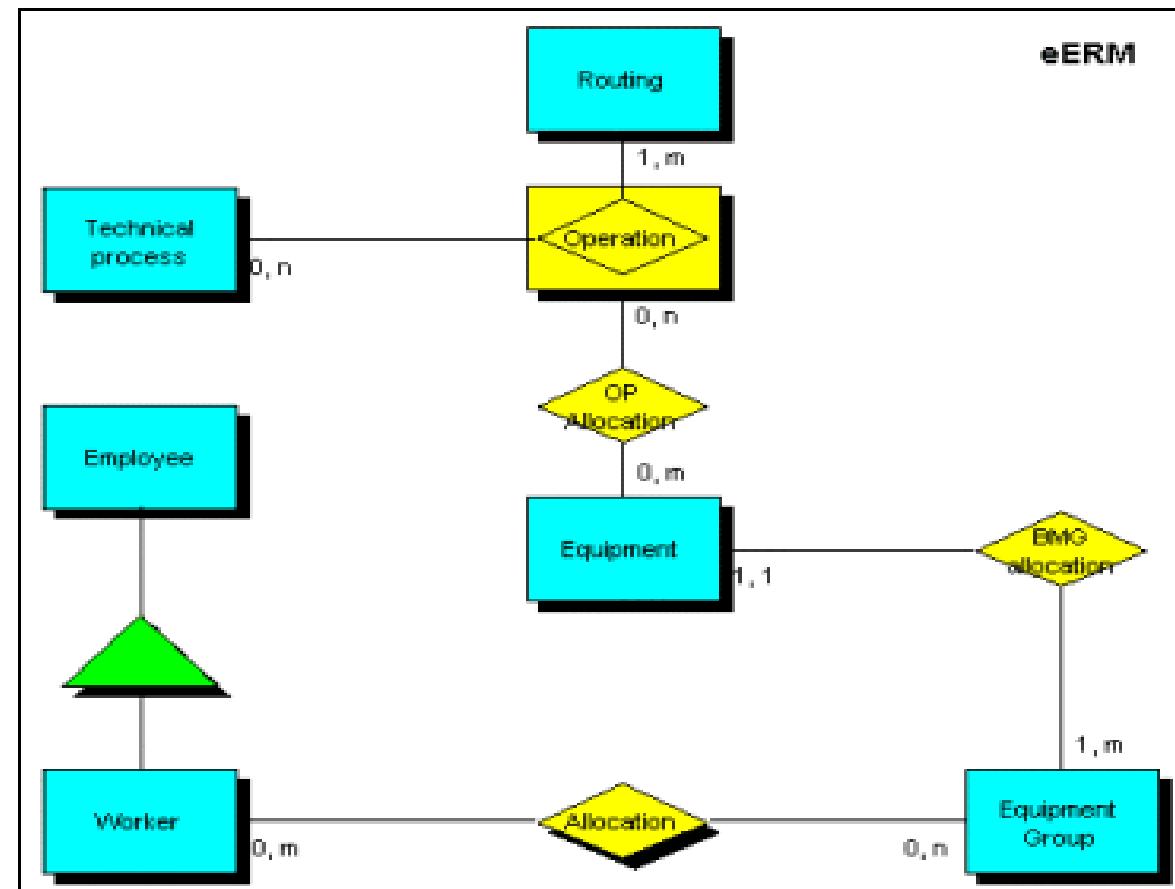




Extending the ERM: eERM

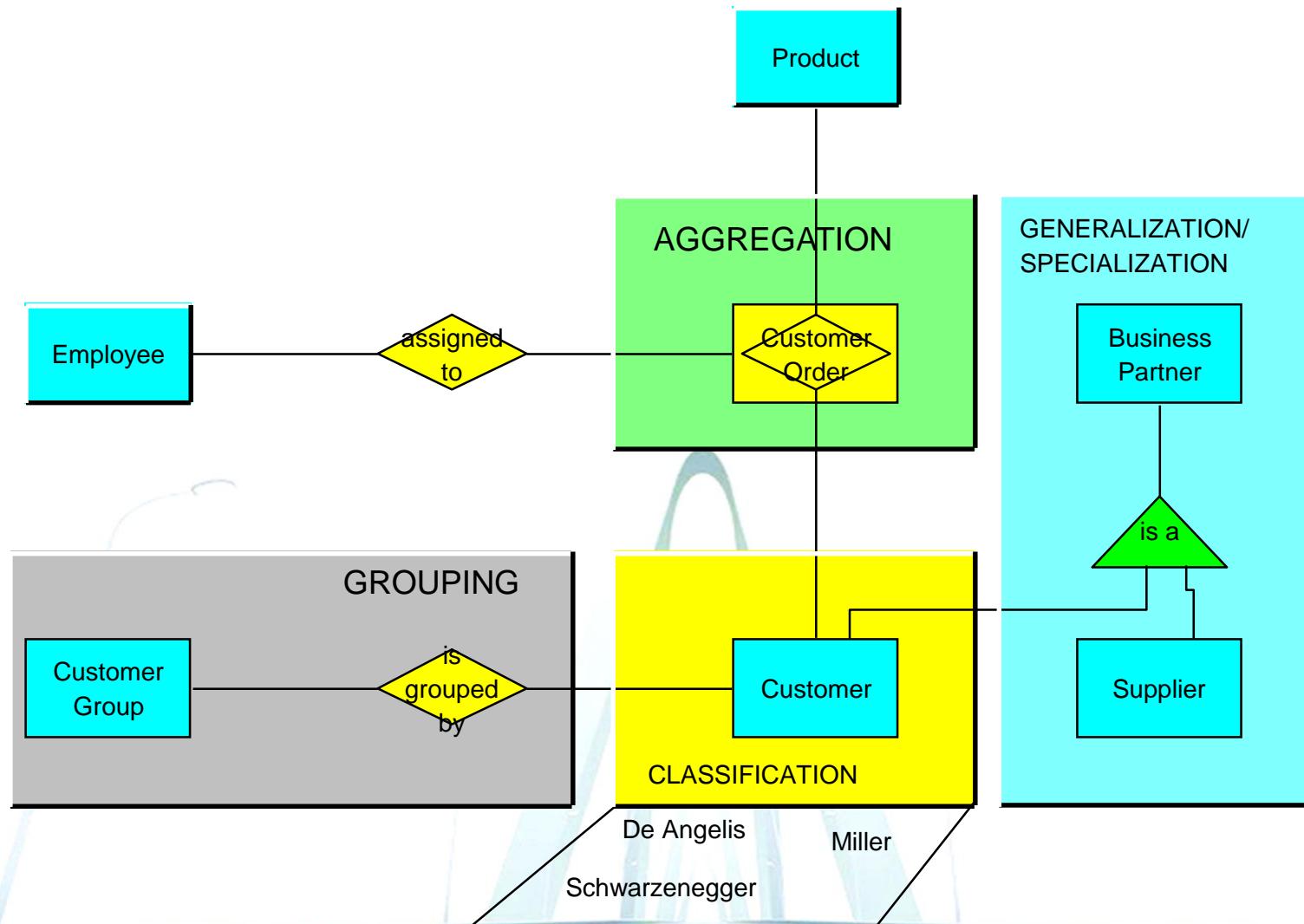
■ Extending the ERM with **construction operators**

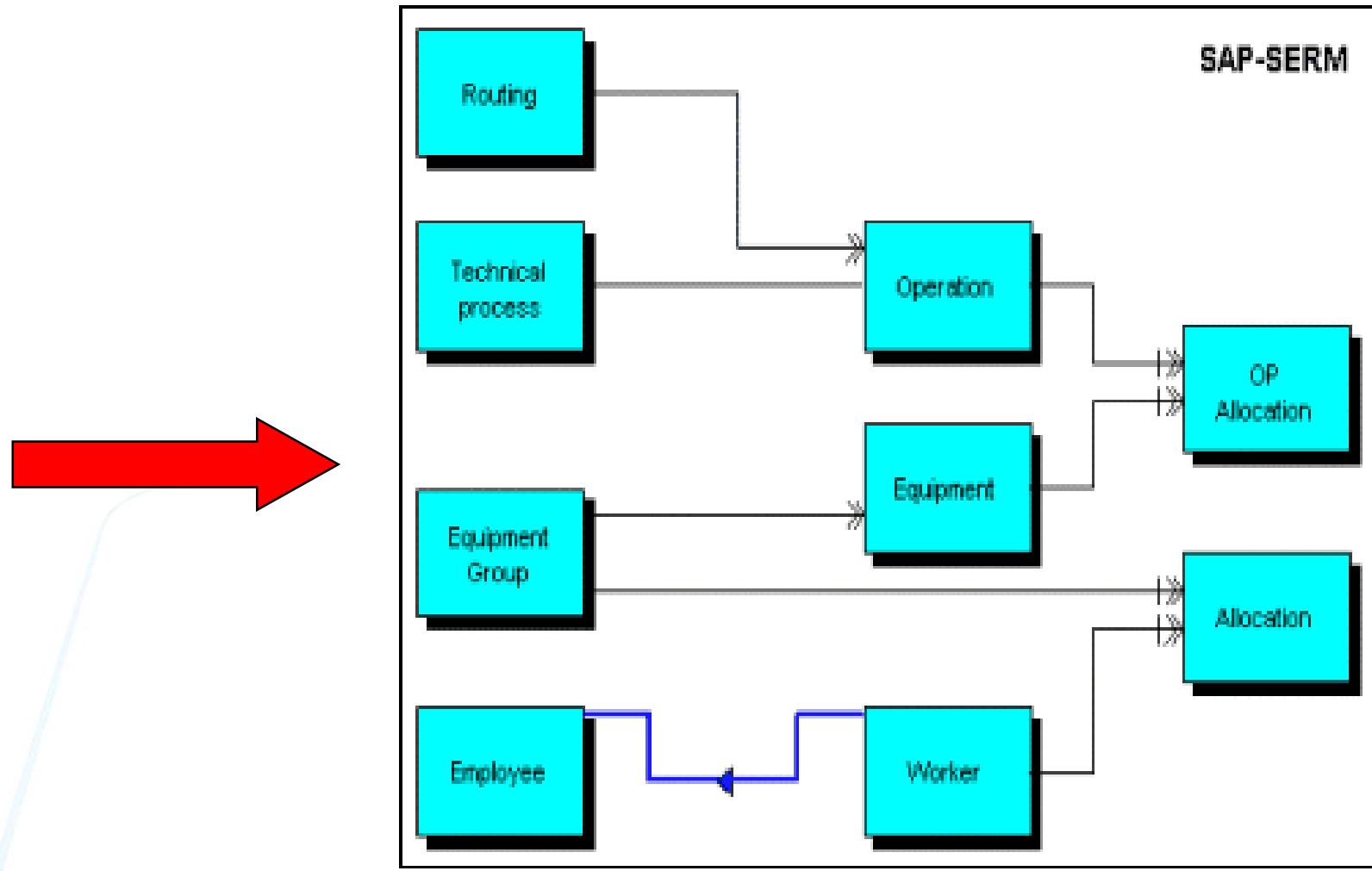
- Classification(分类)
- Generalization / specialization(归纳)
- Aggregation(聚合)
- Grouping(集合)

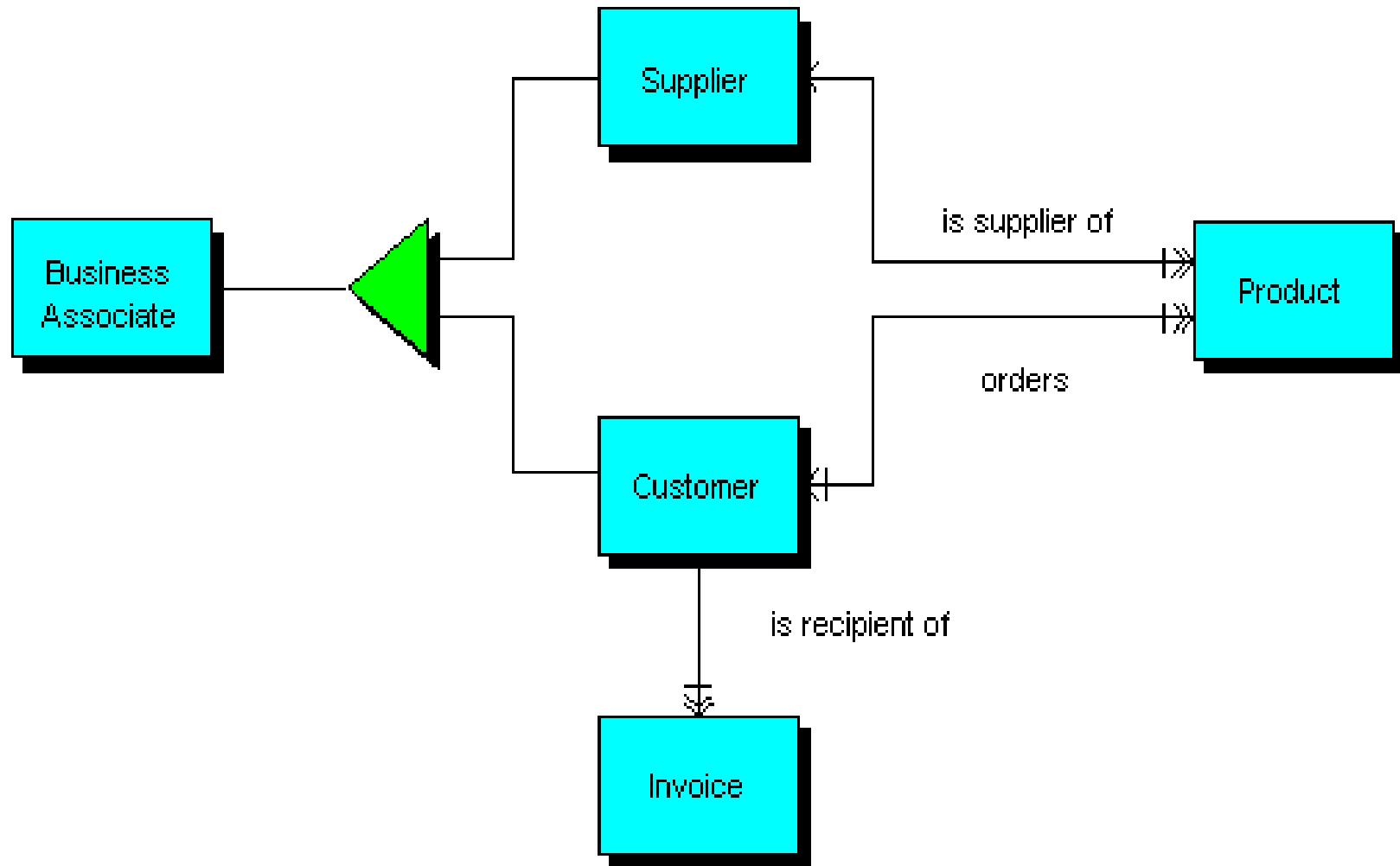




Extending the ERM: Construction Operators









Agenda

1. ARIS框架
2. 基于EPC的过程建模方法
3. ARIS视图
4. ARIS建模过程及实例
5. ARIS软件工具
6. 小结



4 ARIS业务建模过程及实例

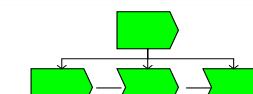
ARIS业务建模过程就是建立各视图的模型及它们之间的联系的过程：

- 组织建模
 - 功能建模
 - 数据建模
 - 过程建模
-
- 这个过程是以控制视图为核心的建模过程，然而各视图的创建过程，可能有一定差异。
 - 一种最佳实践方式是数据视图，功能视图，组织视图，最后是过程视图。

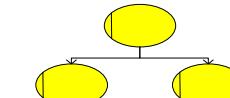


Overview models

Modeling
level 1



Value added chain
diagram



Organization chart

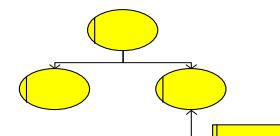
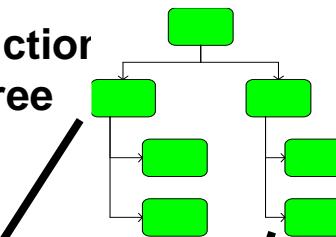
Raw model

Modeling
level 2



eEPC

Function
tree



Organization

Detail model

Modeling
level 3

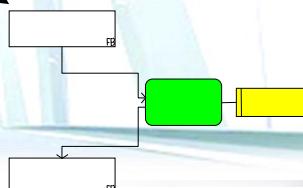


eEPC

Fine model

Modeling
level 4

Function classification
diagram





业务流程架构层次

| 层次 | 模型 | 描述对象 | |
|-----|--------|------|------|
| 第0层 | 企业地图 | | |
| 第1层 | 流程区域地图 | | 流程区域 |
| 第2层 | 主流程模型 | | 主流程 |
| 第3层 | 业务流程模型 | | 业务流程 |
| 第4层 | 流程活动模型 | | 流程活动 |



- 某企业实施ERP系统没有达到预期的效果。
➤ 因此在企业业务流程重组的基础上，优化业务流程，去除企业冗余和一切不增值的环节，以确保企业价值链有效运转。
- 选用了ARIS作为业务建模工具；

ARIS建模常用模型类型

| 模型类型 | 模型类型描述方法 | 用 途 |
|------|----------------|--|
| 组织视图 | 组织图 | 描述组织结构 |
| 过程视图 | 价值增值链 | 描述直接给企业带来价值增值的过程 |
| | eEPC 扩展事件驱动过程链 | 描述由事件触发业务活动，而活动又产生事件结果，由此完成企业流程的过程描述。 |
| 功能视图 | 功能树 | 描述企业流程中面向过程的各个业务活动，每个活动可能就是ERP系统中的一个功能点。 |
| 数据视图 | eERI 扩展实体关系模型 | 描述企业流程中存在的信息及信息之间的关系 |
| | 技术术语模型 | 定义企业内部相互交流中用到的术语 |

某印刷设备制造企业在华投资的分厂

- 需要投入较多精力和成本在市场拓展方面，对客户的订单服务要快速反应，提高准时交货率。
- 作为基于订单生产的中小企业，企业希望以较少的库存占用资金顺利完成整个物料供应与生产交货过程；
- 企业质保的服务质量以及服务水平在极大程度上影响了企业的声誉。
- 着力解决销售业务的效率问题、排产计划的灵活性问题、物料低成本控制的问题和质量保证问题，尽可能实现以最低的库存，占用资金最快，实现客户的各种订单需求。

当前问题

- 调研情况：实施ERP系统没有达到预期的效果。
- 原因：企业的管理体制和ERP系统有一定的差距，业务量参差不齐，生产过程当中基层数据采集不齐全或者不准确，没有形成快速可靠的数据采集系统，不能及时对整个生产过程进行监控，多处（管理）理念处于手工管理阶段。
- 措施：优化业务流程，去除企业冗余和一切不增值的环节，以确保企业的价值链有效运转。
- 难点：在企业进行ERP系统的实施过程中，往往需要对原有的业务流程、组织架构进行调整优化或重组，但是在这个过程中很大程度上依赖于ERP实施顾问的经验，缺少定量的方法和工具来对实施前后的经营过程进行分析、诊断、仿真和评估。



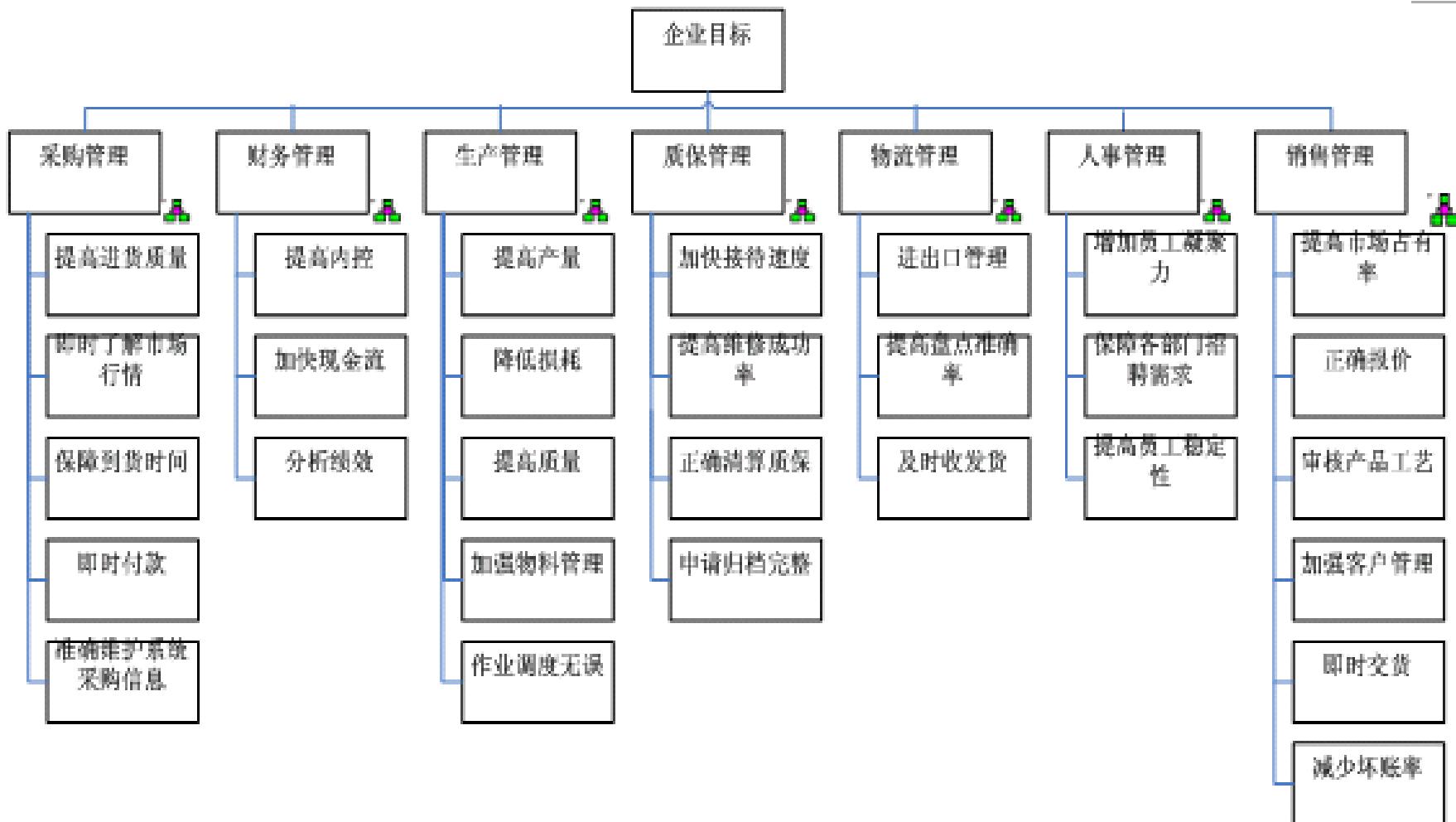
建模范围

结合企业特点，利用ARIS 工具，来定义业务模型范围。

- 结合业务流程建模分析，主要从四个方面对企业业务需求进行分析和描述。
 - 出入库管理
 - 销售采购
 - 质保业务
 - 生产运营
- 涉及部门：销售部、物流部、生产部、采购部、质量部、财务部、人力资源部七部门。

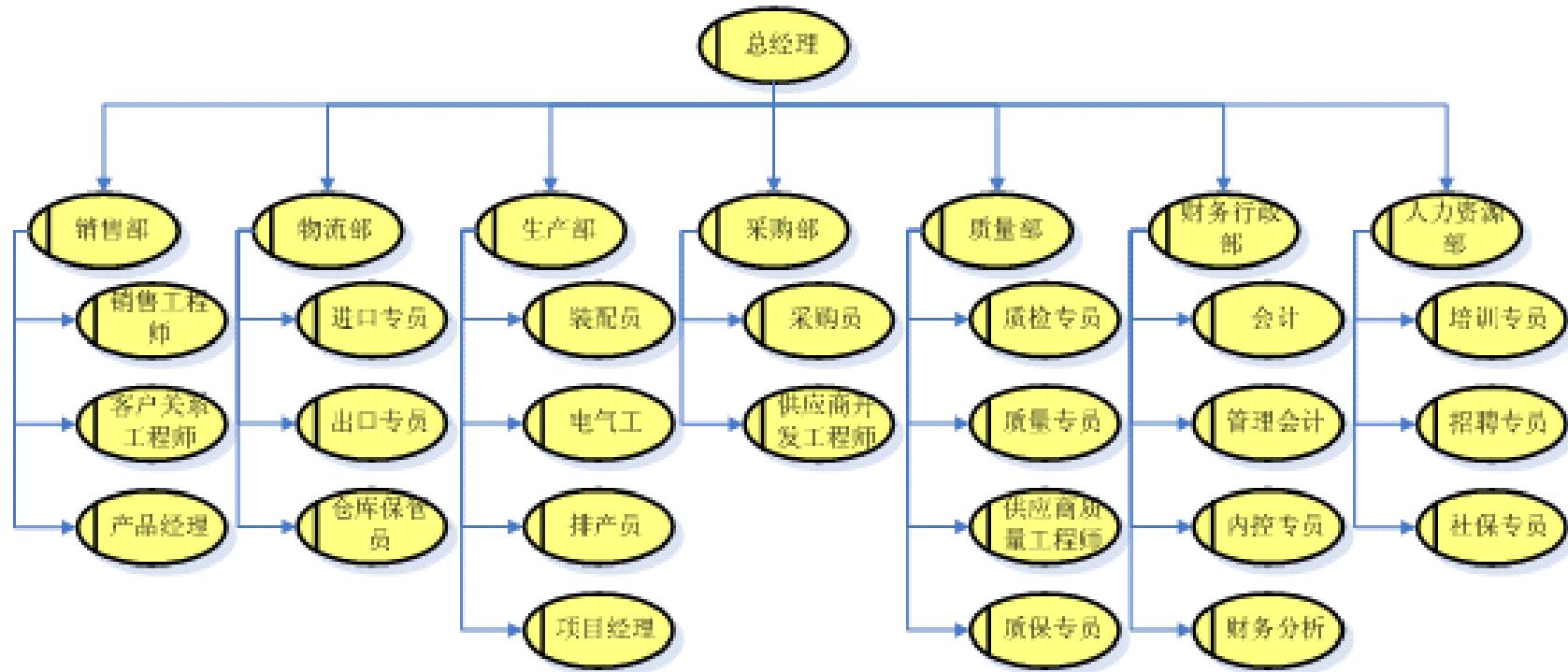


业务目标建模





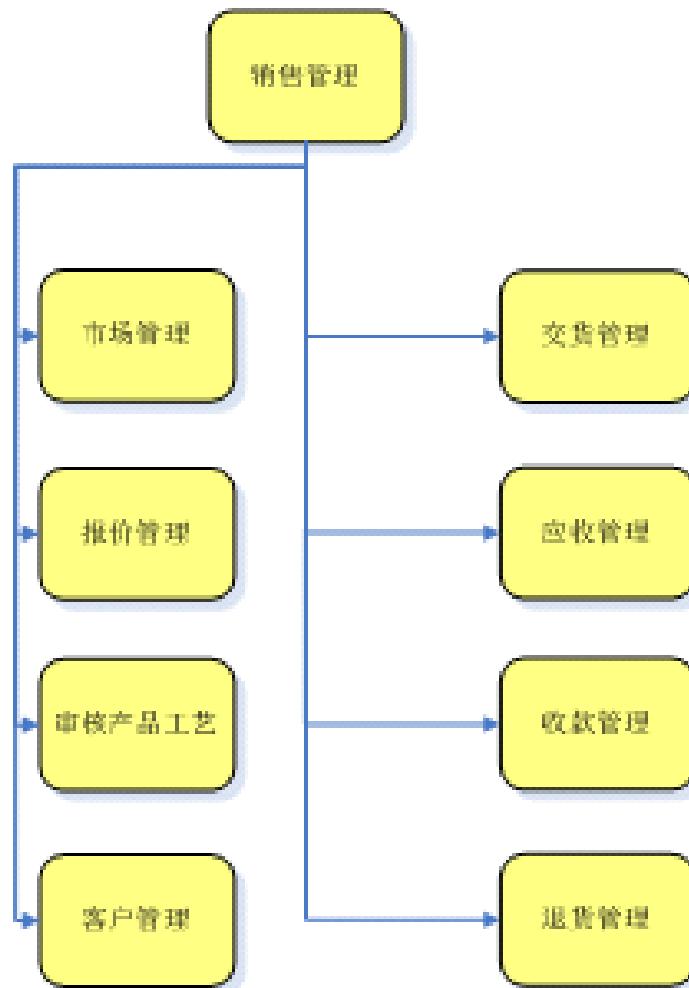
组织建模



ERP 实施前企业的组织结构图



功能建模



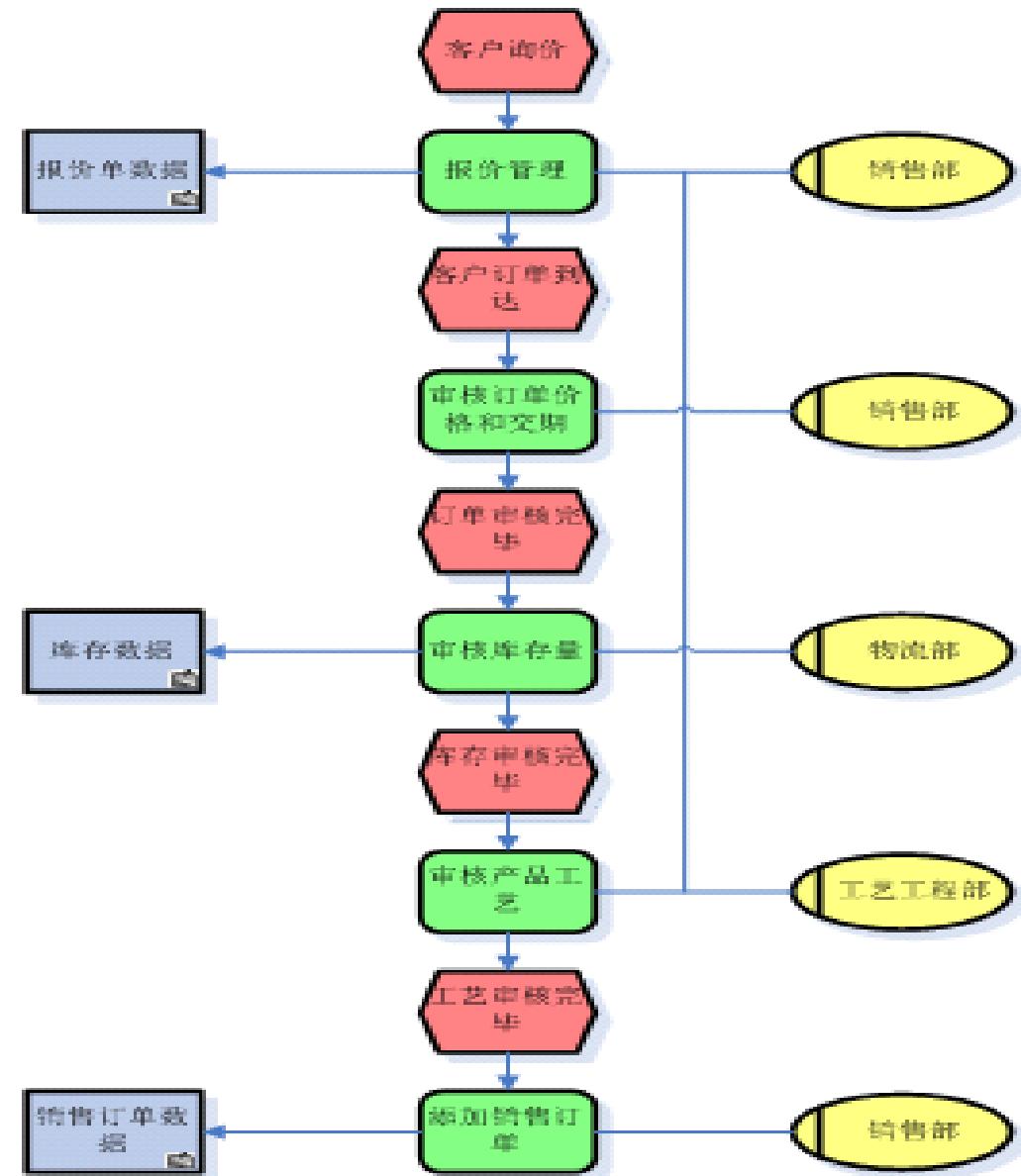
基于功能树模型的业务功能建模

销售管理功能模型+



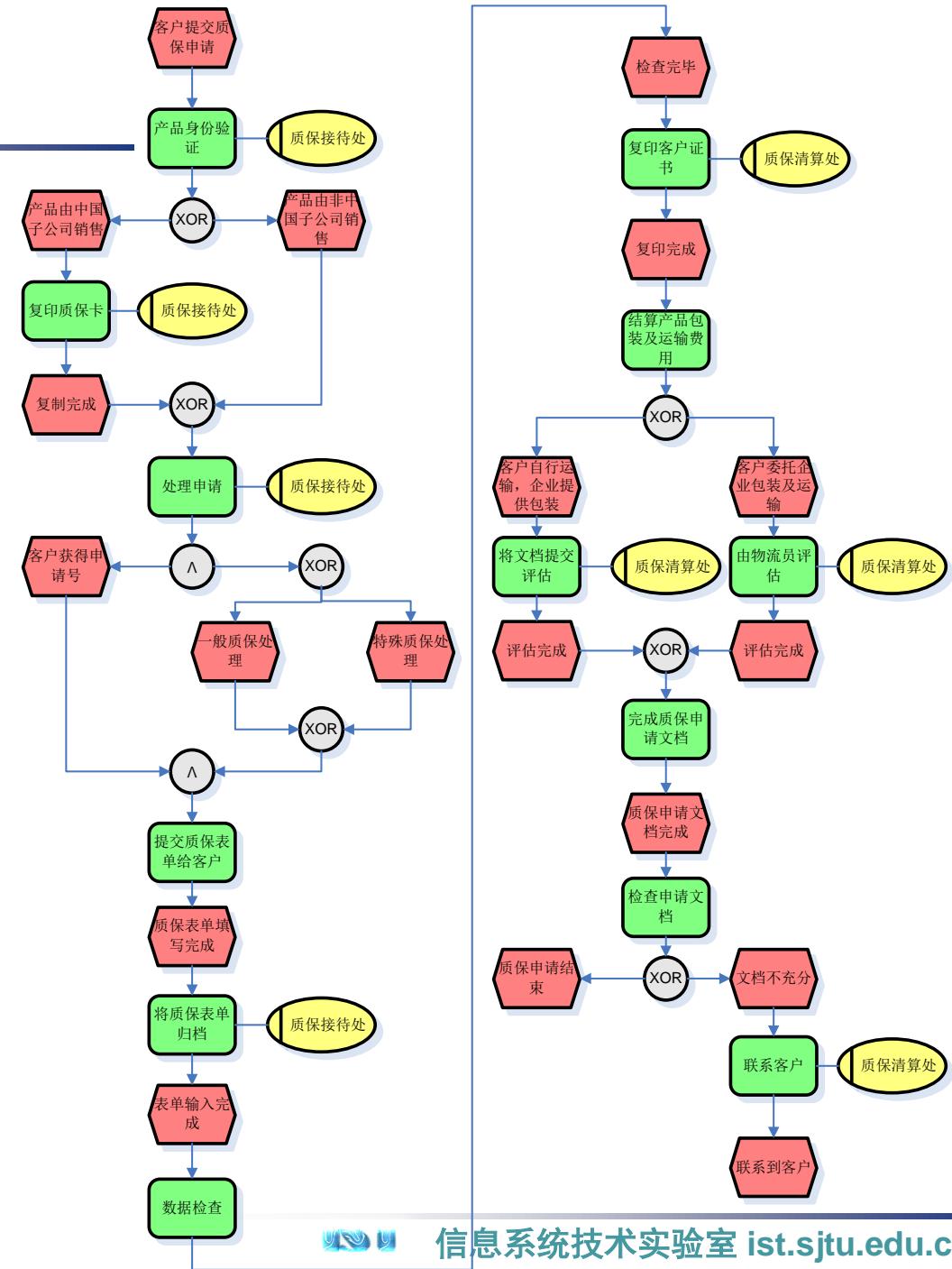
业务流程建模

销售流程



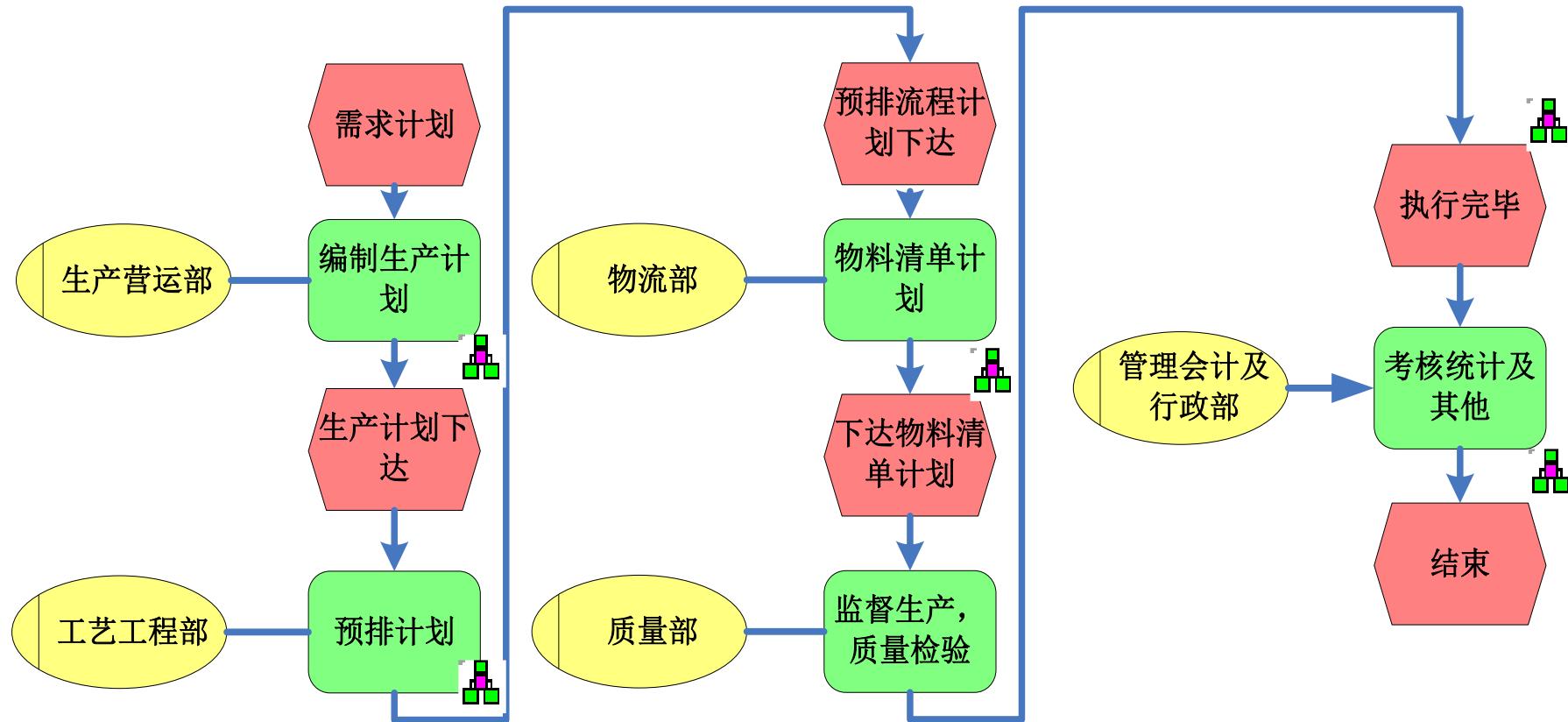


质保流程





车间生产主流程



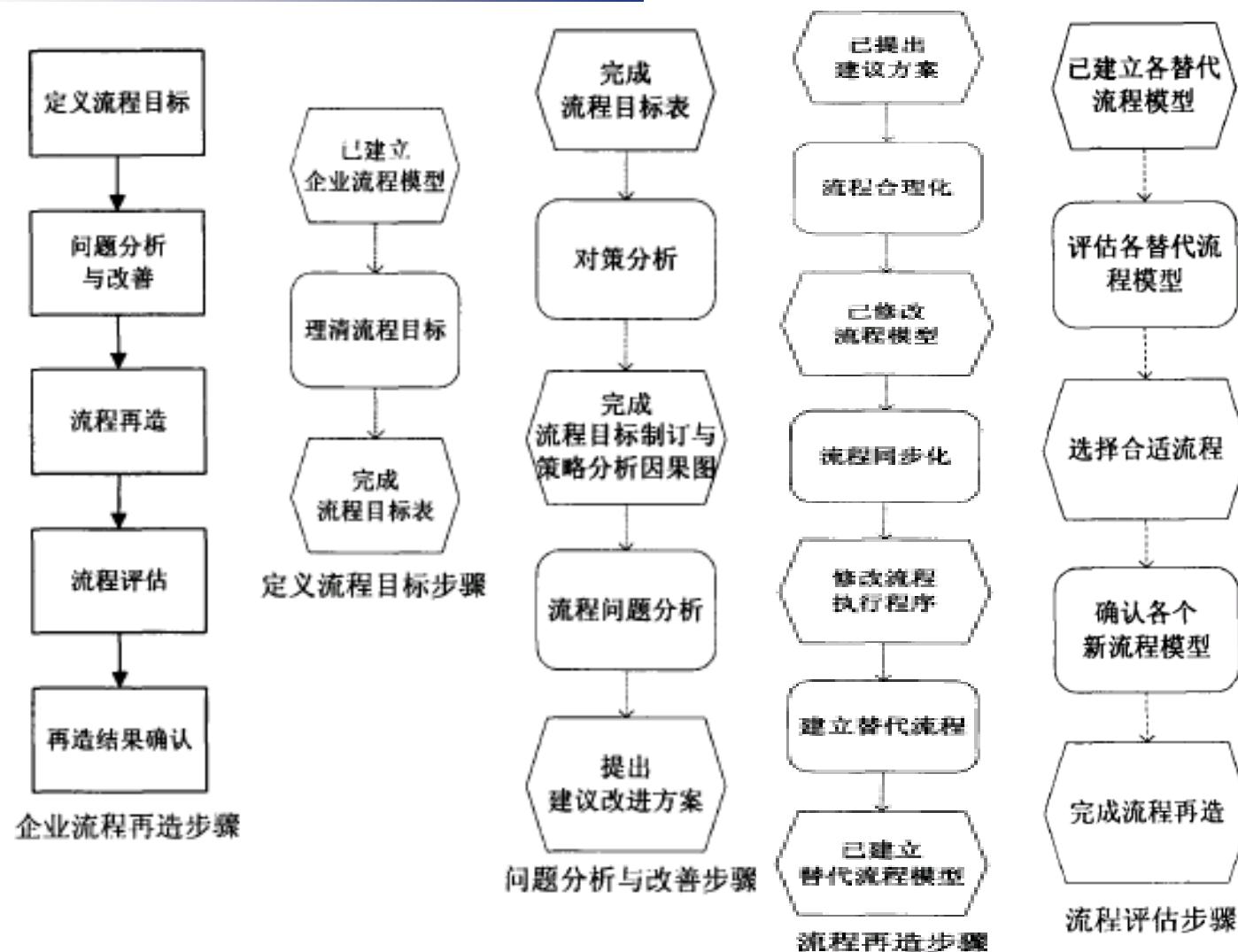


流程优化方向

- 借助ARIS强大的建模功能，给出了系统分析阶段企业模型的建立方法，并从定性的角度分析减少了旧流程步骤多、数据散引发的数据不统一、收集困难的问题。
- 从定性的角度分析得出，新流程利用ERP产品集成性的特点将旧流程中审核客户订单的价格、交货时间、库存数量、订单产品工艺合并为一个步骤，减少了步骤多、数据散引发的数据不统一、收集困难的问题。
- 实现了对企业现有经营过程的优化、仿真、结果分析和评估，为ERP实施中的业务流程优化或再造提供了定量的说明，从而有效降低企业业务分析的复杂度，达到优化ERP实施过程的目的。



基于ARIS的业务流程优化途径





销售订单仿真流程统计数据.

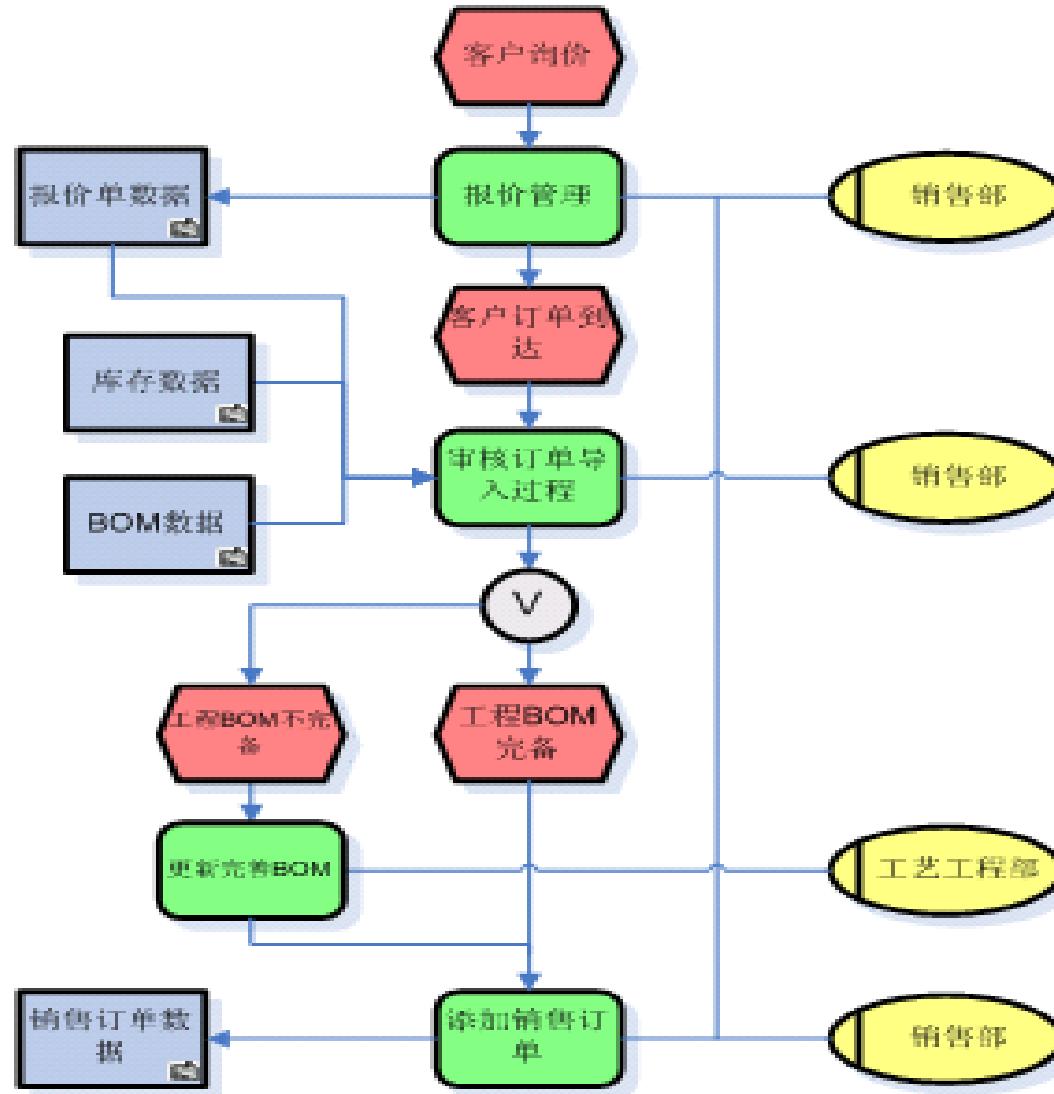
| Items | Numbers |
|----------|---------|
| 创建的事件数 | 5 |
| 创建的功能数 | 5 |
| 创建的组织数 | 3 |
| 技术文档数 | 3 |
| 事件与功能关系数 | 16 |
| 功能间关系数 | 35 |
| 功能与组织关系数 | 21 |
| 功能与文档数 | 5 |

销售订单流程仿真功能统计数据.

| 功能 | 平均等待时间 | 平均定向时间 | 平均处理时间 | 平均总体成本 |
|-----------|------------|------------|------------|-------------------|
| 报价管理 | 4H | 3H | 2H | ¥ 540.00 |
| 审核产品工艺 | 6H | 3H | 12H | ¥ 1,260.00 |
| 审核订单价格和交期 | 2H | 1H | 3H | ¥ 360.00 |
| 审核库存量 | 4H | 1H | 2H | ¥ 420.00 |
| 添加销售订单 | 2H | 2H | 2H | ¥ 360.00 |
| 合计 | 18H | 10H | 21H | ¥ 2,940.00 |

销售订单流程仿真单元统计数据.

| 组织单元 | 参与人数 | 组织利用效率 |
|-------|-----------|-------------|
| 物流部 | 5 | 0.45 |
| 工艺工程部 | 8 | 0.35 |
| 销售部 | 4 | 0.8 |
| 合计&平均 | 17 | 0.53 |



优化后的销售订单流程 eEPC 模型



仿真结果

流程评估——利用ARIS的仿真功能

销售订单优化流程仿真统计数据.

| Items | Numbers |
|----------|---------|
| 创建的事件数 | 4 |
| 创建的功能数 | 4 |
| 创建的组织数 | 2 |
| 技术文档数 | 4 |
| 事件与功能关系数 | 17 |
| 功能间关系数 | 20 |
| 功能与组织关系数 | 8 |
| 功能与文档数 | 6 |

销售订单优化流程仿真功能统计数据.

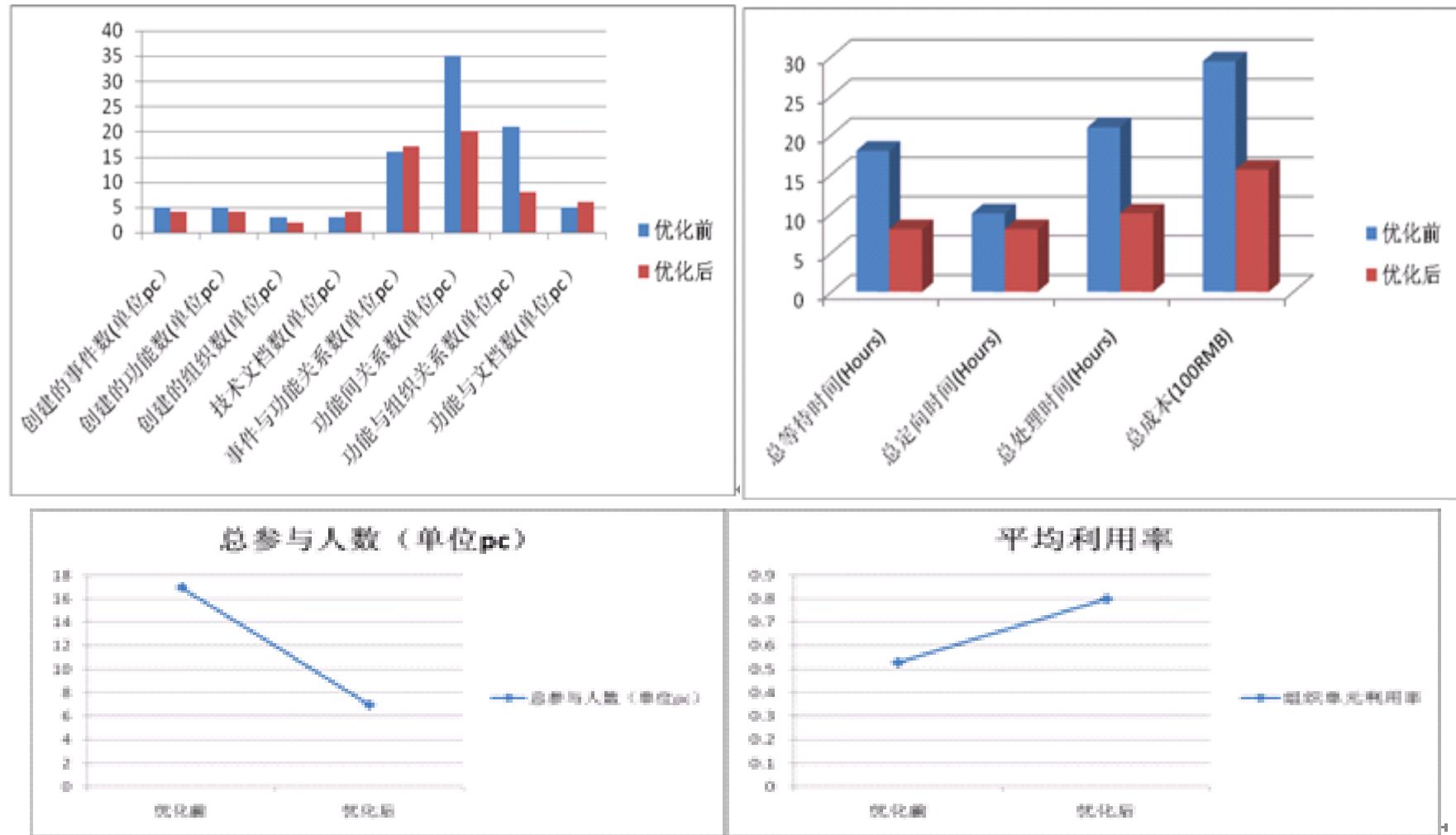
| 功能 | 平均等待时间 | 平均定向时间 | 平均处理时间 | 平均总体成本 |
|----------|-----------|-----------|------------|------------------|
| 报价管理 | 4H | 3H | 2H | ¥ 540.00 |
| 更新完善BOM | 1H | 2H | 3H | ¥ 360.00 |
| 审核订单导入过程 | 1H | 1H | 3H | ¥ 300.00 |
| 添加销售订单 | 2H | 2H | 2H | ¥ 360.00 |
| 合计 | 8H | 8H | 10H | ¥1,560.00 |

销售订单优化流程仿真单元统计数据.

| 组织单元 | 参与人数 | 组织利用效率 |
|-------|----------|-------------|
| 工艺工程部 | 4 | 0.70 |
| 销售部 | 3 | 0.90 |
| 合计&平均 | 7 | 0.80 |



结果分析



销售订单流程优化前后——组织统计数据比较图

结果分析

- ⑥ 利用ARIS的仿真功能，从流程统计、功能统计和组织统计的角度，对业务流程优化前后的仿真数据进行了比较。
- ⑦ 并且从定量的角度分析得出：优化后的流程提高了工作效率，降低了成本，减少了不必要的操作人员。
- ⑧ 通过这些流程的改进，着力解决了印刷设备制造企业销售业务的效率问题、排产计划的灵活性问题、物料低成本控制的问题和质量保证流程业务的效率问题，以较低的库存和占用资金较少来实现客户的需求。
- ⑨ 通过仿真及结果分析，充分说明了基于ARIS的ERP实施方法能够为业务流程再造、组织结构变革提供定量的说明，可以在较大程度上剔除对客户无价值的业务流程，减少成本，降低企业业务复杂度，优化ERP实施过程。从而提升客户满意度，提高企业的经营效益，建立企业竞争优势。



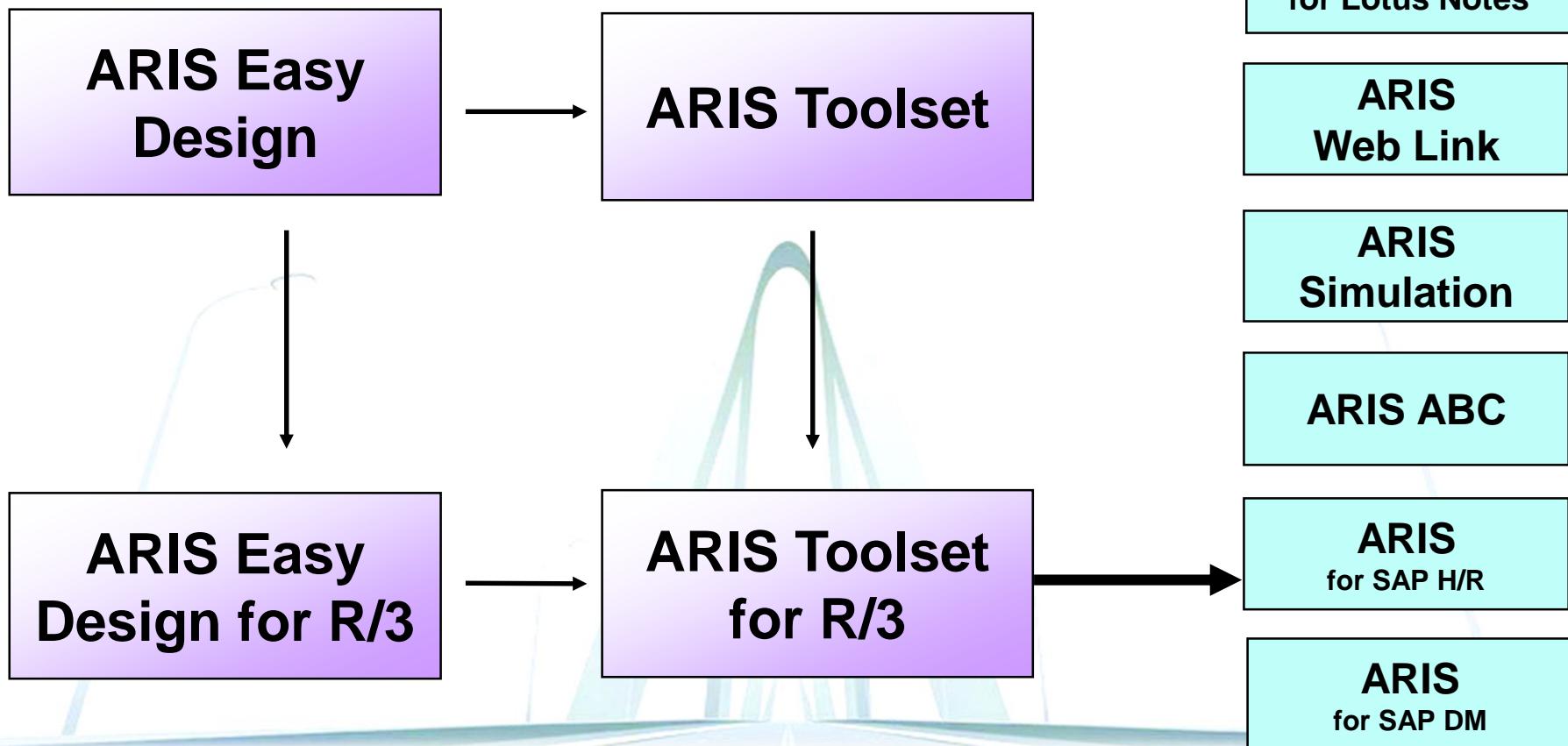
Agenda

1. ARIS框架
2. 基于EPC的过程建模方法
3. ARIS视图
4. ARIS建模过程实例
5. ARIS软件工具
6. 小结



5 ARIS软件工具

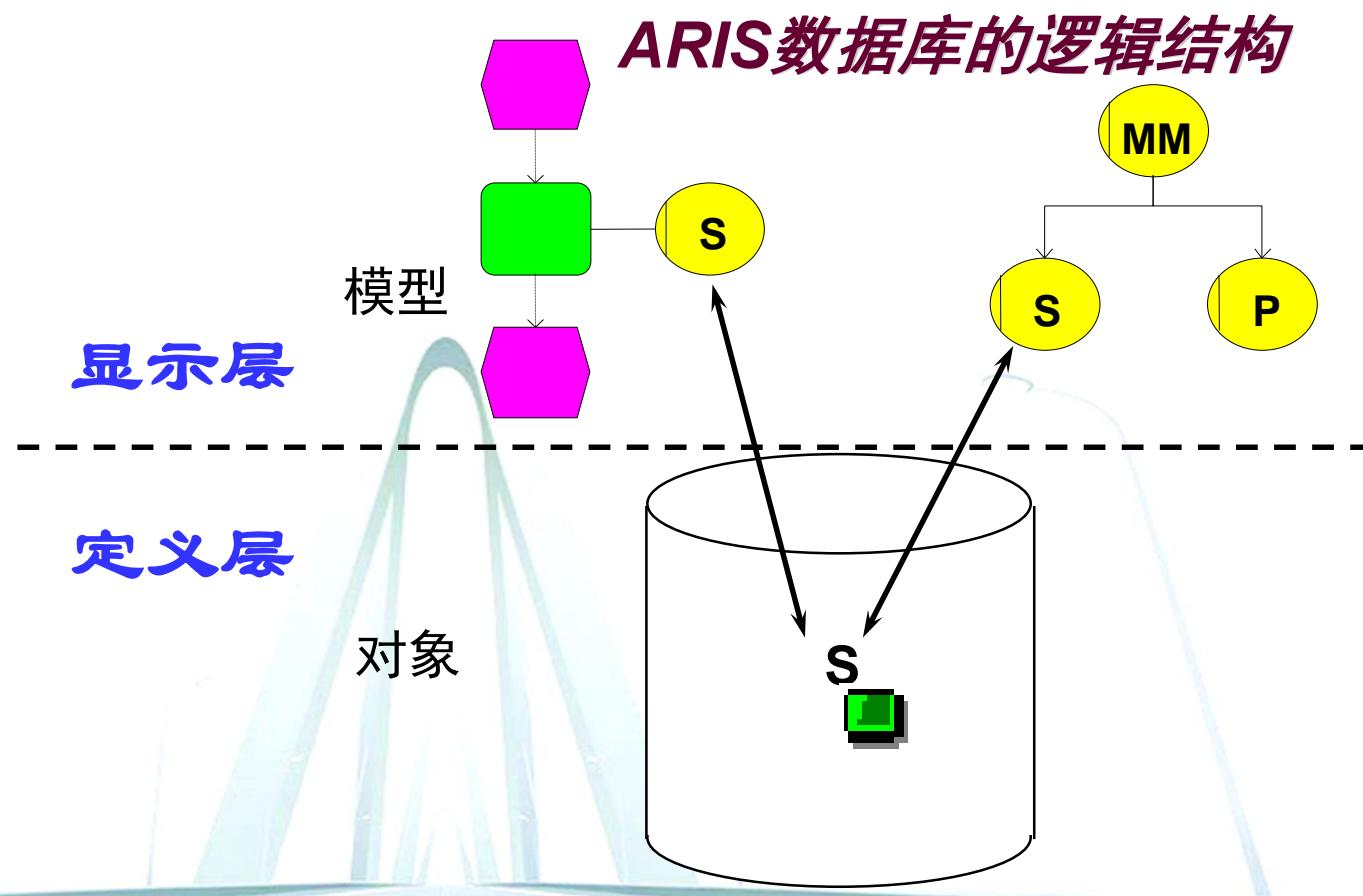
- ARIS有一套标准的软件工具集支持ARIS建模及仿真实施。自1990年以来，IDS Scheer公司一直致力于基于ARIS模型的软件实施及相应工具的发展。





ARIS数据库的层次信息结构：

- ① 服务器
- ② 数据库
- ③ 组结构
- ④ 模型
- ⑤ 对象



ARIS资源管理器：

- ① ARIS资源管理器是ARIS的核心管理组件。
 - ② ARIS资源管理器帮助你管理数据库及其内容(组,模型和对象,还有用户,语言和字体).
 - ③ 选择菜单View中的ARISExplorer菜单项可以激活ARIS浏览器.
-
- ④ 组是一个逻辑目录，模型和对象都存放在其中，组用文件夹符号来表示.
 - ⑤ 例：面向过程的组结构





(1) 创建模型实体

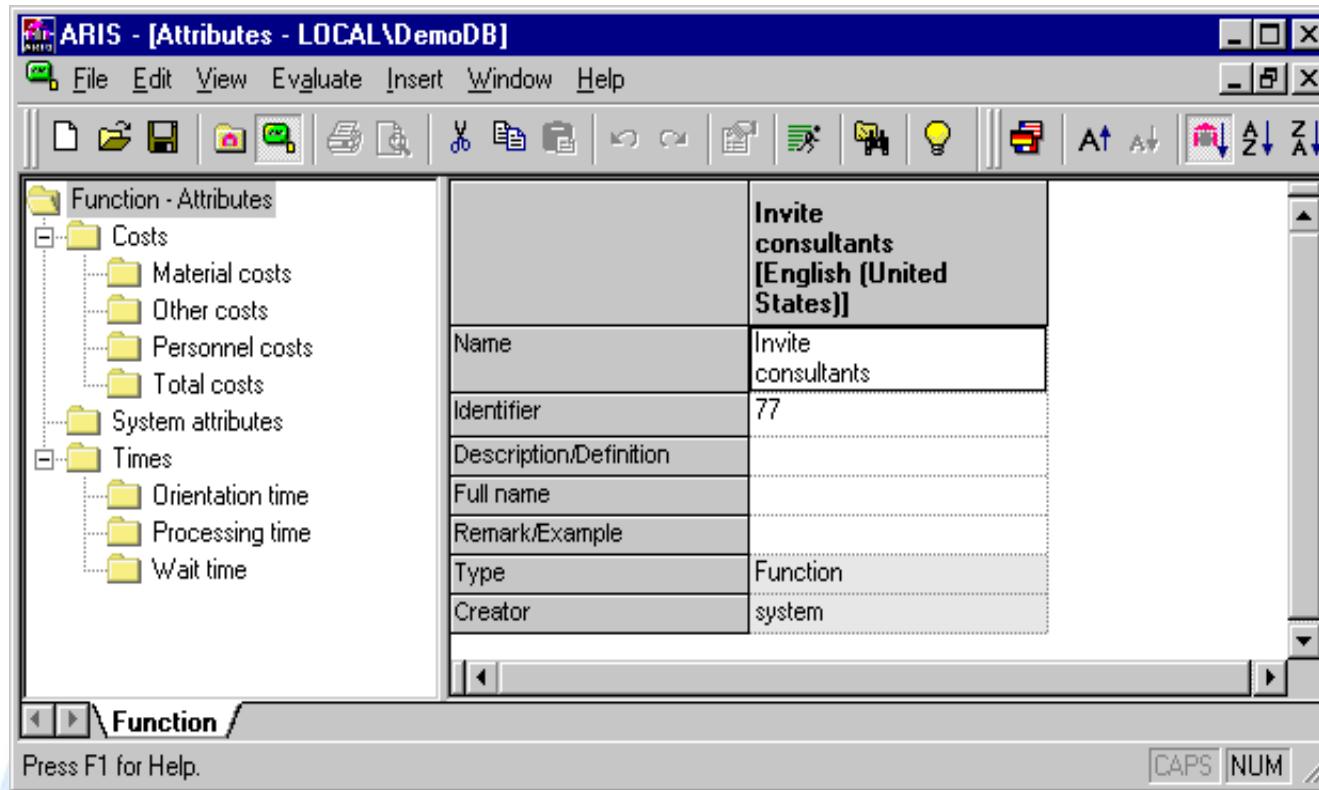
属性

- ① 每个模型类型，对象类型以及关系类型都具有特殊的属性类型。
- ② 通过使用属性，模型、对象和关系都可以用属性详细定义其特性
- ③ 对象属性反映了对象的附加信息，以后可以通过分析功能对这些信息进行评价。
- ④ 附加信息的例子：
 - 执行一个功能需多长时间?
(时间属性组)
 - 执行一个功能需要多少成本?
(成本属性组)
 - 以什么样的方式执行功能(手工， 在线,...)?
(过程类型属性组)



属性窗口

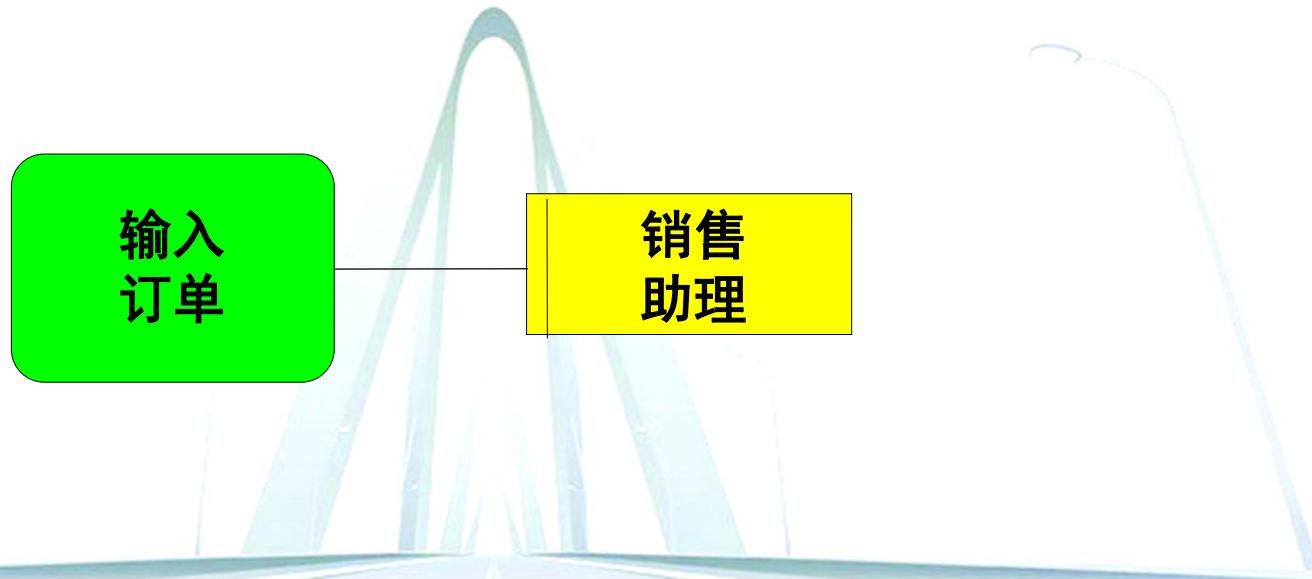
- 对象的属性数量由当前数据库所使用的过滤器方法决定。





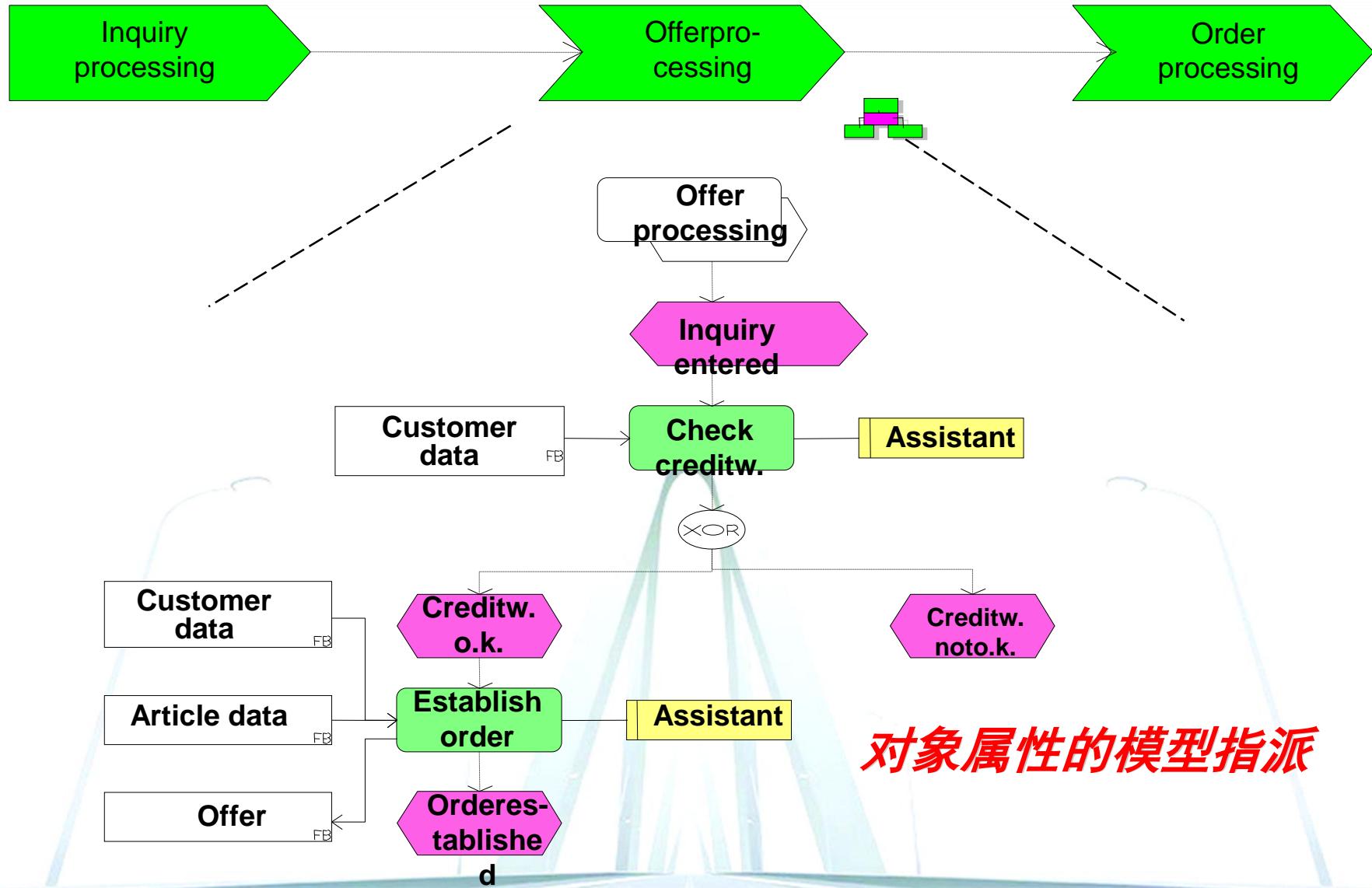
连结的创建：

- 连结表示两个对象之间的关系，因此连结总是始于源对象而终结于目标对象。





(3) 对象的指派、集成

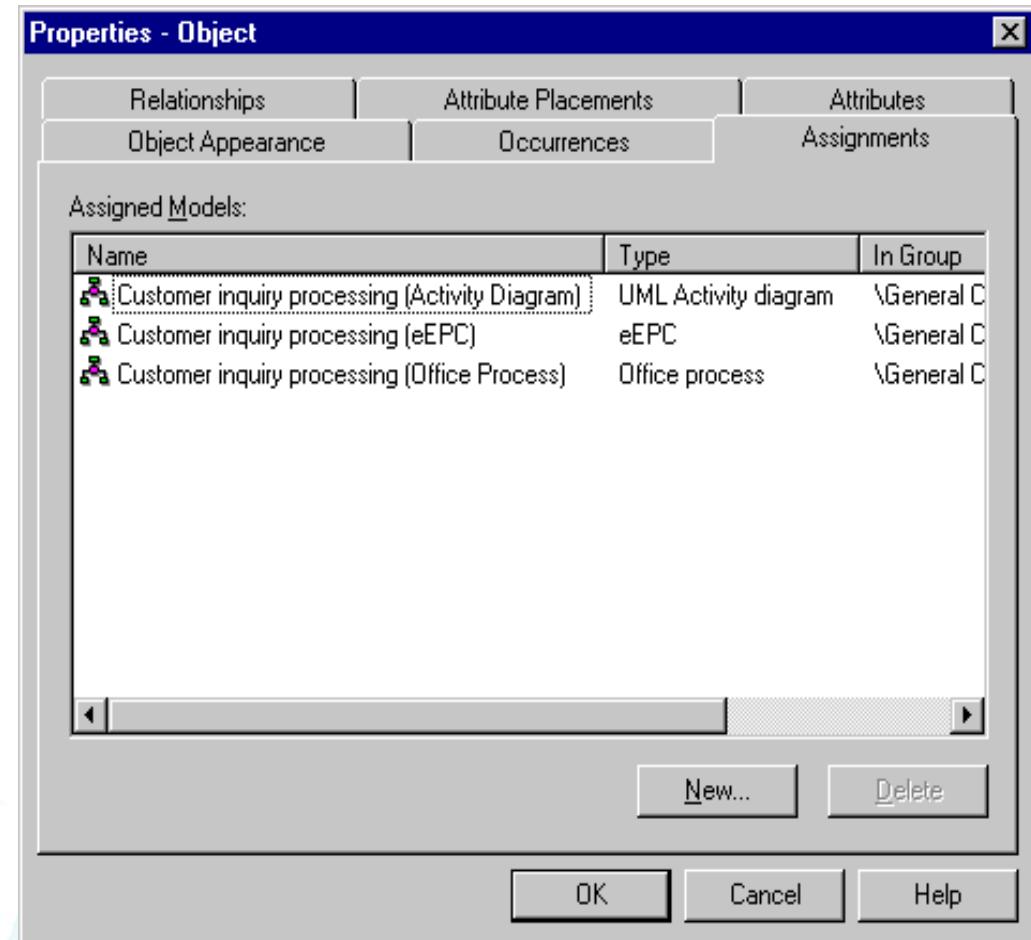


对象属性的模型指派



指派给对象的模型

Customer inquiry processing





OLE对象

- OLE表示对象嵌入与连接.
- 你可以在模型中插入其他应用的文件.
- 你可以在模型中为外部文件显示应用的符号或直接嵌入文档.
- 两种情况下，你后来都可以方便地编辑文件，应用文件可被ARIS工具集自动地激活.
- 如果同时嵌入和连接一个文件，所有后来由他人进行的对原文的修改都会自动在你的ARIS模型中显示出来.

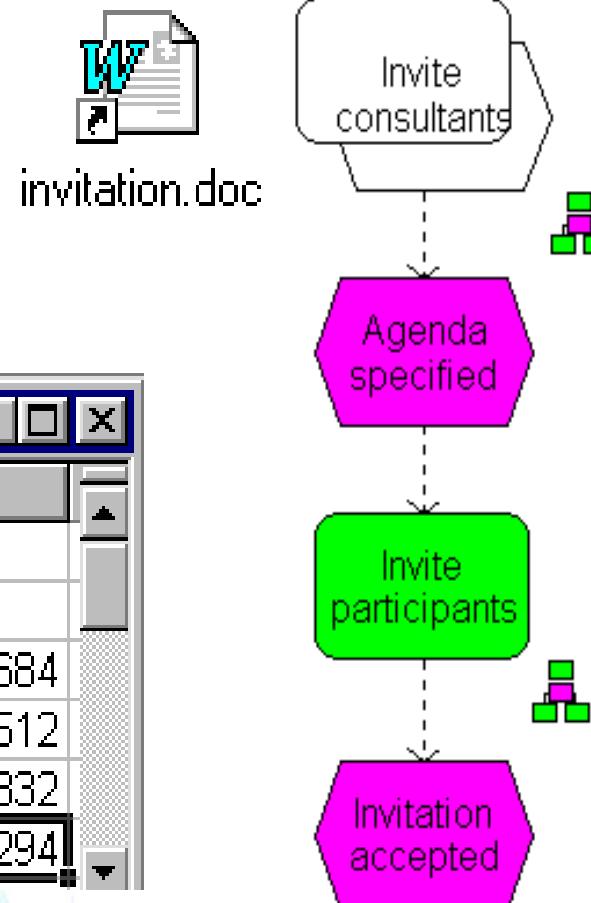


在模型中集成OLE对象？



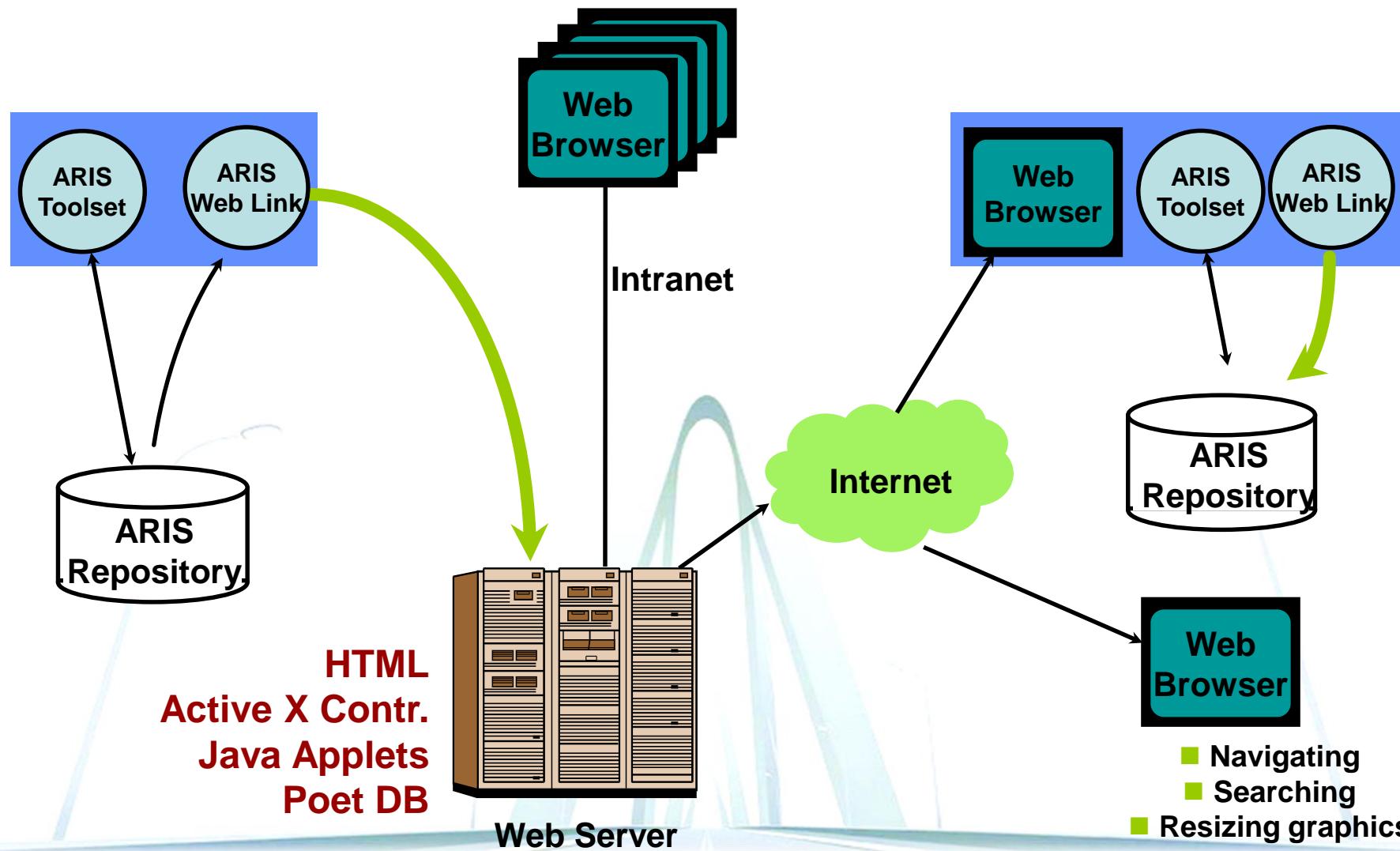
participantlist.xls

| | A | B | C | D |
|---|----------|------------|------------|-------|
| 1 | Name | First name | Depart. | Phone |
| 2 | | | | |
| 3 | Winter | Andrew | Support | 5684 |
| 4 | Miller | Bob | Production | 6512 |
| 5 | Smith | Judy | Sales | 7832 |
| 6 | Williams | Helen | Cust. Care | 4294 |





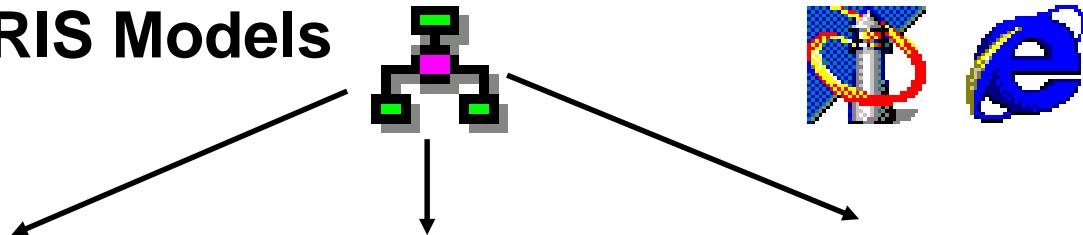
5.2 基于ARIS Web Link 的系统集成





Output types

ARIS Models



| | ActiveX | HTML | Java |
|----------------------------|---|----------|--|
| Browser / Platform: | Internet Explorer Windows 7 | -- | -- |
| Required Software: | ActiveX - Elements | | Webserver |
| Graphic format: | WMF | JPG; GIF | JPG; GIF |
| Functionality: | Download model links external links scaling | | Download model links external links |

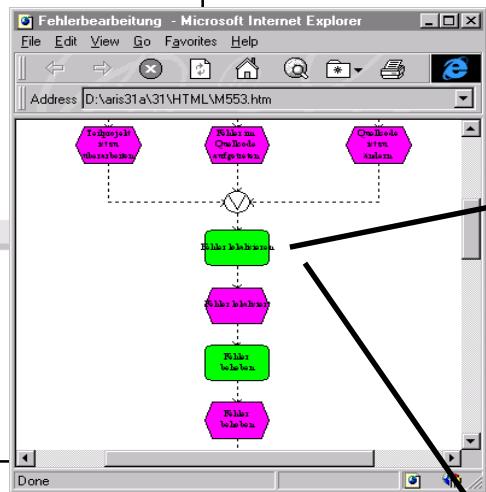


Functionalities in the resulting pages

MODEL LIST:

Organizational chart:

- [sales organization](#)



Model displayed in browser

Left mouse button

Right mouse button

Object page

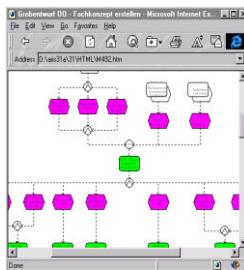
ARIS Web Link 4.0

Object: *check export procedure Function*

Object attributes:

| | |
|-----------------------|-----------------|
| Avg. material costs | 3.00 Euro |
| Avg. orientation time | 5.00 Minute(s) |
| Avg. other costs | 3.00 Euro |
| Avg. personnel costs | 25.00 Euro |
| Avg. processing time | 20.00 Minute(s) |
| Avg. total costs | 46.00 Euro |
| Avg. wait time | 60.00 Minute(s) |
| Creator | system |

Assigned models



Object Page
External Links
Assigned Models

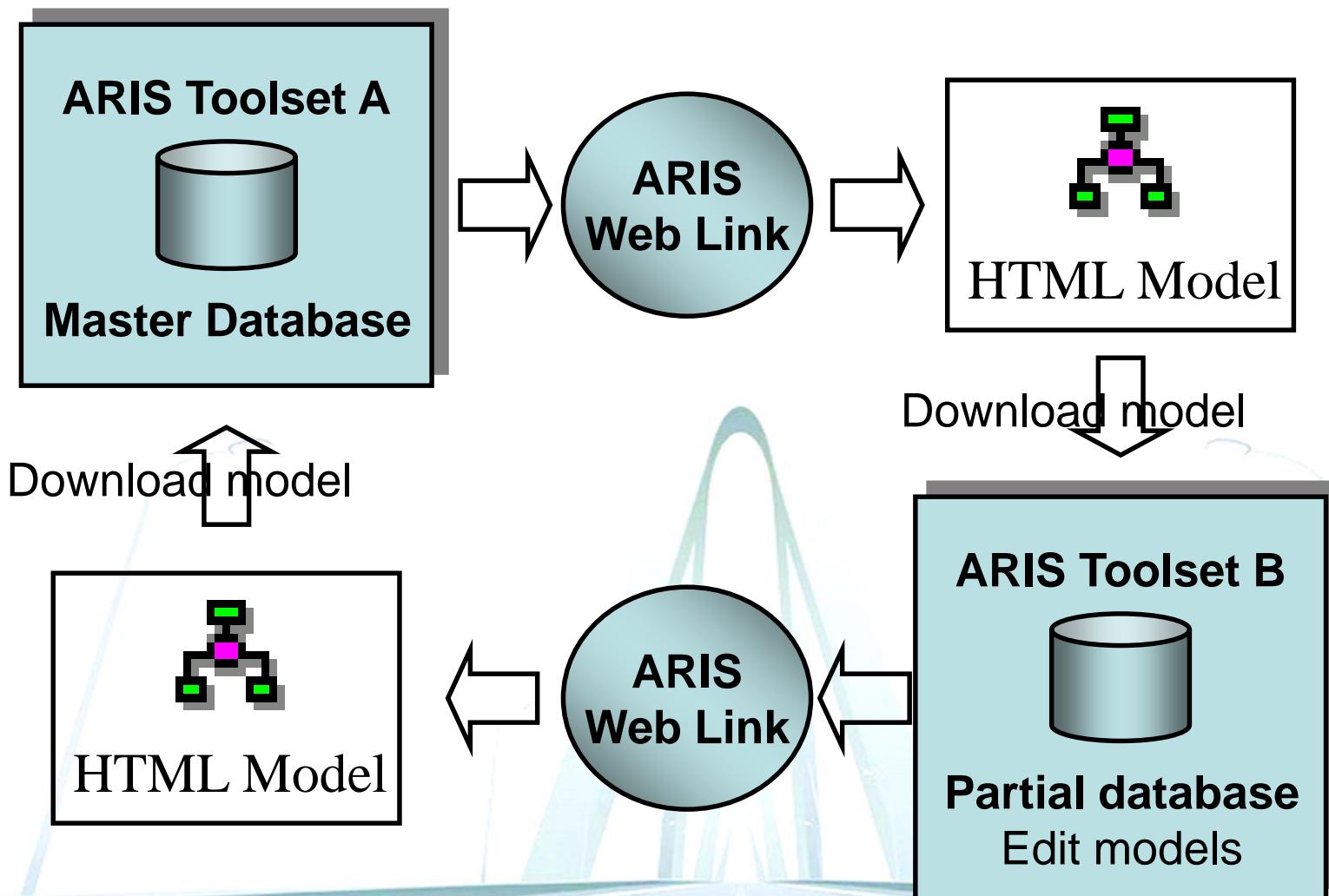
Zoom In +
Zoom Out-
Reset Zoom ×

External link

The screenshot shows a Microsoft Word document titled "ARIS-CORE.DOC" with the URL "D:\Programme\Microsoft\Office\Winword\Documents\ARIS-CORE.DOC". The content of the document discusses the evolution of object management and its impact on network computing.

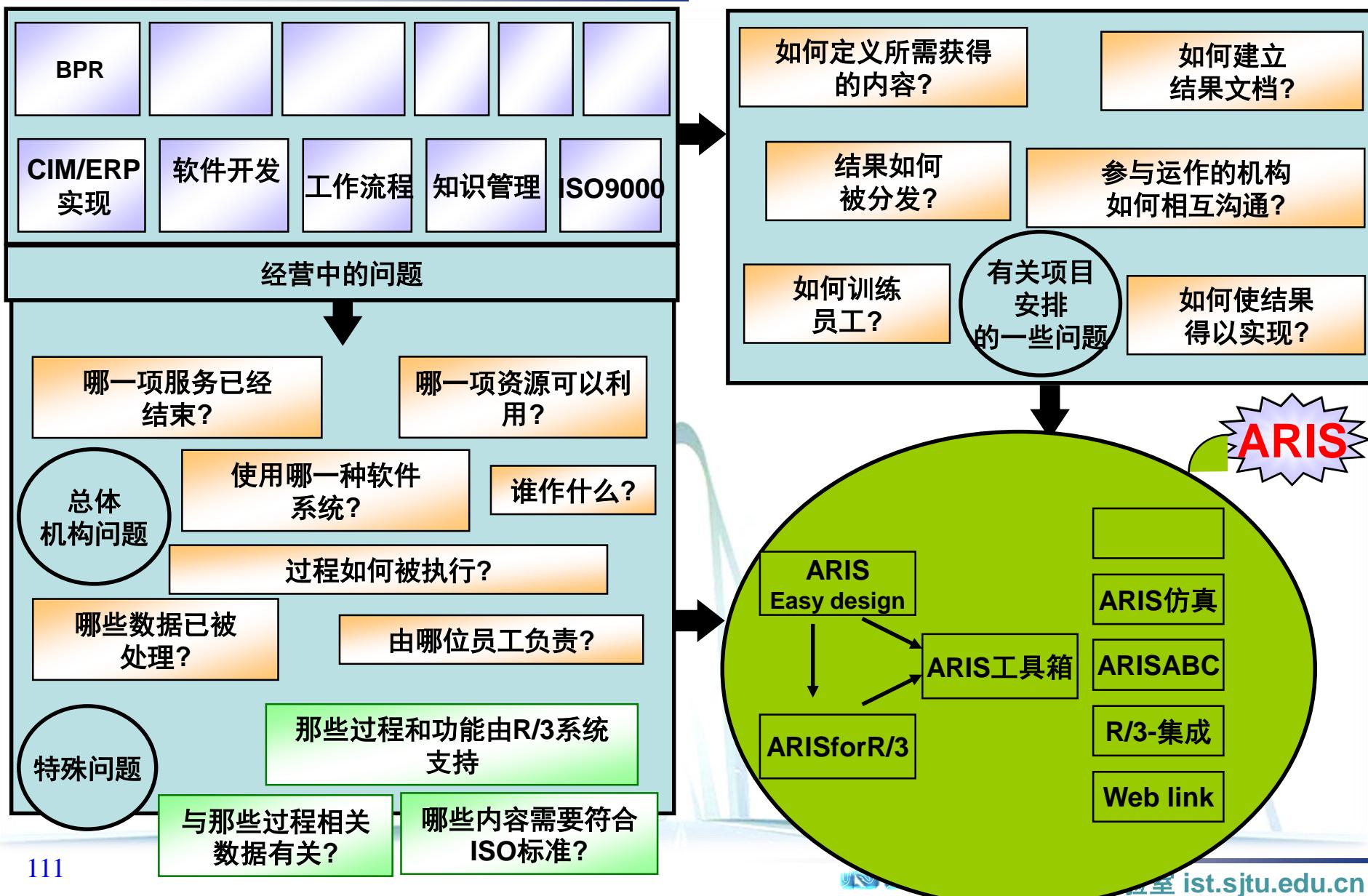


Projects with Web Link support



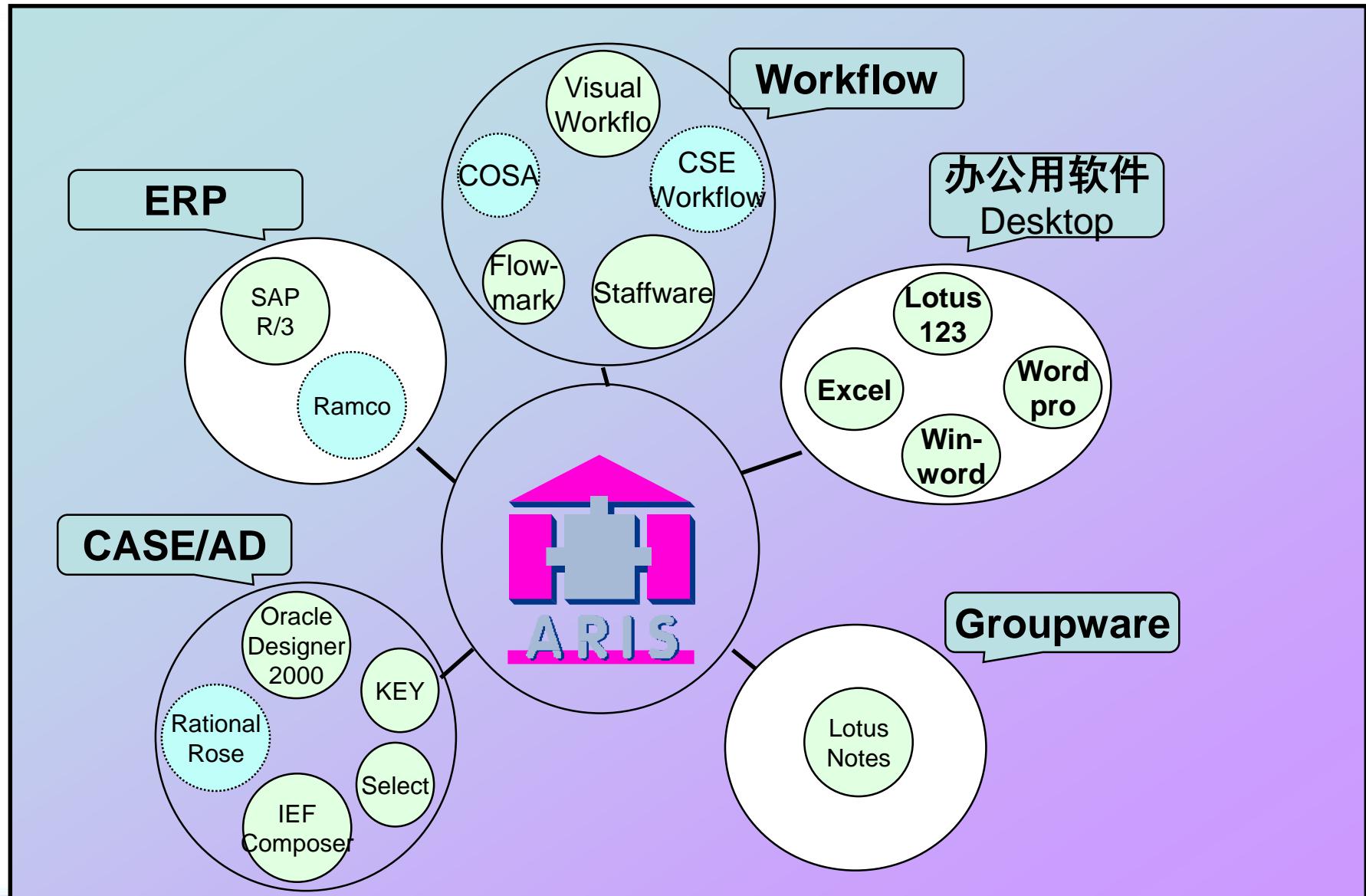


5.3 ARIS工具的综合应用





ARIS与其他软件的接口





Agenda

1. ARIS框架
 2. 基于EPC的过程建模方法
 3. ARIS视图
 4. ARIS建模过程实例
 5. ARIS软件工具
 6. 小结
- A1 BPMN建模

6 小结

- ④ ARIS 是一种面向过程的模型结构，以面向对象的方法描述了企业各个视图以建立完整的企业模型。
 - ARIS方法体系有一些成熟应用工具可以使用，助长了该方法的推广。
 - ARIS方法建立的业务流程模板可以作为蓝图重用，这是其在业务流程改进BPR中得以广泛应用的基础。
 - ARIS是面向过程建模的，所以在过程模型的基础上引入工作流管理系统成为可能。
 - ARIS 上的 Balanced Scorecard, ABC, Knowledge Management都是实际中用途较广的方法工具。
- ⑤ EPC作为一种基于事件的建模方法，被广泛应用于分析和设计过程，也是实现ERP的基石。外部两种趋势对EPC的进一步发展有重大影响：
 - 业务集成的趋势使得EPC成为业务视角和系统视角间的协调者。
 - 智能性、知识性的要求使得EPC成为基于过程模型融入本体、知识、业务知识的重要基础。



- 业务流程建模标注BPMN(Business Process Modeling Notation)，是用流程图形式来描述业务流程的一种方法。
- BPMN从2002年开始制定，2004年5月发布第一个版本，由对象管理组织OMG进行维护管理。目前版本为BPMN 2.0。
- BPMN定义了建立业务流程的图形模式：
- 业务流程图BPD(Business Process Diagram)。BPD由一系列的图形元素组成，所采用的图形化的标注使用一些简单的形状来区分。为了满足对业务流程描述的全面性，BPMN通过对基本元素增加一些额外信息及标注来进行扩展，扩展后的BPD图能够支持相对复杂的业务流程描述。



④ ARIS Express

- ARIS Express是ARIS的免费流程分析和管理工具，提供了对ARIS核心9个模型的支持，其中包括BPMN 2 Collaboration Diagrams, EPC, Organizational charts等的支持图。

⑤ ProcessOn 在线流程设计器

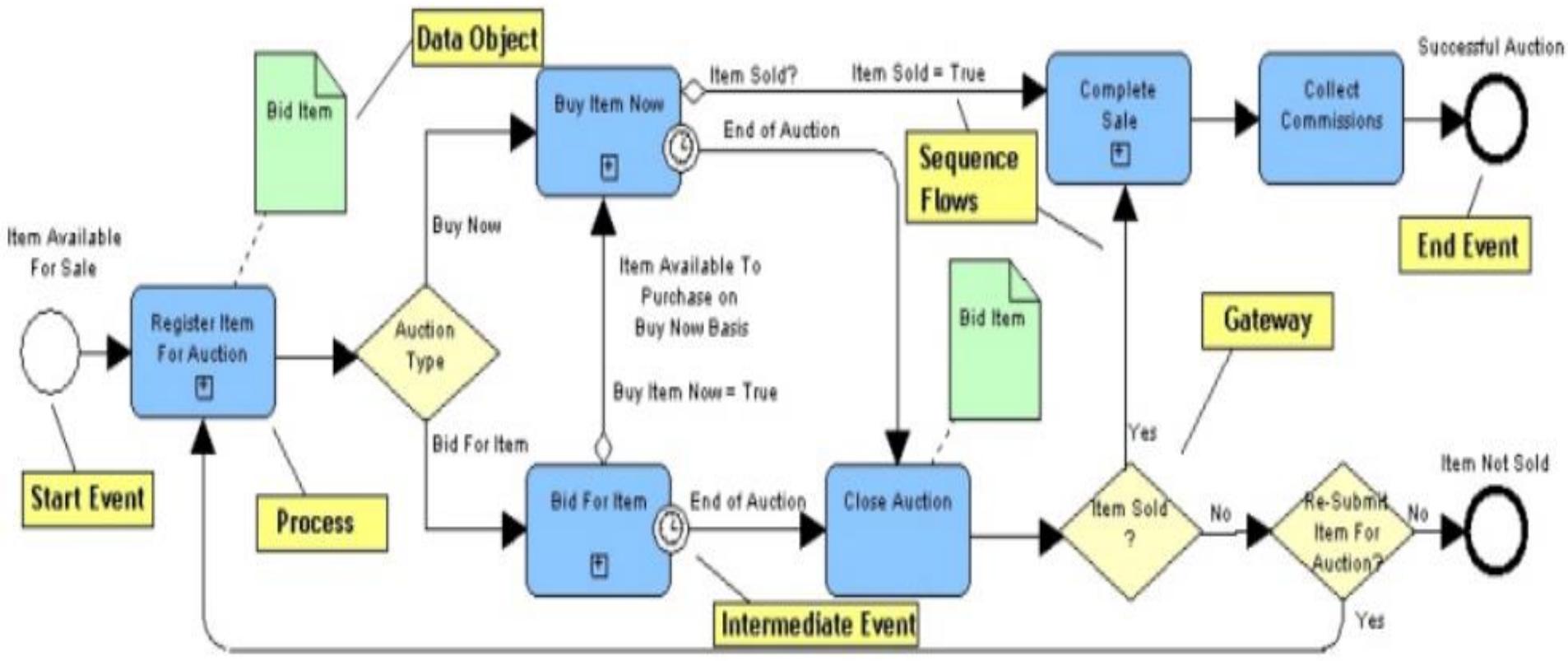
- ProcessOn流程图设计器支持BPMN2.0的三种标准类型建模 - Process, Collaboration和Choreography。用户在画BPMN2.0流程图的同时，还可以设置与特定图形相关的业务属性，ProcessOn内置了BPMN2.0展示层所要求的所有标准业务属性。

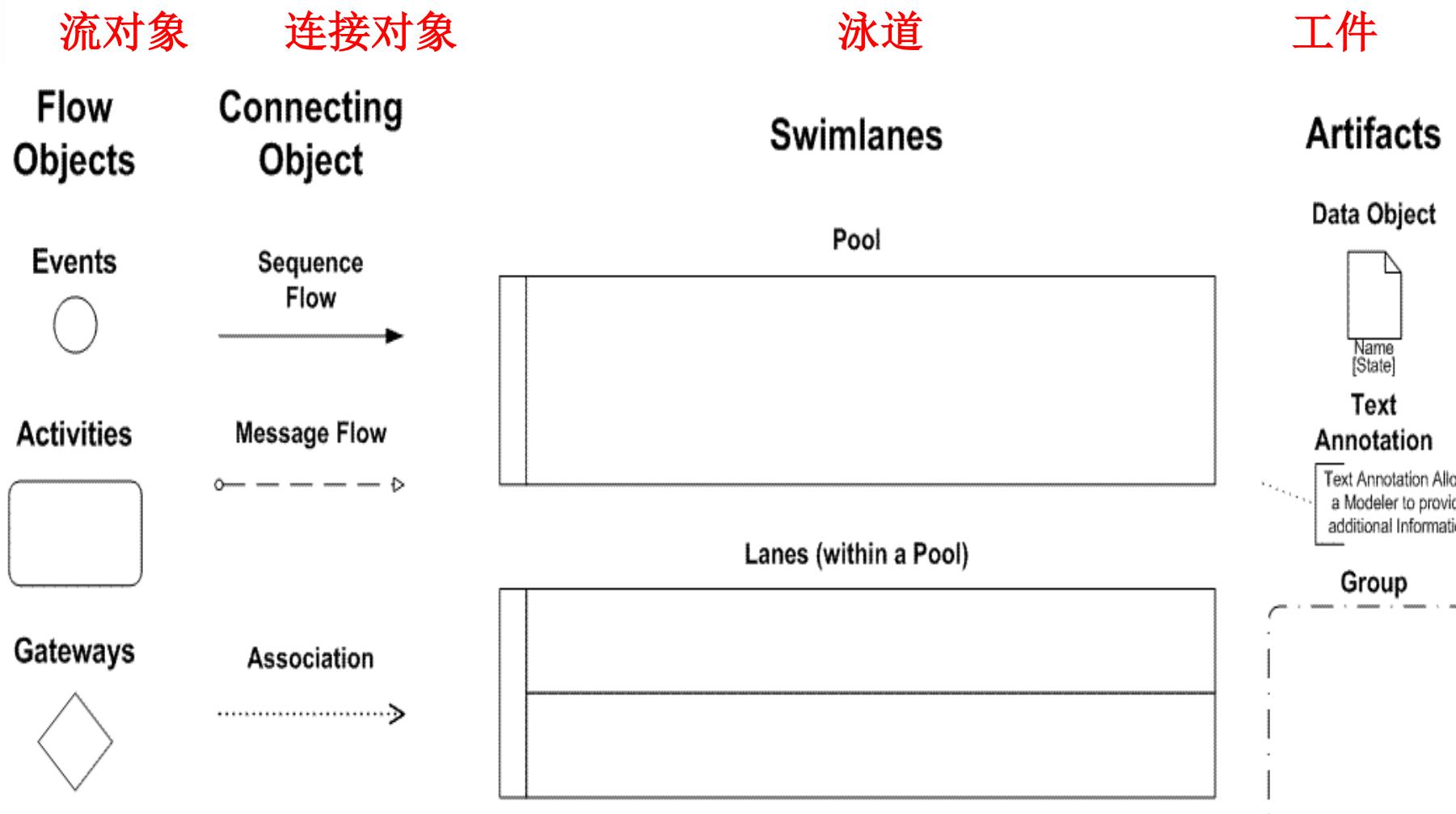
⑥ BPMN 2.0 Modeler for Visio description

- BPMN的2.0建模for Visio是微软Visio附件设计绘制和建模业务流程。这是一个全面的用户友好的软件包。2.0建模的BPMN为Visio支持的BPMN 2.0元素提出一套完整的对象（流对象，连接对象，泳道，工件和数据）。



案例：基于BPMN的在线拍卖业务流程图







流对象的三元素

| | |
|--------------------|--|
| 事件 (开始, 中间, 结束) | |
| 活动 | |
| 判断/网关 | |

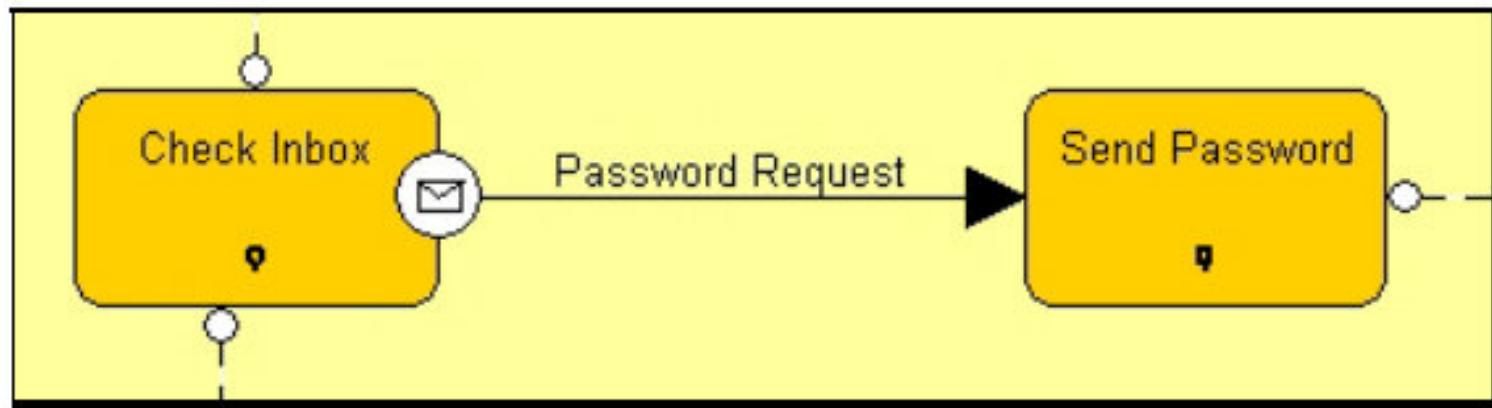


流对象—事件

| Start Events | Intermediate Events | End Events | Description |
|--|--|---|--|
| Start Message  | Message  | End Message  | A start message arrives from a participant and triggers the start of the process, or continues the process in the case of an intermediate event. An end message denotes a message generated at the end of a process. |
| Start Timer  | Timer  | A Timer cannot be an End Event. | A specific time or cycle (for example every Monday at 9am) can be set to trigger the start of the process, or continue the process in the case of an intermediate event. |
| Start Rule  | Rule  | A Rule cannot be an End Event. | Triggers when the conditions for a rule become true, such as "Stock price changes by more than 10% since opening." |
| Start Link  | Link  | End Link  | A link is a mechanism for connecting the end event of one process flow to the start event of another process flow. |



◎ BPMN中的事件触发机制



案例：Check Inbox过程结束触发信息事件，传递 Password Request 信息到Send Password过程。



- 活动元素包括：过程、子过程、任务。

过程

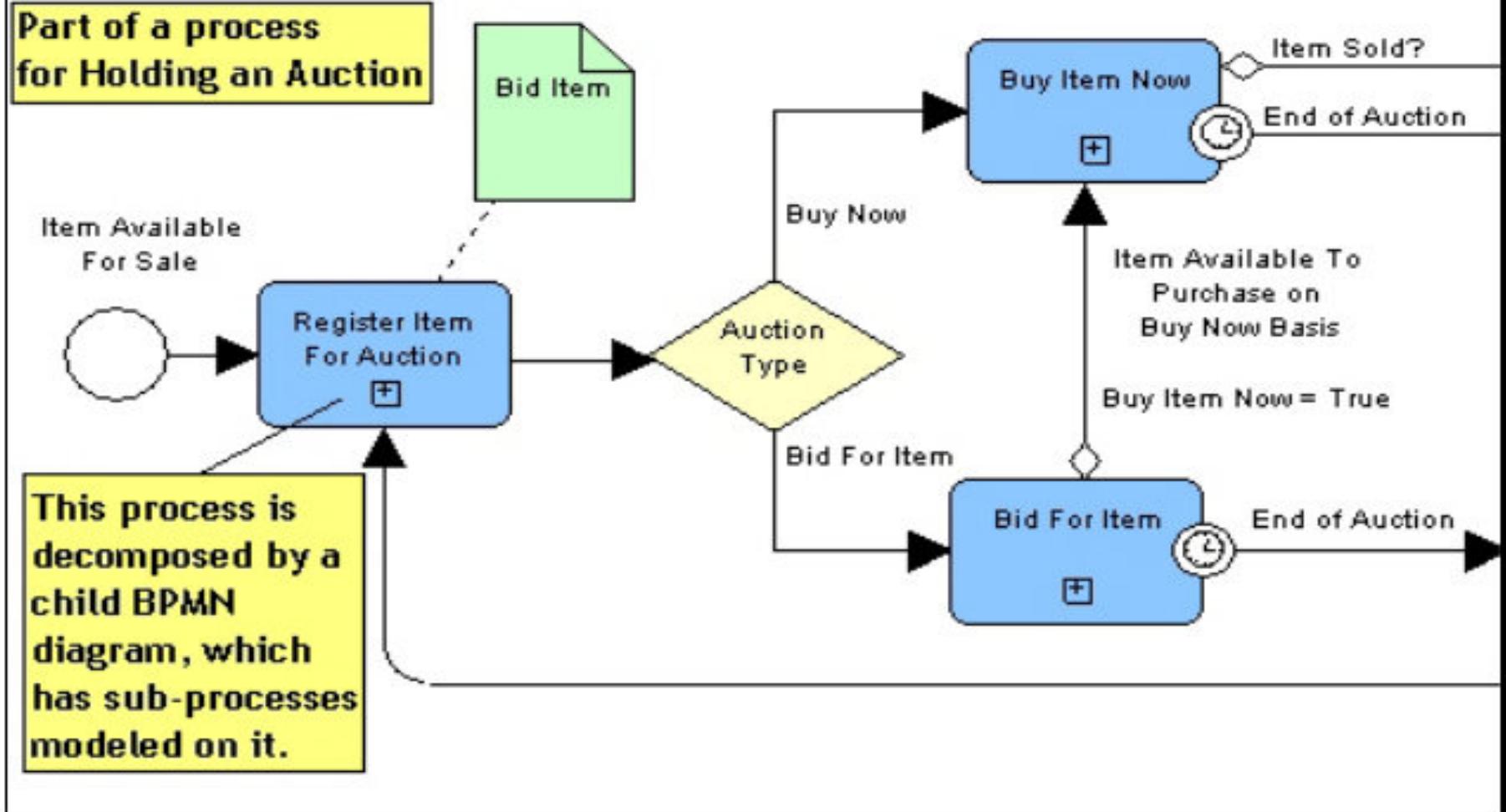
子过程

任务

过程元素对具体业务流程进行描述，根据其对流程描述的详细程度可分为过程、子过程、任务，其中任务作为最小的流程单位，不可再分。

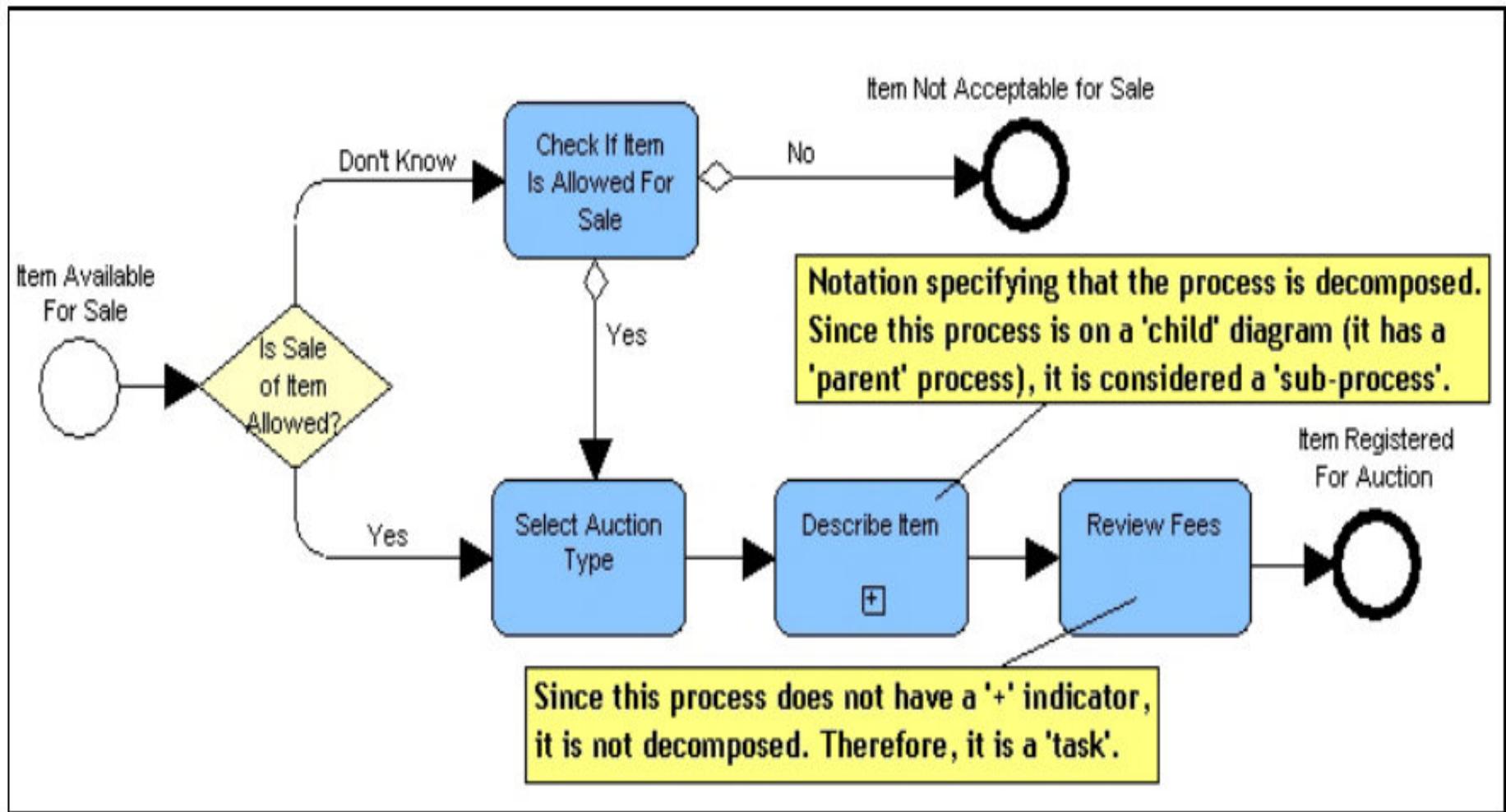


过程与子过程





子过程与任务





网关用来表示流程的分支和聚合。Gateway在BPMN是个很复杂的概念，不同的网关代表了各自的路径选择和控制方式。在Gateway的组合使用时需要注意流程描述的合理性问题。

| | | |
|----------------------------|------|--|
| 异或 - 分支\合并 (XOR) | 基于数据 | |
| | 基于事件 | |
| 排他 - 分支\合并(OR) | | |
| 复杂 - 分支\合并(Complex) | | |
| 并行 - 分支\合并(AND) | | |

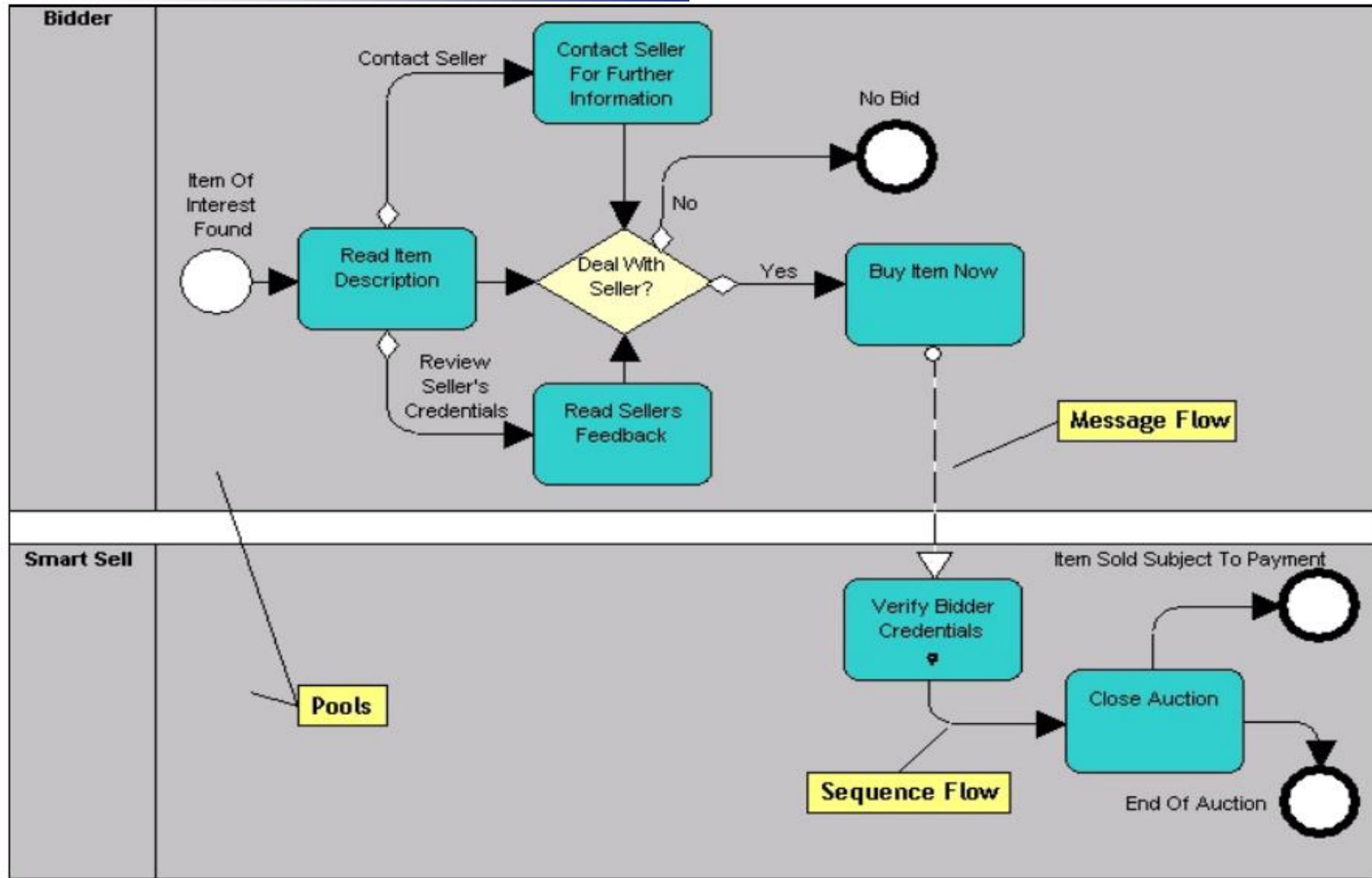


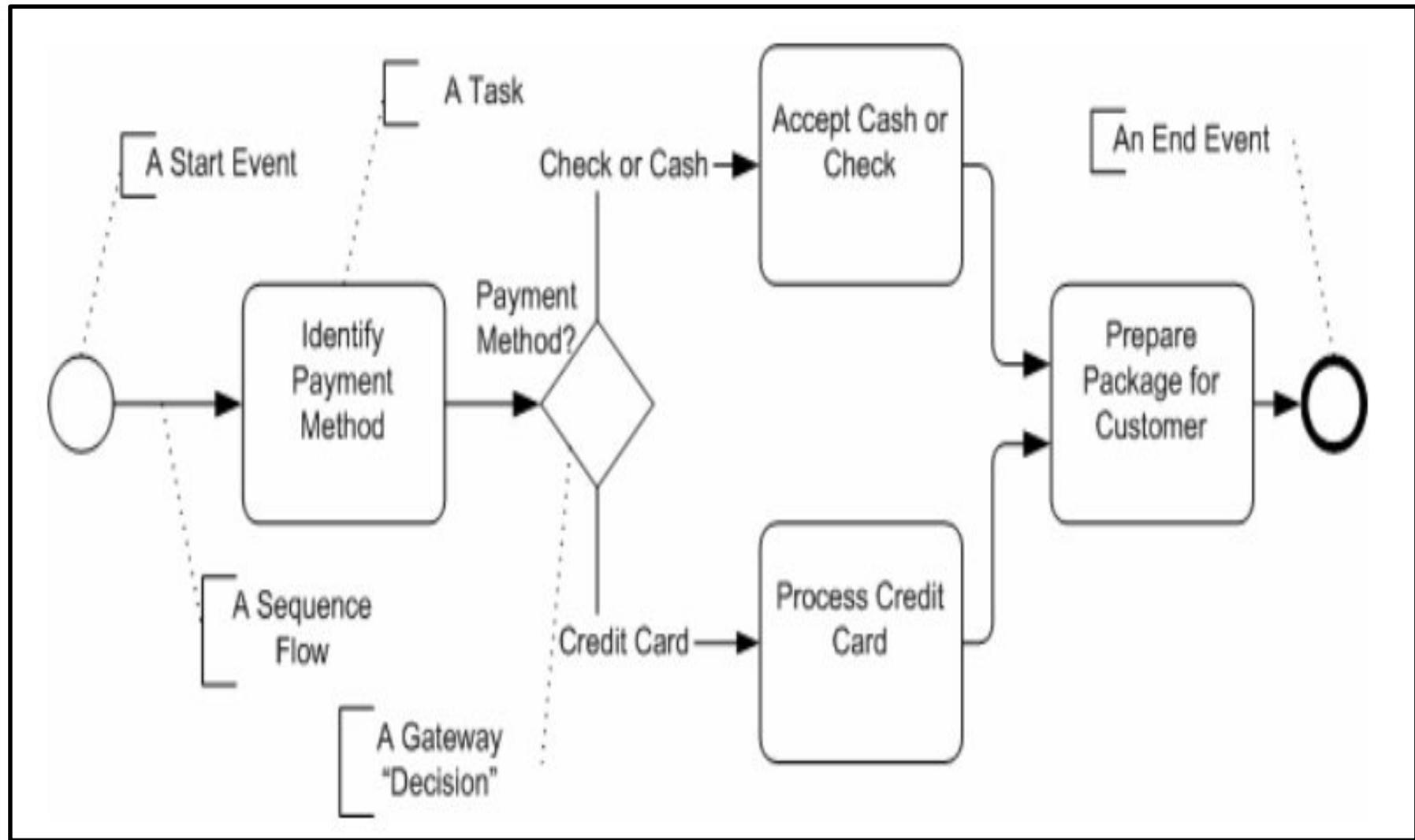
连接对象基本元素

| 名称 | 描述 | 符号 |
|-----------------------------|--|------------|
| 顺序流 Sequence Flow | 在一个业务流程中，顺序流是用于描述活动进行的先后顺序的。在顺序流上可以标注一定的条件。 | → |
| 消息流 Message Flow | 消息流则用于描述两个参与者之间消息的传递。在BPD图中，表述为泳池间的消息传递。 | ○→ |
| 关联 Association | Association的主要作用在于连接BPMN图形元素与附件。带有箭头的关联可以在需要的时候标明数据的流向，一般用于表示Data Association。 | —→ ---- |



连接对象—顺序流、信息流







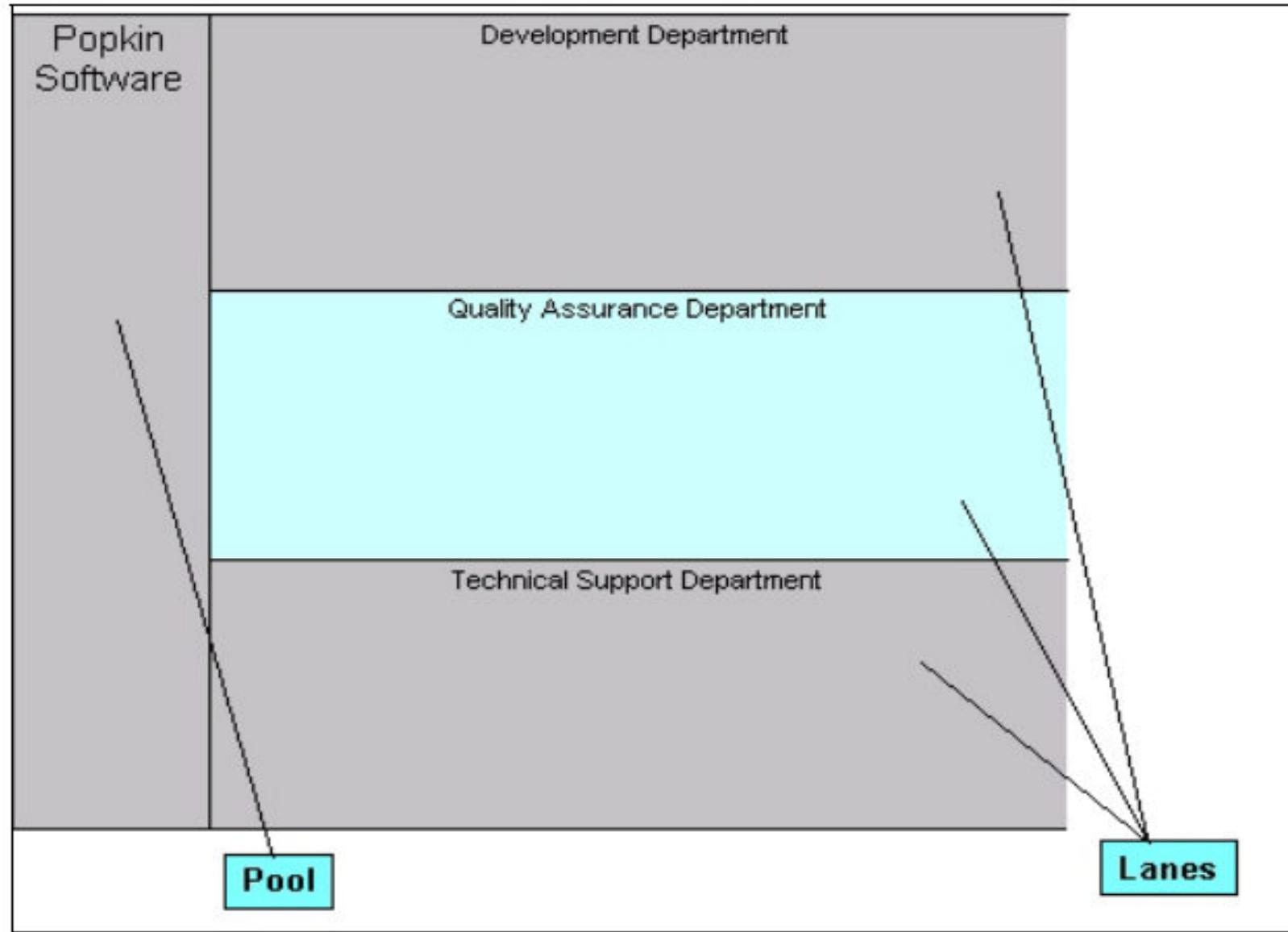
④ 泳道



⑤ 子泳道

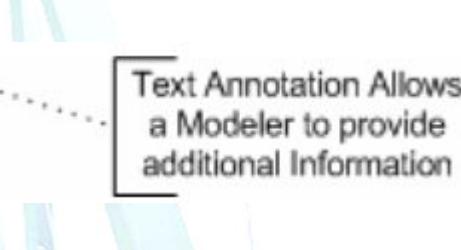


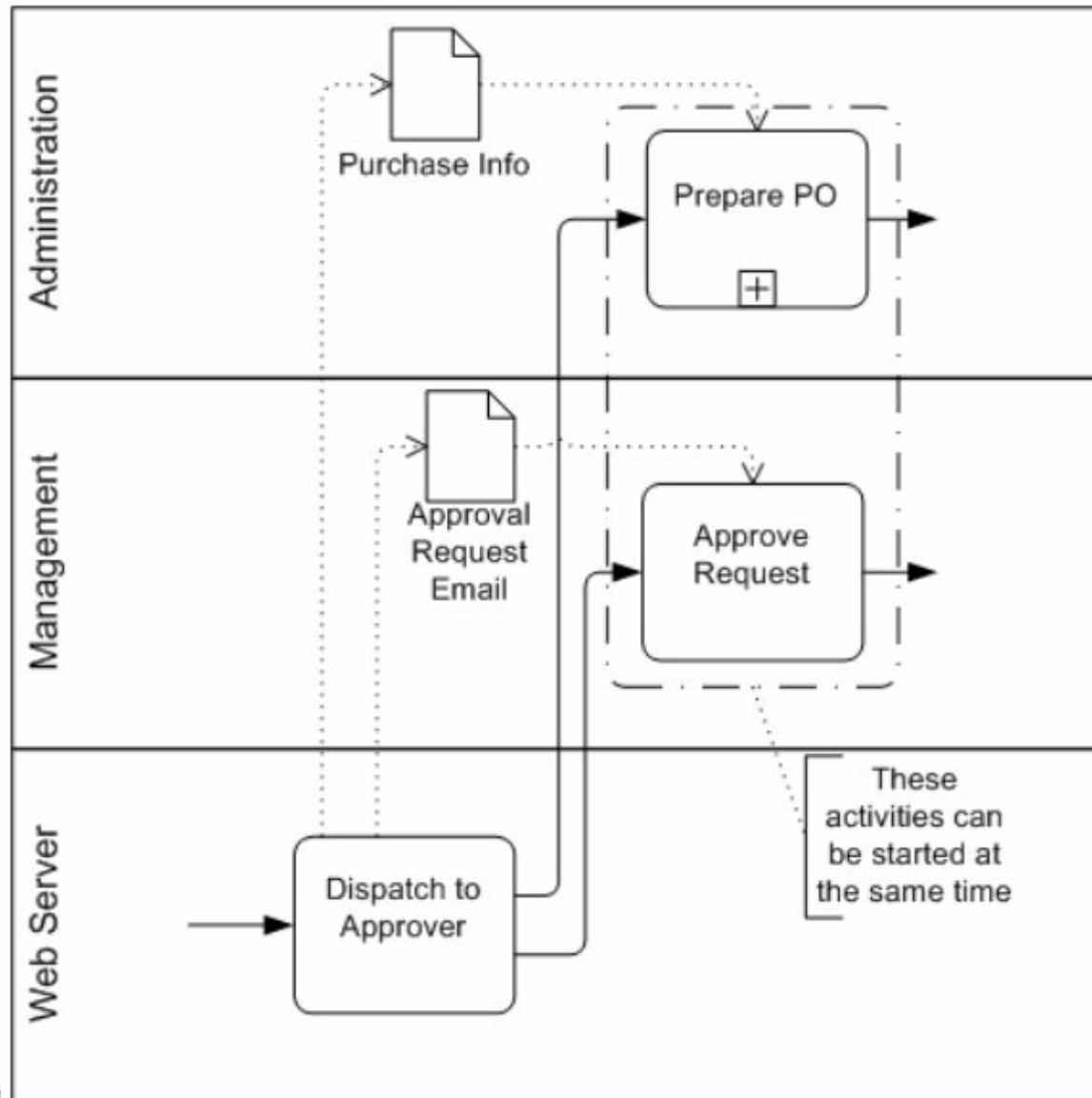
泳道用途多样，可用来划分组织（企业或部门）、职能（企业职能或虚拟组织）、应用（计算机程序）、位置（公司的物理位置）、类（面向对象程序中的模块）、实体（数据库中的逻辑表）等。

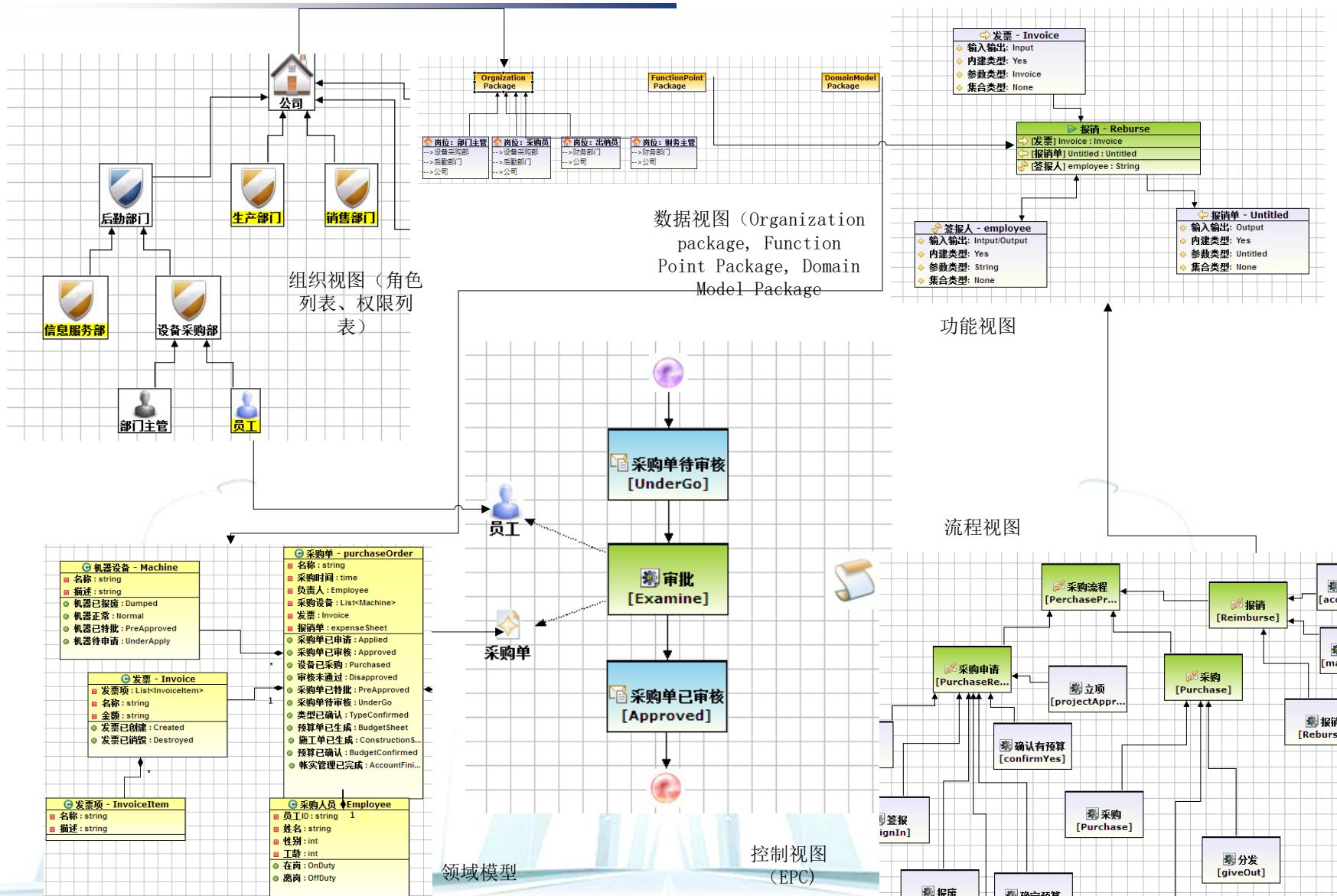




Artifact(工件)的三要素

| | |
|------|---|
| 数据对象 |  |
| 工作组 |  |
| 注解 |  <p>Text Annotation Allows a Modeler to provide additional Information</p> |







信息系统分析与设计

Lecture 5

企业资源管理

蔡鸿明

hmcai@sjtu.edu.cn

软件学院信息系统技术实验室

lst.sjtu.edu.cn





1. 企业资源管理基础
 2. 库存管理
 3. 物料需求计划
 4. 能力需求计划
 5. 供应链管理
 6. 小结
- A1 信息编码方法



1 企业资源管理基础

ERP是从MRP(物料资源计划)发展而来的新一代集成化管理信息系统，是一个整合了企业管理理念、业务流程、基础数据、人力物力、计算机硬件和软件于一体的企业资源管理系统；

ERP (Enterprise Resource Planning)

企业 资源 计划

- ① 企业：通常所说的企业，一般是指从事生产、流通或服务等活动，为满足社会需要进行自主经营、自负盈亏、承担风险、实行独立核算，具有法人资格的基本经济单位。以什么为目的？

- ② 资源 有形资源，无形资源

- ③ 计划 为实现目标而拟定的具体内容和步骤

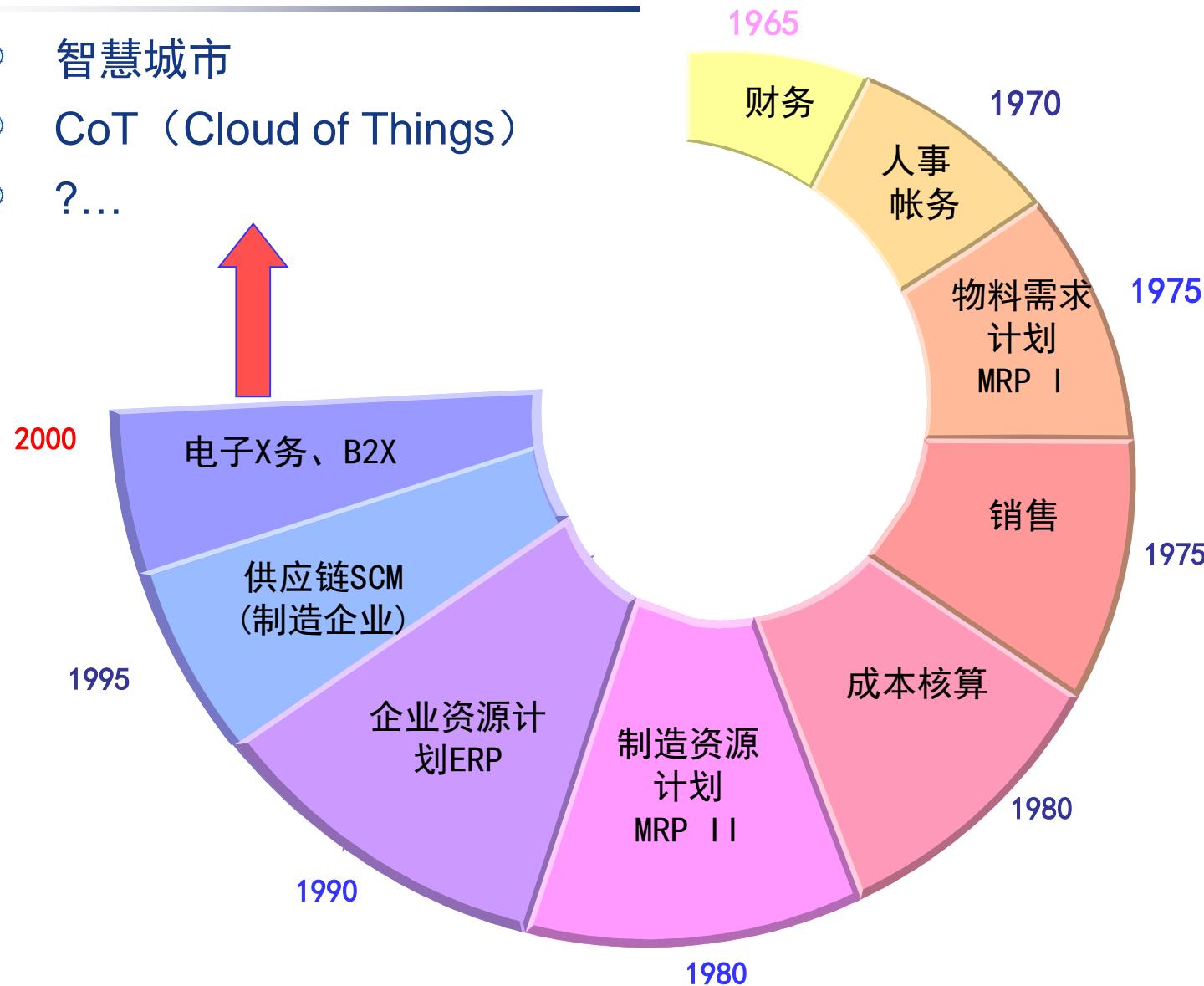
ERP系统主要功能：

- SAP的基础模块：SD,MM,PP, FI&CO
- ERP的三大流程：采购，销售，生产



管理信息系统的发展历程

- ◎ 智慧城市
- ◎ CoT (Cloud of Things)
- ◎ ?...

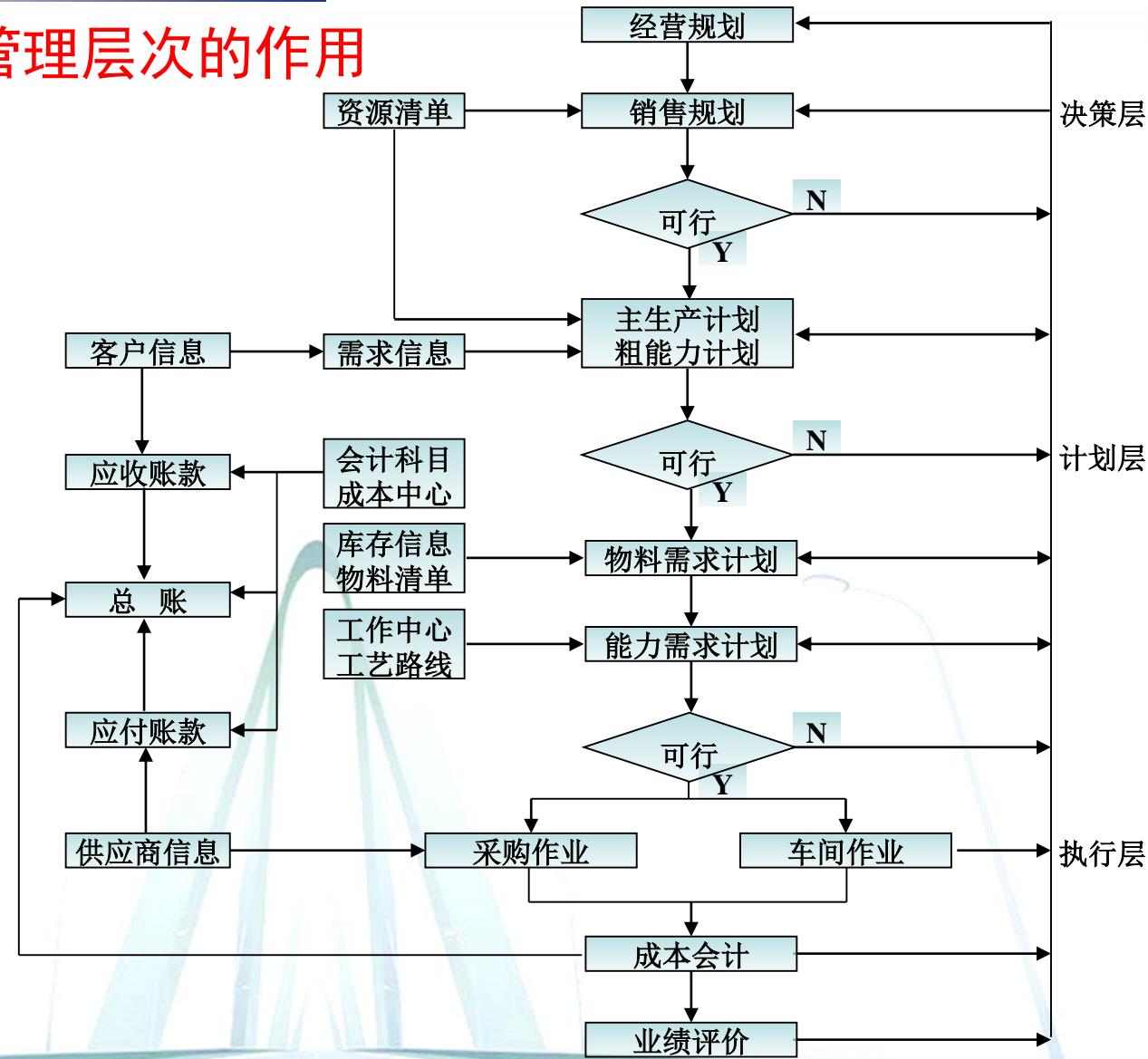




制造资源计划

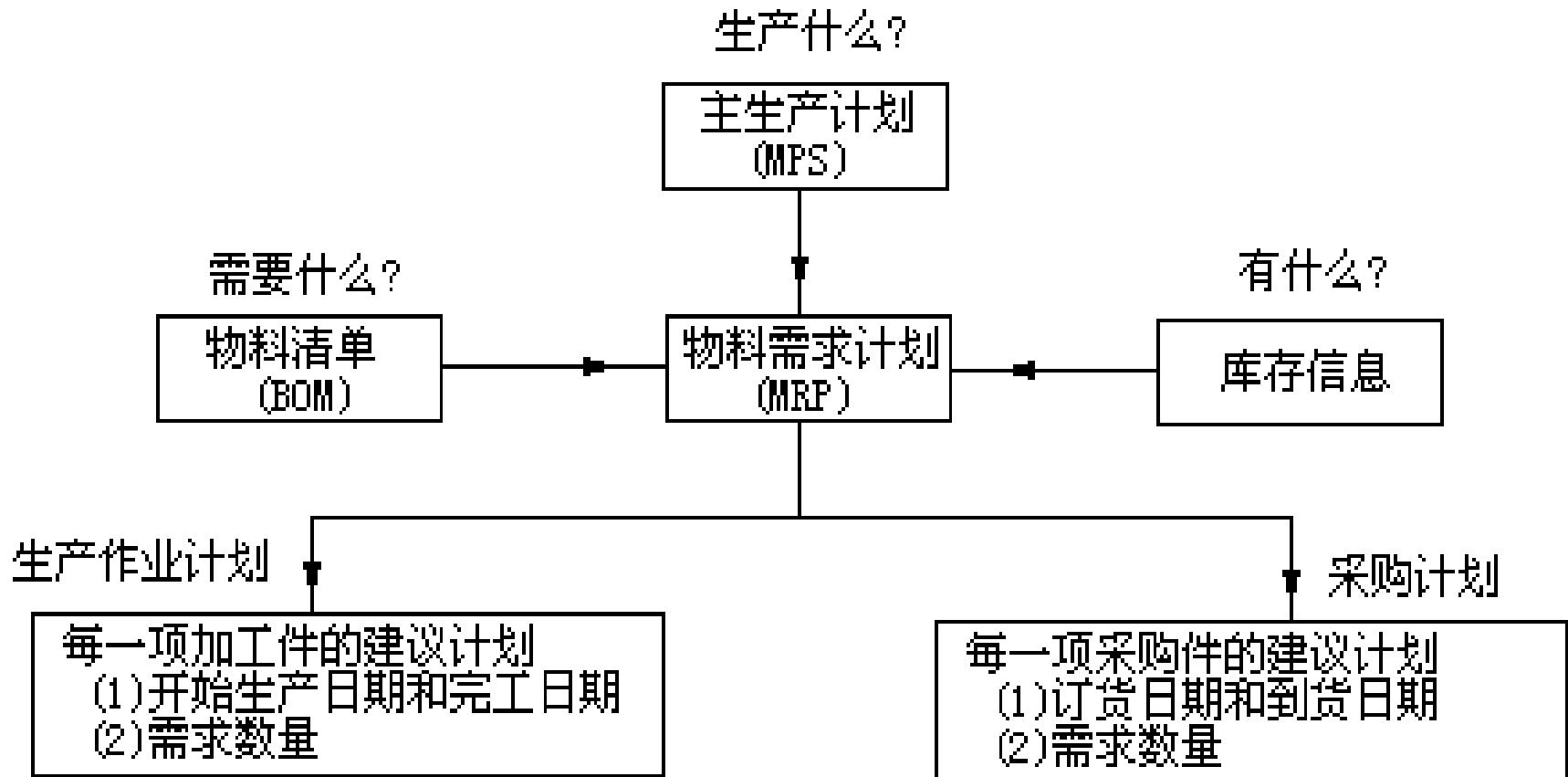
ERP系统在不同的管理层次的作用

- 在决策层，ERP系统可以支持战略计划制定及参考。
- 在计划层，ERP系统可以促进实时管理的实施。
- 在执行层，ERP系统可以降低业务成本，保证计划及时执行。





物料、库存、计划之上的需求管理理论





(1) 生产类型

- 备货生产 (**Make To Stock-MTS**)
- 订货生产 (**Make To Order-MTO**)
- 订货组装 (**Assemble To Order-ATO**)
- 工程生产 (**Engineer To Order-ETO**)

- 不同的生产类型，生产拉动的方式完全不同，带来计划模式差别很大。生产类型是需求制定的首先考虑问题。



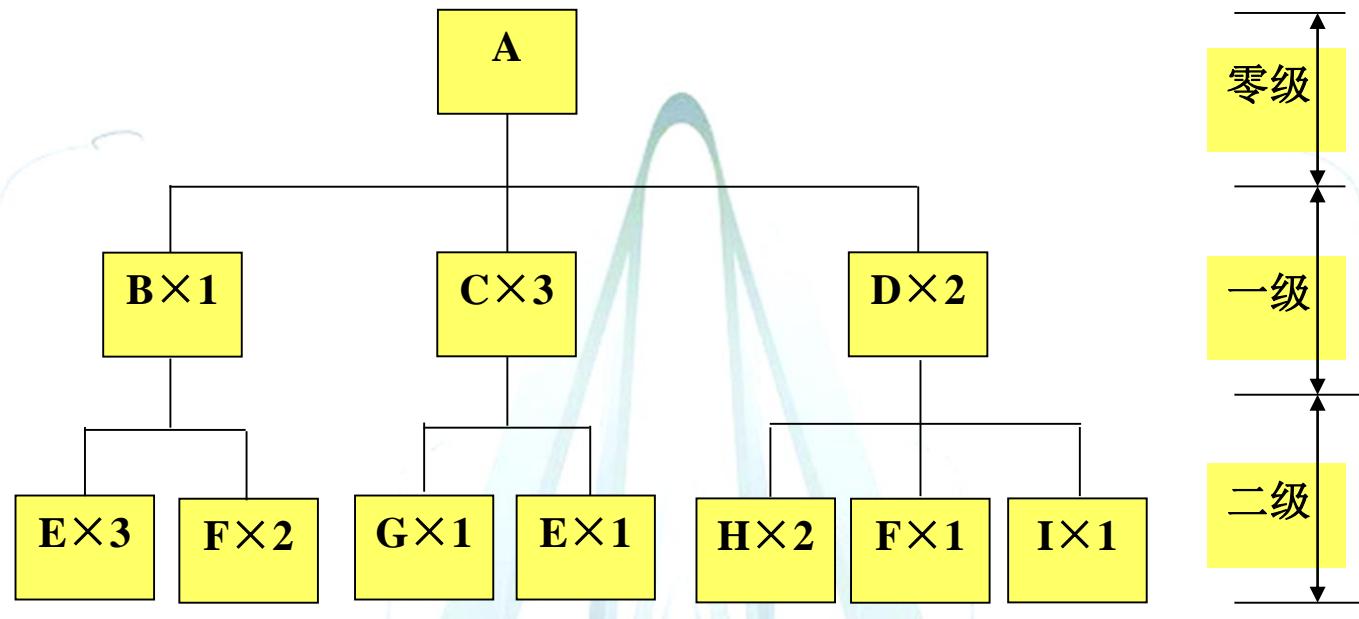
(2) 物料编码

- 物料编码有时也叫物料代码，是计算机系统对物料的**唯一识别代码**。
- 每种物料必须有对应且唯一的物料编码；
- 设计编码体系要科学合理；
- 按国家、企业的规定设置；
- 一经确定不能轻易修改，但要有一定扩充性；
- 便于使用和记忆
- 编码是管理思想的体现，管理在前，分类在后。



(3) 物料清单

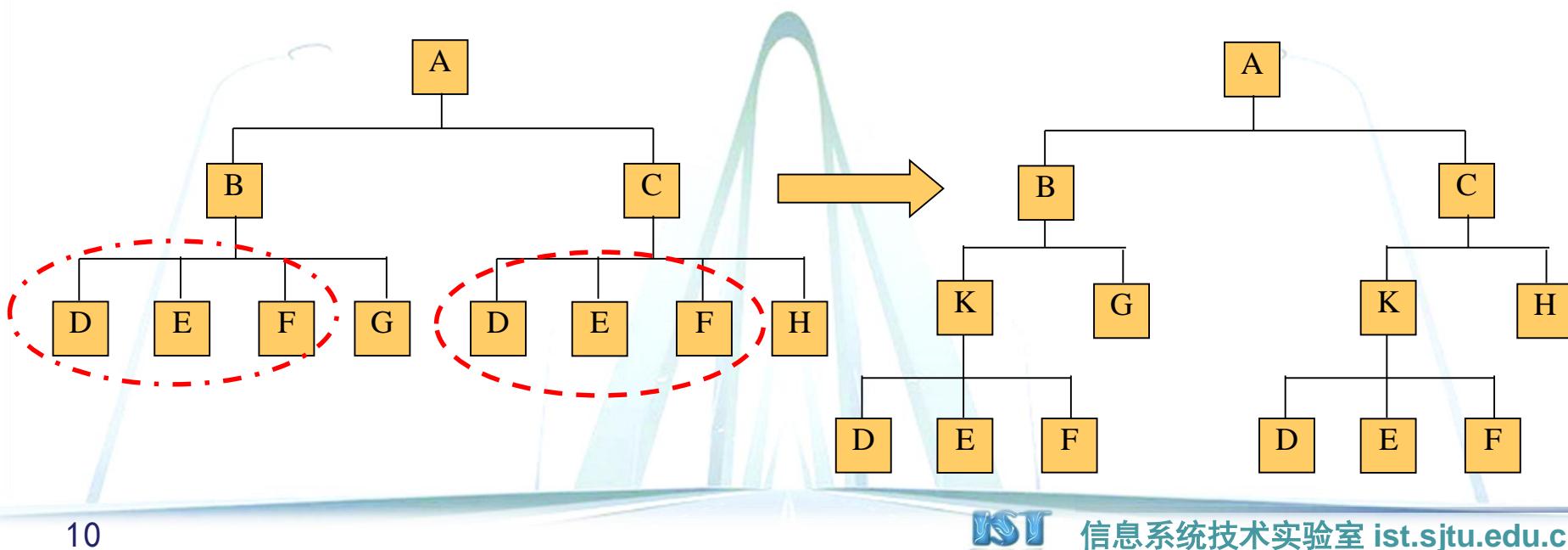
- 物料清单(BOM—Bill of Materials)是描述产品结构的文件。它表示了产品的组件、子件、零件直到原材料之间的结构关系，以及每个组件所需要的各下属子件的数量。





④ BOM中的虚拟件

- ④ 定义：为达到一定的管理目的而虚拟的、实际并不存在的物品。
- ④ 作用：仅用于简化管理，提高管理效率，无提前期，不是计划的对象



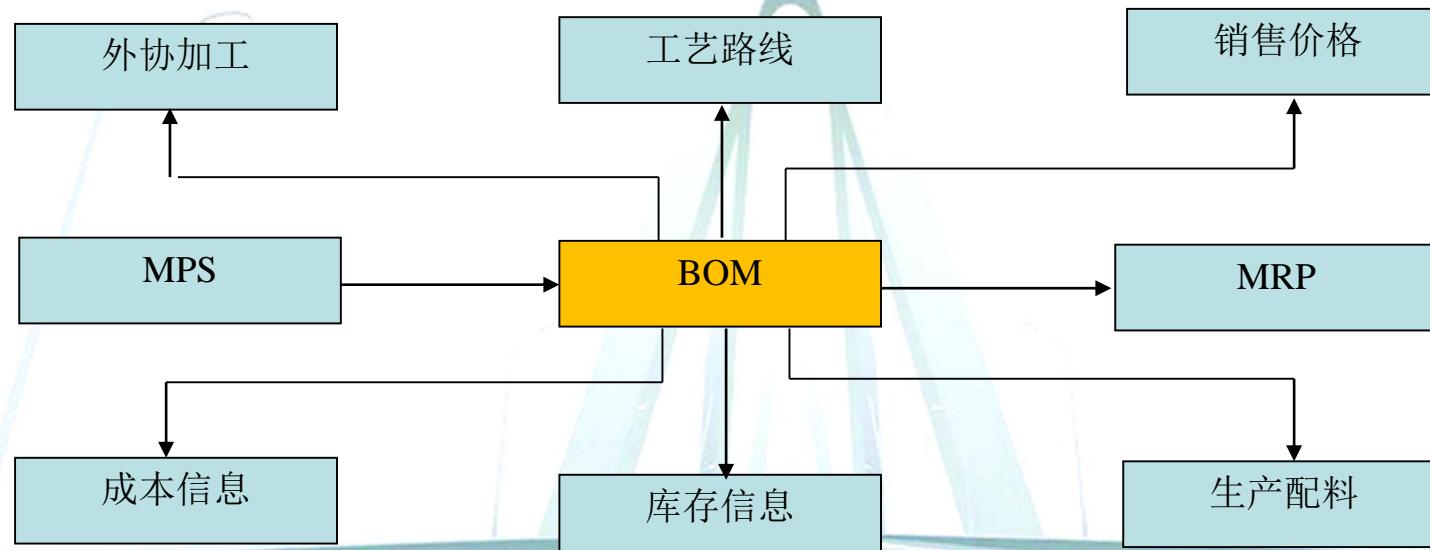


④ BOM种类按照使用方式划分：

- 基本型物料清单
- 计划型物料清单
- 模块型物料清单
- 成本型物料清单

还可以划分为设计BOM，制造BOM等

⑤ 物料清单作为成本控制，信息管理的核心，具有极其重要性，要保证物料清单的准确性在98%以上。



(4) 工作中心/设备

- 工作中心（Working Center,简称WC）是生产加工单元的统称，在完成一项加工任务时同时也发生了加工成本。它是由一台或几台功能相同的设备，一个或多个工作人员，一个小组。
- 关键工作中心（Critical Work Center），有时也称为瓶颈工序（bottleneck），是运行粗能力计划的计算对象。
- 特点：
 - （1）经常满负荷，加班加点工作；
 - （2）需要技术熟练工人操作，不能任意替代；
 - （3）工艺独特的专用设备，不能替代或分包外协；
 - （4）价格昂贵的设备，不能随时增加；
 - （5）受多种限制，如短期内不能随便增加负荷和产量（通常受场地、成本等约束）



工艺路线 (Routing)

主要说明物料实际加工和装配的工序顺序、每道工序使用的工作中心，各项时间定额（如：准备时间、加工时间和传送时间，传送时间包括排队时间与等待时间），及外协工序的时间和费用。

工艺路线也即是加工的流程。

表 3.8 工艺路线

物料代码 80021——定位栓

| 工序号 | 部门 | 工作中心 | 描述 | 准备时间 | 每件加工时间 |
|-----|-------|------|----|------|--------|
| 10 | 00800 | 1 | 下料 | 0.5 | 0.010 |
| 20 | 00832 | 2 | 粗车 | 1.5 | 0.030 |
| 30 | 00833 | 3 | 精车 | 3.5 | 0.047 |
| 40 | 00811 | 1 | 检验 | | |



④ 工艺路线的作用

- 用于MRP计划和能力计划的生成与平衡；
- 计算物料提前期；
- 生成车间生产作业计划；
- 物料成本核算；
- 跟踪控制产品的生产。

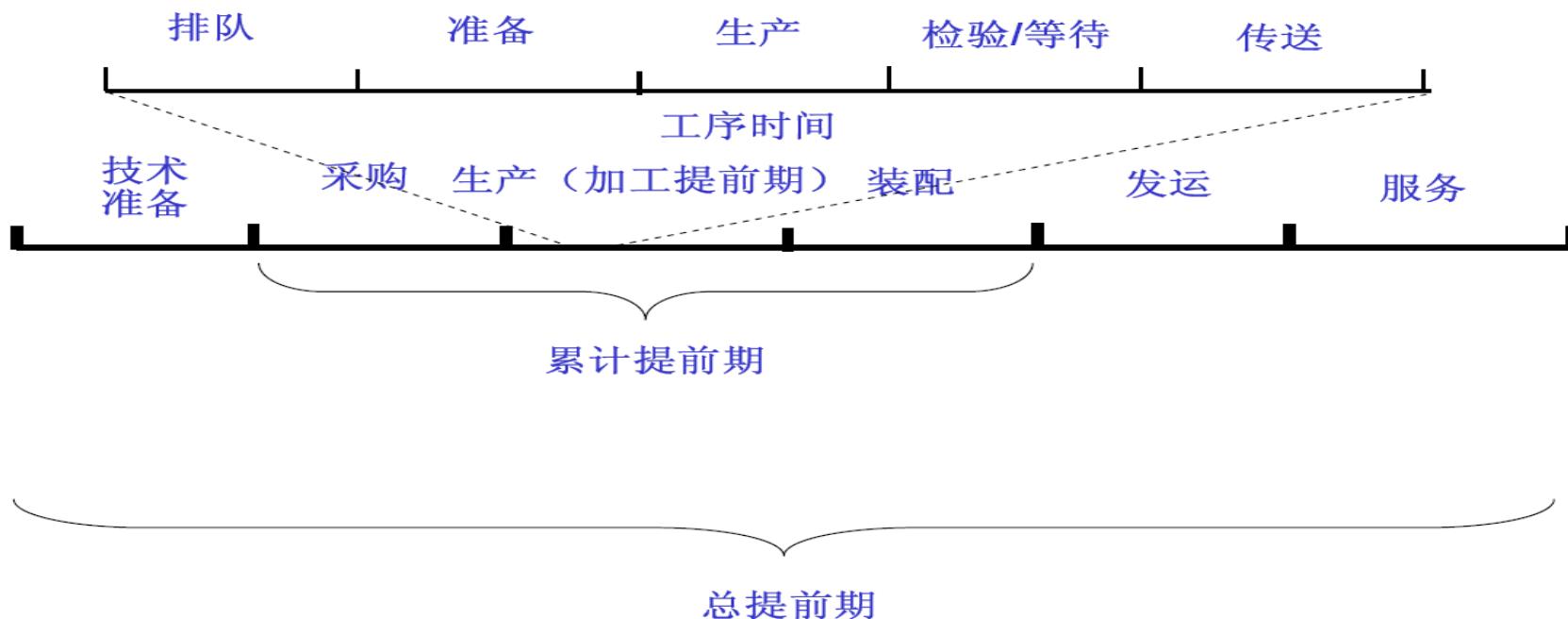
⑤ 专门用作工艺路线应用的**CAPP**（Computer Aided Process Planning）系统。

- **CAPP**是将产品设计信息转换为各种加工制造、管理信息的关键环节，是企业信息化建设中联系设计和生产的纽带，同时也为企业的管理部门提供相关的数据，是企业信息交换的中间环节。



(6) 提前期与计划展望期

- 生产加工（或装配）提前期：
 - 构成（排队、准备、加工、等待和传送）
- 固定提前期：与加工数量无关，常用于物料采购。
- 变动提前期：与加工数量有关，常用于物料加工。
生产提前期=排队时间+准备时间+批量×单位加工时间+等待时间+传送时间
- 其中排队时间、等待时间和传送时间又称为保险期或缓冲期。





(7) 独立需求和相关需求

- 独立需求是MPS下达的，通过预测或用户订单得到；



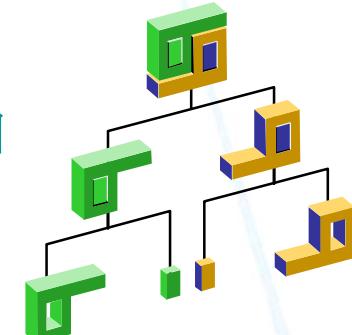
客户订单



预测

- 相关需求是通过物料需求计划运算后得到

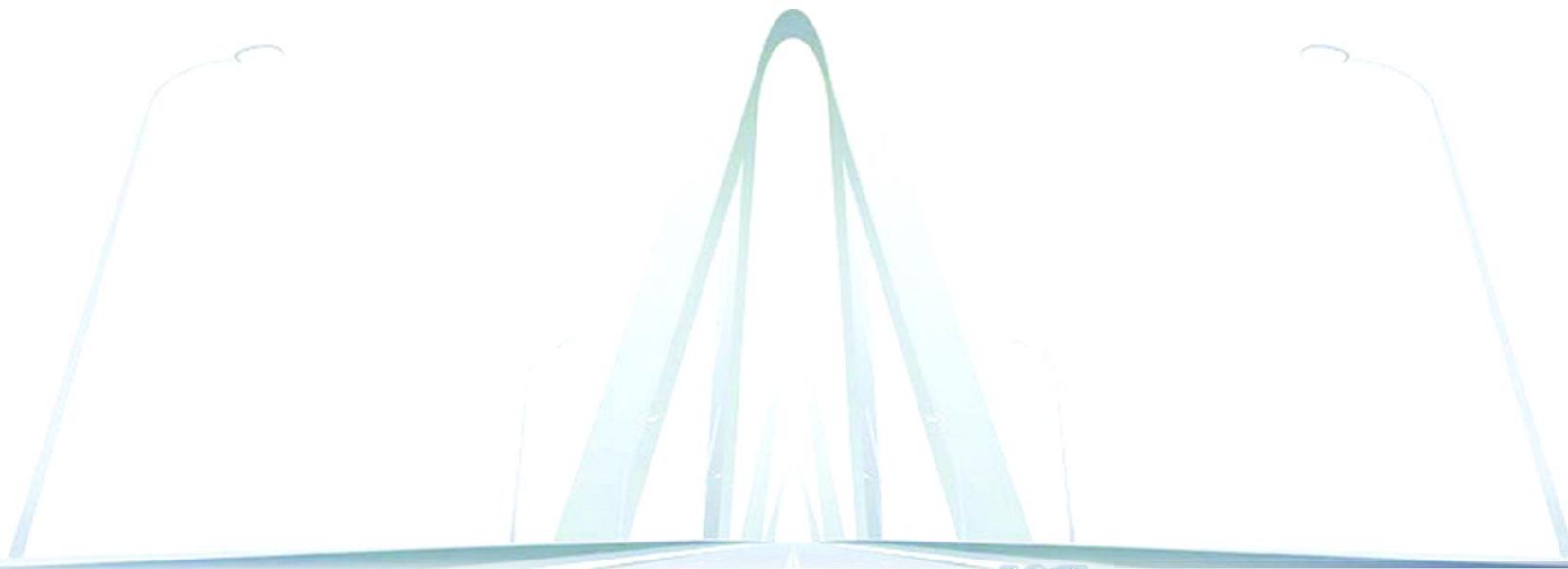
子装配和采购项目





上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

汽车轮毂制造实例



信息系统技术实验室 ist.sjtu.edu.cn

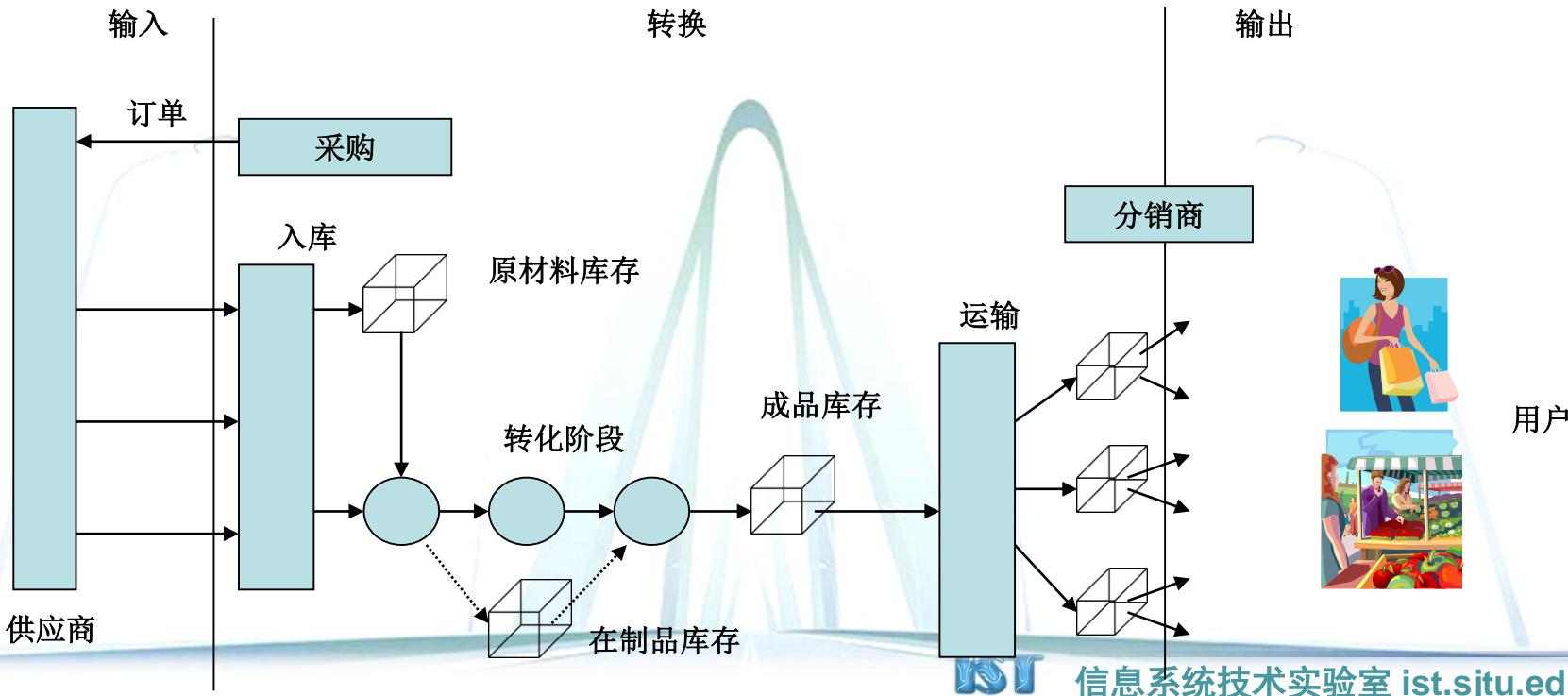


1. 企业资源管理基础
 2. 物料需求计划
 3. 库存管理
 4. 能力需求计划
 5. 供应链管理
 6. 小结
- A1 信息编码方法



2 库存管理

- 库存管理是企业生产过程的重要组成部分，是企业物料管理的核心，是指企业为了生产、销售等经营管理需要而对计划存储、流通的有关物品进行相应的管理，如对存储的物品进行接收、发放、存储保管等一系列的管理活动。
- 库存管理的对象是企业中的所有物料，包括原材料、零部件、在制品、半成品及产品，以及其他辅助物料。



库存管理是企业资源管理的核心问题，也是企业运营的两难问题：

① 好处：

- (1) 快速满足客户期望，缩短交货期，维持销售产品的稳定
- (2) 稳定生产需求，防止发生缺货，
- (3) 防止价格上涨

② 弊端：

- (1) 占用大量资金
- (2) 增加费用支出
- (3) 容易初放管理（计划不周、采购不力、生产不均衡）

相关方法主要有：

2.1 库存量控制策略：定量库存控制模型，定期库存控制模型

2.2 库存的ABC管理方法

2.3 库存定货点理论



独立需求：

● 定量库存控制模型

- 定量库存控制模型是当库存数量下降到某个库存值时，进行补充库存，以此来保证库存供应。这种库存控制方法称为定量库存控制模型。

● 定期库存控制模型

- 定期库存控制模型是按一定周期T检查库存，并进行库存补充，补充到规定的库存量S，这种控制方法称为定期库存控制模型。

相关需求

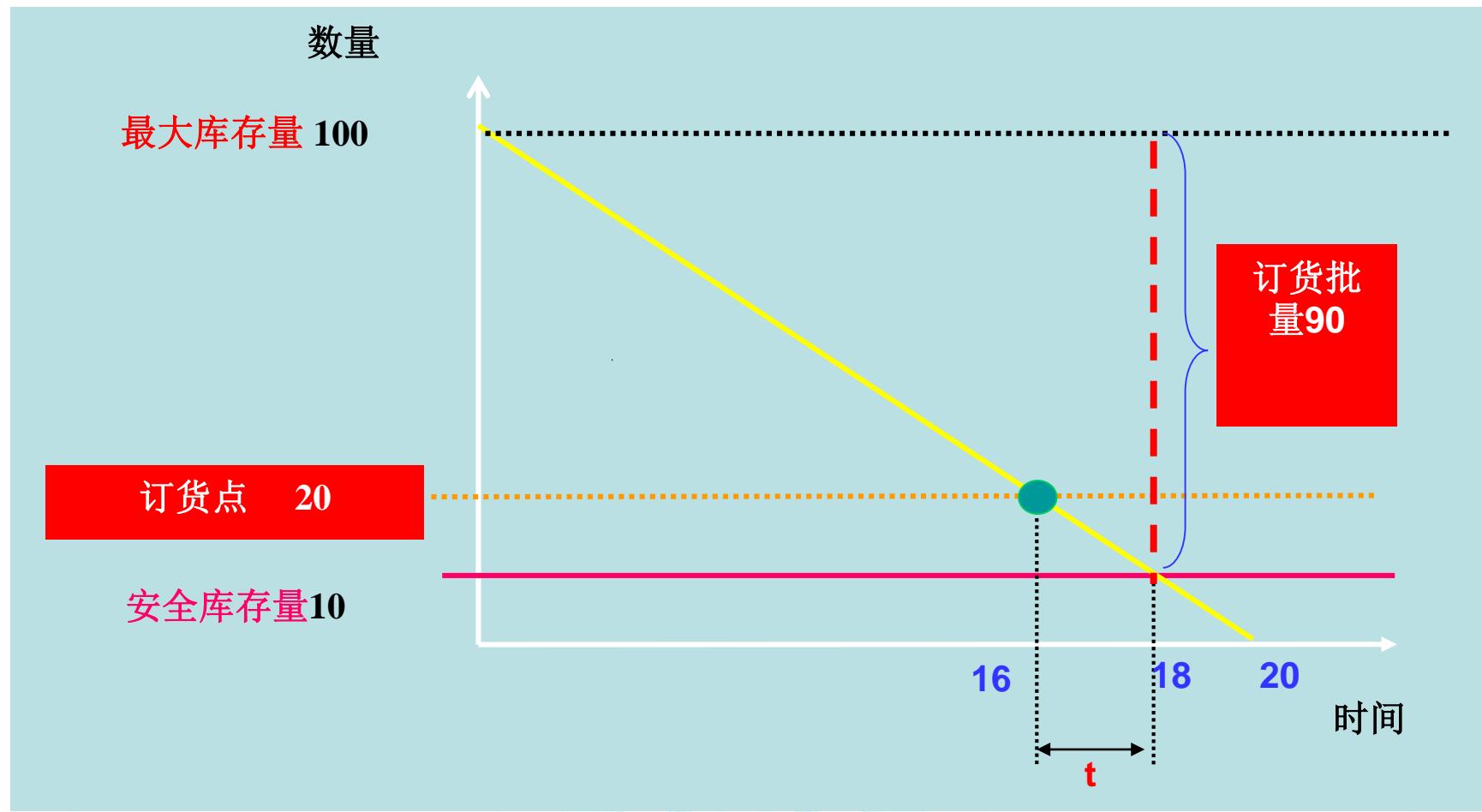
● MRP库存控制方法



- 假设你经营一小店，规模较小，你的仓库最多能放100箱方便面，即最大库存量为100箱。现仓库里已有100箱方便面；小店每天销量为5箱；安全库存量为10箱；由于受地理位置、交通及厂家供应商供货能力等因素的影响，订货之后需2天之后才能到货；
- 问：为了正常经营下去该怎么订货？
- 什么时候订货（R）？每次订多少（Q）？



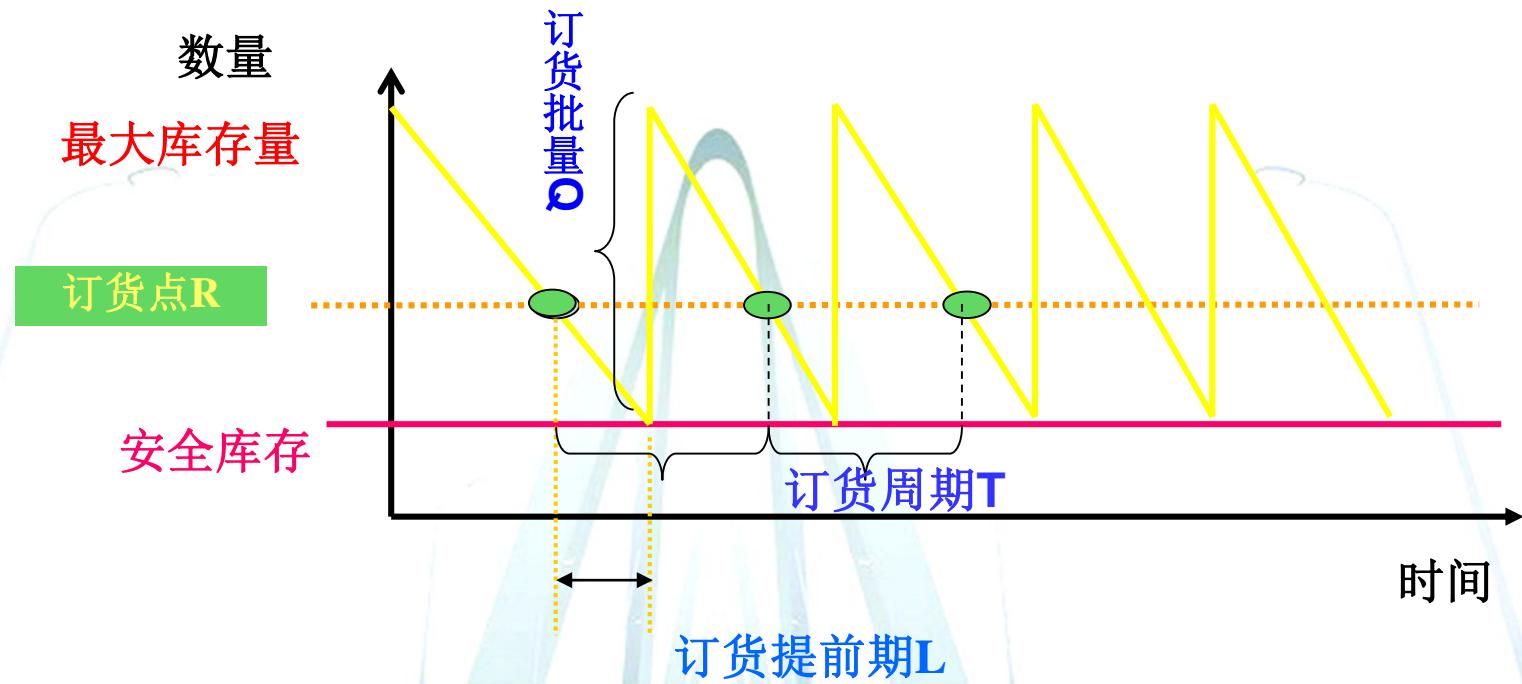
库存定货点理论





定货点理论缺陷：

- (1) 物料的消耗相对稳定;
- (2) 物料的供应比较稳定;
- (3) 物料的需求是独立的;
- (4) 物料的价格不是太高



固定订货周期法

- 按照固定的时间周期（1月或1周等）来订货，而订货数量则是变化的。一般都是事先依据对产品需求量的预测，确定一个比较恰当的最高库存额，在每个周期将要结束时，对存货进行盘点，决定订货量，货物到达后的库存量刚好到达原定的最高库存额

固定订货周期法使用范围：

- 1：消费金额高，需要实施严密管理的重要物品；
- 2：需要经常调整生产或采购数量的物品；
- 3：需求量变动幅度大且变动具有周期性，但可正确判断的物品；
- 4：建筑工程、出口等可以确定的物品；
- 5：设计变更风险大的物品；
- 6：多种商品采购可以节省费用的情况；
- 7：需要定期知道的物品；
- 8：同品种物品分批保管、同品种物品向多家供货商订购、批量订货分批入库等订货、保管、入库不规则物品等。

- 固定订购量系统的简化应用被称为分存控制法，有双堆法和三堆法等。双堆法是将订购点数量从库存量中分出，并单独存放或划以明显标志，当库存只剩下订购点一堆(或一箱)时即提出订购请求，每次订购固定数量的商品。三堆法是将安全库存再从订购点一堆中分出。
- 双堆法和三堆法简便易行，无需经常盘点，没有持续的库存记录，可以直观地识别订购点以便及时组织订购。这种方法比较适合于价值低、备运时间短、供应充足的商品。
- 固定订购量系统的优点：
 - 管理简便，订购时间和订购量不受人为判断的影响，保证库存管理的准确性；
 - 由于订购量一定，便于安排仓库内的作业活动，节约理货费用；
 - 便于按经济订购批量订购，节约库存总成本。

● 缺点

- 不便于对库存进行严格的管理；
- 订购之前的各项计划比较复杂。



(1) 定量库存控制模型

年总成本=年产品成本+年采购成本+年库存成本

$$T_C = D \times P + C \times D/Q + H \times Q/2 \quad (\text{条件 } R=L_T+A)$$

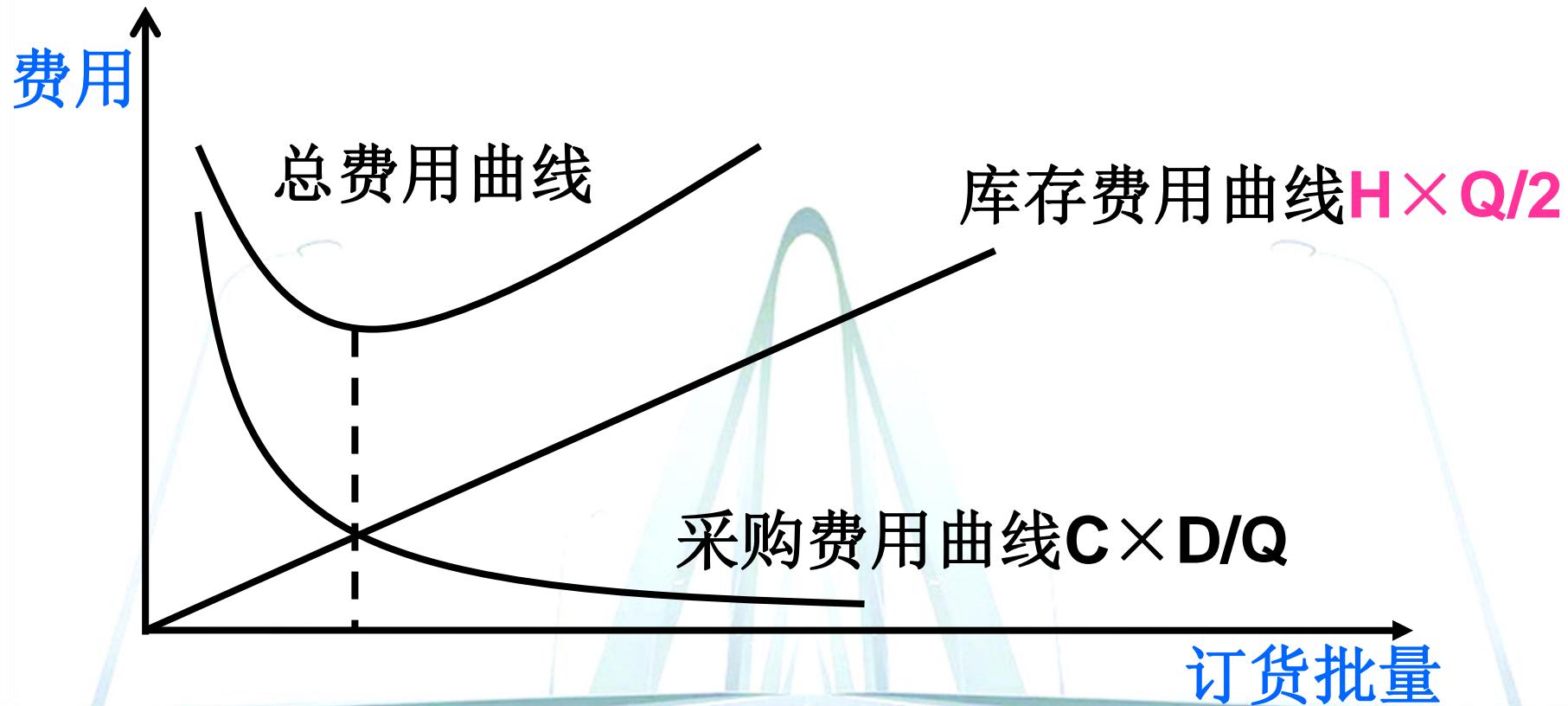
- ① T_C -----年总成本
- ② D -----库存物料的年需求率（件/年）
- ③ P -----单位产品成本（产品价格）
- ④ C -----单位订货费用（元/次）
- ⑤ Q -----订货批量
- ⑥ R -----订货点
- ⑦ H -----单位库存保管费用（元/件·年）
- ⑧ L_T -----订货周期内物料的消耗量
- ⑨ A -----安全库存量
- ⑩ L -----订货提前期
- ⑪ F -----单件库存保管费与单件库存购买费之比，即 $F=H/P$



$$T_C = D \times P + C \times D/Q + H \times Q/2$$

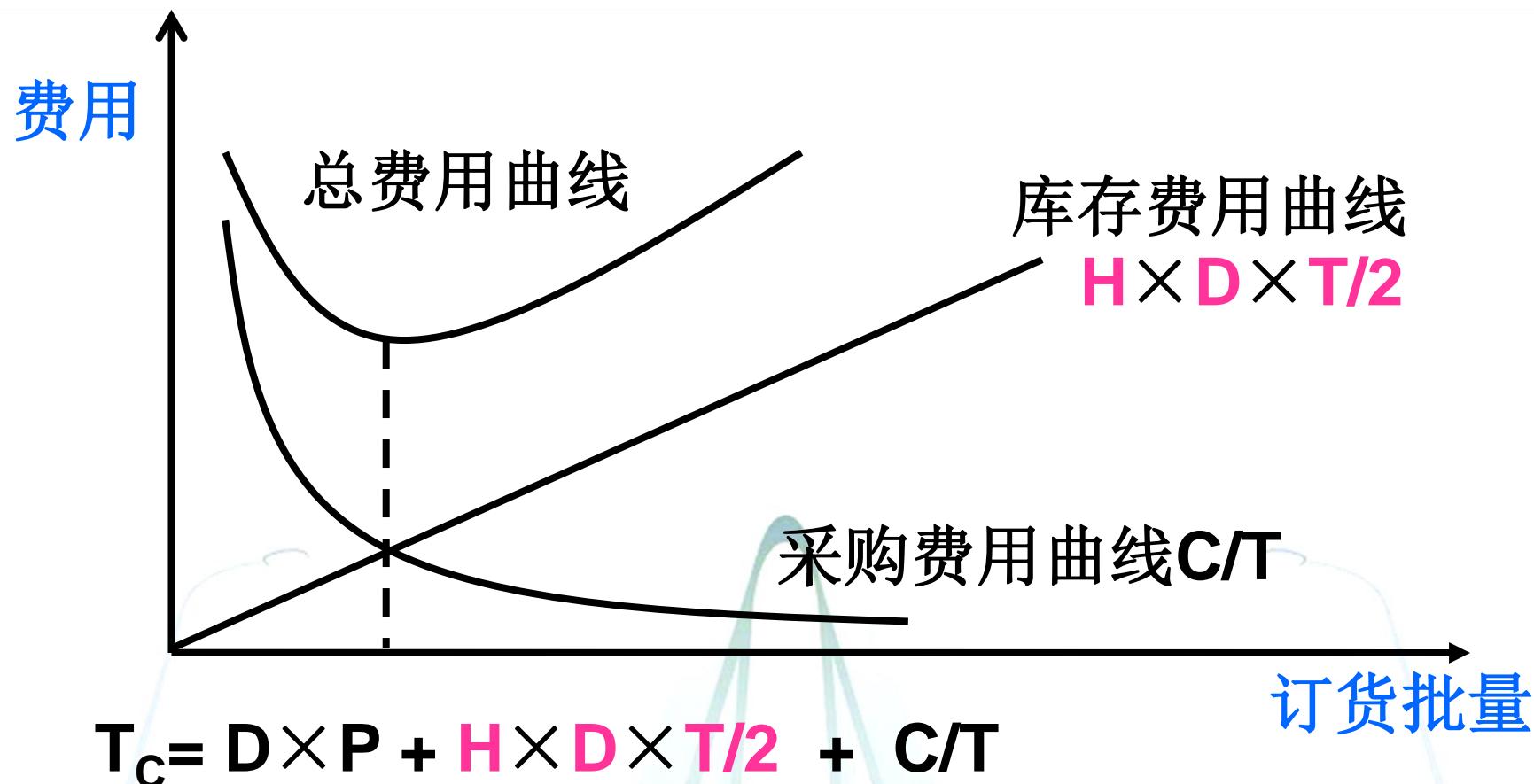
$$Q = 2 \times C \times D / H = 2 \times C \times D / (F \times P)$$

$$R = L_T + A \quad L_T \text{ ----订货周期内物料的消耗量}$$





(2) 定期库存控制模型



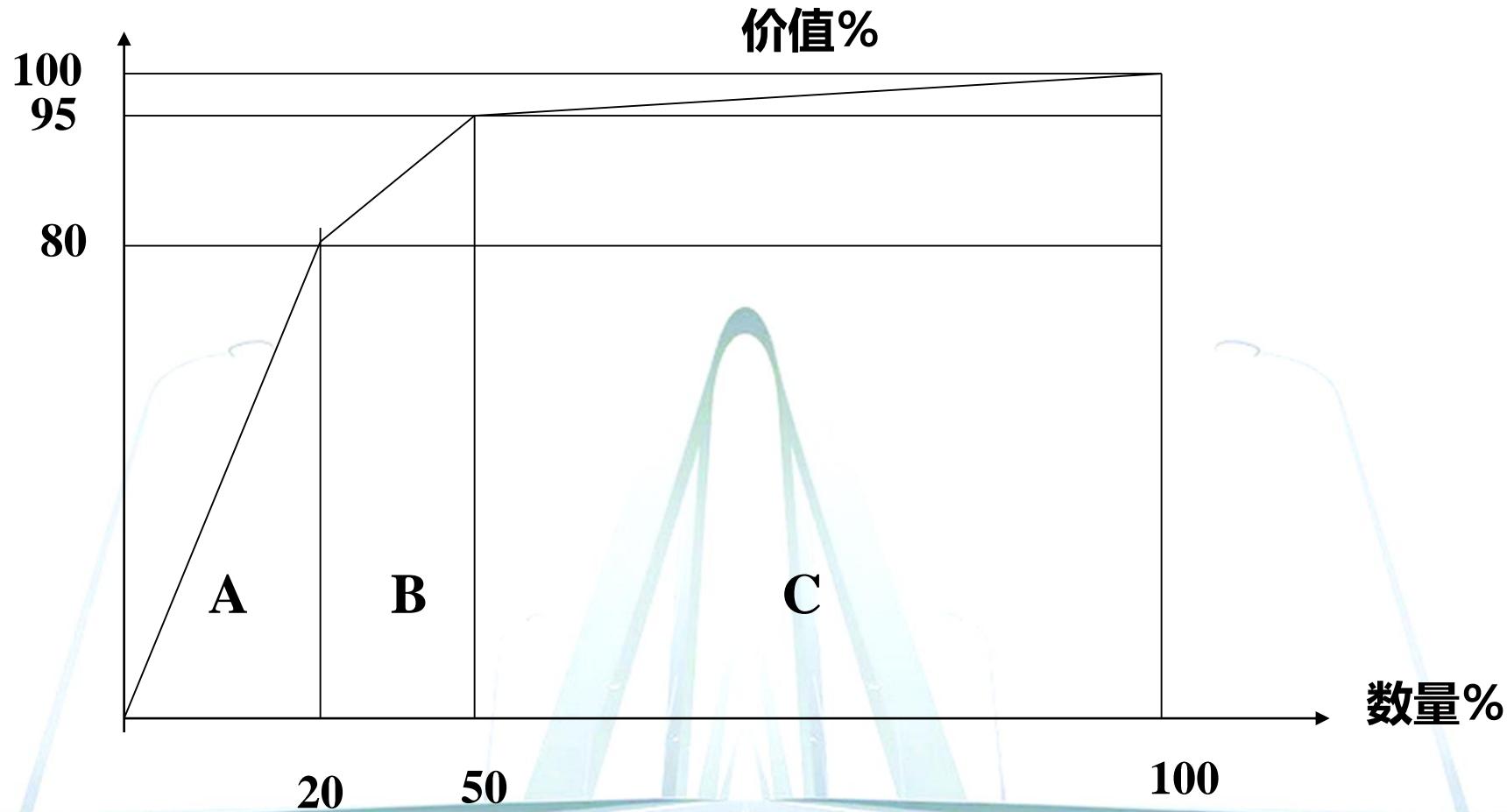
$$T = \sqrt{2 \times C / (D \times H)}$$

$$Q = (T+L) D / 365$$



2.2 库存的ABC管理方法

- 将不同产品按照重要性（主要是成本）不同，分别采用不同的管理方式。





ABC分类管理表

| 项目 分类 | A | B | C |
|----------|------------|----------------|---------------------|
| 价值 | 高 | 中 | 低 |
| 管理要点 | 将库存压缩到最低水平 | 库存控制有时可严些有时可松些 | 集中大量订货，以较高库存来节约订货费用 |
| 订货量 | 少 | 较多 | 多 |
| 订购量计算方法 | 按经济批量计算 | 按过去的记录 | 按经验估算 |
| 检查库存情况 | 经常检查 | 一般检查 | 季度或年度检查 |
| 进出统计 | 详细统计 | 一般统计 | 按金额统计 |
| 保险储备量 | 低 | 较大 | 允许较高 |
| 控制程度 | 严格控制 | 一般控制 | 控制总金额 |
| 控制系统 | 连续型库存观测系统 | 综合控制法或连续、定期法 | 定期型库存观测系统 |



库存ABC分析法的举例

| 序号 | 编码 | 年使用量 | 成本(元) | 年成本(元) |
|----|----|-------|-------|--------|
| 1 | | 200 | 8 | 1600 |
| 2 | | 1000 | 20 | 20000 |
| 3 | | 400 | 7 | 2800 |
| 4 | | 500 | 10 | 5000 |
| 5 | | 400 | 6 | 2400 |
| 6 | | 10000 | 5 | 50000 |
| 7 | | 100 | 20 | 2000 |
| 8 | | 100 | 90 | 9000 |
| 9 | | 600 | 2 | 1200 |
| 10 | | 10000 | 0.6 | 6000 |



分类结果

| 序号 | 年使用量 | 成本(元) | 年成本(元) | 占总成本% | 分类 |
|----|-------|-------|--------|--------|-----|
| 1 | 200 | 8 | 1600 | 1.60% | C |
| 2 | 1000 | 20 | 20000 | 20.00% | A |
| 3 | 400 | 7 | 2800 | 2.80% | C |
| 4 | 500 | 10 | 5000 | 5.00% | B |
| 5 | 400 | 6 | 2400 | 2.40% | C |
| 6 | 10000 | 5 | 50000 | 50.00% | A |
| 7 | 100 | 20 | 2000 | 2.00% | C |
| 8 | 100 | 90 | 9000 | 9.00% | B |
| 9 | 600 | 2 | 1200 | 1.20% | C |
| 10 | 10000 | 0.6 | 6000 | 6.00% | B&C |

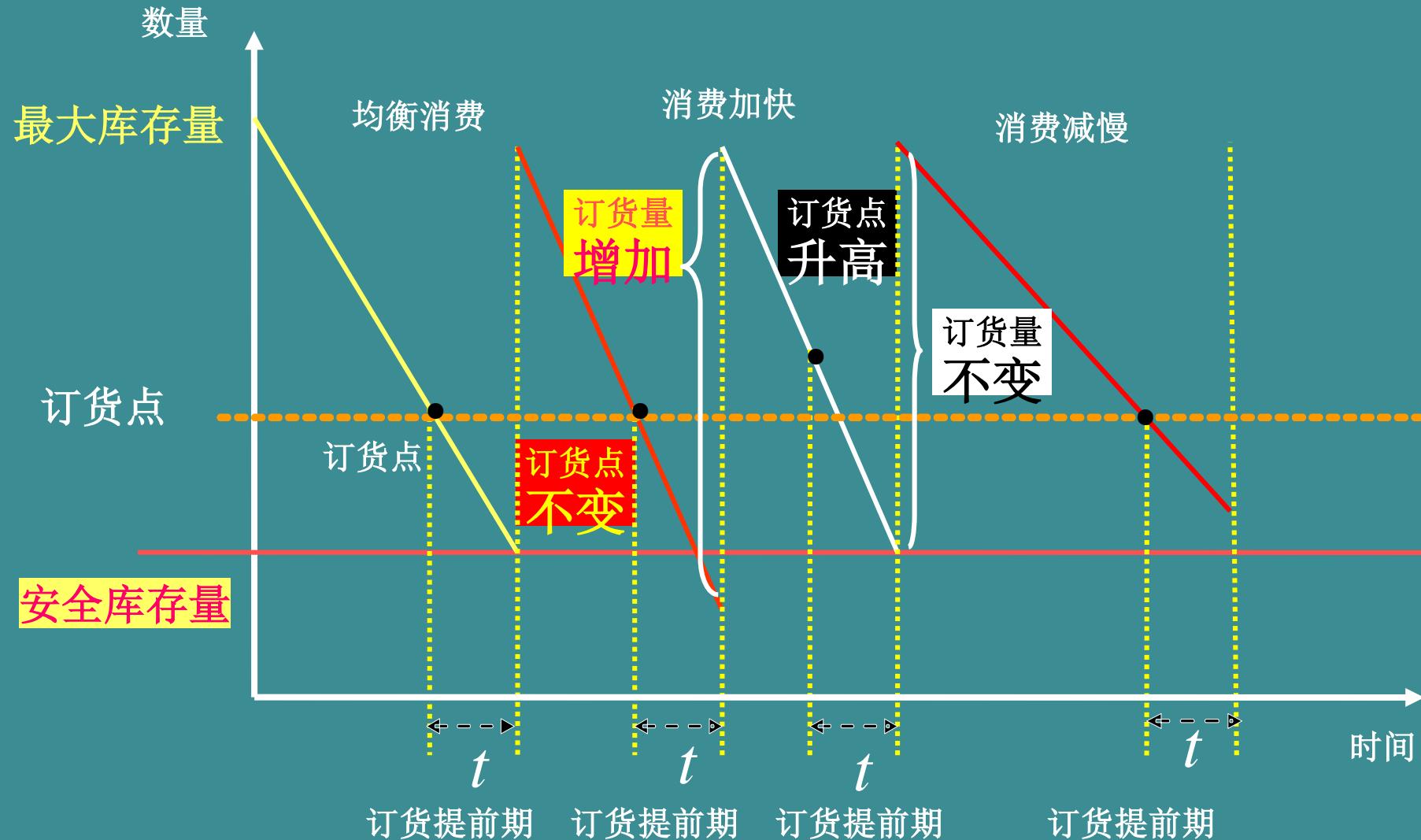


按年成本排序 (年总成本 = \$100,000)

| 项目 | Annual | | 年成本(\$) | 累计年成本 | 累计项目数 |
|----|---------|--------|---------|-------|-------|
| | 年成本(\$) | 百分比 | | | |
| A | 6 | 50,000 | 50.0% | 50.0% | 10% |
| | 2 | 20,000 | 20.0 | 70.0 | 20 |
| B | 8 | 9,000 | 9.0 | 79.0 | 30 |
| | 10 | 6,000 | 6.0 | 85.0 | 40 |
| | 4 | 5,000 | 5.0 | 90.0 | 50 |
| C | 3 | 2,800 | 2.8 | 92.8 | 60 |
| | 5 | 2,400 | 2.4 | 95.2 | 70 |
| | 7 | 2,000 | 2.0 | 97.2 | 80 |
| | 1 | 1,600 | 1.6 | 98.8 | 90 |
| | 9 | 1,200 | 1.2 | 100.0 | 100 |



库存定货点理论





控制库存的管理策略

平均库存值：全部库存物品的价值之和的平均值，表示某一段
时间内库存所占用的资金。

可供应时间：现有库存能够满足多长时间的需求。平均库存量/
单位时间的需求。

库存周转率：年销售额/年平均库存值

$$\text{库存周转率} = \frac{\text{年销售额}}{\text{年平均库存值}}$$

$$\text{成品库存周转率} = \frac{\text{年销售额}}{\text{成品平均库存值}}$$

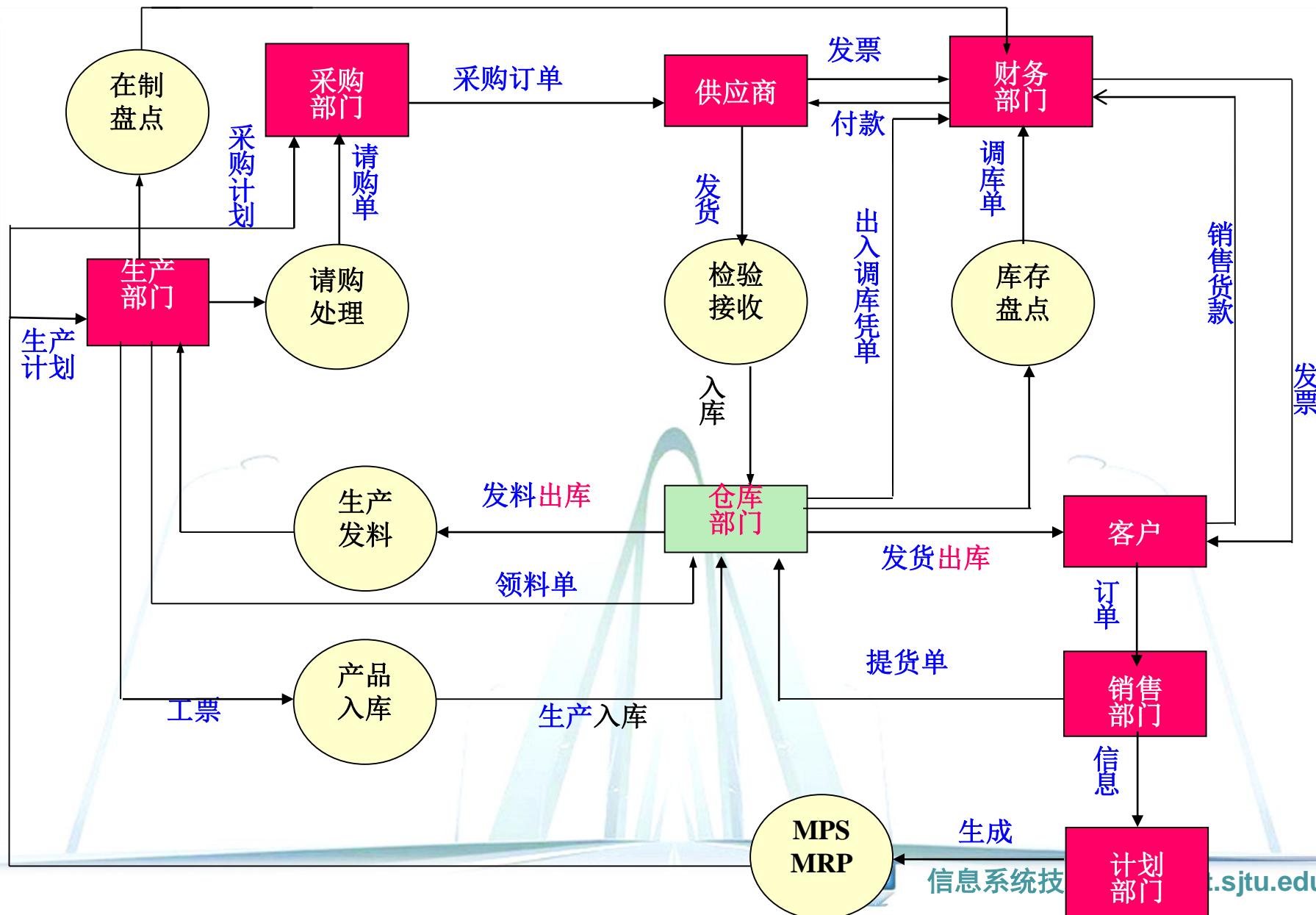
$$\text{在制品库存周转率} = \frac{\text{生产产值}}{\text{在制品平均库存值}}$$

$$\text{原材料库存周转率} = \frac{\text{原材料消耗额}}{\text{原材料平均库存值}}$$



库存管理业务流程分析

调整单据

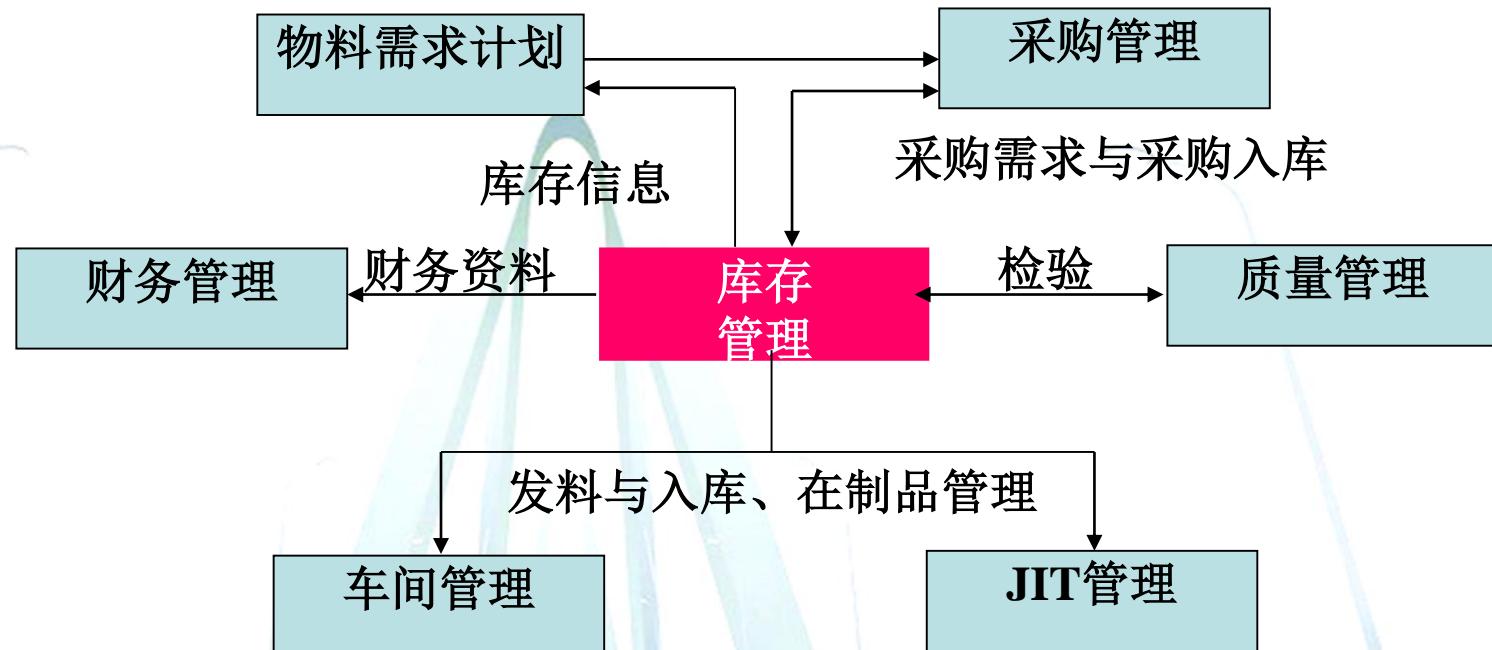




库存管理功能分析

- 物料的入库
- 物料的出库
- 物料的移动管理
- 库存盘点
- 库存物料管理信息分析

库存管理与其他系统关系

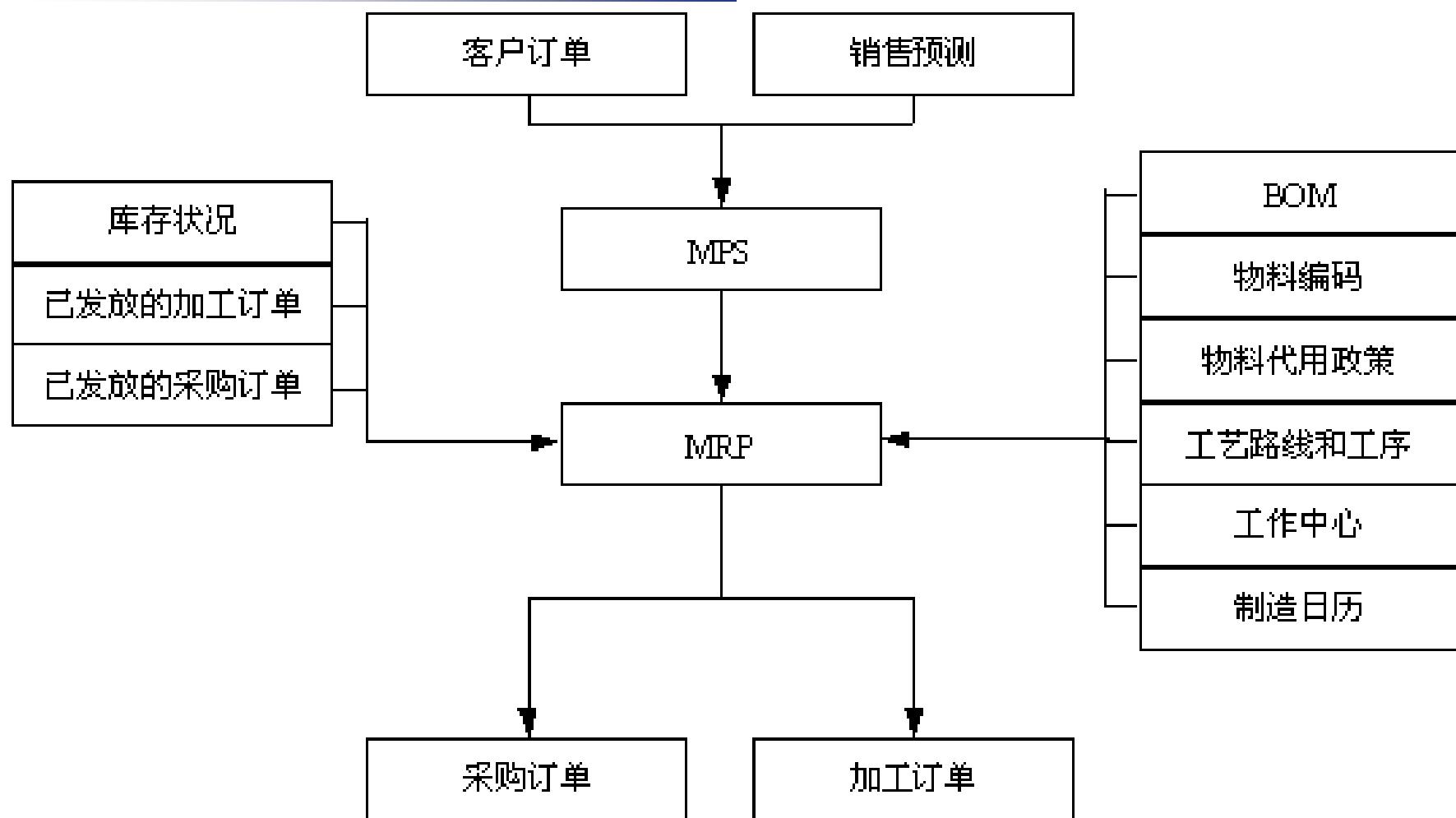




1. 企业资源管理基础
 2. 库存管理
 3. 物料需求计划
 4. 能力需求计划
 5. 供应链管理
 6. 小结
- A1 信息编码方法

3 物料需求计划

- 物料需求计划（Material Requirement Planning，简称为 MRP）是对主生产计划的各个项目所需的全部**制造件**和全部**采购件**的网络支持计划和时间进度计划。
- 物料需求计划主要解决：
 - 要生产什么？生产多少？（来源于MPS）
 - 要用到什么？（根据BOM展开）
 - 已经有了什么？（根据物品库存信息、即将到货或产出信息）
 - 还缺什么？（计算出结果）
 - 何时安排？（计算出结果）



- 将制造什么？用什么东西来制造？具备些什么？还需要些什么？



⑤ MRP处理的问题与所需信息

| 处理的问题 | 需用信息 |
|------------------|---------------------|
| 1. 生产什么？数量多少？时间？ | 1. 现实、有效、可信的MPS |
| 2. 要用到什么？ | 2. 准确的BOM，及时的设计更改通知 |
| 3. 已有什么？ | 3. 准确的库存信息 |
| 已定货量？到货时间？ | 下达订单跟踪信息 |
| 已分配量？ | 配套预料单、提货单 |
| 4. 还缺什么？ | 4. 批量规则、安全库存、成品率 |
| 5. 下达订单的开始日期？ | 5. 提前期 |

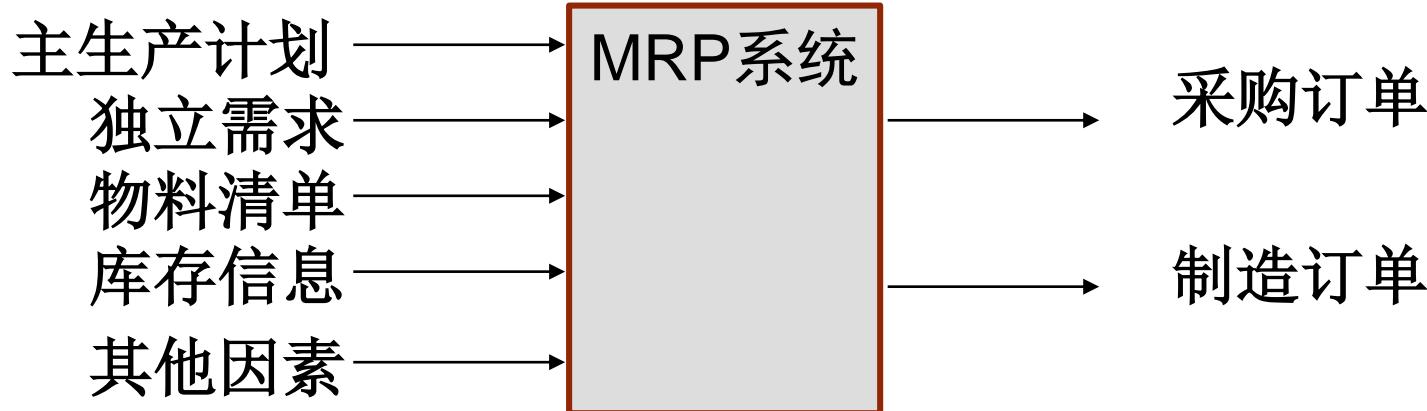


MRP中的基本数量概念

- 在MRP运算中，经常用到的基本数量概念包括描述库存信息的数量概念和描述需求信息的数量概念。
- 描述库存信息的数量概念包括现有库存量、计划收到量、已分配量、安全库存量、可用库存量和预计库存量等。
- 描述需求信息的数量概念包括总需求量、毛需求量、净需求量、计划产出量和计划投入量等。
- 逐层计算原则是指MRP在计算物料需求时，应该采用自顶向下按照产品结构层次逐层计算物料需求量的方式。
- 逐层计算原则是MRP工作原理的重要组成部分，它揭示了MRP计算物料需求量的基本过程形式，是理解MRP工作原理的基础。

逐层计算原则

逐层MRP的输入、处理和输出

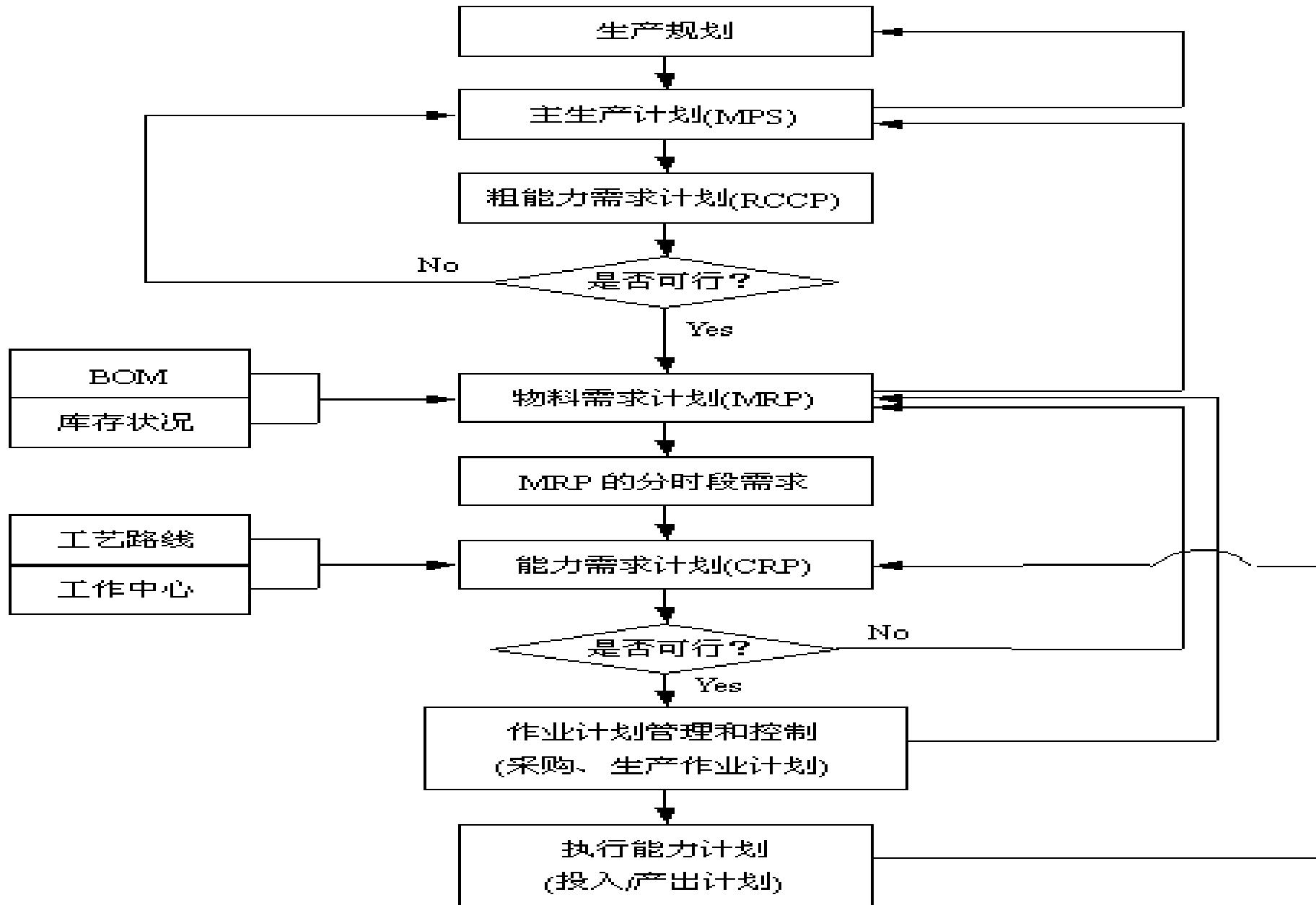


逐层MRP的输入、处理和输出：

- MRP最主要的输入数据是3个，即MPS、BOM和库存状况。
- MRP的处理过程主要包括读取MPS数据、分解BOM、计算物料毛需求、计算物料净需求和下达作业计划。
- MRP的输出主要是可以用于管理和控制的各种计划和报告。



MRP工作原理





低层码

- 物料的低层码是系统分配给物料清单上的每个物品一个从0至N的数码。产品结构中最上层的层级码为0，下一层部件层级码则为1，依此类推。
- 一个物品只能有一个MRP低层码，当一个物品在多个产品中所处的产品结构层次不同或即使处于同一产品结构中的但却处于不同产品结构层次时，则取处在最低层的层级码作为该物品的低层码，也既取数字最大的层级码。
- 低层码作用
 - 低层码是MRP的计算顺序。
 - 在展开MPS进行物料需求计算时，计算的顺序按低层码的顺序从低层码数字小的物料往低层码数字大的顺序进行计算
 - 当计算到该产品的最低层次时，再进行MRP需求计算与原材料（或构成的组件）的库存分配，保证可用的库存量优先分配给了处于最低层的物料，时间上最先需求的物料先得到库存分配，避免了需求较晚的物品提前下达计划，并占用库存

MRP的计算过程

● MRP中每个项目的计算步骤

计算毛需求量

项目毛需求量=项目独立需求+父项的相关需求

父项的相关需求=父项的计划订单数量*项目用量因子

计算净需求量

预计库存量初值=前期库存+计划接收量-毛需求量-已分配量

净需求量=预计库存初值的相反数+安全库存

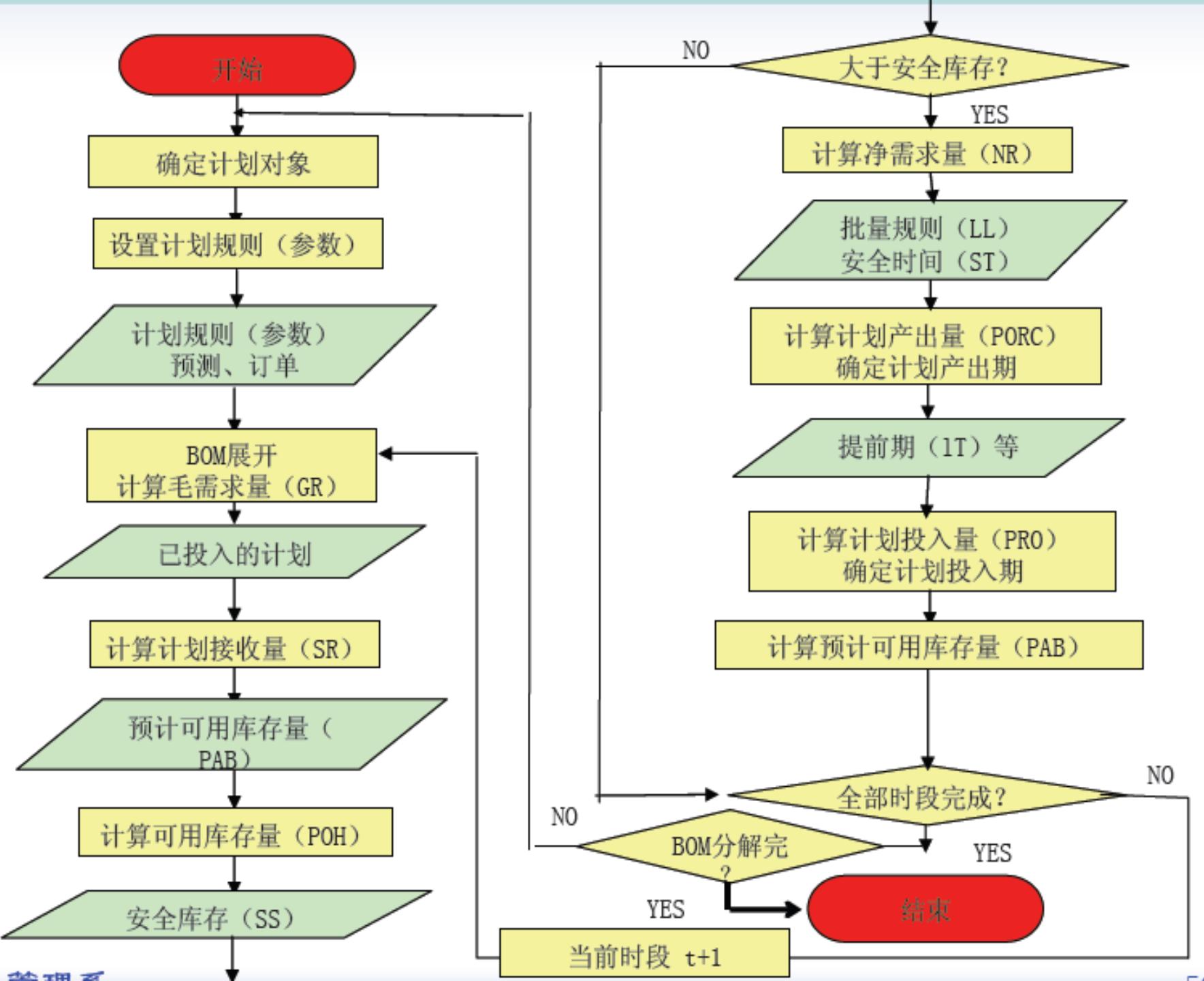
净需求量=毛需求量-前期库存-计划接收量+已分配量+安全库存

生成订单计划和下达订单计划

计划更低一层相关项目的毛需求量，进入下一循环

MRP的编制方法

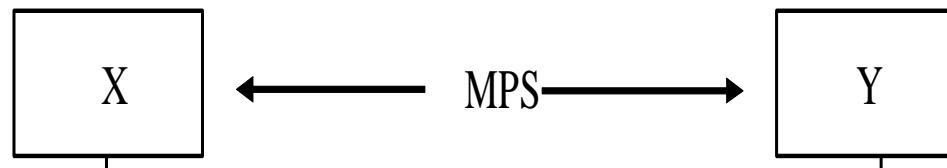
- 第1步推算物料毛需求。考虑相关需求和低层码推算计划期全部的毛需求。
$$\text{毛需求量} = \text{相关需求量} + \text{独立需求量}$$
- 第2步计算当期可用库存量。考虑已分配量计算计划初始时刻当期预计库存。
$$\text{当期预计可用库存量} = \text{现有库存量} - \text{已分配量}$$
- 第3步推算PAB初值。考虑毛需求推算特定时段的预计库存量。
$$\text{PAB初值} = \text{上期末预计可用库存量} + \text{计划接收量} - \text{毛需求量}$$
- 第4步推算净需求。考虑安全库存推算特定时段的净需求。
$$\begin{aligned} &\text{当PAB初值} \geq \text{安全库存}, \text{ 净需求} = 0 \\ &\text{当PAB初值} < \text{安全库存}, \text{ 净需求} = \text{安全库存} - \text{PAB初值} \end{aligned}$$
- 第5步推算计划产出量。考虑批量推算特定时段的计划产出量。
$$\begin{aligned} &\text{当净需求} > 0, \text{ 计划产出量} = N \times \text{批量} \\ &\text{满足: 计划产出量} \geq \text{净需求} > (N-1) \times \text{批量} \end{aligned}$$
- 第6步推算预计可用库存量。推算特定时段的预计库存量。
$$\text{预计可用库存量} = \text{计划产出量} + \text{PAB初值}$$
- 第7步递增一个时段，分别重复进行第三到第六步，循环计算至计划期终止。
- 第8步推算计划投入量。考虑提前期推算计划期全部的计划投入量



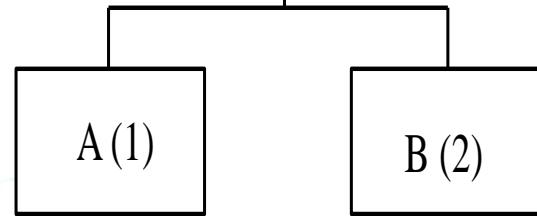


零件在不同部件中的计算示例

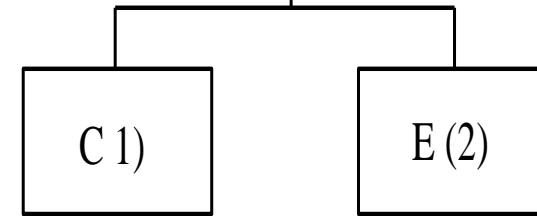
0层



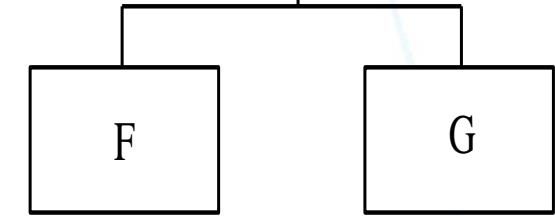
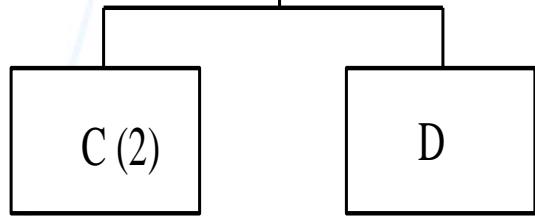
1层



MRP



2层





- 推算Y对C、E的毛需求
- Y的提前期为2周

| 提 前 期 | 低 层 码 | 物 料 | 时段 | 当 期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------------|-------------|--------|----------|--------|----|----|---|---|----|----|----|---|
| | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 2 | | Y | MPS计划产出量 | | | 10 | | | 20 | | 15 | 5 |
| | | | MPS计划投入量 | | 10 | 20 | | | 15 | 5 | | |
| 2 | 1 | C | 毛需求 | | 10 | 20 | | | 15 | 5 | | |
| 3 | 1 | E | 毛需求 | | 20 | 40 | | | 30 | 10 | | |



- 推算X对A、B、C的毛需求
- X的提前期为1周

| 提 前 期 | 低 层 码 | 物 料 | 时段 | 当 期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------------|-------------|--------|----------|--------|----|---|----|----|----|----|----|----|
| 1 | | X | MPS计划产出量 | | | 7 | | 10 | | 15 | | 12 |
| | | | MPS计划投入量 | | 7 | | 10 | | 15 | | 12 | |
| 1 | 1 | B | 毛需求 | | 14 | | 20 | | 30 | | 24 | |
| 2 | 1 | A | 毛需求 | | 7 | | 10 | | 15 | | 12 | |
| 2 | 2 | C | 毛需求 | | 14 | | 20 | | 30 | | 24 | |



● 推算X、Y对C的毛需求

| 提 前 期 | 低 层 码 | 物 料 | 时段 | 当 期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-------------|-------------|----------|--------|--------|----|----|----|----|----|----|----|---|
| | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 1 | X | MPS计划产出量 | | | 7 | | 10 | | 15 | | 12 | |
| | | | | | 7 | | 10 | | 15 | | 12 | |
| 2 | Y | MPS计划产出量 | | | | 10 | | 20 | | 15 | 5 | |
| | | | | | 10 | | 20 | | 15 | 5 | | |
| 2 | 2 | C | X对C毛需求 | | 14 | | 20 | | 30 | | 24 | |
| 2 | 1 | C | Y对C毛需求 | | 10 | | 20 | | 15 | 5 | | |
| 2 | 2 | C | C总毛需求 | | 24 | | 40 | | 45 | 5 | 24 | |



推算C的PAB初值、净需求、计划产出量计划投入量和预计库存量

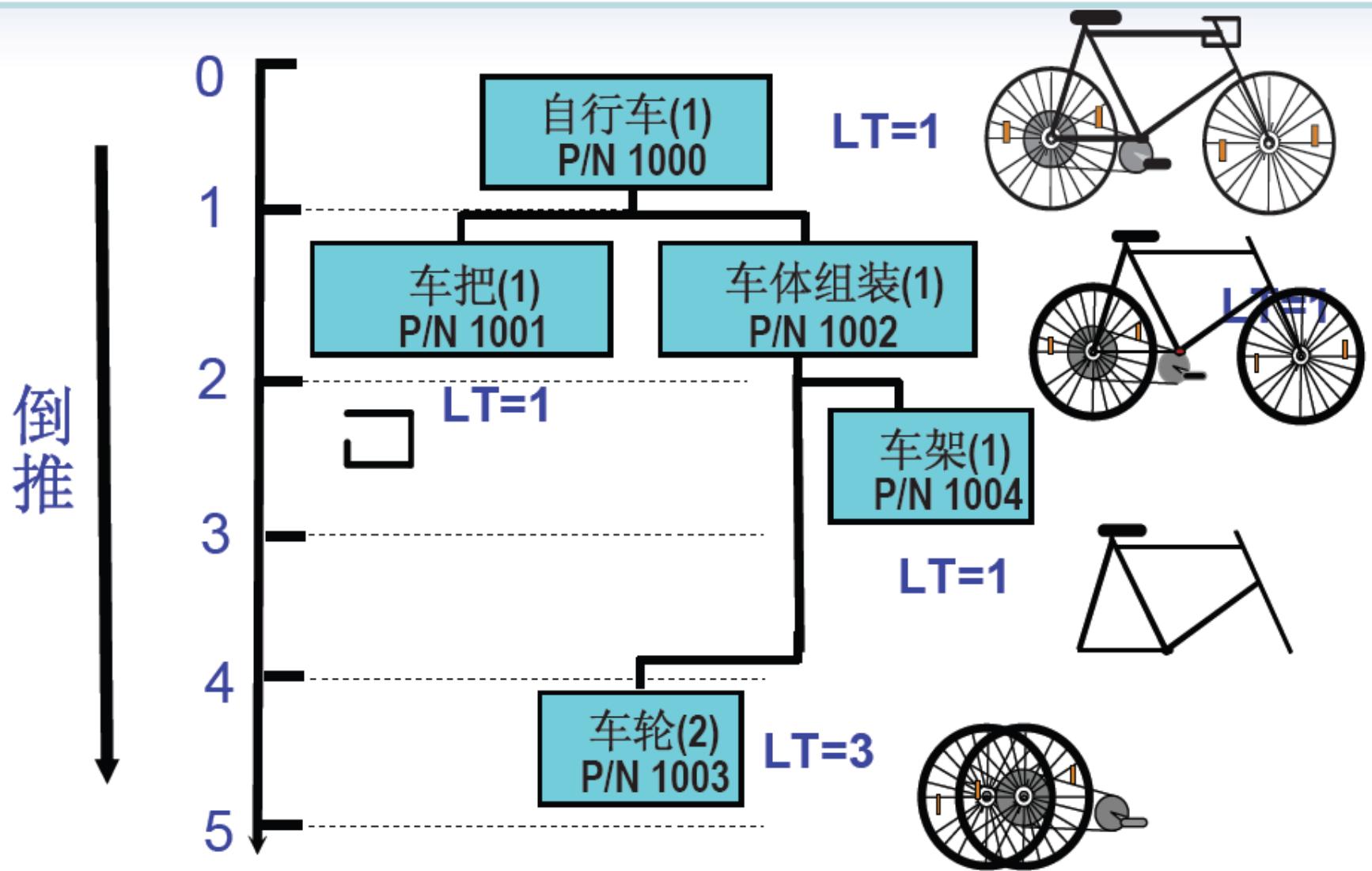
| 批量 | 提 前 期 | 现 有 量 | 分 配 量 | 安 全 库 存 | 低 层 码 | 物 料 | 时段 | 当 期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
|----|-------------|-------------|-------------|------------------|-------------|--------|------------|--------|----|----|----|----|-----|----|-----|----|---|
| | | | | | | | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | |
| 1 | 1 | | | | | X | MPS计划产出量 | | 7 | | 10 | | 15 | | 12 | | |
| | | | | | | | MPS计划投入量 | | 7 | 10 | | 15 | | 12 | | | |
| 1 | 2 | | | | | Y | MPS计划产出量 | | | 10 | | 20 | | 20 | | 15 | 5 |
| | | | | | | | MPS计划投入量 | | 10 | 20 | | 15 | 5 | | | | |
| 20 | | 40 | 5 | 10 | 2 | C | 毛需求 | | 24 | | 40 | | 45 | 5 | 24 | | |
| | | | | | | | 计划接收量 | | 50 | | | | | | | | |
| | | | | | | | (PAB) 初值 | 35 | 61 | 61 | 21 | 21 | -24 | 11 | -13 | | |
| | | | | | | | 预计可用库存量 | 35 | 61 | 61 | 21 | 21 | 16 | 11 | 27 | | |
| | | | | | | | 净需求 | | | | | | 34 | 0 | 23 | | |
| | | | | | | | 计划产出量 | | | | | | 40 | | 40 | | |
| | 2 | | | | | | 计划投入量 | | | | | 40 | 40 | | | | |



- 同样可以推出A、B、E 的PAB初值、净需求、计划产出量计划投入量和预计库存量等。做出A、B、E物料需求（MRP）的报表.



例子 自行车产品结构





- 根据上述**MPS**及相关情况，试编制**MRP**计划方案
- 注： **POR**为计划投入量

项目号: P/N 1000 批量规则: L4L
项目名: 自行车 安全库存: 50
提前期: 1周

| 类别 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----|---|---|---|---|-----|-----|-----|---|-----|---|----|
| POR | | | | | 200 | 200 | 200 | | 400 | | |



物料需求（车把）

项目号: P/N 1001

批量规则: L4L

安全库存: 0

项目名: 车把

提前期: 1周

| 类别 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------------|-----|-----|-----|-----|----------|----------|----------|-----|----------|---|----|
| 毛需求量 GR | | | | | 200 | 200 | 200 | | 400 | | |
| 计划接收量 SR | | | | | | | | | | | |
| 计划在库量 POH | 100 | 100 | 100 | 100 | - 100 | - 200 | - 200 | | - 400 | | |
| 预计可用库存 PAB | 100 | 100 | 100 | 100 | | | | | | | |
| 净需求量 NR | | | | | 100 | 200 | 200 | | 400 | | |
| 计划产出量PORC | | | | | 100 | 200 | 200 | | 400 | | |
| 计划投入量 POR | | | | 100 | 200 | 200 | | 400 | | | |



物料需求（车架）

项目号: P/N 1004
项目名: 车架
50

批量规则: F0Q=100 倍增
提前期: 1周 安全库存:

| 类别 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------------|-----|-----|-----|-----|----------|----------|-----|----------|-----|-----|-----|
| 毛需求量 GR | | | | 150 | 200 | 200 | | 400 | | | |
| 计划接收量 SR | | | | | | | | | | | |
| 计划在库量 POH | 150 | 150 | 150 | | - 100 | - 100 | 100 | - 300 | 100 | 100 | 100 |
| 预计可用库存 PAB | 150 | 150 | 150 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 净需求量 NR | | | | 50 | 150 | 150 | | 350 | | | |
| 计划产出量PORC | | | | 100 | 200 | 200 | | 400 | | | |
| 计划投入量 POR | | | 100 | 200 | 200 | | 400 | | | | |



物料需求（车轮）

项目号: P/N 1003
项目名: 车轮

批量规则: FOQ=150 倍增
提前期: 2周 安全库存: 50

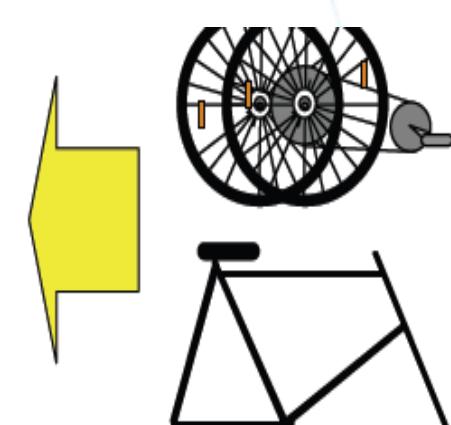
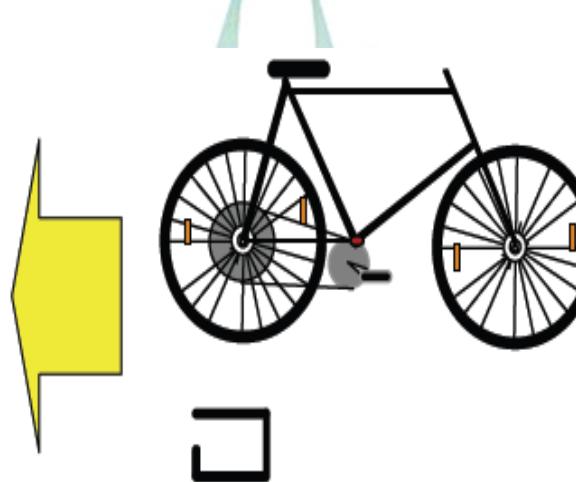
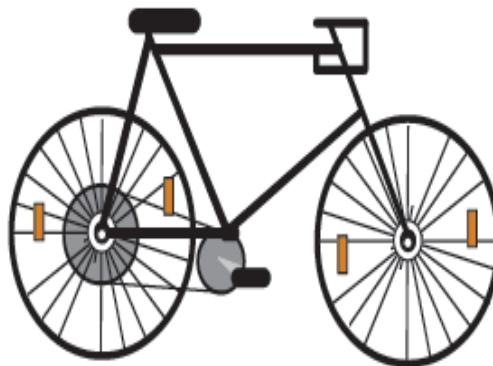
| 类别 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 毛需求量 GR | | | | 300 | 400 | 400 | | 800 | | | |
| 计划接收量 SR | | | | | | | | | | | |
| 计划在库量 POH | 200 | 200 | 200 | - | - | - | 150 | - | 100 | 100 | 100 |
| 预计可用库存 PAB | 200 | 200 | 200 | 100 | 350 | 300 | 150 | - | 100 | 100 | 100 |
| 净需求量 NR | | | | 150 | 400 | 350 | | 700 | | | |
| 计划产出量PORC | | | | 150 | 450 | 450 | | 750 | | | |
| 计划投入量 POR | | 150 | 450 | 450 | | 750 | | | | | |

注意安全库存



物料需求POR计算结果

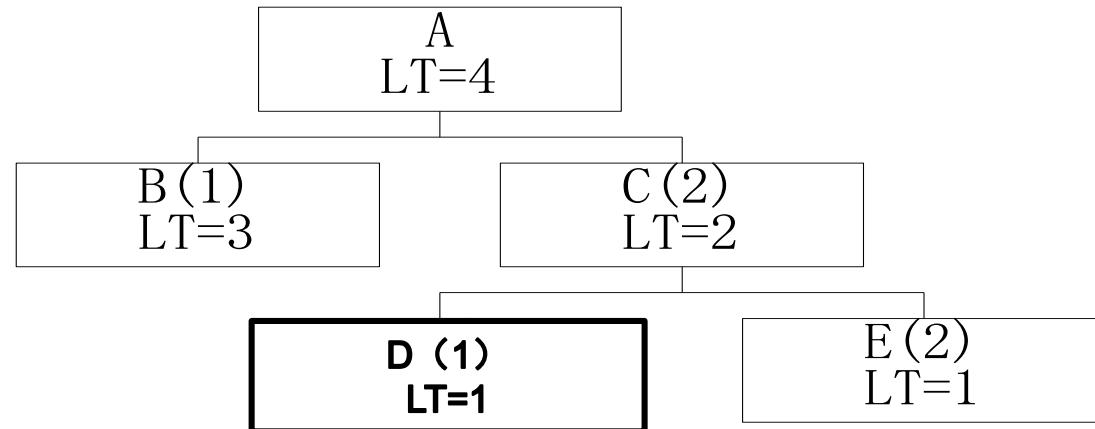
| 类别 | 时段 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|----|
| P/N 1001 | 自行车 | | | | 200 | 200 | 200 | | 400 | | |
| P/N 1001 | 车把 | | | 100 | 200 | 200 | | 400 | | | |
| P/N 1002 | 车体 | | | 150 | 200 | 200 | | 400 | | | |
| P/N 1003 | 车轮 | 150 | 450 | 450 | | 750 | | | | | |
| P/N 1004 | 车架 | | 100 | 200 | 200 | | 450 | | | | |





例1 求毛需求及发出订货计划

已知MPS为在第8个计划周期时产出100件A产品，其中A产品的BOM图见下图。



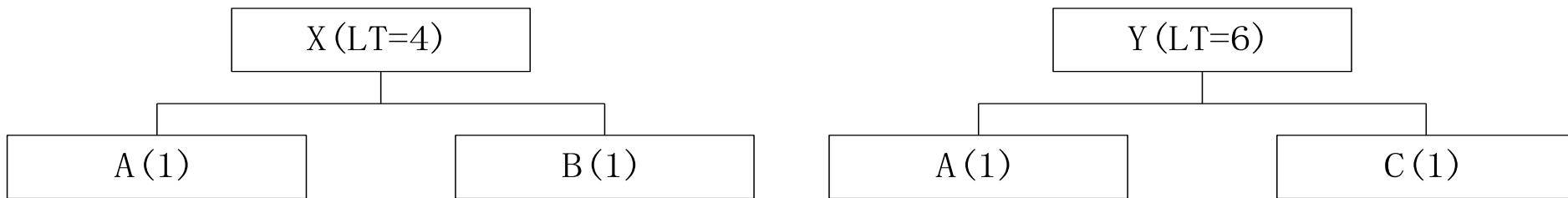
| 提前期 | | 计划时段 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----|---|--------|-----|-----|---|-----|---|---|---|-----|
| 4 | A | 毛需求 | | | | | | | | 100 |
| | | 发出订货计划 | | | | 100 | | | | |
| 3 | B | 毛需求 | | | | 100 | | | | |
| | | 发出订货计划 | 100 | | | | | | | |
| 2 | C | 毛需求 | | | | 200 | | | | |
| | | 发出订货计划 | 200 | | | | | | | |
| 1 | D | 毛需求 | | 200 | | | | | | |
| | | 发出订货计划 | 200 | | | | | | | |
| 1 | E | 毛需求 | | 400 | | | | | | |
| | | 发出订货计划 | 400 | | | | | | | |

注意：上一层物料“发出订货计划”的时间即为下一层物料的”毛需求“时间



例2 物料需求的计算 (当独立需求与相关需求同时存在时)

- 已知：物料A既是产品X的组件又是产品Y的组件，BOM图见下图。所以，A的需求为相关需求；此外，A作为配件又有独立需求。因此，物料A的总的毛需求应为其独立需求和相关需求之和。其算例见下表。



产品X、Y的BOM图（物料X的需求关系）

物料A的毛需求计算表

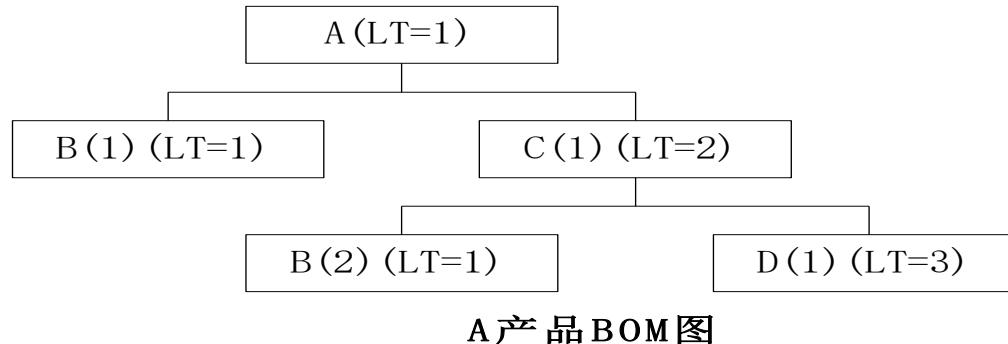
| 周 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| X (1t=4) | | | | | | 25 | | 30 | | | 15 | | |
| Y (LT=6) | | | | | | | | | 40 | | 15 | | 30 |
| 相关需求X-A | | 25 | | 30 | | | 15 | | | | | | |
| 相关需求Y-A | | | 40 | | 15 | | 30 | | | | | | |
| 独立需求A | 15 | 15 | | | | | | | | | | | |
| A的毛需求 | 15 | 40 | 40 | 30 | 15 | | 45 | | | | | | |



例3 物料净需求的计算

(当同一零件分布在同一产品的不同层次上时，即低位码的应用)

- 已知MPS为在第8个计划周期时产出100件A产品，各物料的计划接收量和已分配量均为零，求物料B的净需求。其中A产品的BOM图见下图。



用低位码计算零件B的净需求表

| 提前期 | 物料名 | 现库存 | 需求 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|-----|-----|-----|---|---|-----|---|-----|-----|-----|-----|
| 1 | A | 50 | 毛需求 | | | | | | | | 100 |
| | | | 净需求 | | | | | | | | 50 |
| 3 | C | 60 | 毛需求 | | | | | | 100 | | |
| | | | 净需求 | | | | | | 40 | | |
| 2 | D | 70 | 毛需求 | | | 100 | | | | | |
| | | | 净需求 | | | 30 | | | | | |
| 用低位码运算 | B | 120 | 毛需求 | | | | | 200 | | 100 | |
| | | | 净需求 | | | | | 80 | | 100 | |

注意：(1) 低位码的定义；(2) 如不按低位码运算，则结果是：第7周的净需求为0，第5周的净需求为180



例4 MRP计划的编制

- 已知产品A的BOM见图1，MRP的四个输入分别见表1～表4，请编制项目B、C的物料需求计划（或为项目B、C编制MRP）（虚框部分为答案）

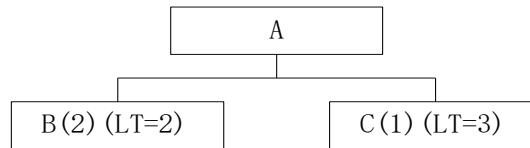


图1 产品A的BOM图

表1 项目A的主生产计划

| 周期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----|----|----|----|----|----|----|----|----|
| 项目A | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |

表2 项目C的独立需求计划

| 周期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|-----|---|---|---|---|---|---|---|---|
| 项目C | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |

表3 物料清单

| 项目 | 层 | 用量 |
|----|---|----|
| A | 0 | |
| B | 1 | 2 |
| C | 1 | 1 |

表4 库存信息

| 项目 | 计划收到 (周) | | | | | | | | 现有库存 | 已分配量 | 提前期 | 订货批量 |
|----|----------|---|----|----|---|---|---|---|------|------|-----|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | | | |
| B | | | | 40 | | | | | 65 | 0 | 2 | 40 |
| C | | | 30 | | | | | | 30 | 0 | 3 | 30 |

项目B的物料需求 (需求量和时间)

| 周期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|----|----|----|----|----|----|----|----|
| 毛需求 | 20 | 20 | 20 | 20 | 20 | 20 | 20 | 20 |
| 计划收到 | | | | 40 | | | | |
| 预计库存 | 65 | 45 | 25 | 5 | 25 | 5 | 25 | 25 |
| 净需求 | | | | | | 15 | | 15 |
| 计划定单入库 | | | | | | 40 | | 40 |
| 计划定单下达 | | | | 40 | | 40 | | |

项目C的物料需求 (需求量和时间)

| 周期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|----|----|----|----|----|----|----|----|
| 毛需求 | 15 | 15 | 15 | 15 | 15 | 15 | 15 | 15 |
| 计划收到 | | | | 30 | | | | |
| 预计库存 | 30 | 15 | | 15 | | 15 | | 15 |
| 净需求 | | | | | | 15 | | 15 |
| 计划定单入库 | | | | | | 30 | | 30 |
| 计划定单下达 | | | | 30 | | 30 | | |



例5 MRP的编制简例（当考虑批量规则时）

- 已知：某产品的毛需求和到货计划见表1，该产品的已分配量为零，提前期为2周，现分别采用按需订货法和固定批量法编制MRP（见表2和表3）。

1. 某产品的毛需求与计划收到量表

| 计划周期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|----|----|----|---|----|---|---|----|
| 毛需求 | 5 | 10 | 18 | 0 | 10 | 6 | 0 | 14 |
| 计划收到量 | | 20 | | | | | | |
| 现有库存 | 20 | | | | | | | |
| 净需求 | | | | | | | | |
| 发出订货计划 | | | | | | | | |

2. 采用按需订货法计算订货量表

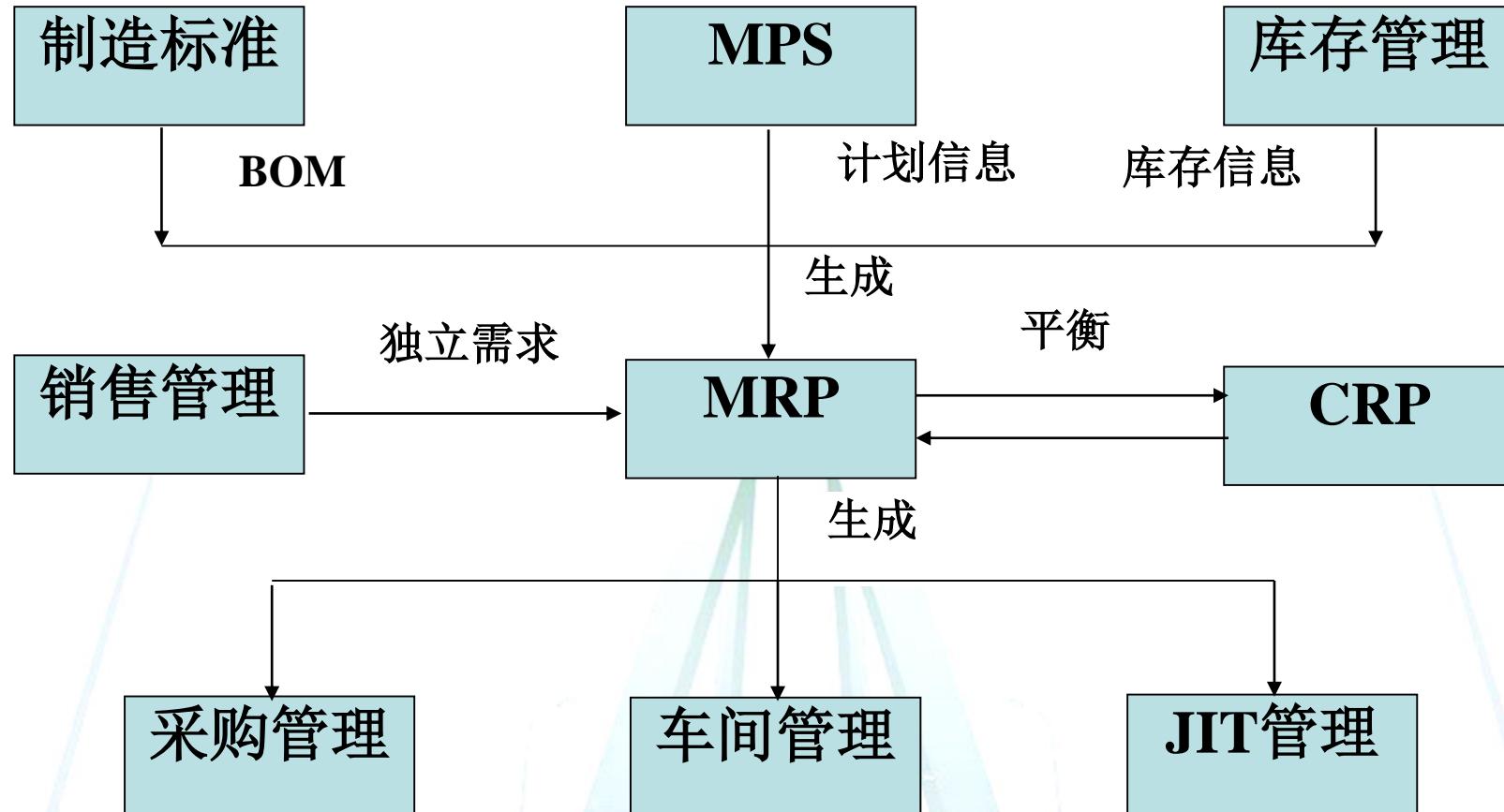
| 计划周期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|----|----|----|---|----|----|---|----|
| 毛需求 | 5 | 10 | 18 | 0 | 10 | 6 | 0 | 14 |
| 计划收到量 | | 20 | | | | | | |
| 现有库存 | 20 | 15 | 25 | 7 | 7 | 0 | 0 | 0 |
| 净需求 | | | | | 3 | 6 | | 14 |
| 发出订货计划 | | | 3 | 6 | | 14 | | |

3. 采用固定批量法计算订货量表

| 计划周期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|----|----|----|---|----|----|---|----|
| 毛需求 | 5 | 10 | 18 | 0 | 10 | 6 | 0 | 14 |
| 计划收到量 | | 20 | | | | | | |
| 现有库存 | 20 | 15 | 25 | 7 | 7 | 12 | 6 | 6 |
| 净需求 | | | | | 3 | | | 8 |
| 发出订货计划 | | | 15 | | | 15 | | |



⑤ MRP子系统与其他子系统的关系图

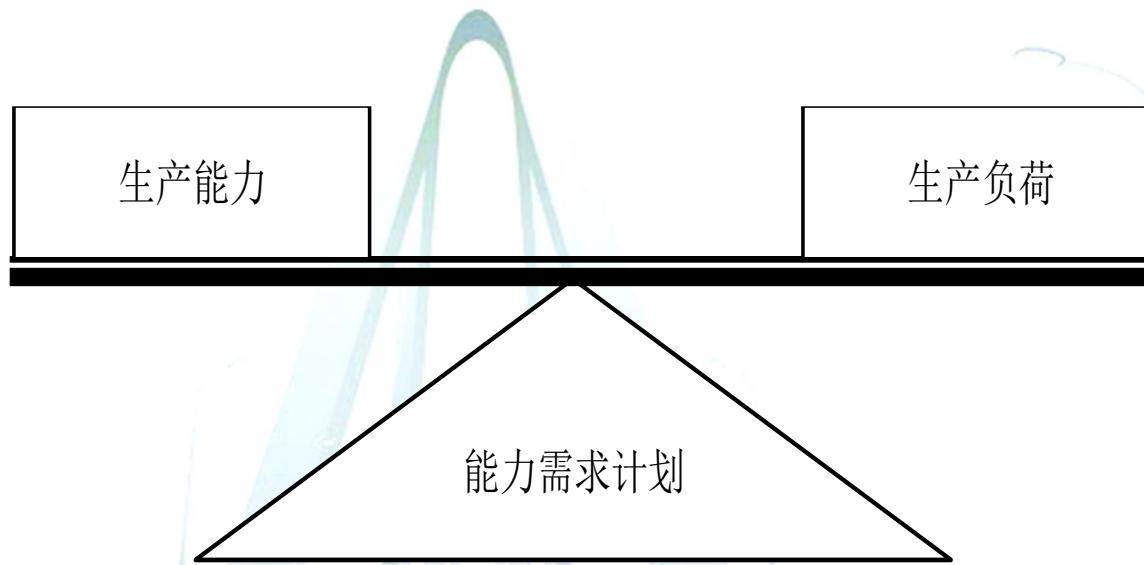




1. 企业资源管理基础
 2. 库存管理
 3. 物料需求计划
 4. 能力需求计划
 5. 供应链管理
 6. 小结
- A1 信息编码

4 能力需求计划

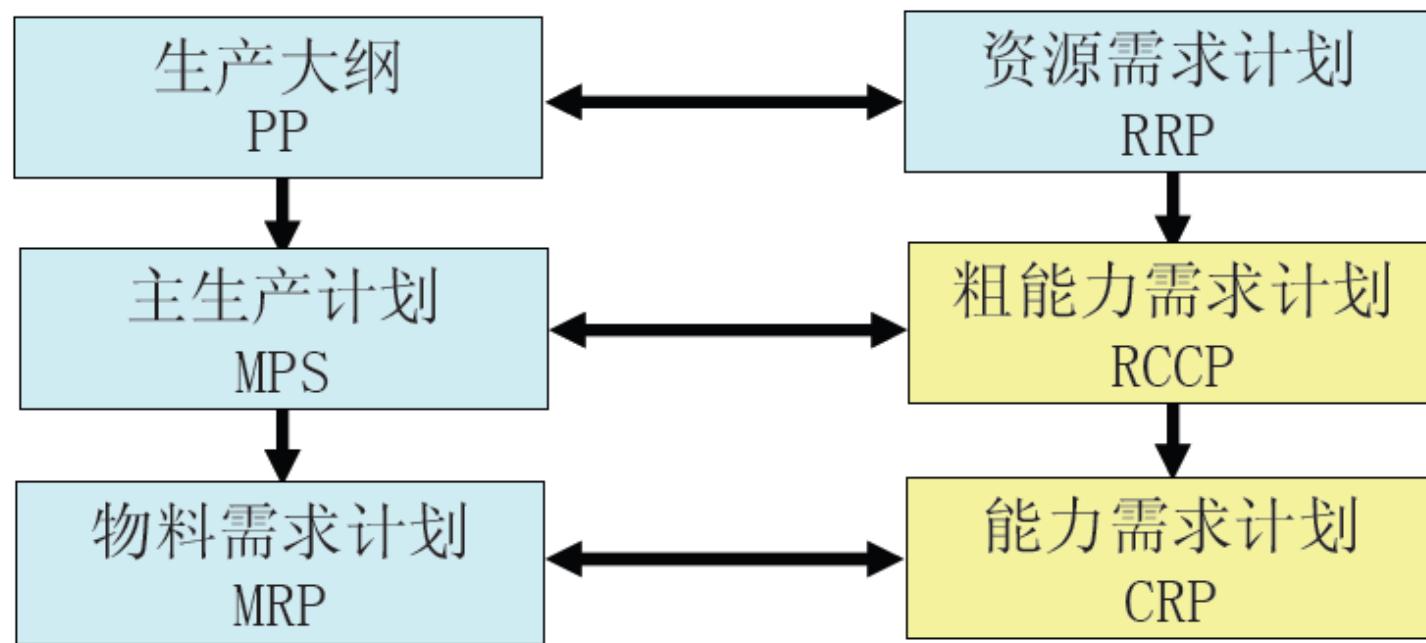
- 能力需求计划(Capacity Requirements Planning, CRP)是对各生产阶段和各工作中心（工序）所需的各种资源进行精确计算，得出人力负荷、设备负荷等资源负荷情况，并做好生产能力与生产负荷的平衡工作，最后制定出能力需求计划。
- 能力需求计划旨在通过分析比较MRP的需求和企业现有生产能力，及早发现能力的瓶颈所在，以协调生产和生产负荷的差距。





- 物料经过哪些工作中心加工
- 工作中心的可用能力和负荷是多少
- 工作中心各个时段的可用能力和负荷是多少

能力计划种类





RCCP与CRP比较

| 项目 | RCCP | CRP |
|------|---------------|-----------|
| 计划阶段 | MPS | MRP |
| 计划对象 | 关键工作中心 | 全部工作中心 |
| 负荷计算 | 独立需求物料 | 相关需求物料 |
| 工作日历 | 工厂日历、关键工作中心日历 | 加工中心日历 |
| 提前期 | 以计划周期为最小单位 | 物料开始与完工时间 |



⑤ 有限能力计划

思路：是在做能力需求计划时暂不考虑能力的限制，对各个工作中心的能力、负荷进行计算，得出工作中心的能力与负荷状况，产生能力需求计划方案，当负荷>能力时，对超负荷的工作中心进行调整。

特征：适用于普通物料；按物料完工日期倒排；需要做能力平衡与调整。

⑥ 无限能力计划

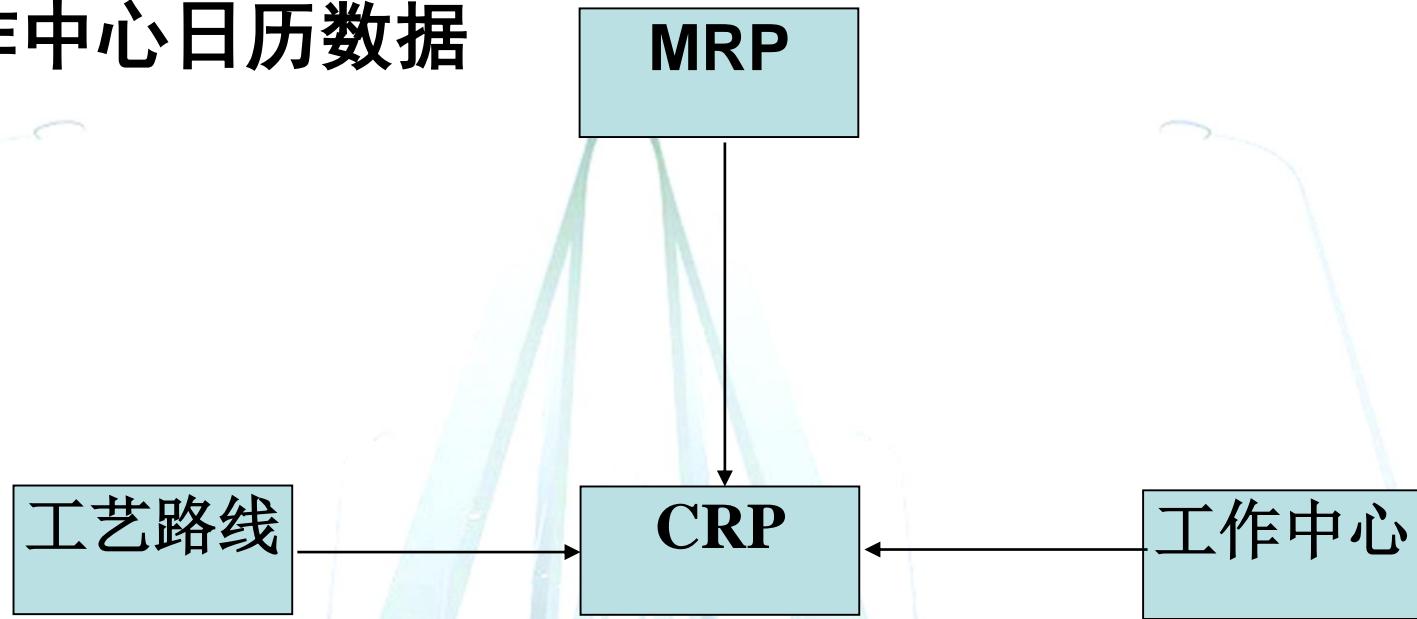
思路：是在有限可用能力限制下，按照计划的优先级进行安排，先把能力分配给优先级高的物料，当工作中心负荷已满时，优先级低的计划被延迟。按该方法制定的计划方案可以不进行负荷与能力平衡。

特征：适用于重要物料；按物料开始日期 顺排；不需要做能力平衡与调整。



CRP的输入数据

- 加工订单数据
- 工艺路线数据
- 工作中心数据
- 工作中心日历数据





- 收集数据（任务单、工作中心、工艺路线、工厂生产日历数据）
- 计算负荷
- 分析负荷情况
- 能力/负荷调整
- 确认能力需求计划

工作中心加工物品的负荷计算如下：

负荷=该物品产量×占用该工作中心的标准工时（或台时）

- 能力-负荷 ≥ 0 ，则满足加工要求，能力富余（或刚好）。
- 能力-负荷 < 0 ，则不能满足加工要求，能力不足。

工作中心W01的物料需求计划 (已下达及确认的)

| 周份 | 1 | 2 | 3 | 4 | 5 |
|-----|----|----|---|----|---|
| 物品A | 10 | | 5 | 10 | |
| 物品B | | 10 | 6 | | 5 |

工作中心W01的物料需求计划 新的计划

| 周份 | 1 | 2 | 3 | 4 | 5 |
|-----|---|---|----|----|---|
| 物品A | 5 | | 10 | | |
| 物品B | | 5 | | 10 | |

工艺路线

| 工作中心 | 物品 | 能力类别 | 能力数据 | 能力单位 |
|------|-----|------|------|------|
| WC01 | 物品A | 工时 | 10 | 小时 |
| WC01 | 物品B | 工时 | 5 | 小时 |

能力数据

工作中心: WC01
 能力类别: 工时
 能力数据: 100
 能力单位: 小时/周

工作中心WC01能力-负荷计算

| 周份 | 1 | 2 | 3 | 4 | 5 |
|---------|-----|-----|------|------|-----|
| 过去需求负荷 | 100 | 50 | 80 | 100 | 25 |
| 计划需求负荷 | 50 | 25 | 100 | 50 | 0 |
| 总负荷 | 150 | 75 | 180 | 150 | 25 |
| 平均能力 | 100 | 100 | 100 | 100 | 100 |
| 余/欠能力 | -50 | 25 | -80 | -50 | 75 |
| 累计余/欠能力 | -50 | -25 | -105 | -155 | -80 |

工作中心日历



信息

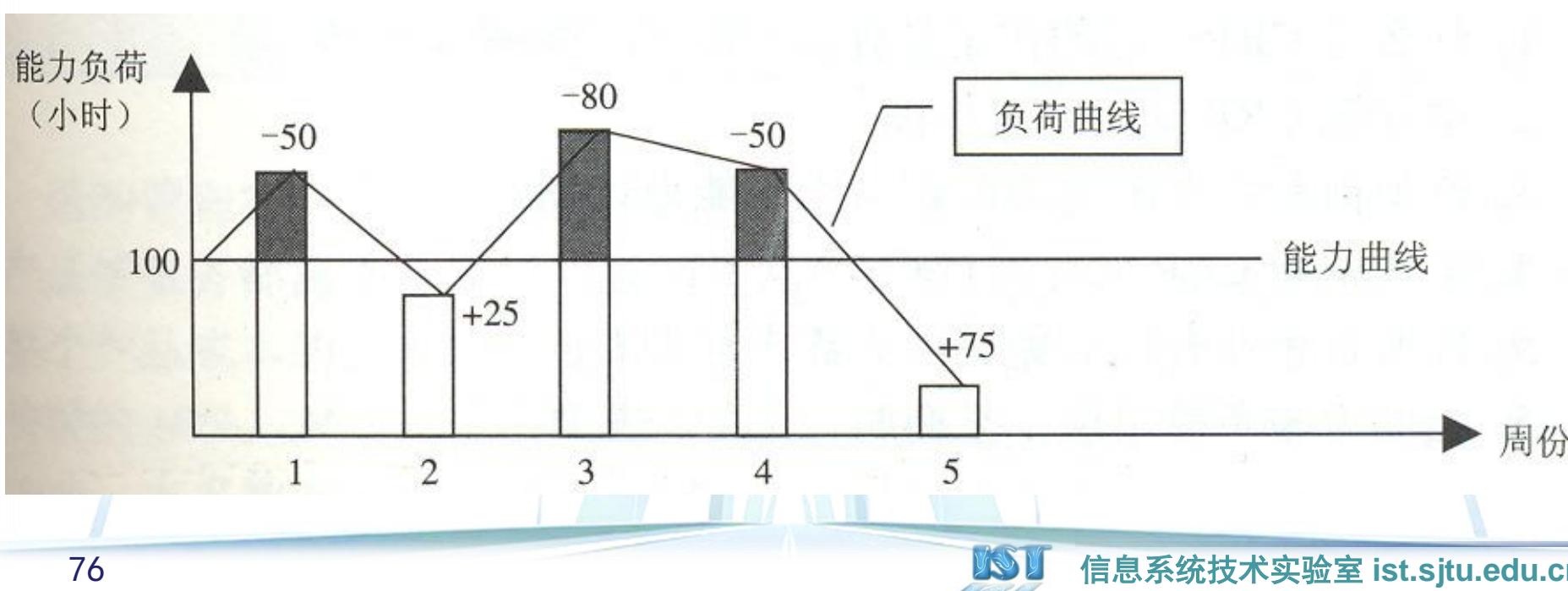


调整能力

- 加班
- 增加人员、设备
- 提高工作效率
- 更改工艺路线
- 增加外协处理等

调整负荷

- 调整生产批量
- 修改MRP、MPS计划
- 推迟交货期
- 撤消订单等





平衡后的能力与负荷图

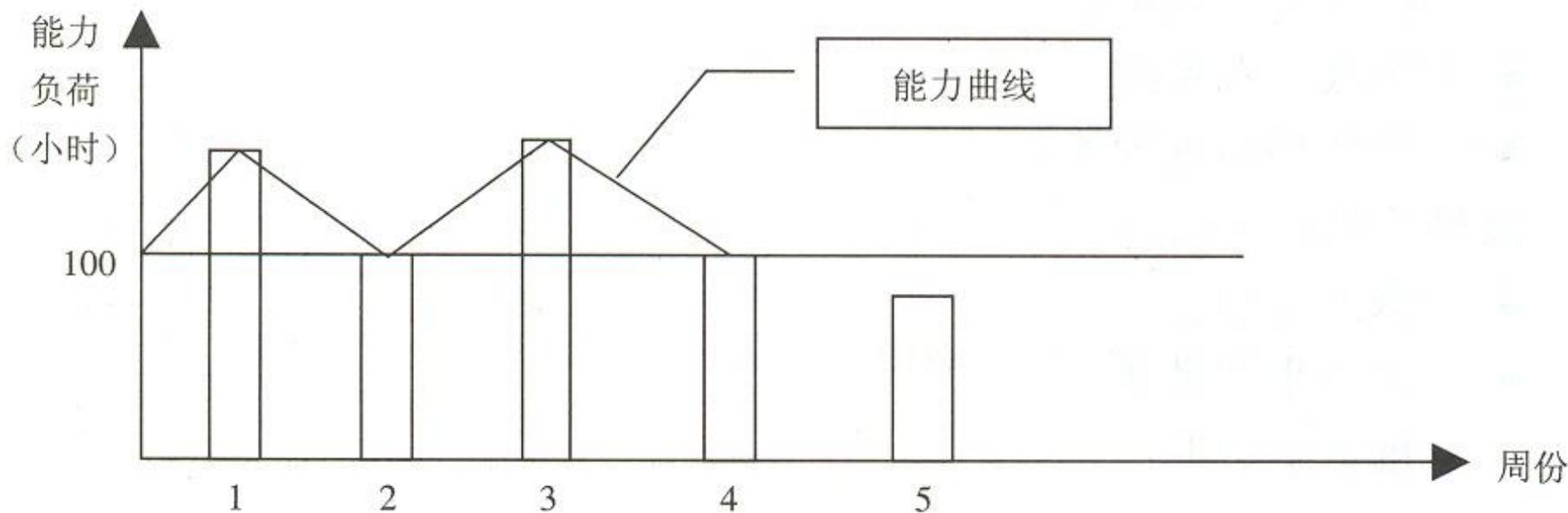
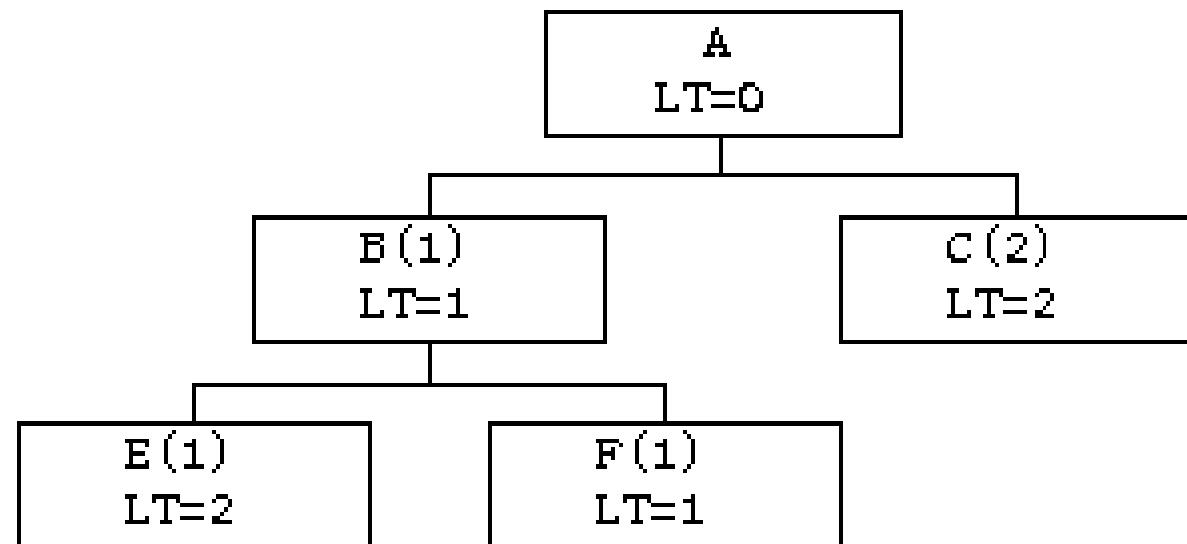


图 7.4 平衡后的能力与负荷曲线



CRP的编制案例

- 某产品A的物料清单见图3,其主生产计划、库存信息、工艺路线及工作中心工时定额信息和工序间隔时间见表2(a)~(d); 零件B、C的批量规则均是: 2周净需求, 零件E的批量规则是: 3周净需求, 零件F的批量规则是: 固定批量80; 每周工作5天, 每天工作6小时, 每个工作中心有一位操作工, 所有的工作中心利用率和效率均为95%。试编制其能力需求计划(CRP) 并分析其能力情况。





编制能力计划CRP

(a) 项目A主生产计划清单

| 周期 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|-----|----|----|----|----|----|----|----|----|----|----|
| 项目A | 25 | 25 | 20 | 20 | 20 | 20 | 30 | 30 | 30 | 25 |

(b) 库存信息

| 项目 | 计划收到量(计划周期) | | | | | | | | 现有库存 | 已分配量 | 提前期 | 固定批量 |
|----|-------------|---|----|---|---|---|---|---|------|------|-----|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | | | | |
| A | | | | | | | | | | | | |
| B | 38 | | | | | | | | 14 | | 1 | 2周 |
| E | | | 76 | | | | | | 5 | | 2 | 3周 |
| F | | | | | | | | | 22 | | 1 | 80 |
| C | 72 | | | | | | | | 33 | | 2 | 2周 |

(c) 工艺路线及工作中心工时定额信息

| 项目 | 工序号 | 工作中心 | 单件加工时间(h) | 生产准备时间(h) | 平均批量 | 单件准备时间(h) | 单件总时间(h) |
|----|-----|------|-----------|-----------|------|-----------|----------|
| A | 10 | 30 | 0.09 | 0.40 | 20 | 0.0200 | 0.1100 |
| B | 10 | 25 | 0.06 | 0.28 | 40 | 0.0070 | 0.0670 |
| C | 10 | 15 | 0.14 | 1.60 | 80 | 0.0200 | 0.1600 |
| | 20 | 20 | 0.07 | 1.10 | 80 | 0.0138 | 0.0838 |
| E | 10 | 10 | 0.11 | 0.85 | 100 | 0.0085 | 0.1185 |
| | 20 | 15 | 0.26 | 0.96 | 100 | 0.0096 | 0.2696 |
| F | 10 | 10 | 0.11 | 0.85 | 80 | 0.0106 | 0.1206 |

(d) 工作中心工序间隔时间

| 工作中心 | 工序间隔时间(天) | |
|------|-----------|------|
| | 排队时间 | 运输时间 |
| 30 | 2 | 1 |
| 25 | 2 | 1 |
| 20 | 1 | 1 |
| 15 | 1 | 1 |
| 10 | 1 | 1 |
| 库房 | — | 1 |



- 先编制MRP计划，再编制能力需求计划。
- 能力需求计划的编制则包括三个步骤
 - 计算工作中心能力
 - 用倒序排产法计算每道工序的开工日期和完工日期
 - 编制负荷图。



第1步：计算工作中心能力

- 为了编制能力需求计划，首先要计算工作中心的负荷能力。各工作中心所需的负荷能力可用下式计算：

工作中心负荷能力=件数×单件加工时间+准备时间

- 例如，对于工作中心30，由于最终产品A在工作中心30加工，单件加工时间和生产准备时间分别为0.09h和0.4h。因此对于主生产计划的每一批计划量25、20及30件在该工作中心的负荷能力分别是： $25 \times 0.09 + 0.4 = 2.65\text{h}$ 、 $20 \times 0.09 + 0.4 = 2.20\text{h}$ 和 $30 \times 0.09 + 0.4 = 3.10\text{h}$ 。
- 基于上述方法，可以得到其它加工中心所需的能力负荷情况。



产品A的MRP计划

| 项目 | | 计划周期 | | | | | | | | | |
|-----------|--------|------|----|----|-----|-----|-----|-----|----|-----|----|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| A | 主生产计划 | 25 | 25 | 20 | 20 | 20 | 20 | 30 | 30 | 30 | 25 |
| B LT=1 | 毛需求 | 25 | 25 | 20 | 20 | 20 | 20 | 30 | 30 | 30 | 25 |
| | 计划接收量 | 38 | | | | | | | | | |
| | 现有库存 | 14 | 27 | 2 | 20 | 0 | 20 | 0 | 30 | 0 | 25 |
| | 净需求 | | | 18 | 0 | 20 | 0 | 30 | 0 | 30 | 0 |
| | 计划定单入库 | | | 38 | | 40 | | 60 | | 55 | |
| | 计划定单下达 | | 38 | | 40 | | 60 | | 55 | | |
| E LT=2 | 毛需求 | | 38 | | 40 | | 60 | | 55 | | |
| | 计划接收量 | | 76 | | | | | | | | |
| | 现有库存 | 5 | 43 | 43 | 3 | 3 | 55 | 55 | 0 | 0 | 0 |
| | 净需求 | | | | | | 57 | | 55 | | |
| | 计划定单入库 | | | | | | 112 | | | | |
| | 计划定单下达 | | | | 112 | | | | | | |
| F LT=1 | 毛需求 | | 38 | | 40 | | 60 | | 55 | | |
| | 计划接收量 | | | | | | | | | | |
| | 现有库存 | 22 | 64 | 64 | 24 | 24 | 44 | 44 | 69 | 69 | 69 |
| | 净需求 | | 16 | | | | 36 | | 11 | | |
| | 计划定单入库 | | 80 | | | | 80 | | 80 | | |
| | 计划定单下达 | 80 | | | | 80 | | 80 | | | |
| C LT=2 | 毛需求 | 50 | 50 | 40 | 40 | 40 | 40 | 60 | 60 | 60 | 50 |
| | 计划接收量 | 72 | | | | | | | | | |
| | 现有库存 | 33 | 55 | 5 | 40 | 0 | 40 | 0 | 60 | 0 | |
| | 净需求 | | | 35 | 0 | 40 | 0 | 60 | 0 | 60 | 0 |
| | 计划定单入库 | | | 75 | | 80 | | 120 | | 110 | |
| | 计划定单下达 | 75 | | 80 | | 120 | | 110 | | | |



能力需求计划

| 零件 | 工作中心 | 耗期 | 计划周期 | | | | | | | | | |
|----|------|----|-------|-----|------|-------|-------|-----|------|------|------|-----|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| A | 30 | 0 | 265 | 265 | 220 | 220 | 220 | 220 | 3.10 | 3.10 | 3.10 | 265 |
| | 小计 | 0 | 265 | 265 | 220 | 220 | 220 | 220 | 3.10 | 3.10 | 3.10 | 265 |
| B | 25 | 0 | 0 | 265 | 0 | 268 | 0 | 388 | | 358 | 0 | 0 |
| | 小计 | 0 | 0 | 265 | 0 | 268 | 0 | 388 | | 358 | 0 | 0 |
| C | 20 | 0 | 635 | 0 | 6.75 | 0 | 950 | 0 | 880 | 0 | 0 | |
| | 小计 | 0 | 635 | 0 | 6.75 | 0 | 950 | 0 | 880 | 0 | 0 | |
| | 15 | 0 | 12.1 | 0 | 12.8 | 0 | 18.4 | 0 | 17.0 | | | |
| E | 15 | 0 | 20.72 | 0 | 0 | 0 | 30.38 | 0 | 0 | 0 | 0 | 0 |
| | 小计 | 0 | 32.82 | 0 | 12.8 | 0 | 48.78 | 0 | 17.0 | 0 | 0 | 0 |
| | 10 | 0 | 0 | 0 | 0 | 13.17 | 0 | 0 | 0 | 0 | 0 | 0 |
| F | 10 | 0 | 965 | 0 | 0 | 0 | 965 | 0 | 965 | 0 | 0 | 0 |
| | 小计 | 0 | 965 | 0 | 0 | 13.17 | 965 | 0 | 965 | 0 | 0 | 0 |

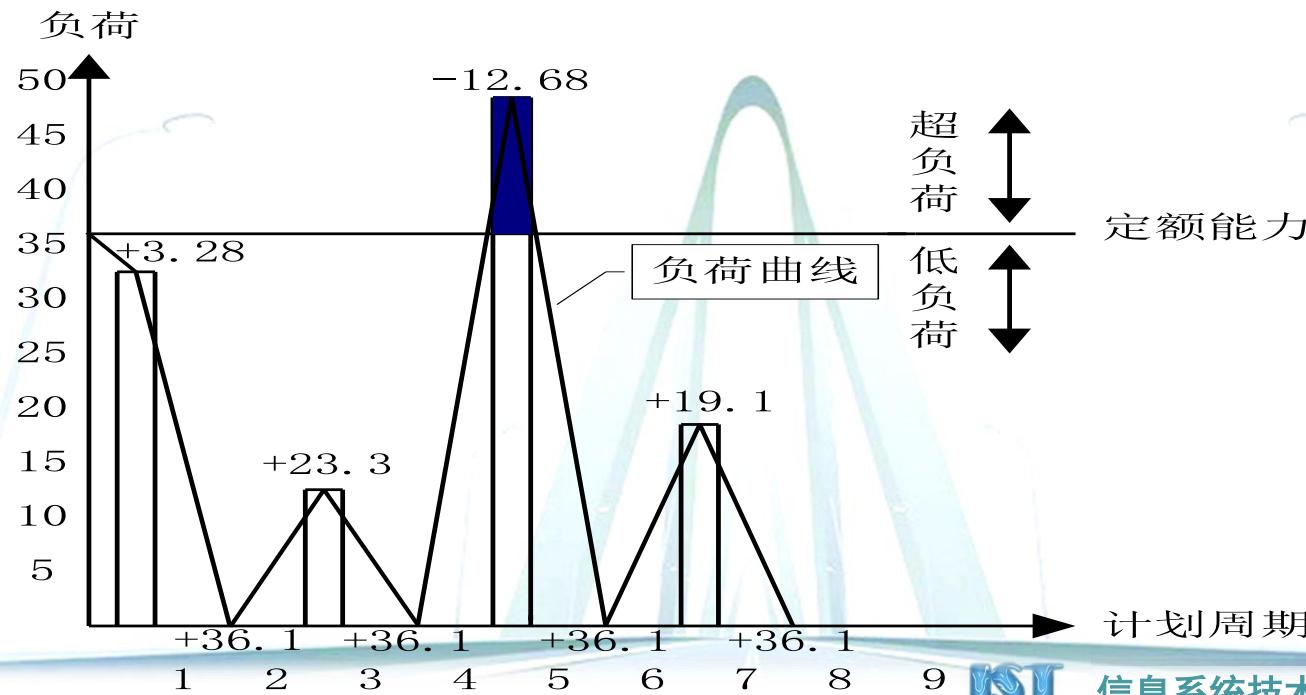
第2步：用倒序排产法计算每道工序的开工日期和完工日期

- ⑤ 所谓倒序排产法，是指将MRP确定的定单完成时间作为起点，然后安排各道工序，找出各工序的开工日期，进而得到MRP定单的最晚开工日期。
- ⑥ 以下仅以零件C为例，说明如何用倒序排产法计算每一批定单每道工序的开工日期和完工日期。
- ⑦ 由表可知，零件C的加工有两道顺序工序：工序10和工序20，这两道工序分别在工作中心15和20上完成。
- ⑧ 根据该题的已知条件：每周工作5天，每天工作6小时，每个工作中心有一位操作工，所有的工作中心利用率和效率均为95%，可以得到各工作中心每天的可用能力为： $8 \times 1 \times 0.95 \times 0.95 = 22\text{h}$ ；一周的最大生产能力为： $22 \times 5 = 36.1\text{h}$
- ⑨ 根据MRP计划表，第1批C零件应在第3周周1早上已经在库房中，因此其完成时间应该是第2周周5下午下班之前完成；由于最后一道工序20完成后，按计划还要安排1天运输时间，所以工序20的完工时间定在第2周周4的下班时间；工序20的生产时间为6.35h（见能力表），工作中心20每天的可用工时为22h，因此在工序20上的生产时间为 $6.35/22=0.88\text{天}$ （取整为1天）；而在在工作中心20前的排队时间和将工作中心15运输到工作中心20的时间都是1天，所以第2周的第2天和第3天用于运输和排队等待；工序10的完成时间应在第2周的第1天下班时间；工序10的生产时间为2天（ $12.10/22=1.68$ ，取整为2天）。工序10的开工时间为第1周第5天，由于还要有1天运输和1天等待，这样，这批定单的开工时间为第1周的第3天。



第3步：编制负荷图

- 以下以工作中心15为例，说明如何编制工作中心能力负荷曲线图。
- 根据已知上述的有关计算知，工作中心15的额定可用能力为36.1，在第1、3、5、7周的能力需求分别为32.82、12.80、478、10，因此除了第5周因其能力-负荷= -12.6<0，因此其负荷处于超负荷状态（或其能力处于欠能力状态），其余各周均处于超能力或低负荷状态。工作中心15的负荷曲线图如图4所示。
- 图4为工作中心15的能力负荷曲线图。





1. 企业资源管理基础
 2. 库存管理
 3. 物料需求计划
 4. 能力需求计划
 5. 供应链管理
 6. 小结
- A1 信息编码

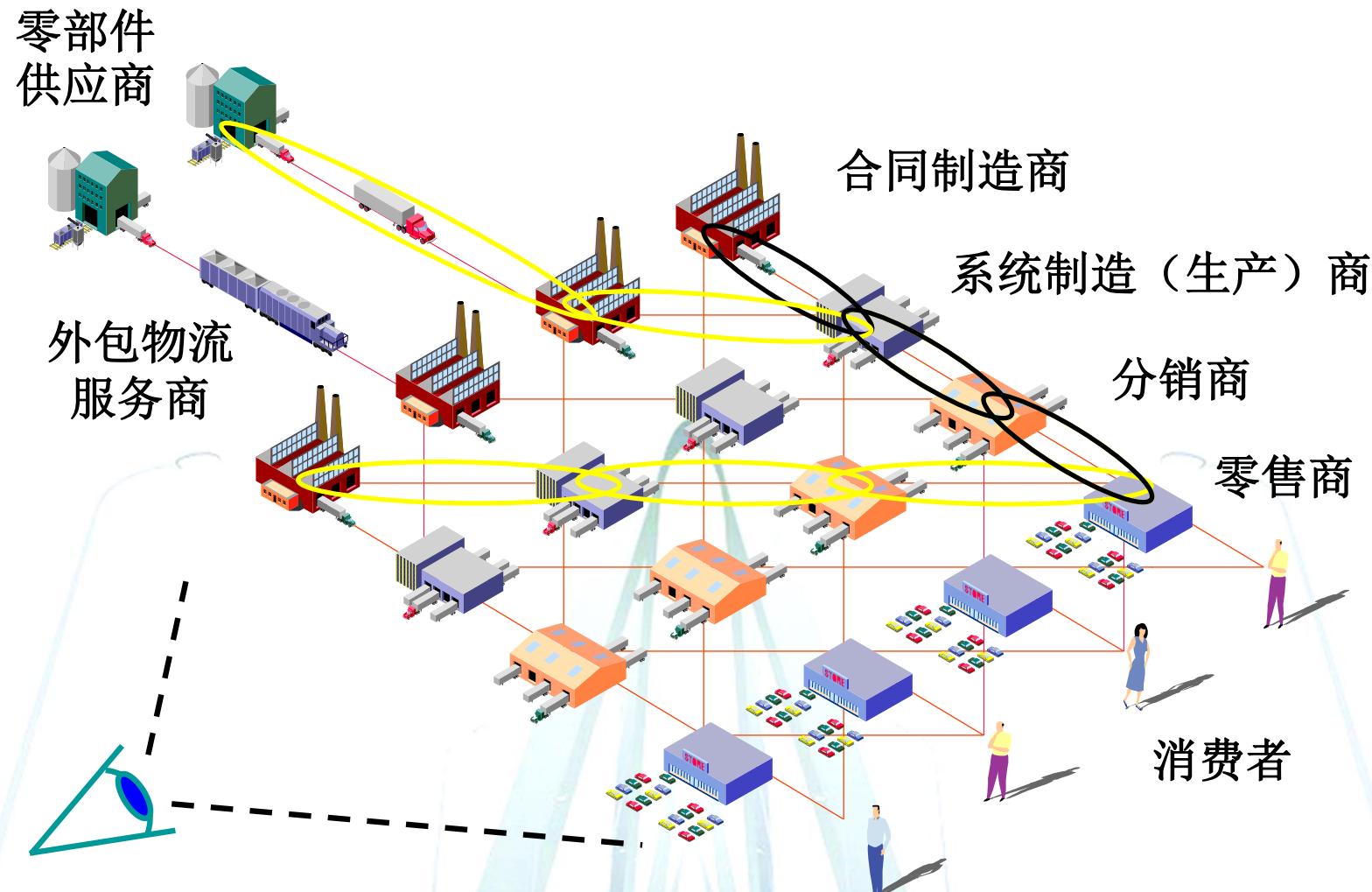


5 供应链管理

- ① 供应链是在相互关联的部门或业务伙伴之间所发生的物流、资金流和信息流，覆盖从产品（或服务）设计、原材料采购、到交付给最终消费者的全过程。
- ② 供应链不仅仅面向制造企业，服务性行业也存在供应链。
- ③ 供应链管理指借助信息技术和电子商务，将供应链上业务伙伴的业务流程相互集成，从而有效地管理从原材料采购、产品制造、分销、到交付给最终消费者的全过程，在提高客户满意度的同时，降低成本、提高企业的效益。
 - 供应链中的信息畸变（长尾效应）
 - 基于供应链的存货管理
 - 供应链管理的实践举例
 - 案例研究：自动补货系统
 - 供应链管理的效益

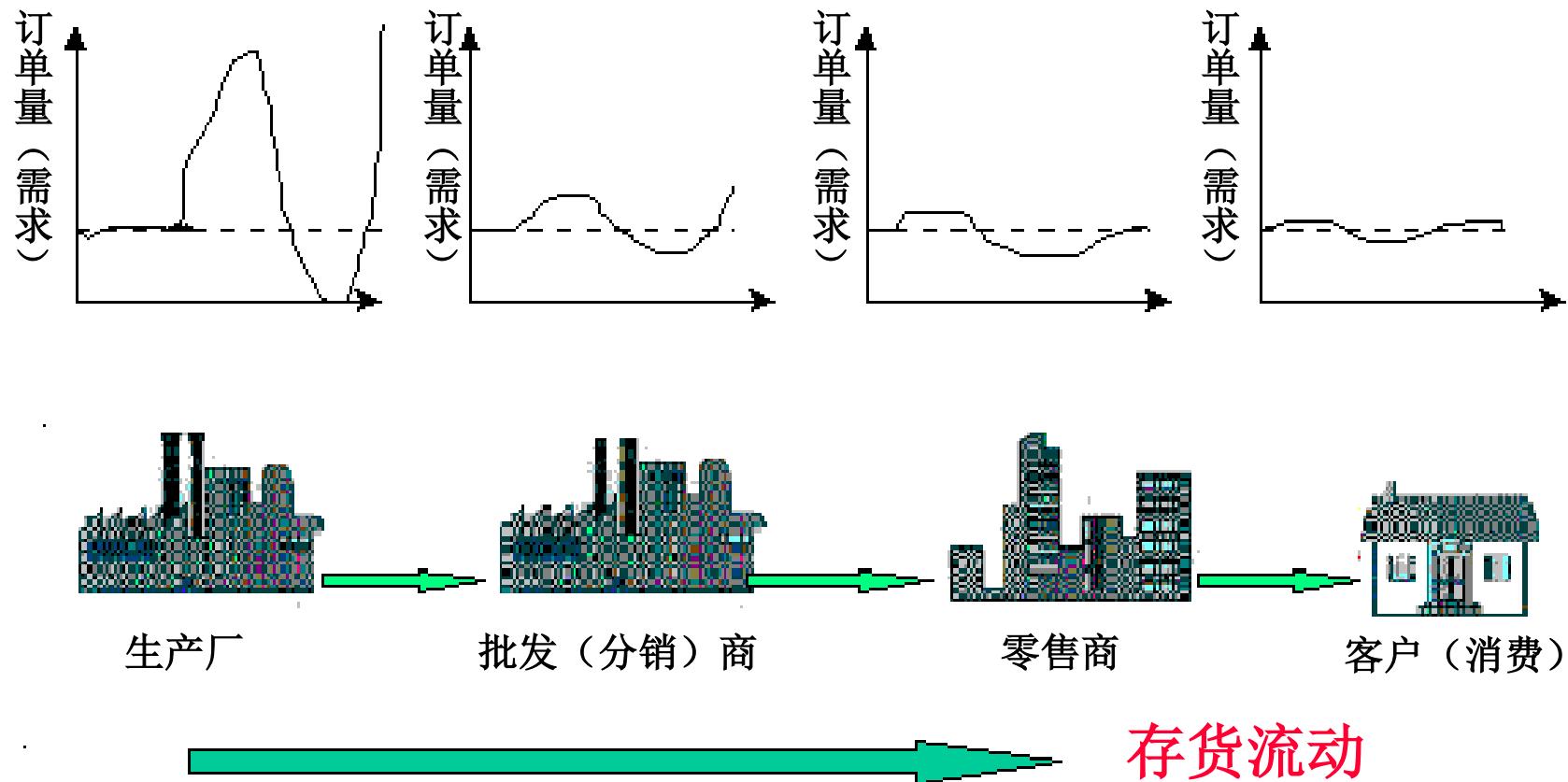


典型供应链的构成



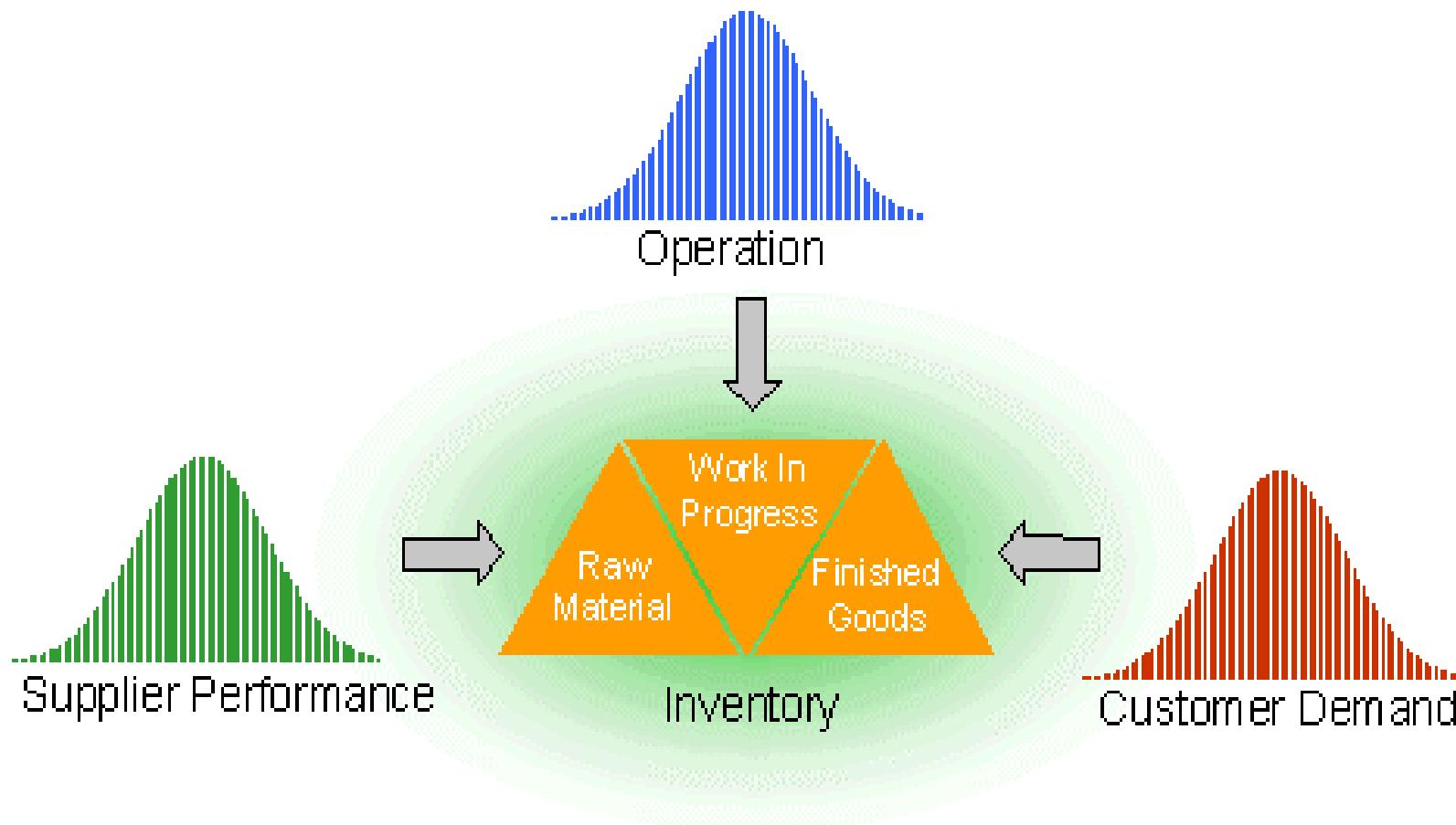


供应链中的信息畸变：长尾效应





抵御供应链中的不确定性





决定存货水平的因素

供应商的
不确定性

- 平均补充时间
- 标准差

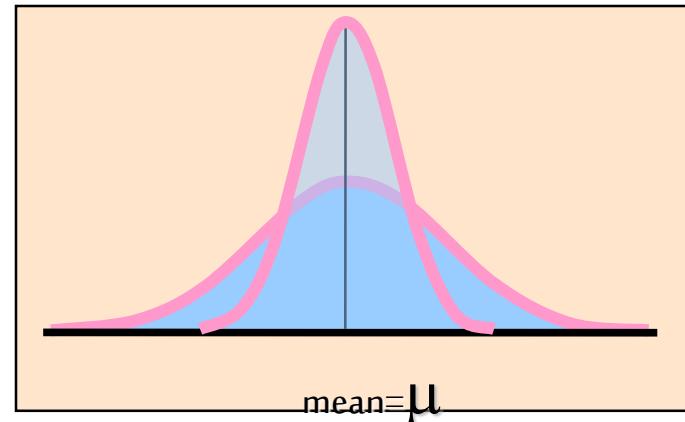
需求的
不确定性

- 平均消费量
- 标准差



运营策略

- 查货周期
- 交付频度
- 服务水平

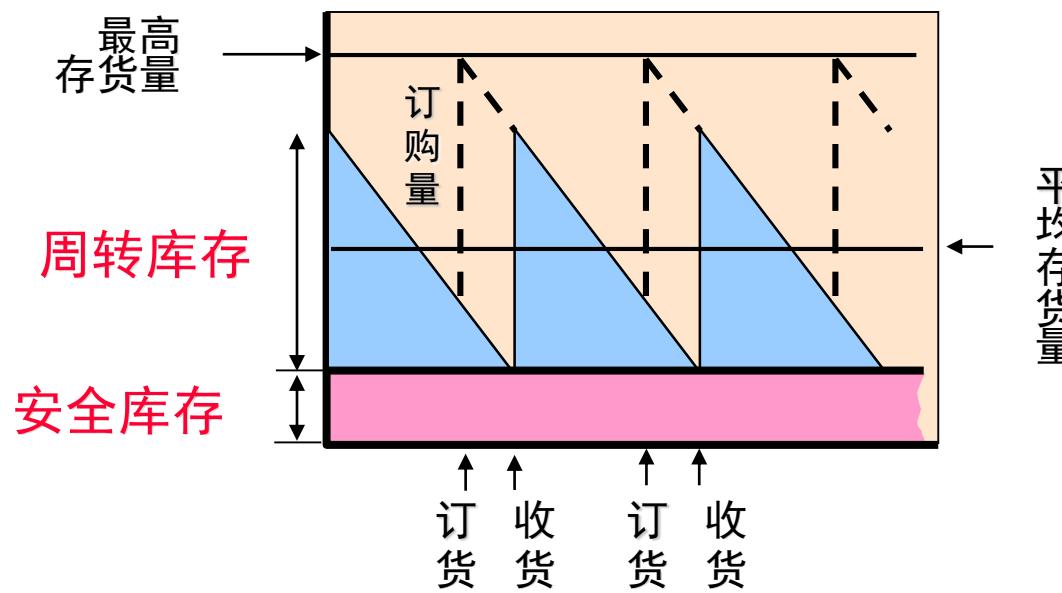




经济定货量给我们的启示

$$EOQ = \sqrt{\frac{2DS}{H}}$$

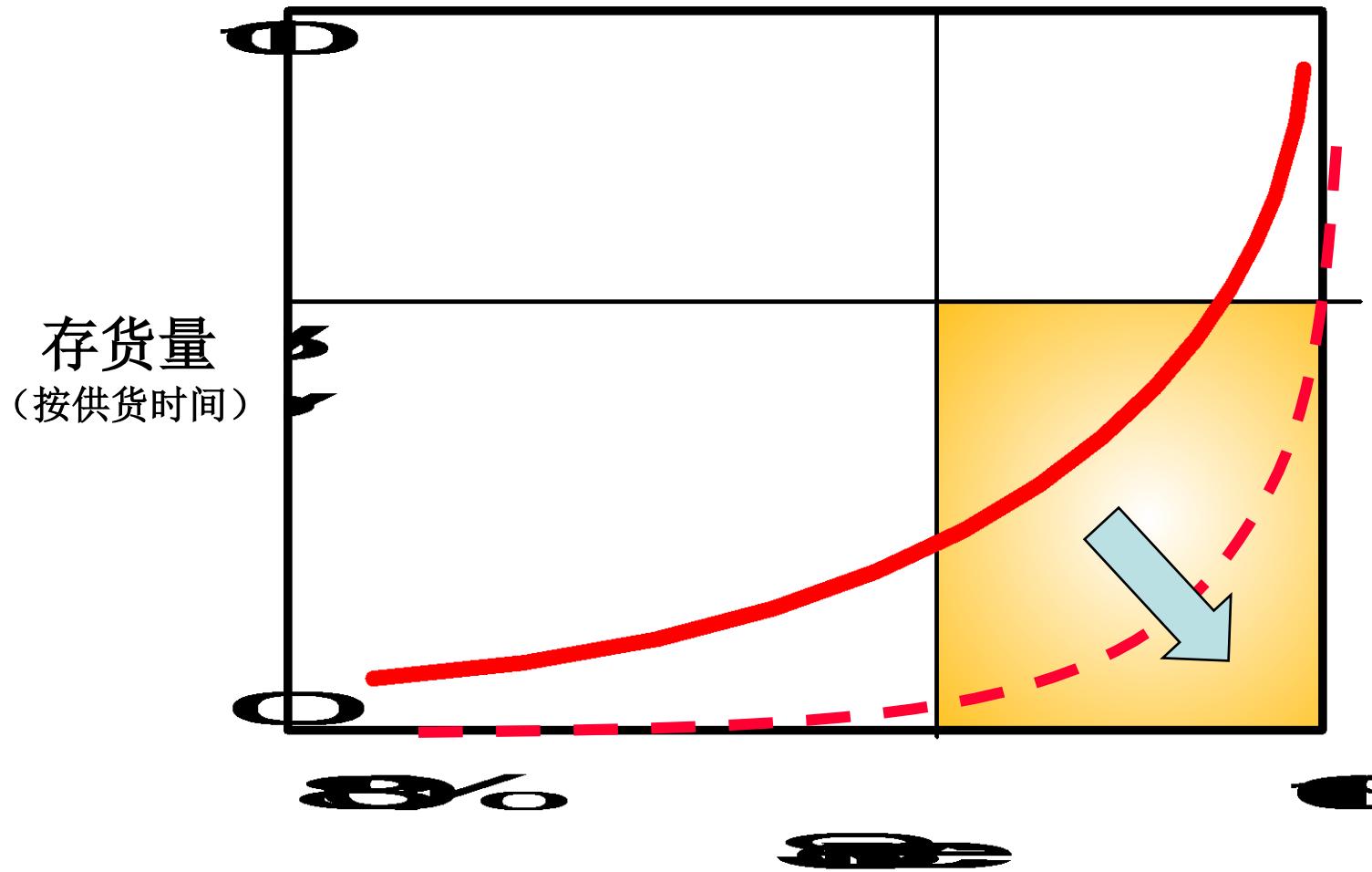
D: 年需求量 (单位/年)
S: 订货成本 (元/订单)
H: 单位储存成本 (元/单位)





供货水平与存货量之间的关系

(供应链管理临界效率曲线)





克服长尾效应

原因

需求数据处理
(独立预测)

订单批量限制

价格波动

短缺屯货

消除策略

需求信息共享，建立业务伙伴，
联合制定联合预测

运输工具共享与集中管理；
采用第三方物流服务（3PL）

采购订单与收货同步，建立伙伴关系

重新设定配货规则，仓库和存货共享



① 改进预测准确性

- POS 数据
- 生产计划制定

② 改进质量一致性

- 减少波动/平衡生产

③ 降低成本

- 信息替代了实际库存

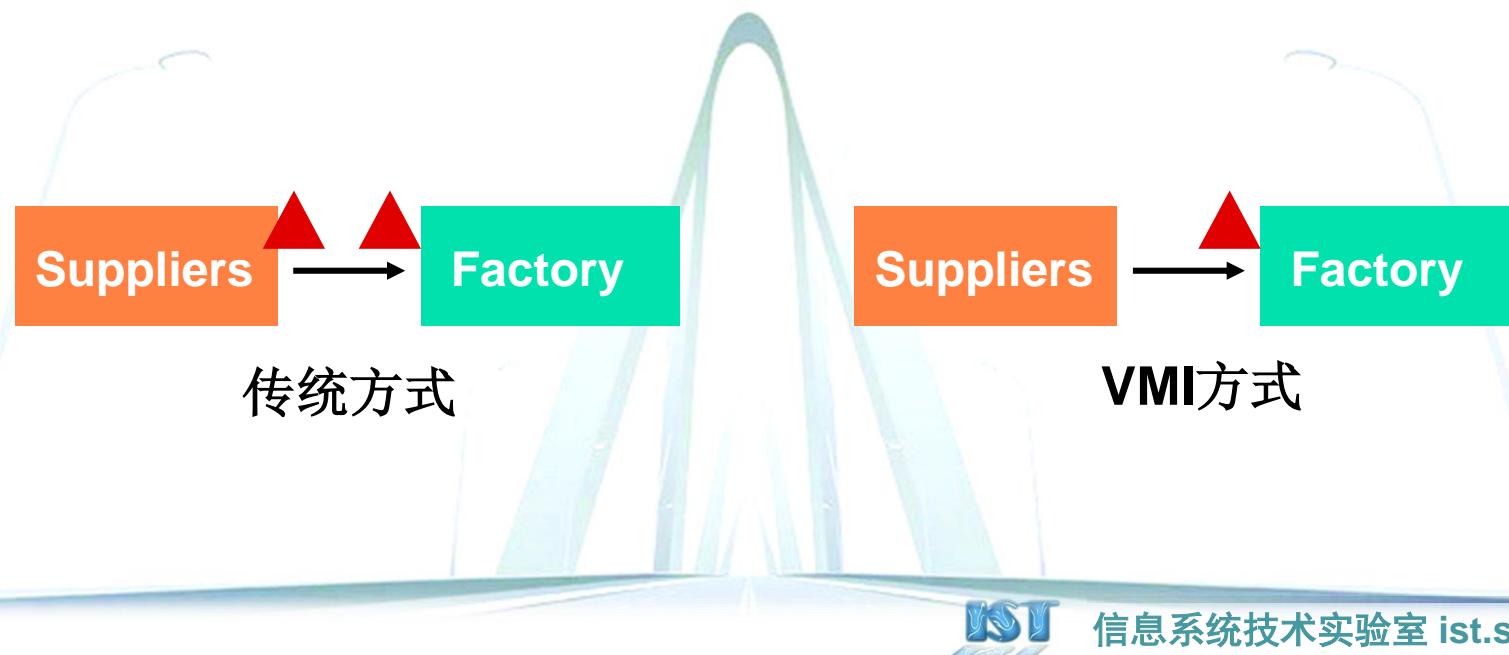
④ 提高设备利用率

⑤ 实现业务伙伴互联互通是供应链的基础



供应商管理存货策略（VMI）

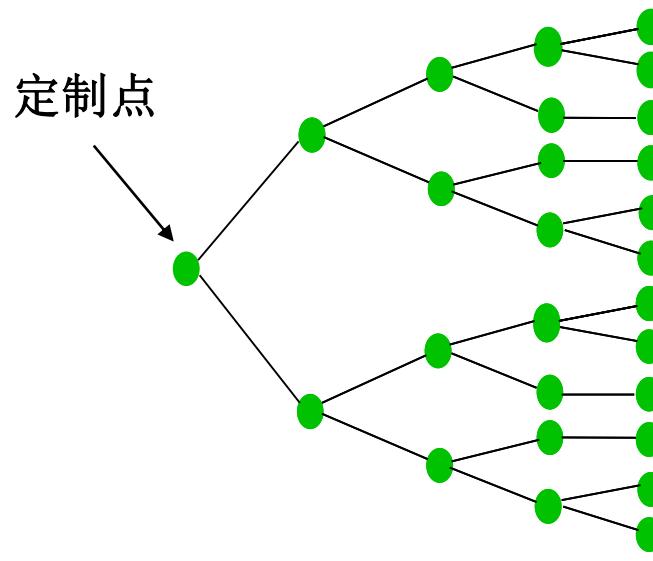
- ① VMI 是一种新型的物料分配和操作方式，从而达到改进供应响应速度、稳定生产过程、缩短供货周期的目的。
- ② 供应链上双赢的合作方式，可以同时满足零售商、批发商、制造商对降低存货成本、提高供货服务水平的需要。



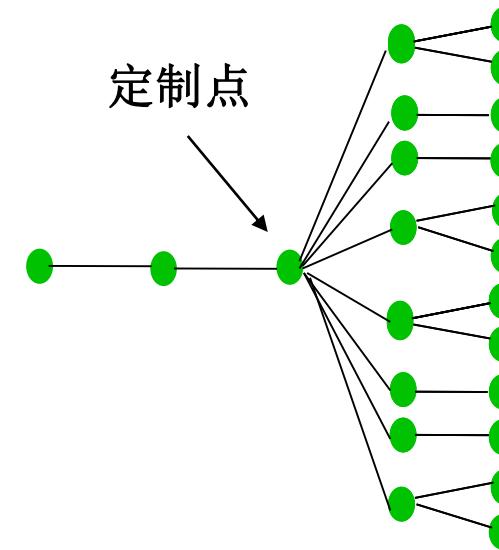


推迟定制

推迟定制是指在设计产品时，将产品的制造和订单执行过程尽可能地标准化，推迟客户差异点，在提高客户价值的同时，提高了资产利用率。



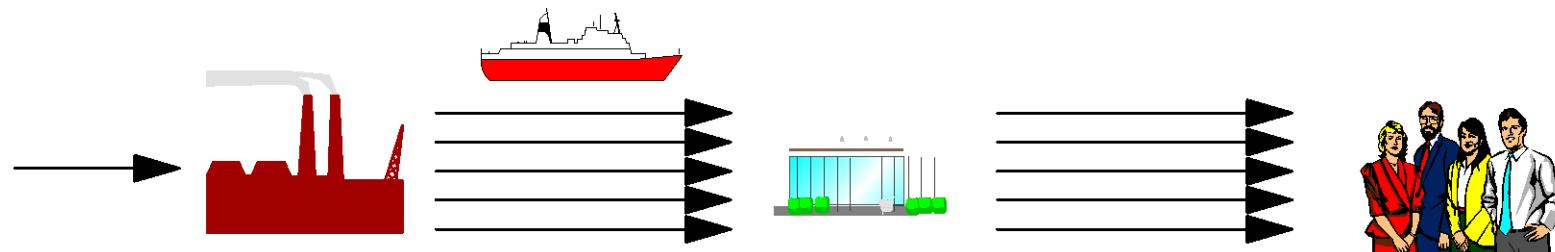
产品标准化之前



产品标准化之后

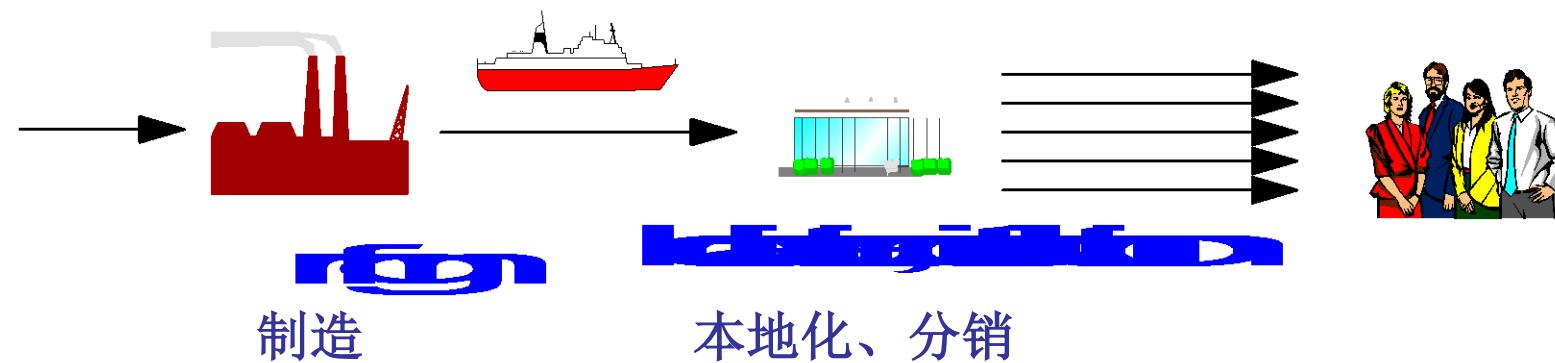


HP 喷墨打印机的推迟定制



制造、本地化

分销



制造

本地化、分销



供应链管理的杠杆作用

经济效益 + 客户满意度

- 市场占有率 / 销售收入
- 改进的 ROA
- 减少流动资产
- 节省物流费用（成本）

- 流程 + 信息系统 + 组织
 - 改善供应链管理



① 降低成本

- 库存降低 10 – 50%
- 减少削价处理和报废 40 – 50%
- 提高资源利用率 10 – 20%

② 改善客户服务水平

- 改进交付可靠性达 95 – 99. 9%
- 缩短交付时间 10 – 20%

③ 加快资金周转

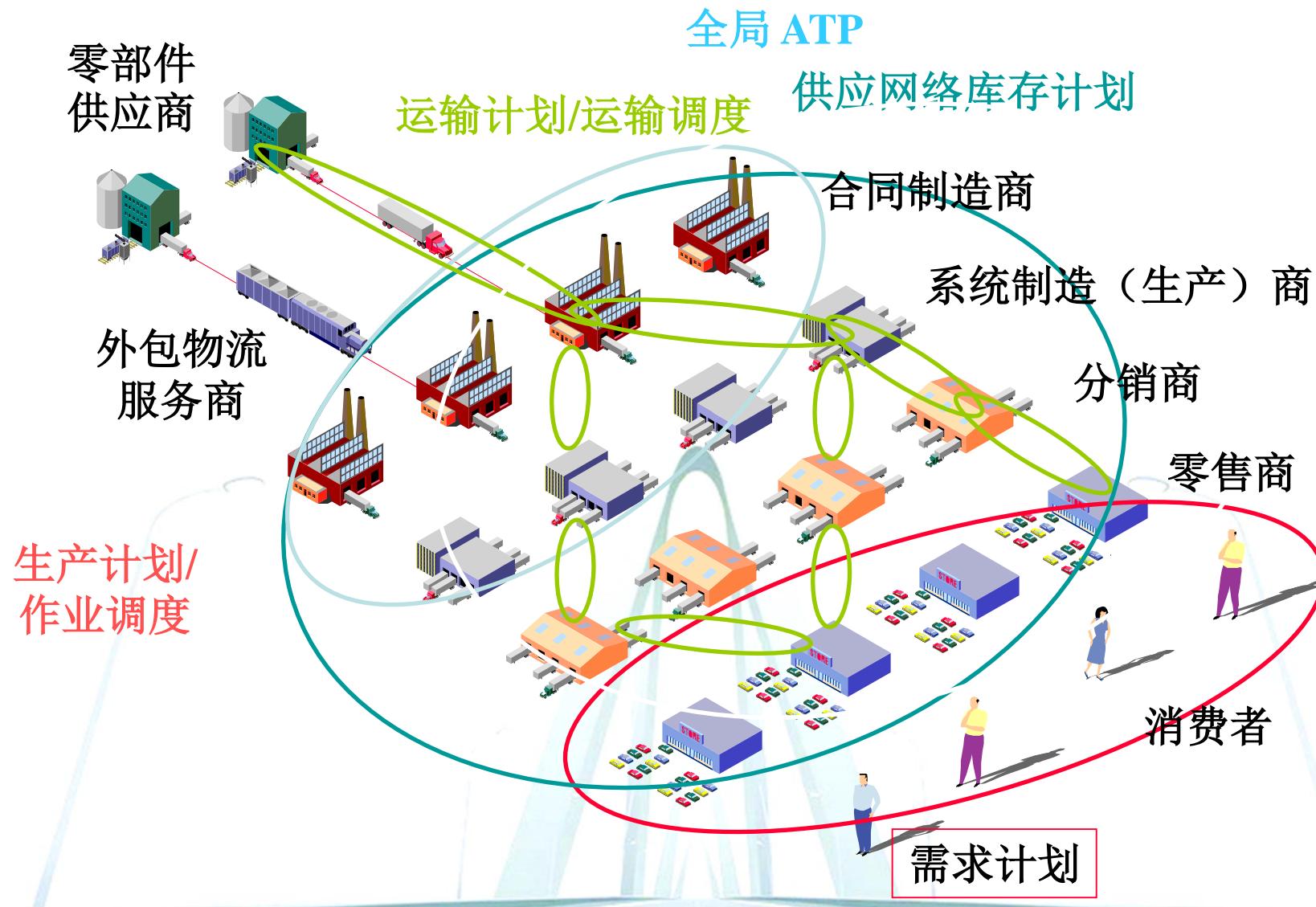
- 比一般企业的资金周转时间短 40 – 60%

④ 增加市场占有率

⑤ 成为受欢迎的业务伙伴



供应链软件方案举例





1. 企业资源管理基础
 2. 库存管理
 3. 物料需求计划
 4. 能力需求计划
 5. 供应链管理
 6. 小结
- A1 信息编码



6 小结

- 企业的资源主要有物料，设备等，资源的信息表示是计算机应用的基础；
- 企业内容资源具体的管理功能主要有库存管理，物料需求，设备能力管理等部分；
- 供应链管理是企业产业链条件下的资源动态管理过程。



A1 信息编码方法

- 信息分类编码是利用计算机辅助企业管理必要的前提条件。
- 信息编码应遵循相应的信息分类方法。一般情况下，信息的**分类在先，编码在后**。
- 编码的目的

唯一化

为事物提供一个概要而不含糊的认定，
便于数据的存储和检索。

规范化

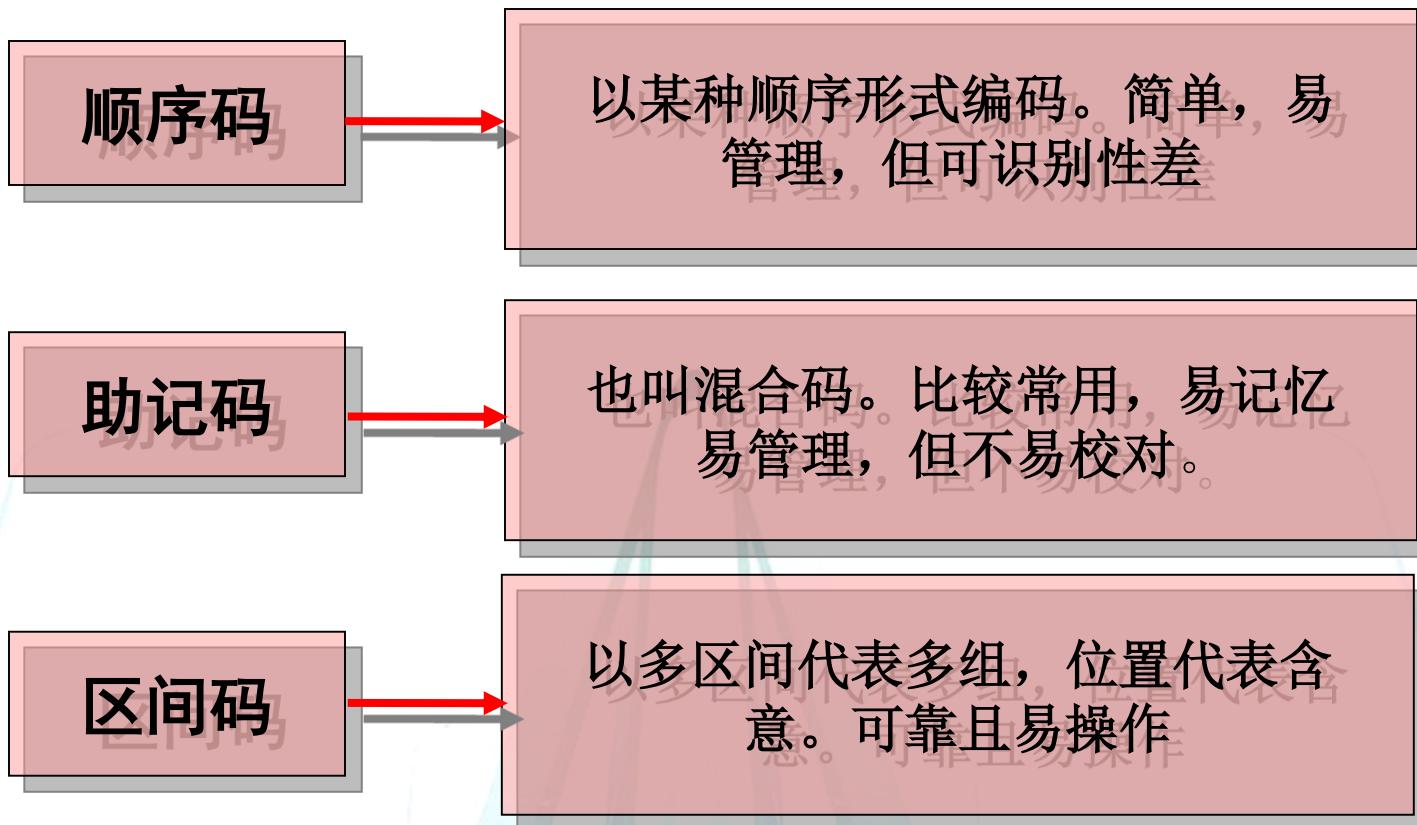
即编码要有规律，符合某一类事物的
聚集，提高处理的效率和精度。

系统化

也即标准化，符合国家或行业标准，
提高数据全局一致性。



◎ 编码的种类



(一) 顺序码

- 顺序码又称系列码，它是一种用连续数字代表编码对象的码，例如：用1代表厂长，2代表科长，3代表科员，4代表生产工人等。
- 顺序码的优点是短而简单，记录的定位方法简单，易于管理。但这种码没有逻辑基础，它本身不能说明任何信息的特征。
- 此外，新加的编码只能列在最后，删除则造成空码。通常，顺序码作为其他码分类中细分类的一种补充手段。

(二) 助忆码

助忆码用文字、数字或文字数字结合起来描述，其特点是，可以通过联想帮助记忆。

例如，

- ① 用TV—B—20代表20英吋黑白电视机
- ② 用TV—C—25代表25英吋彩色电视机。

- ③ 助忆码适用于数据项数目较少的情况（一般少于50个），否则可能引起联想出错。此外，太长的助忆码占用计算机容量太多，也不宜采用。

(三) 区间码

- ① 区间码把数据项分成若干组，每一区间代表一个组，**码中数字的值和位置都代表一定意义。典型的例子是邮政编码。**
- ② 区间码的优点是：信息处理比较可靠，排序、分类、检索等操作易于进行。但这种码的长度与它分类属性的数量有关，有时可能造成很长的码。在许多情况下，码有多余的数。同时，这种码的维修也比较困难。
- ③ 根据需要，也可在区位中某部分区别不同的块。如100-299表示企业管理人员，300-899表示生产人员、900-999表示专家或临时人员等。
- ④ 区间码又可分为多面码、上下关联区间码和十进位码三种。



1. 多面码

一个数据项可能具有多方面的特性。如果在码的结构中，为这些特性各规定一个位置，就形成多面码。

- ① 例如，对于机制螺钉，可作如下表那样的规定。编码2342表示材料为黄铜的Q1.5mm方形头镀铬螺钉。
- ② 多面码示例

| 材料 | 螺钉直径 | 螺钉头形状 | 表面处理 |
|-------|--------|--------|-------|
| 1—不锈钢 | 1-Q0.5 | 1—圆头 | 1—未处理 |
| 2—黄铜 | 2-Q1.0 | 2—平头 | 2—镀铬 |
| 3—钢 | 3-Q1.5 | 3—六角形状 | 3—镀锌 |
| | | 4—方形头 | 4—上漆 |



2. 上下关联区间码

上下关联区间码由几个意义上相互有关的区间码组成，其结构一般由左向右排列。

例如，会计核算方面，用最左位代表核算种类，下一位代表会计核算项目。

● 资产类——现金（101）

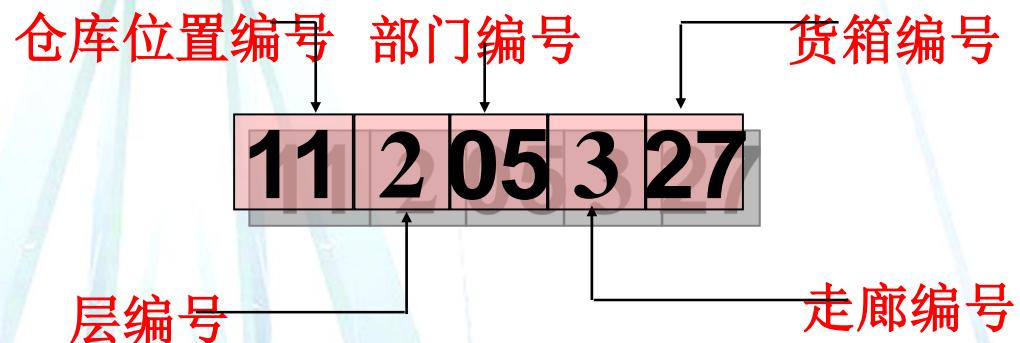
——人民币10101

——美元10102

——欧元10103

另一个例子

物流中的库存编号11205327



3. 十进位码

- 此法相当于图书分类中沿用已久的十进位分类码，它是由上下关联区间码发展而成的。如 610. 736，小数点左边的数字组合代表主要分类，小数点右边的指出子分类。
- 子分类划分虽然很方便，但所占位数长短不齐，不适于计算机处理。显然，只要把编码的位数固定下来，仍可利用计算机处理。



◎ 编码结构中的校验位

- 保证正确输入，有意识地在编码设计结构中原有编码的基础上，另外加上一个校验位，使它事实上变成编码的一个组成部分。

◎ 校验位可以发现以下各种错误：

- 抄写错误，例如1写成7；
- 易位错误，例如1234写成1324；
- 双易错误，例如26913写成21963；
- 随机错误，包括以上两种或三种综合性错误或其他错误。



校验位的确定示例

校验位通过事先规定的数学方法计算出来。

- 如某编码设计方法中 共XXXX X，其中第五位为校验位，令A1 A2 A3 A4 表示该编码的1-4位；则不妨取
- $a_5 = \text{MOD}[(a_1 + 2a_2 + 3a_3 + 4a_4), 9]$

例如：

$$\begin{array}{ccccc} a_1 & a_2 & a_3 & a_4 & a_5 \\ 5 & 1 & 3 & 2 & 6 \\ \times) & 1 & 2 & 3 & 4 \\ & 5+2+9+8=24 \end{array}$$

则 $a_5 = (24/9)$ 的余数=6

当编码51326错写为57326时，计算中便可得到 $a_5=0$ ，因此51326、57320为正确情况，可得出57326不能通过校验。



校验的方法

确定校验位值的方法

1. 算术级数法。

原代码

12 3 4 5

各乘以权 65432

乘积之和

$$6+10+12+12+10=50$$

以 11 为模去除乘积之和，把得出的余数作为校验码：

$$50 / 11 = 4 \cdots 6$$

因此代码为 123456。

2. 几何级数法。

原代码

12 3 4 5

各乘以权

32 16 8 4 2

$$\text{乘积之和 } 32+32+24+16+10=114$$

以 11 为模去除乘积之和，把得出的余数作为校验码：

$$114 / 11 = 10 \cdots 4$$

因此代码为 123454。

3. 质数法。

原代码回 2345

各乘以权

17 13 7 5 3

乘积之和

$$17+26+21+20+15=99$$

以 11 为模去除乘积之和，把得出余数作为校验码：

$$99 / 11 = 9 \dots 0$$

因此代码为 123450

注意，以 11 为模时，若余数是 10，则按 0 处理。

还有一种常见的确定方法是
位数累加法。



身份证校验方法

十七位数字本体码加权求和公式

$$S = \sum A_i * W_i, i=1, \dots, 17, \text{ 其中 } W_i = 2^{(i-1)} \pmod{11}, i=1, \dots, 17;$$

$$Y = \text{mod}(S, 11);$$

i : 表示号码字符从左至右的位置序号, A_i : 表示第 i 位置上的身份证号码字符值, W_i : 表示第 i 位置上的加权因子, 位置序号与加权因子对应关系如下:

| i : | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|
|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|

| | | | | | | | | | | | | | | | | | |
|---------|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|
| W_i : | 1 | 2 | 4 | 8 | 5 | 10 | 9 | 7 | 3 | 6 | 1 | 2 | 4 | 8 | 5 | 10 | 9 |
|---------|---|---|---|---|---|----|---|---|---|---|---|---|---|---|---|----|---|

校验码字符值的计算:

| | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|----|
| Y: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|----|---|---|---|---|---|---|---|---|---|---|----|

| | | | | | | | | | | | |
|------|---|---|---|---|---|---|---|---|---|---|---|
| 校验码: | 1 | 0 | X | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 |
|------|---|---|---|---|---|---|---|---|---|---|---|



编码策略

① 1 编码的精简性

- 如需要编码来识别250个客户，就不需要6位数字编码。

② 2 编码的可扩充性

- 考虑编码数量的合理增长。

③ 3 编码的稳定性

- 编码的改变可导致一致性问题以及要求数据的更新，因此稳定性较为重要。如转换到新编码时，有时甚至需要新旧共存一段时间。

④ 4 编码的唯一性

- 如系统同时有hardware及Housework，使用HW便没有意义。



- ⑤ 5 采用不易出错的代码结构，数字、字母不占用不同的位置，避免混淆编码。
 - 如5ZO8I,52081,或者S2OB1等。
- ⑥ 6 使用分类编码。
 - 如100和300是一类产品，200是另一类，可增加指导性位（0, 1）保证编码正确分类。
- ⑦ 7 编码有意义。
 - 便于记忆使用，如EN比XQA或者132更好。
- ⑧ 8 合理的代码结构。
 - 当代码长于4个字母或5个数字字符时，应分成小段；
 - 如注册码：J2MV9-JYYQ6-JM44K-QMYTH-8RB2W



信息系统分析与设计

Lecture 6

UML业务建模扩展

蔡鸿明

hmcai@sjtu.edu.cn

软件学院信息系统技术实验室

lst.sjtu.edu.cn



Agenda

1. UML业务建模扩展

2. 业务视图

 2.1 业务愿景视图

 2.2 业务过程视图

 2.3 业务结构视图

 2.4 业务行为视图

3. 业务规则

4. 业务模型到软件系统

5. 小结



1 UML的业务建模扩展

- 业务建模的目的就是为了建立**企业核心业务功能**的描述。从而更好地理解业务的关键机制、改进业务、确定新的业务机会、明确外购需求等。

- 对于企业，将所有的信息系统都建立在一个基础业务模型之上，有以下几方面的好处：
 - 为了更好地了解现有业务的关键机制。
 - 信息系统自然成为总体业务的一个内在组成部分，为企业提供业务支持基础并提高工作的效率。
 - 系统之间更易于整合，便于信息的交换和共享。
 - 系统更易于随着业务模型的变更进行相应的升级及修改，这些变更可能来自周围环境变化、业务目标或者业务的改进和革新。这大大减少了信息系统维护和业务过程持续更新的成本。
 - 业务逻辑可以在多个系统之间重用。
 - ...

- 业务模型的风格差距很大...
 - 业务模型永远不可能完全精确或完整，对于相同的业务，不同的人有不同看法，没有一个所谓的完全精确的模型能同时得到大家的一致认同。
 - 一个试图覆盖所有细节的模型，将会面临自身与所描述业务本身同样复杂和难以理解的风险。
 - 由于建模语言以及建模概念本身的局限，也并不是每个细微之处都能够被捕捉到。
- 模型应该主要集中表现业务的核心任务及其关键机制。
- 模型构建者的职责就是洞悉业务核心任务并确定模型内容。

业务模型一般由以下几部分组成：

- ① **视图**：一个业务模型由多个不同的视图表示，每个视图专注于表示业务一个或者多个特定属性。视图是一个特定观点的抽象描述，同时忽略与此无关的诸多细节。
- ② **图表（模型）**：每个视图都由数个图表组成，每个图表都表达了业务结构的特定部分或者特定的业务状态。引入多个图表对于将业务模型可视化是非常必要的，因为每种类型的图表都有不同的目的，表达了业务模型视图的不同方面或机制。一个图表可以表达一个结构或者一些动态的协作。这些图表包含并表述了业务状态中的对象、过程、规则、目标和愿景。
- ③ **对象和过程**：通过使用不同的对象和过程，图表中不同概念相互关联起来。
 - 对象就是业务中的“事物”；它们可能是物理存在的，如人、机器、产品、原料，或者更加抽象，如债务、指示和服务。对象也可以通过包含业务中其他内容的相关信息来描述其他对象。
 - 过程是业务中的活动，这些活动消费、提炼或者使用某些对象去影响或者生产其他对象。

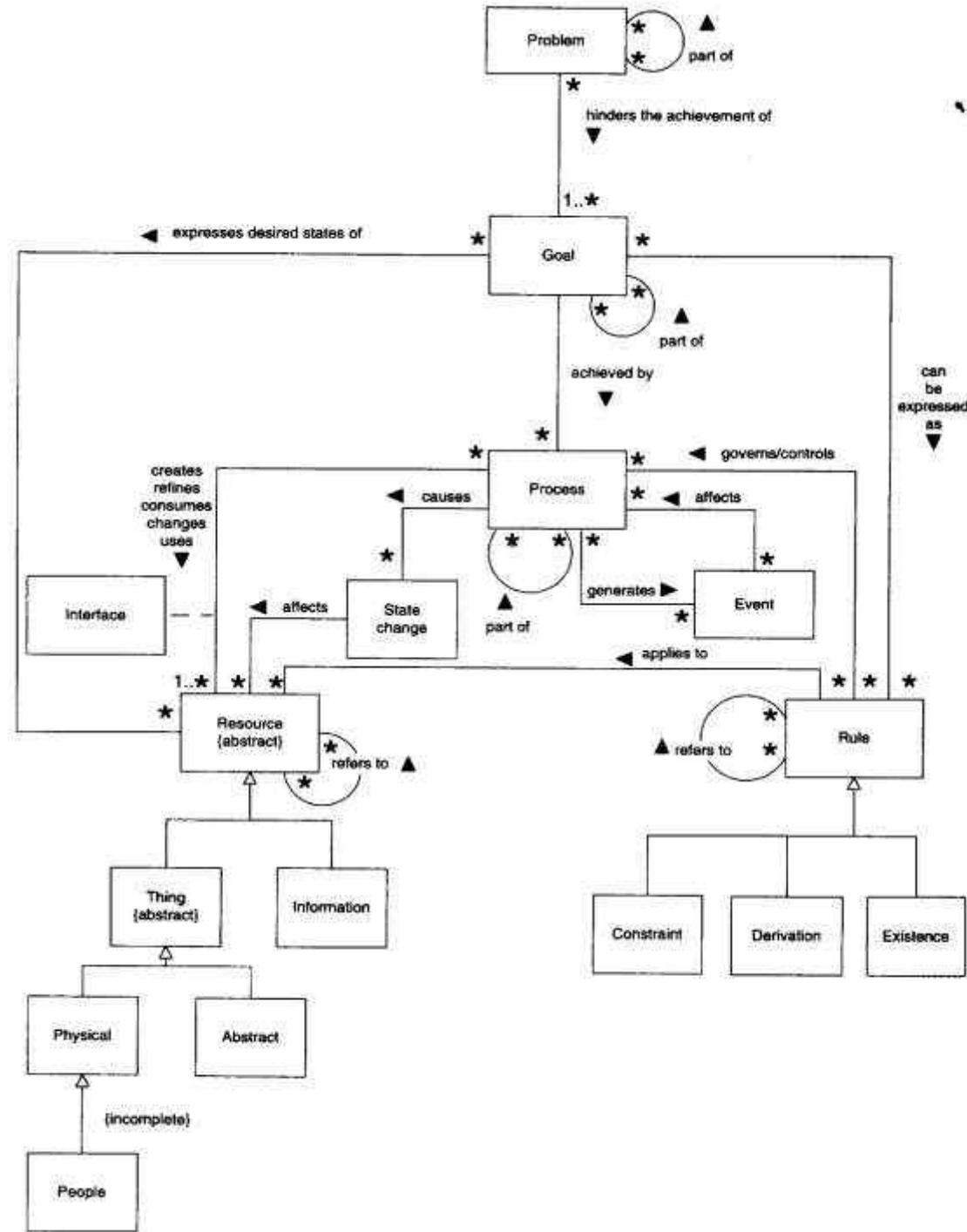
用来定义业务系统的概念有：

- ① **资源** 业务中的对象，如人、原料、信息和产品，这些业务中使用的或者产出的东西。资源放置在各个结构之中并彼此关联，资源在业务中使用的或者产出。资源可以分类为物理、抽象和信息形式。
- ② **过程** 业务资源状态变化过程中所执行的活动。过程描述了业务中的操作如何完成，受规则的支配。
- ③ **目标** 业务的目标，或者业务要实现的整体目标。目标可以被分解为子目标并分派给业务的各个部分，如过程和对象。目标表达了资源的期望状态，这些状态通过各个过程的运作实现，可以以一个或者多个规则表达。
- ④ **规则** 对业务的某些方面进行定义或者约束，并表现业务知识。规则决定业务如何运作或者各种资源怎样进行组织并相互联系。



④ 业务建模概念的元模型

元模型是基本业务概念及其相互间关联的模型。





② 元模型展示了过程如何实现目标。

- 目标确定了一个或多个需要解决的问题，并展示了一项或者多项资源所期望的状态。
- 目标可以以过程控制规则进行表达。
- 过程通过产生或者捕获事件与其他过程进行交互。
- 资源可以是物理形式(如人或者机器)、抽象形式(如账号或者发票)，或者是某种信息，这些信息包含了其他资源的相关数据(如信息系统中的数据库记录)。
- 过程与资源通过一个接口相互作用并导致资源的状态发生改变。

业务建模元模型的构成基础是
过程、目标、资源和规则。

- ④ 业务元模型描述了业务架构，或者说表述了业务建模的**组成要素**，一个架构捕捉了系统中一系列元素，它们之间通过各种**关系**相互连接。
- ④ 元模型归纳了业务建模所使用的概念及其关系。这个面向基础概念的元模型演示了模型在描述复杂结构和关系方面的强大功能。
- ④ 元模型将业务概念图形化，而不是仅仅通过阅读文本描述来记忆它们，这使业务更加易于理解和掌握。
- ④ 这个元模型是一个UML类图，其中每个概念都以一个类的形式进行描述，概念之间的关系或是关联或特化。



UML的业务建模扩展

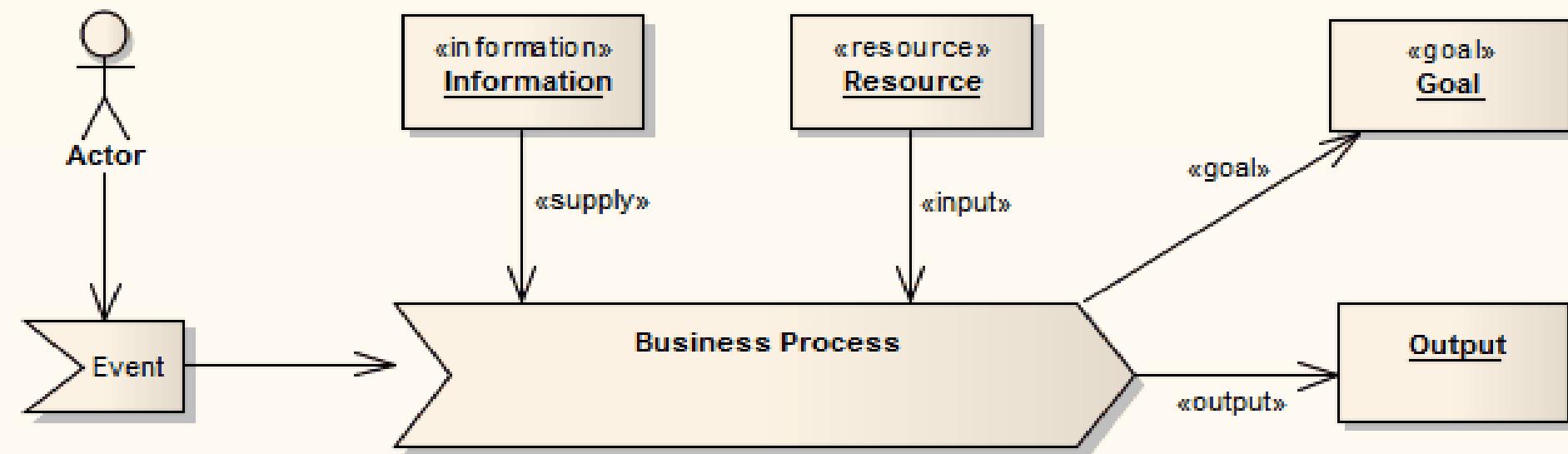
- ④ 业务建模与软件建模处于生命周期的不同阶段，对现实中的业务要素和业务思想进行捕捉，并将其直接转化为程序设计语言很危险，它将扰乱业务建模的重点。这里重点关注**UML创建高层业务模型**，并不是指代码规格说明。
- ④ UML是为软件系统建模而定义的，尽管软件系统和业务系统之间存在许多相似之处，但是它们之间也存在一些差异。
- ④ 标准UML符号和图表已经能满足建模的功能需求。静态图、动态图无论是对于软件系统还是业务系统都同样适用。但业务建模中的一些常用概念却并未包含在标准UML定义中。
- ④ 业务模型可以转变为相应的信息系统模型，但这一步骤是不是业务建模阶段的任务。

UML拥有一套内置的扩展机制，通过这些机制可以在UML中引入并定义业务建模的相关概念。

- ◎ 这些扩展机制内容如下：

- 构造型：作为UML词汇表的一个扩充，可以在现有的构件基础上建立新的构件，并可以拥有自己的图标。
- 属性：UML元素属性的扩展使我们可以在原有元素规格说明中增加新的属性。
- 约束：UML元素语义的一个扩展使我们对现有模型元素的规则进行添加或者修改。

- 基于现有UML建模元素创建了一系列扩展，这些扩展被笼统地称为Eriksson-Penker业务扩展，通过在UML的核心建模元素上扩展来满足业务建模的需要。
- Ehksson-Penker业务扩展为业务建模奠定了基础框架，在这个框架之上根据业务需要添加相应的构造型或者属性。
- 扩展后的UML能与许多常见的通用**业务模式**结合使用。
- 扩展为业务系统的过程、资源、规则和目标建模提供了相应的符号标识。



Eriksson-Penker业务扩展将简易性作为最优先的设计原则。

- **版型** 版型是业务构造型扩展的主要形式，不是给元素增加新的属性或约束，而是直接在已有元素中增加新的语义，这种机制可以看作是已有元素进行专有化。版型的表示方法是在模型元素的旁边添加一个版型的名称，版型名称使用双括号括起来，《Stereotypes》。
- Eriksson-Penker业务扩展扩展中惟一使用到活动图的地方，在这里活动图被命名为装配线图。一个**装配线图**是一个活动图中所展现的过程模型，过程中所使用的各个对象进行链接。这些对象可以是信息系统中的信息对象，也可以是与过程相关的其他一些资源对象。

④ Eriksson-Penker业务扩展

- 这些扩展为UML业务扩展奠定了基础框架，业务架构可以在这个框架之上根据业务需要添加相应的构造型或者属性。
- 扩展不是业务扩展的最终形式，它们作为开发的基础以满足特定建模情况的需要或者进一步改进UML业务建模。
- 这些扩展以标准UML扩展机制定义并与标准UML完全兼容，将UML和过程建模相结合，使得UML业务建模将更加简单。
- 这些扩展并非是试图创造一种新的建模语言。

⑤ Eriksson-Penker业务扩展的相关工具支持包括：

- [sparxsystems\(\[www.sparxsystems.com\]\(http://www.sparxsystems.com\)\)](http://www.sparxsystems.com)的Enterprise Architect
- Qualiware 公司以前是Qualiware CASE
 - QEA (QualiWare Enterprise Architecture)
 - QEM (QualiWare Excellence Manager)
 - QBM (QualiWare Business Modeler)

...



JUDE

File Edit Diagram Alignment View Collaboration Tool Window Help

Man Diagram Search Alias

Structure Inheritance

Eriksson-Penker_english

Activity Diagram

Class 0

Information

People

Physical

Quantitative Goal

Activity Diagram

Namespace

Name Activity Diagram

Frame Visibility

Definition

Activity Diagram

Diagram

Search

Alias

Structure

Inheritance

Activity Diagram

Activity Diagram

Process Goal : Quantitative Goal

<<people>>
People Obj

<<goal>>
Process Goal : Quantitative Goal

<<control>>

<<active>>

<<process>>
Process

<<physical>>
Output Obj

<<physical>>
Input ObjA

<<physical>>
Input ObjB

<<physical>>
Physical Obj

<<supply>>

<<supply>>

<<information>>
Information Obj

Eriksson-Penker Sample
From Business Modeling with UML P.108

15

Close

```
graph TD; subgraph Eriksson_Penker [Eriksson-Penker_english]; subgraph Activity_Diagram [Activity Diagram]; direction TB; PG[Process Goal : Quantitative Goal] -- "<<control>>" --> P[Process]; P -- "<<active>>" --> PG; PA[People Obj] -- "dashed" --> P; IA[Input ObjA] -- "dashed" --> P; IB[Input ObjB] -- "dashed" --> P; PO[Physical Obj] -- "<<supply>>" --> P; IO[Information Obj] -- "<<supply>>" --> P; PO -- "dashed" --> OO[Output Obj]; end; Note[Eriksson-Penker Sample<br/>From Business Modeling with UML P.108]
```

Agenda

1. UML业务建模扩展

2. 业务视图

 2.1 业务愿景视图

 2.2 业务过程视图

 2.3 业务结构视图

 2.4 业务行为视图

3. 业务规则

4. 业务模型到软件系统

5. 小结

A1.信息编码方法

2 业务视图

复杂的业务建模需要使用多个视图。但模型定义中需要达到怎样的细化程度？

- 如果模型描述了业务在较高层次的全局概览，描述可以非常的概括，只需表达出最主要的因素即可。
- 如果模型试图捕获业务中的大量细节，那么它就是一个非常复杂的模型，使用、扩充和阅读也将比较困难。
- 理想方案是介于两个极端之间，模型细节层次取决于目的。
 - 如果此模型将用于构建相应的信息系统，那么使用的信息及其格式就应该作为重点加以强调。
 - 如果此模型将用于业务的改进或者革新，那么就应该突出业务之间的交互作用，同时确定那些可能有助于整体性能提升的改变。

UML业务建模的基础是四种业务视图：

- ④ **业务愿景视图**：业务的整体愿景。描述了公司的目标结构和实现目标所必须解决的问题。
- ④ **业务过程视图**：描述了业务中的活动及其创造的价值，同时也展示了为实现过程的目标，过程之间的相互作用以及涉及到的各种资源，也展示了不同过程间的相互作用。
- ④ **业务结构视图**：业务中各种资源之间的结构关系，业务组织关系或者创建的产品结构。
- ④ **业务行为视图**：业务模型中各个重要资源和过程的独立行为。

业务视图

业务过程 (Business Process) 视图

视图：描述了业务中的活动及其创造的价值，也展示了实现过程的各个目标，过程之间的相互作用以及涉及到的资源，以及不同过程间的相互作用。

业务过程 视图

业务结构 视图

业务行为 (Business Behavior) 视图：业务模型中各个重要资源和过程的独立行为。

业务行为视图

业务愿景 (Business Vision) 视图：业务的整体愿景。描述了公司的目标结构和实现目标所必须解决的问题。

业务结构 (Business Structure) 视图：业务中各种资源之间的结构关系，例如业务组织关系、产品结构。

- **业务愿景视图展示了业务将来发展的策略。**
 - 愿景陈述由一个概念模型和目标/问题视图作为补充说明。
 - 概念模型展示了业务中所有的重要概念及其相互联系。
 - 目标/问题视图展示出愿景陈述中高层目标被分割为各个子目标，也指出了阻碍目标达成的问题所在。
- **业务过程视图是业务建模的核心。**
 - 过程以**过程图**描述，它从本质上而言是一个UML活动图，同时结合了一些构造型以将资源和目标与过程相连接。
 - 业务中的组织关系可以通过泳道在过程的上下文中加以展示。当业务事件成为过程之间的通信渠道时，对应的过程图也包含了业务事件的发送和接收符号。

- **业务结构视图**展示了业务中各种资源的不同结构。
 - 描绘了产品或者服务是怎样由各种资源组合而成，可以是业务中的信息，或者公司的组织，类图和对象图为这些情况分别定义了各自的结构。
- **业务行为视图**描述了独立资源的行为、资源或过程之间的交互作用。
 - 资源的状态、对资源产生影响的事件、在某个特定状态或是当接收到某个特定事件时资源所执行的动作，通过UML状态图进行相应的模型化。
 - 过程内部的交互作用可以由序列图或协作图展示。
 - 而过程之间的交互作用可以在装配线图中展示。

2.1 业务愿景视图

在 "业务愿景" 视图中使用了以下三种技术：

- **策略定义** 通过对当前以及未来一段时间内周边环境的分析而确立的公司定位，以及业务发展的战略目标和必须进行的变更。
 - (1) SWOT矩阵； (2) 愿景陈述
- **概念建模** 业务中使用的重要概念定义及其之间的相互联系。
 - 概念模型；
- **目标/问题建模** 公司目标的定义，其中包含了各个子目标以及实现所有这些目标可能遇到的潜在问题的定义。
 - 目标/问题图表

“业务愿景” (Business Vision)视图描述了公司的目标。

- Business Vision是公司发展方向的蓝图，制定了业务的整体发展策略，定义了业务目标，同时也担当起其他视图建模的参考指南。
 - 视图所包含的许多目标将间接影响着信息系统的功能；
 - 目标已经被分解为多个更加细致的子目标。
 - 这些业务过程致力实现的目标，将指导建模的全过程，并且在信息系统构建过程中扮演着决策参照点的作用。
- “业务愿景” 视图同样用作动机描述工具。可以与业务人员共享，提供相应的信息和资源，同时也赋予相应的职责以实现最后的期望目标。

在创建“业务愿景”视图时，有几个重要因素：

- **公司使命** 公司的最高目标。
- **目标** 更加具体的目标，并且在时间段内可量化。
- **强项** 业务的优势所在。
- **弱点** 业务中需要加以改进的方方面面。
- **机会** 业务未来的发展领域。
- **威胁** 对业务产生负面影响的外部因素(如竞争者、技术变化等)。
- **关键要素** 业务成功和发展的必备因素(如较短上市周期、对新技术的快速适应能力等等)。
- **策略** 如果付诸实施就可以实现目标的行动计划。
- **核心竞争力** 业务中最为关键的部分。
- **角色** 业务中相关人员的特定功能。
- **组织单元** 业务分割而成的各个组群。
- **关键过程** 实现目标的关键步骤。

“业务愿景”视图最终得到的结果就是公司期望的状态定义，以及达到此状态的方法与策略。其结论以一个愿景陈述以及一个或者多个目标/问题模型进行表述。

- **愿景陈述**是一个简短的文本描述，它表述了对公司未来若干年的“愿景”。是高层管理人员的职责。愿景陈述是愿景、目标、问题模型的表述形式，它表述了那些附加于业务之上的具体目标及其子目标的定义。
- **目标/问题模型**作为一个正式的图表，将业务的主体目标划分为若干子目标，并标明为了实现这些目标所需解决的问题。将目标分解并对过程建模是业务架构或者业务过程建模师的职责，而高层管理人员在这里充当了模型支持者的角色，他们不断对雇员强调模型的重要性。

(1) SWOT矩阵

- ④ SWOT (SWOT “威胁”、“机遇”、“弱点”和“强项”)矩阵是用来对企业业务状态进行评估的技术。
 - 威胁和机遇是业务的外部属性，弱点和强项是业务的内部属性。
 - SWOT矩阵用作愿景陈述的编写依据，用文本描述了业务的未来。
 - 矩阵内部以各种策略进行填充以描述业务内外部属性之间的交互(如矩阵的一部分展示了如何充分利用内部力量和外部机遇，另一部分描述了如何处理与外部机遇交互的内部弱点)。
 - 矩阵中所提议的策略是更加正规化策略的基础。如果一个策略在矩阵中不同的地方反复出现，则表明此策略对于业务有着特别重大的意义，重复次数越多则表明越重要。

Sample公司

- Sample公司是一家在Internet上营销并提供金融服务的公司。它使用Internet作为与客户交流的主要渠道。
- 客户使用公司网站上提供的公告板自由讨论不同的股票信息并阅读股票市场的一些相关文章。网页上也显示了当前的股票市场价格，提供上市公司的相关简档，同时也提供了一些针对个人投资者的其他服务。
- 公司充当在线股票经纪人的角色，使客户能够维护股票信息并通过Internet在市场上买卖股票。经纪人服务与其他的更高级服务一样，仅面向那些每月支付服务费的注册客户。其他的服
务，例如公告板，对于访问者都是免费开放的。
- 公司同样也将通过销售网页上的广告获取附加利润。广告客户根据网页的访问量支付一定的费用。



SWOT

矩阵示例

| 业务整体策略 | 内部优势 | 内部弱点 |
|--|---|--|
| 通过提供全面的服务和具有竞争力的价格，引领基于因特网的金融服务 | 1. 对Web设计的良好把握 2. 对Internet程序设计的良好理解 3. 对金融市场和服务的良好理解 4. 与股票代理商的良好关系 | 1. 销售力量不足 2. 金融资源不足 3. 公司知名度不高——没有品牌效应 |
| 外部机遇 | 策略 | 策略 |
| 1. Internet上对于金融服务的浓厚兴趣 2. 市场上没有完备的提供商 3. 低廉的营销渠道 4. 未探查国际市场 5. 对Internet公司的投资兴趣 | 1. 开发应用性好，内容完整的Web站点 2. 瞄准对Internet有投资兴趣的国际客户 | 1. 通过外部代理销售广告 2. 寻找基于版税的转包商 3. 吸引法人股东 4. 吸引国际客户 |
| 外部威胁 | 策略 | 策略 |
| 1. 目前市场上已经存在的同类公司的竞争 2. 可能的经济萧条 3. 客户只会使用免费的服务 4. 在Internet上难以提高知名度 | 1. 为金融站点设定标语和促销活动 2. 在其他网站投放广告 3. 制定价格策略，减少客户的初期投入 | 1. 雇佣好的销售经理 2. 吸引法人股东 3. 启动促销活动加大网站知名度 |

- 通过对Sample公司的SWOT矩阵进行分析，重复出现或者相互之间非常类似的策略往往是最关键的部分。
- 图中可以看出业务策略的三个重要部分：
 - 强调它是一个完整的在线金融服务提供商(矩阵中数次提及这是一个完整的服务策略)。
 - 面向那些想要在美国市场上进行投资的国际客户(国际市场是提到的一个外部机遇)。
 - 争取法人股东以保证充裕的启动资金(必须注明资金不足的弱点)。
- 这三个策略是最具影响的，矩阵中的其余众多策略也应该在愿景陈述中加以引用。这意味着这些目标的方法和途径必须加以确定。
- SWOT矩阵就是在编写愿景陈述之前常用的一项辅助性技术。



(2) 愿景陈述

- 策略在愿景陈述中加以总结，一段文本文档概述了业务的发展方向。
- 它包括了当前业务上下文的描述、业务的需求、可以描述的问题以及可能发生的场景。
- 愿景陈述描述了公司将变成什么样子，它将如何操作，以及将会得到怎样的结果。
- 愿景陈述应包括清晰明确的高层目标，高层目标稍后被分解为若干可操作的目标，这些可操作的目标链接到各个过程，愿景陈述同样给出了判断高层目标是否实现的系统或者途径。
- 愿景陈述常常也会包含过程工作计划，即愿景的相关支持性业务模型如何进行组织的描述。



愿景陈述示例

Bob邮购公司

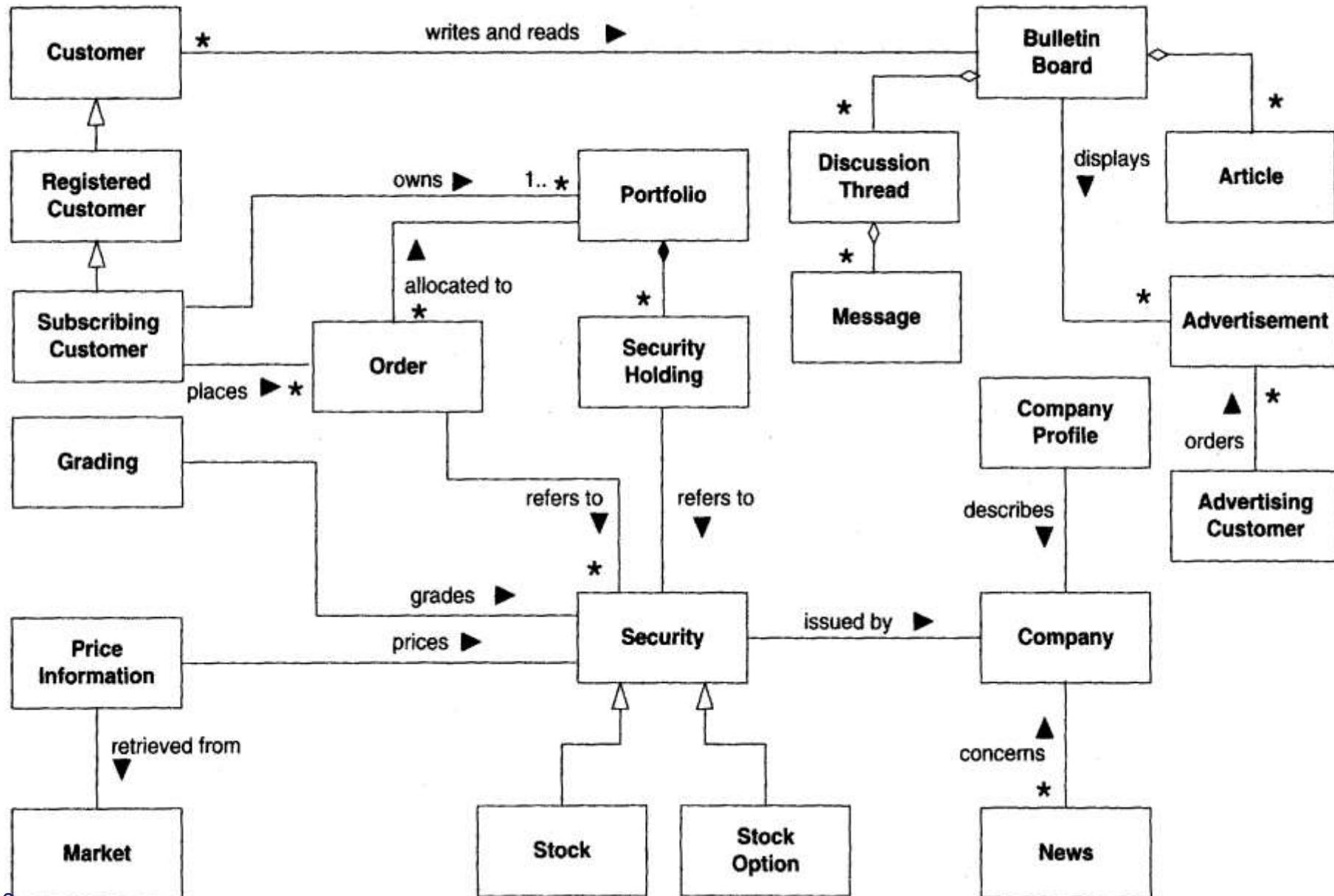
- 我们应该是在线金融服务的主要代理公司。我们应该提供给客户有吸引力的方案并使他们的获得的金融服务物有所值。在证券所交易服务的基础上，我们可以将我们的销售过程和客户的理财服务整合起来。而要想整合这些过程，我们必须提供一些接口，实现因特网、交易平台和电话、传真、邮件等的综合渠道。我们还可以进一步整合，通过提供一些额外的服务，与银行的理财金信息的自动更新和自动购买等。

(3) 概念建模

- ④ 概念模型定义了业务中使用的重要概念，该模型为所有概念建立了通用的词汇表，同时展示了不同概念之间的关系。
- ⑤ 基础概念的清晰定义是理解业务或者进一步建立业务模型的基础。如果没有一个通用词汇表，那么不同的个体就对同一个概念有着不同的理解和阐述。
- ⑥ 概念模型是概念的高层模型，它与信息系统中的数据库设计或者类设计并没有什么关系，即使在概念模型的创建过程中使用了类图并将概念建模为类。



Sample公司的概念模型



Sample业务公司的概念模型

客户被分为以下几种类型：

- 普通客户(Ordinary Customer), 那些访问因特网站点的匿名用户。
- 注册客户(Registered Customer), 那些注册了姓名和电子邮件信息的客户。
- 订购客户(Subscribing Customer), 那些每月支付相应费用以获得更多高级服务的用户。

(4) 目标/问题建模

- ④ 目标模型描述了业务的目标，以及实现目标过程中可能遇到的问题。
- ⑤ 目标控制业务行为，同时也展示业务工作中相关资源的期望状态。
 - 目标模型明确了业务存在的原因，业务所致力实现的目的以及为了实现这些目的将要采取哪些业务策略。
 - 目标模型同样指明了改进业务或者解决目标之间冲突的途径。
 - 通过不断定义子目标，业务中的每个目标都被分解为若干个子目标。
 - 目标由业务过程及相关资源共同实现。
 - 在目标/问题图表中，目标和问题的模型化表达是基于UML对象图和Eriksson-Penker业务扩展的构造型元素。

目标

- ④ 目标描述了一项或者更多资源期望的某种状态。
 - 目标可以针对整个业务，也可以针对独立的业务过程。
 - 目标以标志着某种优化和改进的形式进行规定，如增加产出、提高质量或者缩短产品上市时间，所有这些过程、资源和规则所取得的成效使得目标得以实现。
- ⑤ 业务可以有一个总体的远景规划或目标，目标层次越深入，相应的目标就越具体越量化。目标越具体越量化，就越容易判断目标是否实现。

- ④ 一个目标可以被分解为若干子目标。较高层次目标的实现依赖于各个子目标的实现。
- ④ 子目标同时也可对其他由于某些原因导致实现失败或未能实现的子目标进行替换或补充。
 - 例如，如果高层目标是 "增加15%的利润"，并且三个关于利润增加的子目标中有两个得以实现，即使另外一个没能带来任何利润增加，那么高层目标也可以达成。这两个子目标的成功实现弥补了另外一个目标实现中的失败。
- ④ 将目标分解为子目标的过程称为**目标建模**。目标分解的依据取决于建模者的判断。
 - 例如，销售目标可以按不同的过程或者不同的产品线(或者也可以是两者的组合)将其分割为若干个子目标。

目标总是与各种问题紧密相连。

- ④ 因为问题就是实现目标途中的障碍。问题可以是目标实现途中的紧急状况，也可以是阻碍目标实现的对象实体。
- ④ 因此，通常在目标建模的同时也对其附带的一些问题进行建模。通过目标与问题这两个概念的同时使用，可以更简单地对目标进行描述。
 - 例如，当描述一个目标的时候，有人会问"实现这个目标会遇到那些问题或者障碍"，而又有会问 "如果这个问题得到解决，我们可以实现怎样的目标?"
- ④ 通过定义这些问题，子目标也得以确定，同时也将避免这些问题的出现。此时子目标就可应用于业务过程以排除实现过程中的障碍。

UML中描述目标和问题：

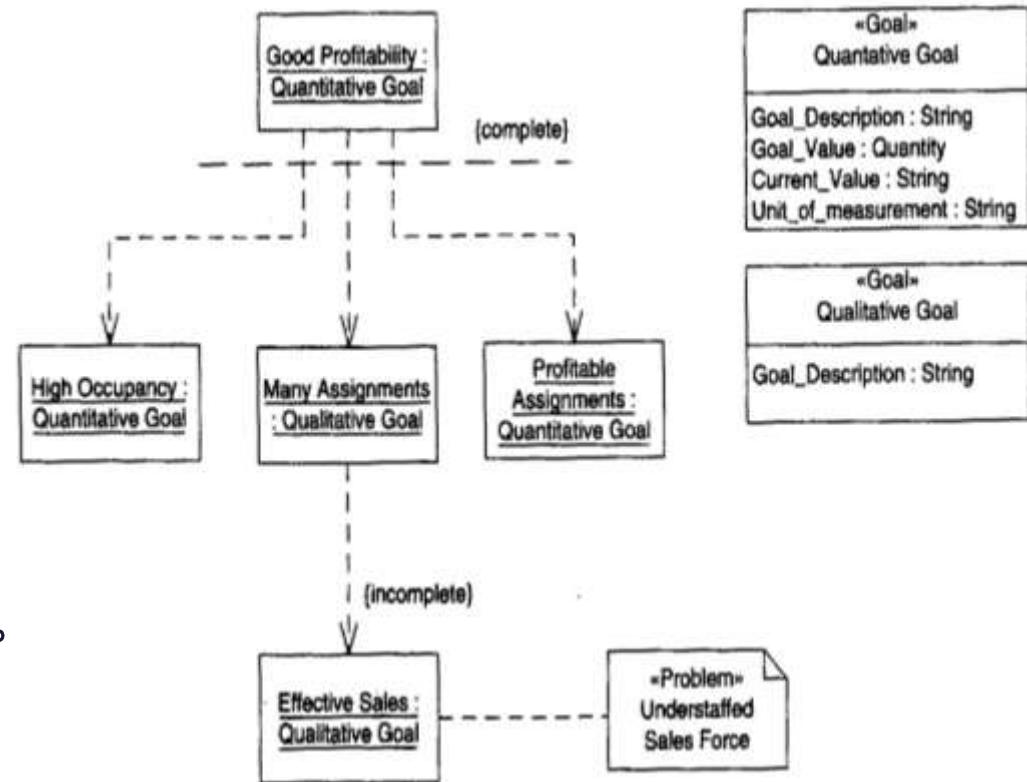
- ④ Eriksson Penker业务扩展中的解决方案是将目标以对象的形式进行展现，同时使用一个对象图以展示目标和子目标之间的依存关系。
- ④ 目标对象以一个被构造为Goal的目标类的目标对象表示在Eriksson-Penker业务扩展中，有两个预定义的目标类：
- ④ 定性目标(Qualitative Goal)和定量目标 (Quantitative Goal)，它们都属于构造型Goal。
- ④ 一个定量目标可以用一个指定计量单位的目标值进行描述(定量)，而定性目标则以一个相对宽松的形式描述，并没有一个特定的目标值(定性)。
 - 必须有一个过程对目标是否实现进行判定。可是，当使用定性目标的时候，相对一个特定的数值而言，过程更加依赖于人为的判断。



- 问题以构造为Problem的注解表示，同时附加文本描述。注解被附加在问题所关联的目标之上。

目标、子目标和问题

- 问题“销售人力不足”进一步指明了一个子目标“加强销售队伍的建设”。
- 当目标及其子目标之间的依赖关系完整无缺时，所有子目标的实现必定保证此目标的实现。
- 如果关系不完整，所有子目标的完成也无法保证高层目标得以实现。



- 规则普遍应用在所有的视图之中，同时也可以与其他规则相连接，因此必须小心避免不同视图中的规则产生冲突。
- 通过定义约束的泛化形式，许多UML图都对规则定义提供了内置的支持。
 - 类图中的关系中包含了结构性约束（如关联的多重性）。
 - 状态图的状态和状态变换中包含了行为约束（如当一个状态发生变换时执行的相应动作）。
 - 派生规则可以被定义为UML图的一个补充约束（如当一个属性的值是由另外一些属性的值通过计算得到）。
 - 活动图中的活动流中包含了行为约束（如某个活动必须在某个活动之前执行）。
- UML中的约束是以紧贴被约束模型元素放置的符号所包围的内容表示。

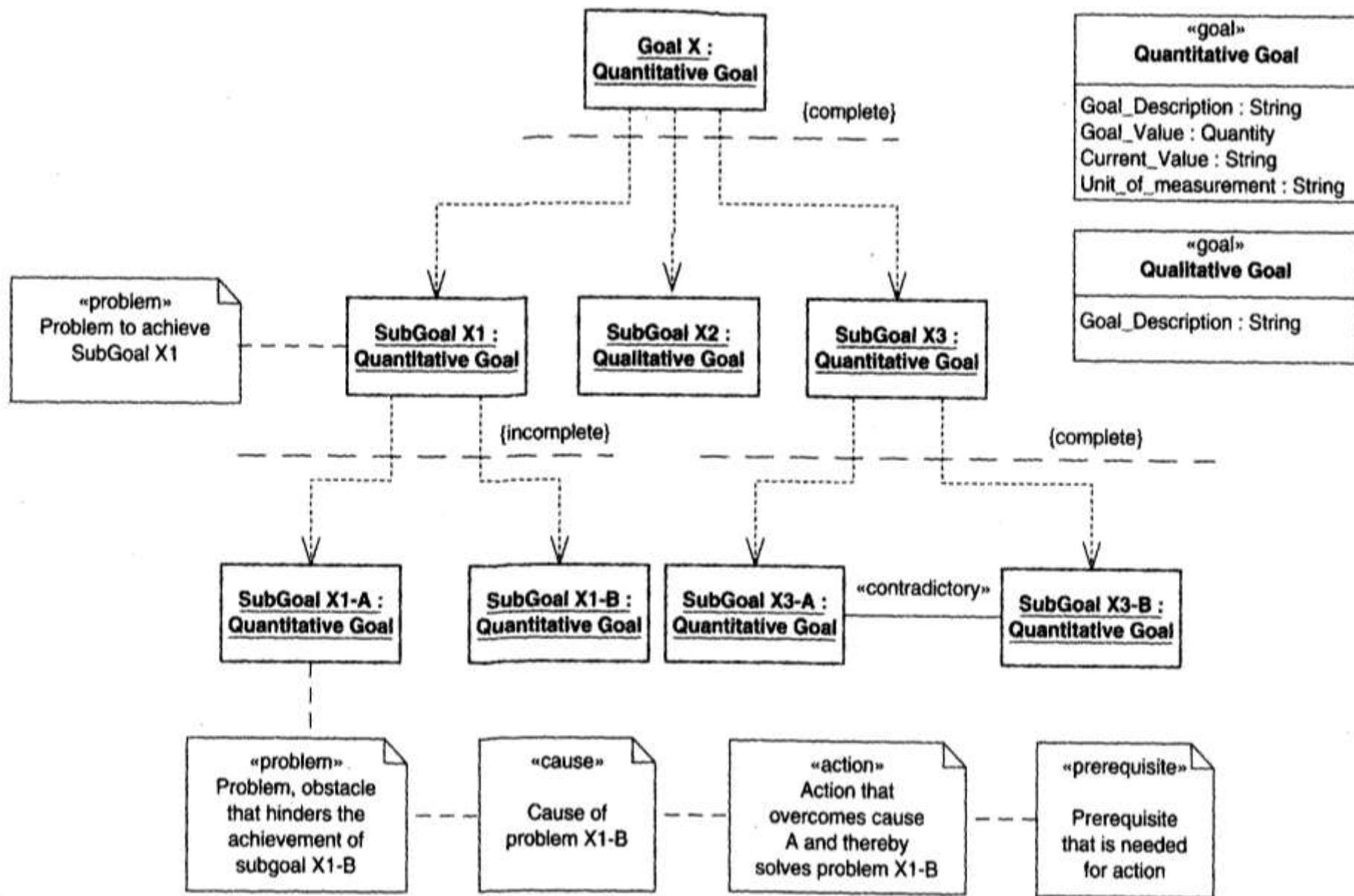


目标/问题图表

- ④ 目标/问题图表是描述对象及其相互关系的UML对象图。
- ④ 完整的目标模型是多个对象图的组合，其中每个图表都展示了某个特定的高层目标如何被分解为各个子目标。
- ④ 目标以类的对象描述，这些对象拥有构造型元素<goal>。在图中这样的对象用于展示目标，目标之间的相互依赖，目标与它们所解决问题之间的关系。
- ④ 目标/问题图表是描述对象及其相互关系的UML对象图。完整的目标模型实际上 is 多个对象图的组合，其中每个图都展示了某个特定的高层目标如何被分解为各个子目标。
- ④ 目标以类的对象描述，这些对象拥有构造型元素<goal>。在图表中这样的对象用于展示目标，目标之间的相互依赖，目标与它们所解决问题之间的关系。



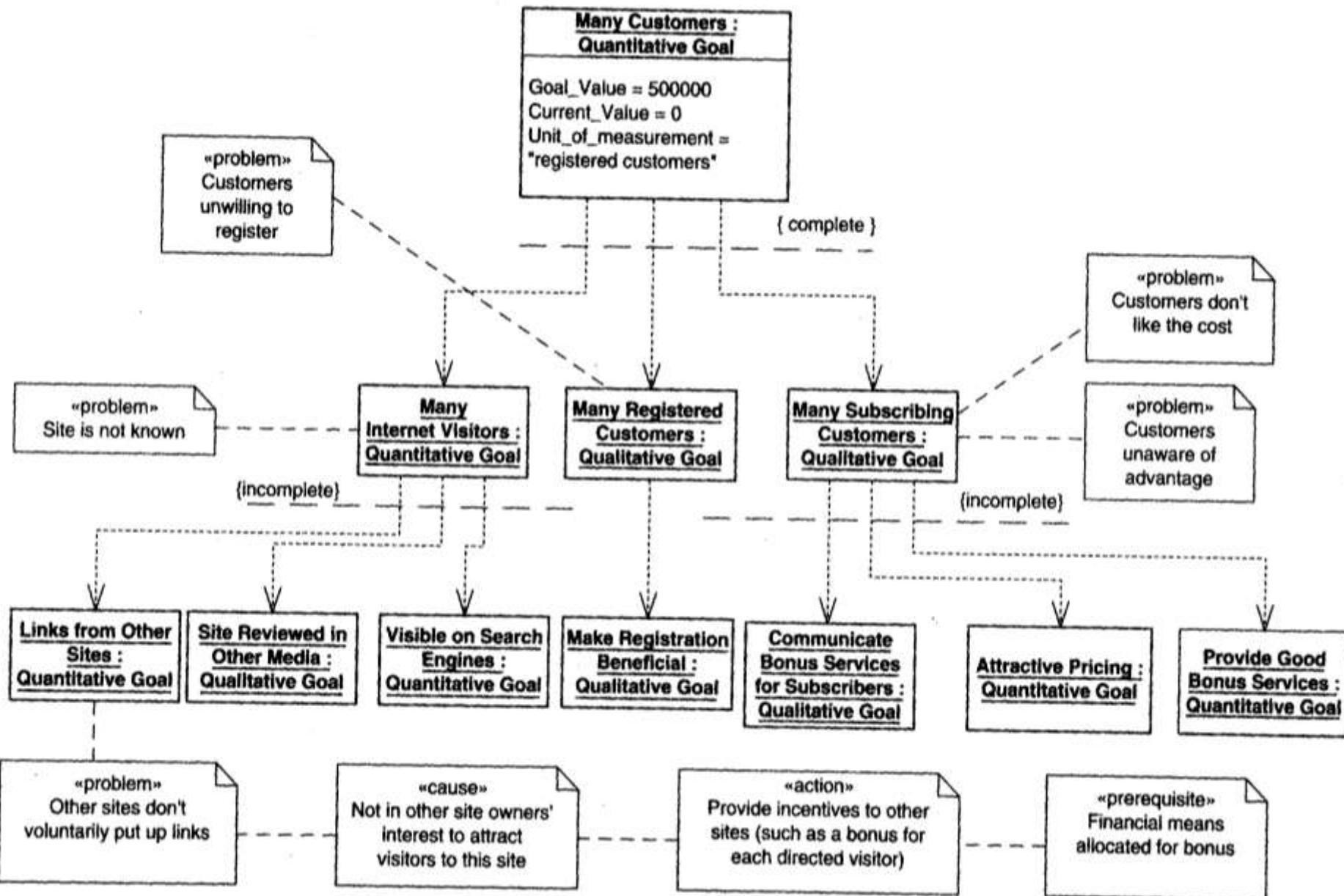
目标/问题图



- ④ Sample业务公司的一个更加具体的目标/问题图表。
 - 此图的最高目标是争取大量的客户。
 - 目标值被定为500000个注册客户（注册姓名和地址的客户）。
- ④ 这个目标被分解为三个子目标，它们也描述了以下几个客户类别：
 - 不知道其名称的因特网访问者。
 - 注册客户，那些注册了姓名和电子邮件信息的客户。
 - 订购客户，那些每月支付相应费用以获得更多高级服务的用户。
- ④ 所有这些都是要争取的客户类型，而业务理念就是通过提供高质量的服务吸引尽可能多的新的订购客户。



目标/问题图示例



- ④ 所有这些动作都已经以子目标的形式建模。图中的目标/问题图表仅仅展示了此业务的主要目标之一。其余的目标则拥有各自的图表。
- ④ 业务的各个主要目标可以在一个图表中加以总结，而此图表中也表现了目标之间的冲突(互斥目标)。
- ④ 每个主要目标则可以通过自身的对应图表，结合相应的子目标进行描述。
 - 例如吸引大量客户的目标可能就与从广告中获取利润的目标相冲突，如果网站上的广告内容过多，那么访问者就会产生反感而不访问这些网页。

2. 2 业务过程视图

“业务过程”视图是业务建模的核心内容。

- 过程展示了那些为了达成目的而必须采取的活动及其在过程中涉及的资源之间的关系。
- 资源包括人员、原料、能源、信息和技术，资源可以在过程中被消费、提炼、创建或者使用。
- 过程及其资源之间，以及不同的过程之间都存在这相互作用，同时多个过程进行耦合以实现预定目标。各个过程都有目的和特定目标，而所有过程的共同努力达成了业务的总体目标。
- 在“业务愿景”视图中所陈述的业务目标是过程建模的基础。现有的过程描述将作为建立新模型的基础。
- 过程建模者协助业务架构师创建业务过程模型。

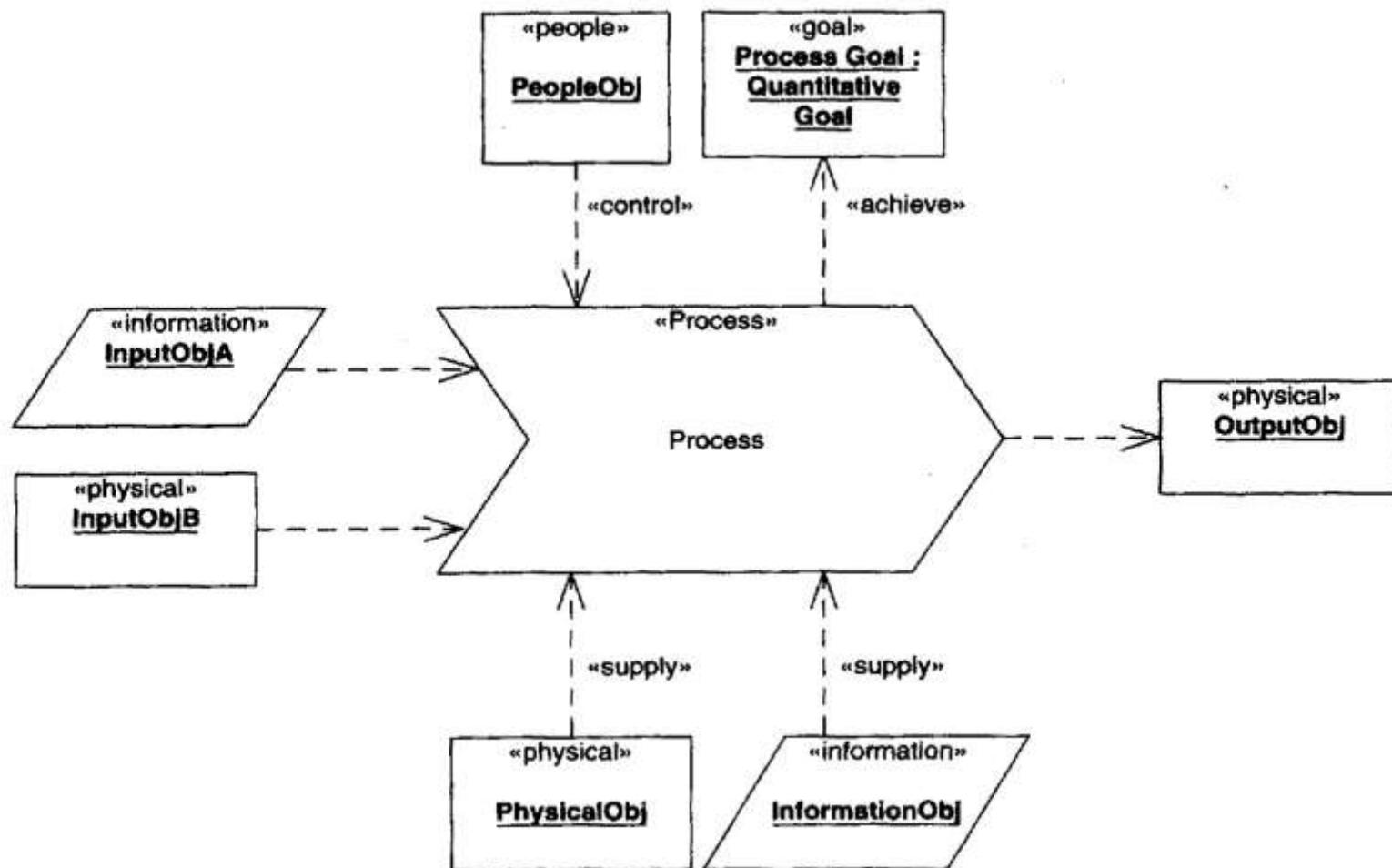
业务过程建模时定义以下内容是很必要的：

- 哪些活动是需要的？在过程图中以过程或者活动的形式说明。
- 活动什么时候得以执行，以怎样的顺序执行？在过程图中通过控制流进行说明。
- 过程的目标是什么？过程图中，通过附加目标对象以及一个将此目标与其他目标相关联的目标图进行说明。
- 活动如何执行？在过程图中，常常通过将过程分解为进一步对活动细节进行描述的子过程来进行说明。
- 活动执行过程中涉及到哪些人和物？这里引用了参与过程的各种资源。
- 消耗了什么，产出了什么？这里引用了过程中所消耗或者产出的资源。
- 活动必须如何执行？通过过程图中的控制流或者通过业务规则进行定义。
- 谁将控制这个过程？这里引用了过程所有者，也就是那些运营业务或者负责业务顺利运行的人。
- 过程如何与业务组织相关联？在过程图中可以通过泳道的使用进行展示。
- 过程如何与其他业务相关联？这可以通过"行为视图"部分中所讲述的交互建模进行展示。



(1) 过程图

- 过程图是附加了一系列构造型的UML活动图，
 - 描述了过程中所执行的活动以及这些活动之间的交互关系、输入输出对象、过程中的资源供给和



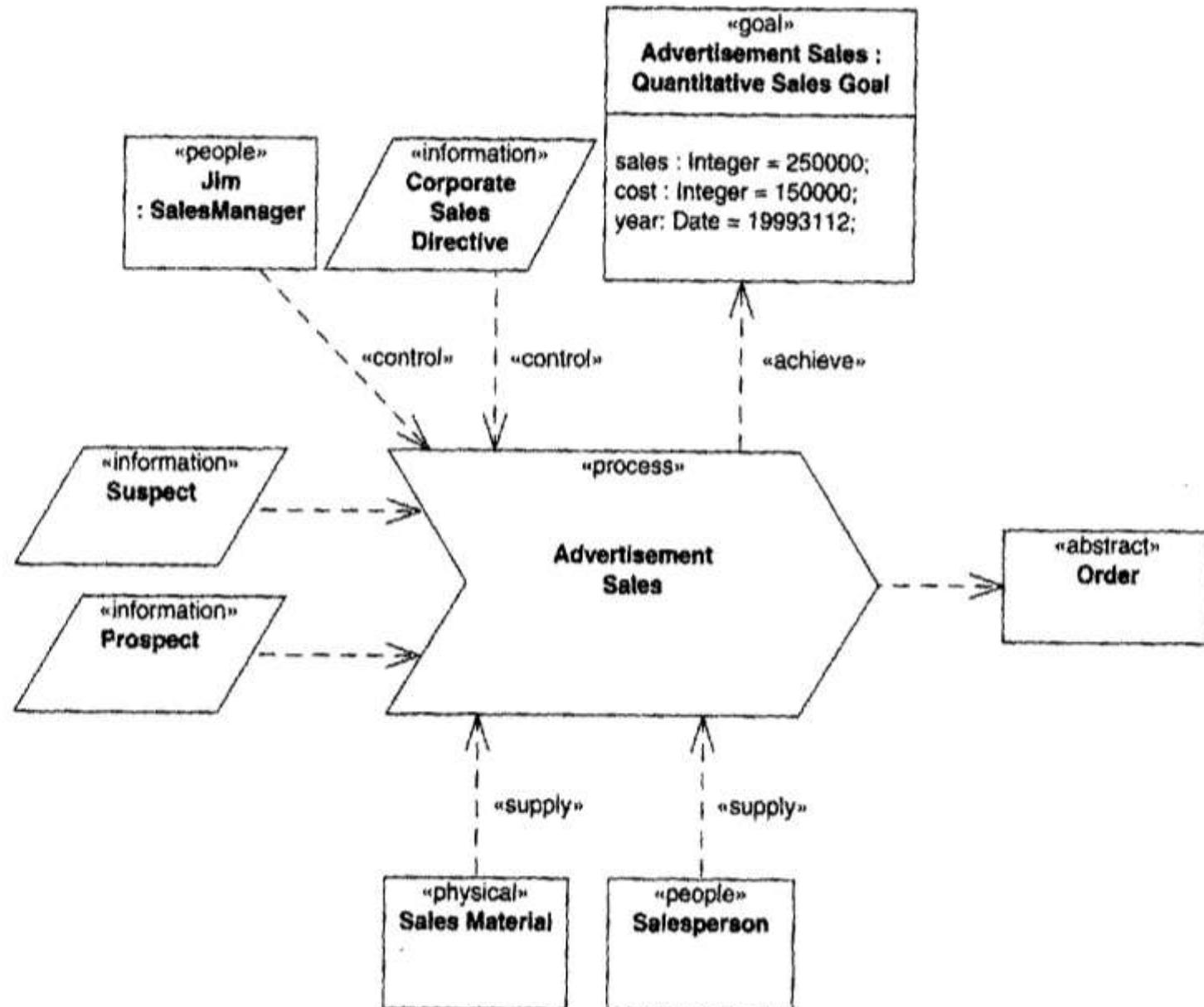


过程图涉及的资源和目标对象分别是：

- ① **目标对象** 目标/问题图中被分配给过程的目标对象。目标对象绘制于过程图之上，并通过构造型为<<achieve>>的依赖关系将过程和目标相连，连接方向为由过程到目标(说明过程要努力实现此目标)。
- ② **输入对象** 过程中消耗的或者加工的各种对象。输入对象是各种资源，同样也可以被构造为<<physical>>、<<abstract>>、<<people>>或者<<information>>。它们以一条虚线连接到过程(即输入对象在活动图中的标记)，输入对象通常放置于过程的左侧。
- ③ **输出对象** 过程产出的对象或者由一个或者多个输入对象经过加工得到的输出对象。输出对象同样也是资源，它们在自过程到输出对象的方向上通过虚线与过程相连接。输出对象被放置于过程的右侧。
- ④ **供给对象** 过程中那些并没有被消耗或者改变的资源。在过程的下方，这些对象以依赖关系(一条虚线)连接到相应的过程。这里的依赖关系被构造为<<supply>>。
- ⑤ **控制对象** 控制或者运行过程的资源。此类对象通常绘制于过程的上方，通过一条虚线连接到过程。这里的依赖关系的构造型为<<control>>。



过程图的示例



示例：广告销售过程

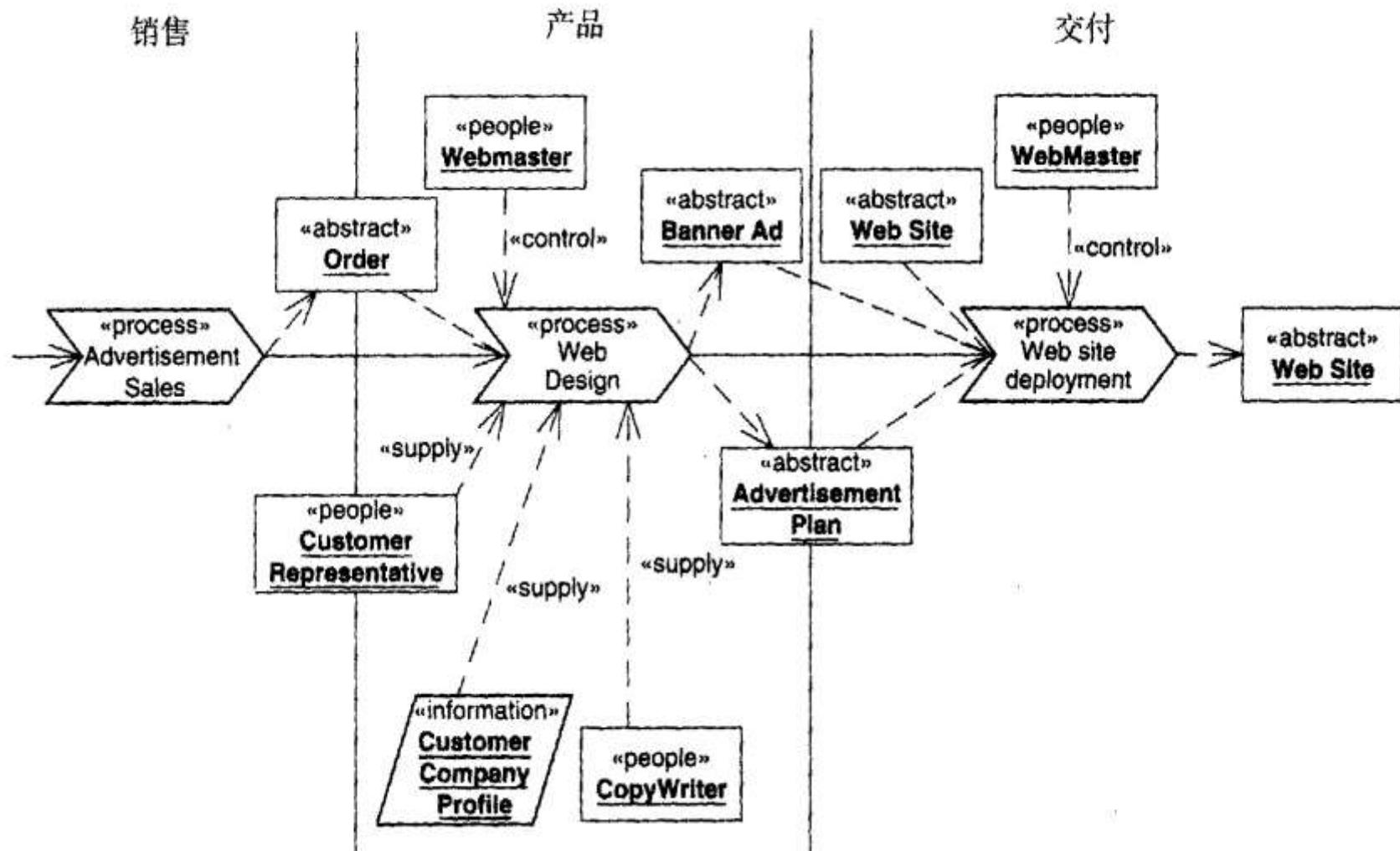
示例展示了销售网页上广告空间的广告销售过程。

- 它拥有一个特定的定量目标，此目标包含了销售总额、成本总额以及年度预算。
- 左侧是输入对象：信息资源猜测 (Suspect) 和预期 (Prospect)。猜测是愿意做广告的公司的相关信息。期望则是那些已经表示对广告有购买兴趣的公司的相关信息。
- 销售过程的成功就是订单 (Order)。订单是抽象资源，因为它是客户与 Sample 业务公司之间的合约。
- 参与销售过程的是销售人员 (SalesPerson) 和销售物资 (SalesMaterial)。
- 销售过程由销售经理和销售条款所控制，销售条款通常是一本指导手册，说明在公司中应如何指导销售工作。



组织界限

加入了泳道的过程图

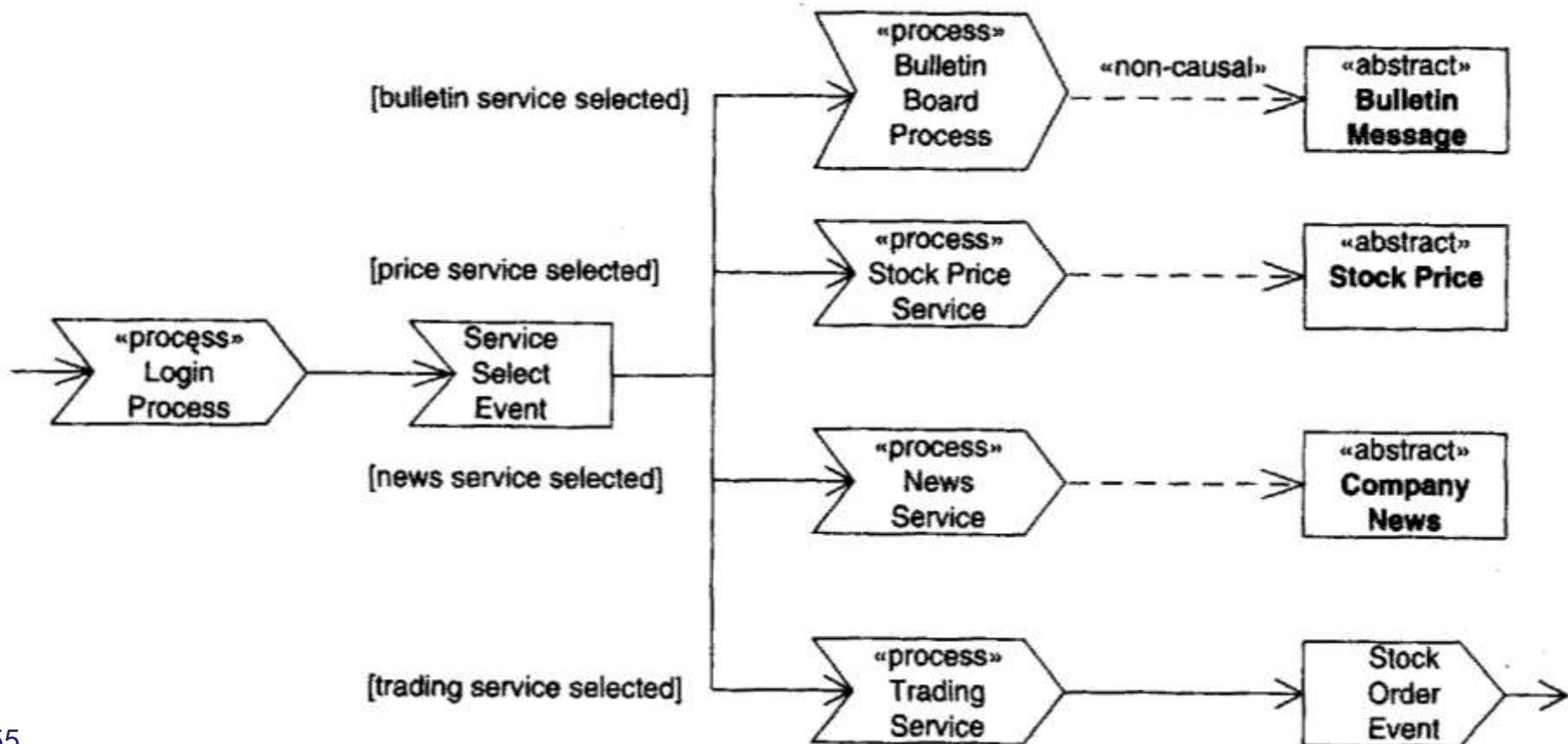


- ④ 泳道用于描述活动在业务组织在哪里执行。
- ④ 除了机构组织之外，泳道同样可以展示对象属于哪个活动或者过程。例如，它可以表示哪台机器或者负责特定活动的人员这些过程的执行跨越了多个组织单元。
- ④ 在UML规格说明中泳道应该位于垂直方向，但是在绘制大型过程图时，水平绘制可能更加实际一些。泳道使图表可以表示出过程之间的整合或者过程如何在若干组织之间划分。
- ④ 通过展示过程中的输入和相关对象，可以展现过程的输出对象如何用作另外一个过程的输入对象或者供给对象。
- ④ 结合泳道的过程图可以展示出活动的输入和输出对象以及活动正在由哪个对象所执行。



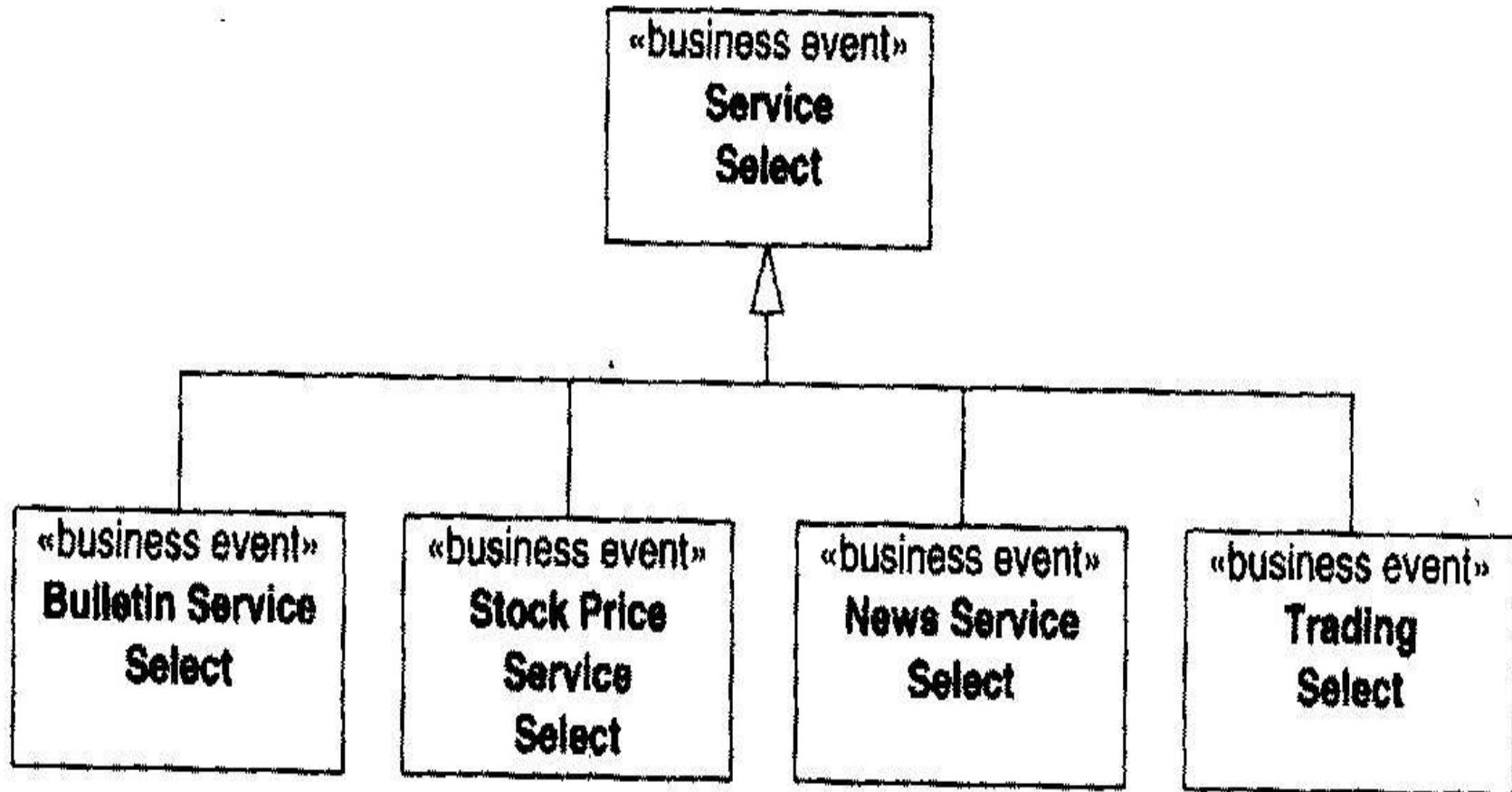
过程的连接

- 过程图中可以运用的另一种技术就是将过程所接收和产生的业务事件进行展示。
 - 客户登录过程的事件描述的过程图。登录之后，过程即开始等待来自客户的服务选择事件。





④ 业务事件的类层次



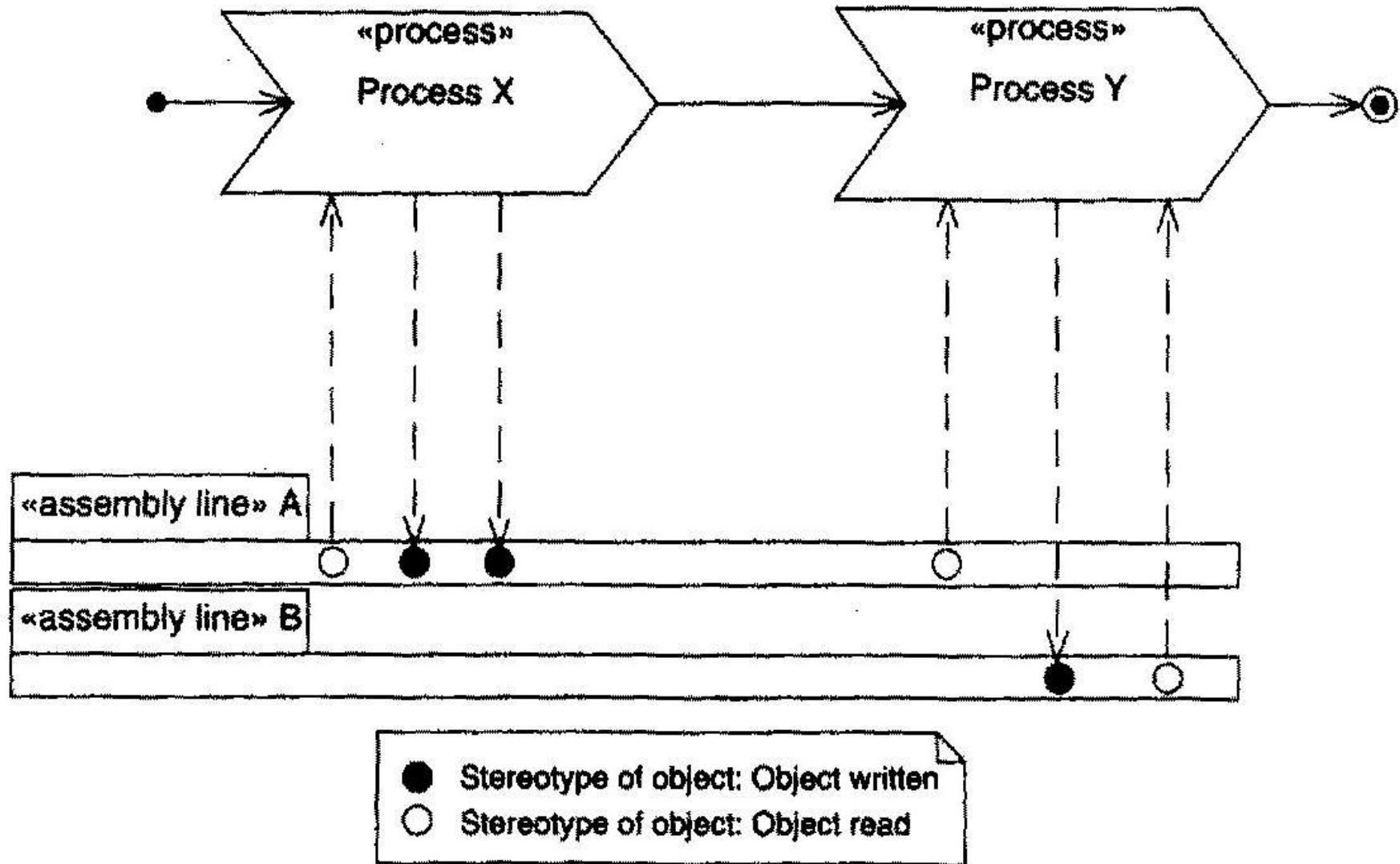


(2) 装配线图

- ④ 装配线图是Eriksson-Penker业务扩展中惟一的图表。与过程图类似，它基于对UML活动图的大量扩展。
- ⑤ 装配线已经成功地运用于过程建模，尤其是当建模的意图是出于对业务支持信息系统的构建。
- ⑥ 因为它将各个过程放置在一条装配线上，因此称之为装配线图。



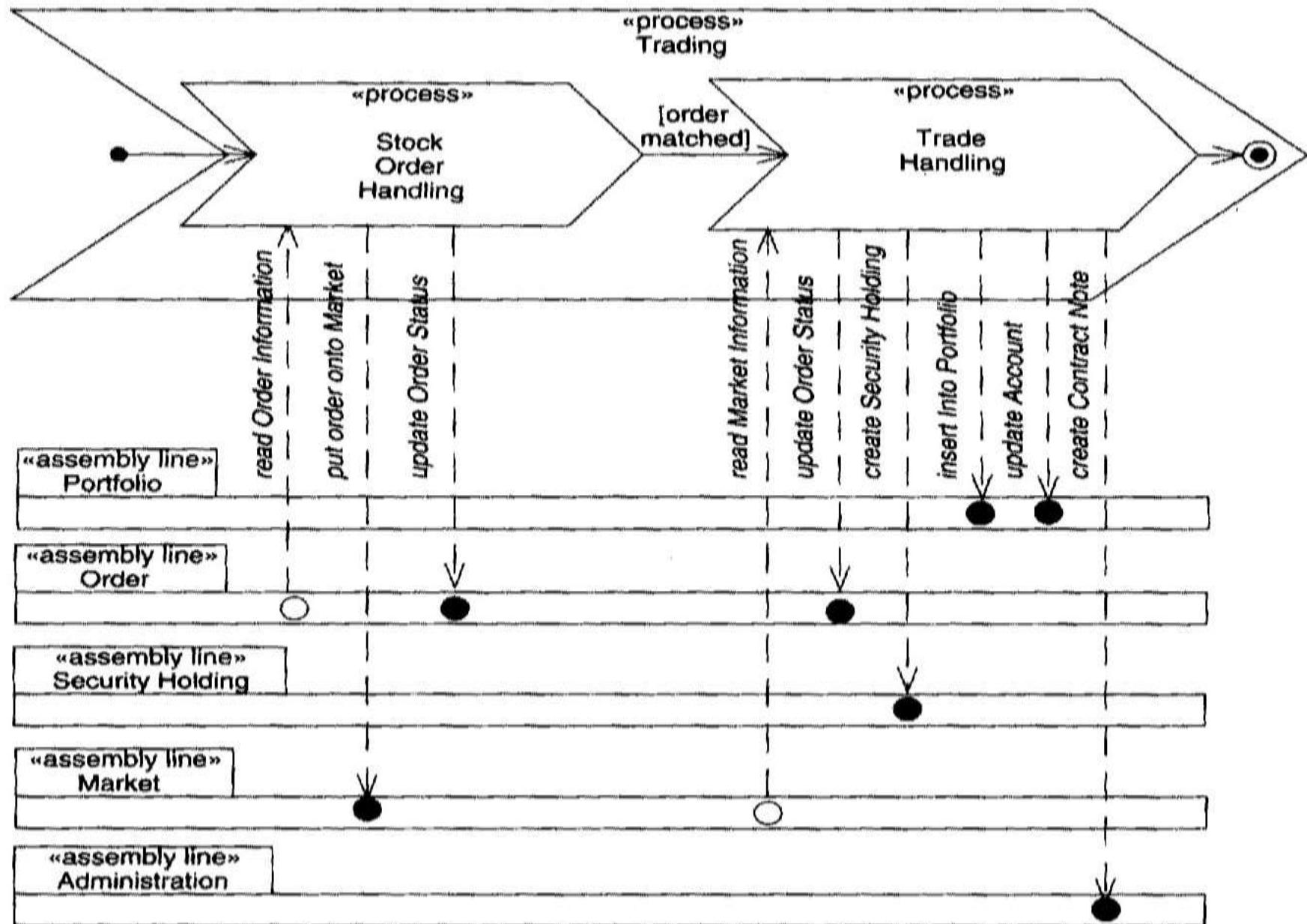
装配线图



- ④ 在装配线图的顶端是一个过程图。在过程图下方是数个水平放置的包，这些包被称为**装配线包**，每个装配线包都代表一组对象(包中的对象可以属于某个特定类也可以属于不同的类)。
- ⑤ 装配线包被构造为<<assemble>>并绘制为一个长的水平矩形，此图演示图中上部的各个过程如何对装配线上的对象进行读写。
- ⑥ 过程到装配线包的**引用关系**通过过程和装配线包内的对象之目的虚线标明(对象流)。在此对象上执行的操作类型在沿着虚线的位置标明 (作为对象流的名称)。
- ⑦ 图中装配线包的引用关系序列按照**从左向右**的次序进行阅读。



装配线图示例





装配线图示例说明

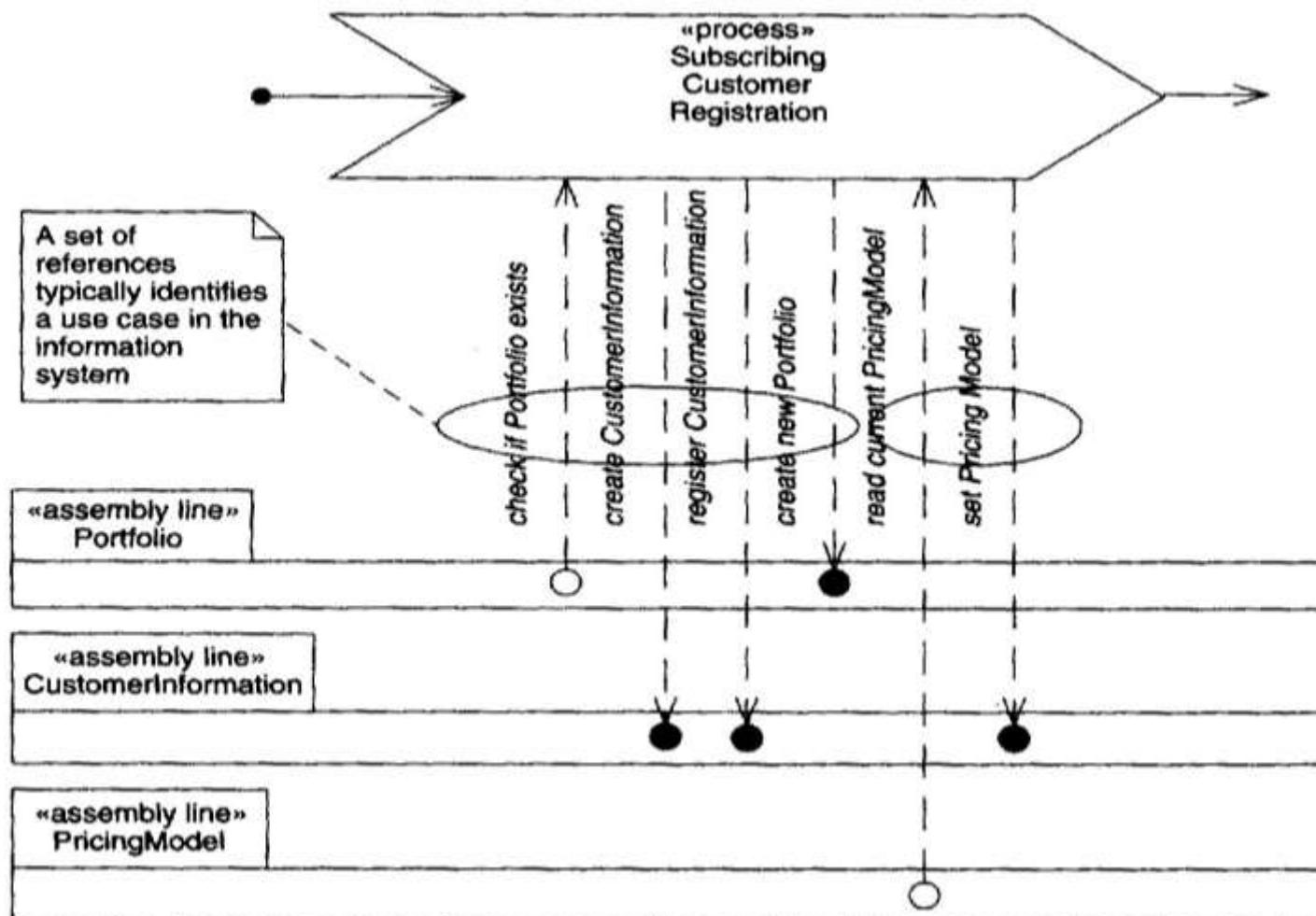
- 装配线图示例以装配线上的一系列资源展示了股票订单处理(Stock Order Handling)以及交易处理(Trade Handling)过程及其交互作用。
- 本例中有三个装配线包，分别对应投资组合(Portfolio)、订单(Order)和证券持有(Security Holding)；装配线包市场(Market)和行政管理(Administration)用于对组织中的市场和行政管理进行划分。
 - 例如，交易处理过程在行政管理包中创建了合约注解(Contact Note)作为它与装配线的最后引用，这里的合约注解(买卖双方作为交易凭证的文档)是行政管理对象，也就是说此对象位于行政管理包中。



(3) 装配线图和用例

- 装配线图中所有装配线包都是信息系统对象。
 - 此图展示了业务过程与信息系统之间的交互关系。
 - 连接到装配线包的引用关系中包含了信息系统的输入输出信息流，同时也表明了业务过程以及信息系统之间的接口。此接口以面向对象建模技术中的用例来描述；
 - 装配线图中的一系列引用关系也同时转变为信息系统中必须提供的用例。
- 这一点非常重要，因为它将业务过程映射到系统用例，而系统用例描述了信息系统的功能需求；它同时也确定了系统用例中的参与者(由使用装配线的过程扮演的角色)。
- 装配线图在业务建模和软件系统需求之间建立了沟通的桥梁。

- 定义信息系统功能需求时，装配线包的引用关系映射为系统用例。





2. 3 业务结构视图

- ④ 业务结构视图展示了业务中的资源、产品和信息的结构，也包括传统的公司组织。
- ④ 业务结构视图的基础是传统的组织图及其描述，以及公司的产品或者服务的相关描述。
- ④ 业务结构视图被看作是过程视图的补充，它描述了那些在过程图中并没有表述但对公司的运作至关重要的信息。
- ④ 来自过程视图的信息也将得到应用，因为它表述了过程所用的资源。
- ④ 模型同样是由业务架构师所构建，同样也可能会有一组过程模型构建师对其进行支持和维护。
- ④ 注意，一般情况下，这两种视图的建模工作将并行展开，因为它们相互促进并且必须保持一致。很难将这两种视图中的某个完全完成之后再转移到另一个视图的建模工作，因为它们的开发过程是相互作用的。

(1) 资源

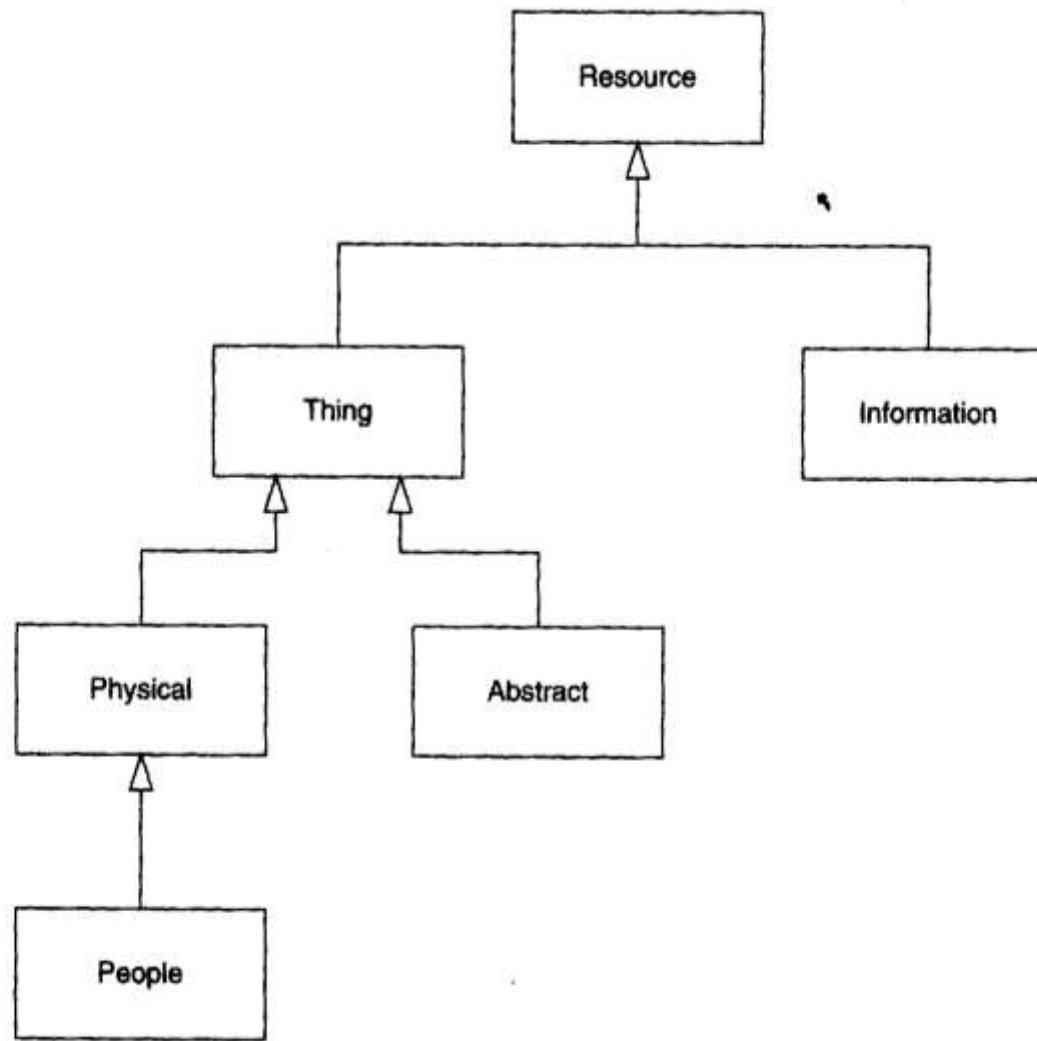
- 资源就是参与业务或者为业务所用的对象，是一些被业务过程消耗、产出、转化，或使用的概念。例如，原料、能源、产品、人、信息和服务。在面向对象建模中建模为类的元素通常可看作是业务建模中的资源。
- 关于资源的定义有：
 - 一个资源即一个实体，此实体在某个类别的任务中扮演某一个角色 [Vernadat 1996]。
 - 资源是业务中使用的一个概念，表示所有我们选择的作为整体来评估的事物。
- 资源类型以类的形式进行描述，资源实例则以对象的形式表示。

- ④ 资源模型展示了各种资源的结构。
- ④ 泛化的模型将以类图的形式描述，而其实际的结构配置信息则在对象图中展示。
- ④ 那些通常是作为公司所提供的产品或服务的资源的内部结构可以通过资源模型加以描绘。
- ④ 资源模型和概念模型之间的差异在于资源模型集中于更加具体的资源结构，如产品或者服务，而概念视图集中于业务定义时所用到的重要概念的含义和联系。

四种具体的资源类型：

- **物理的** 占据一定空间的物质实体。日用品、原材料、零件或者过程中使用的产品都是物理资源的实例。一个物理对象往往是基于其他不同的物理对象所构建的。
- **抽象的** 一种思想或者概念，往往来源于其他对象的合体(如一份定单是与一系列选购的商品所联系的)。此类事物或者概念并非物理存在，但是对于业务而言却很重要。合同、角色、账户以及能源都是抽象资源的一些例子。
- **人** 业务中参与的人员。这是物理资源的一个特例以突出过程中所涉及的人员。在元模型中用一个专有的类来对人进行描述。人将被看作是一种特殊的物理资源。
- **信息对象** 某个概念、事物或者其他信息对象的表现。它包含了其他资源的相关信息，信息对象保存了业务中其他对象的实际情况或者相关知识。

不同资源类型之间层次关系的元模型





- 不不同类型资源的一些例子，这里不同的类被构造为各自相应的类别。

«Abstract»
Stock
Option

«Physical»
Drill

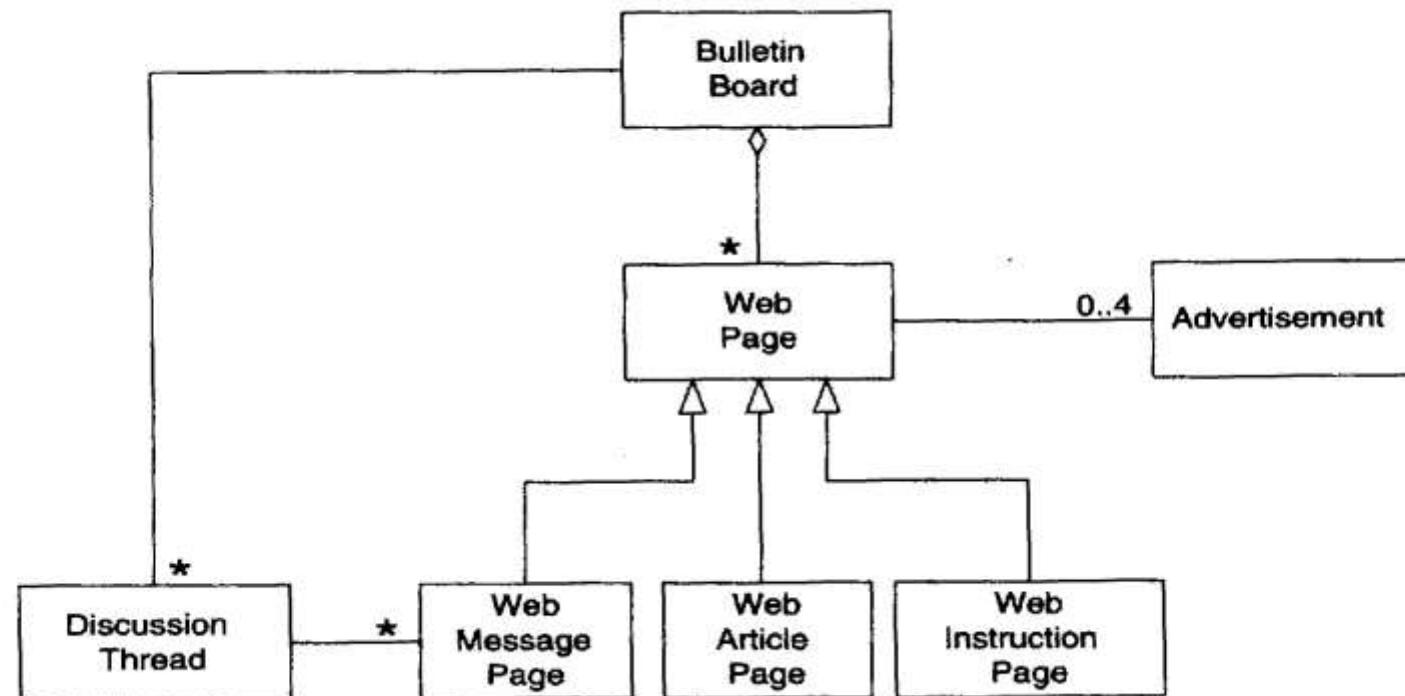
«People»
Trader

«Information»
Business
News

- 一般情况下，资源之间有定义良好的结构和关系。
 - 业务所产出的产品将是某些其他组成的聚合(如一个物理资源基于其他物理资源而创建);
 - 业务中的人以功能性机构的方式组织，例如一个部门 (如一个人力资源是一个抽象资源，正如一个组织单元的一部分)，客户与抽象对象(如订单和合约)之间同时存在着许多关系，这些关系由业务来处理。

Sample业务公司公告板资源结构的类图。

- 公告板由包含了消息(WebMessagePage)、文章(ArticlePage)或者相关说明(WebInstructionPage)的不同网页组成。
- 所有的消息都以不同的讨论主线(DiscussionThread)组织。而一个网页最多可以有4个广告(Advertisement)。



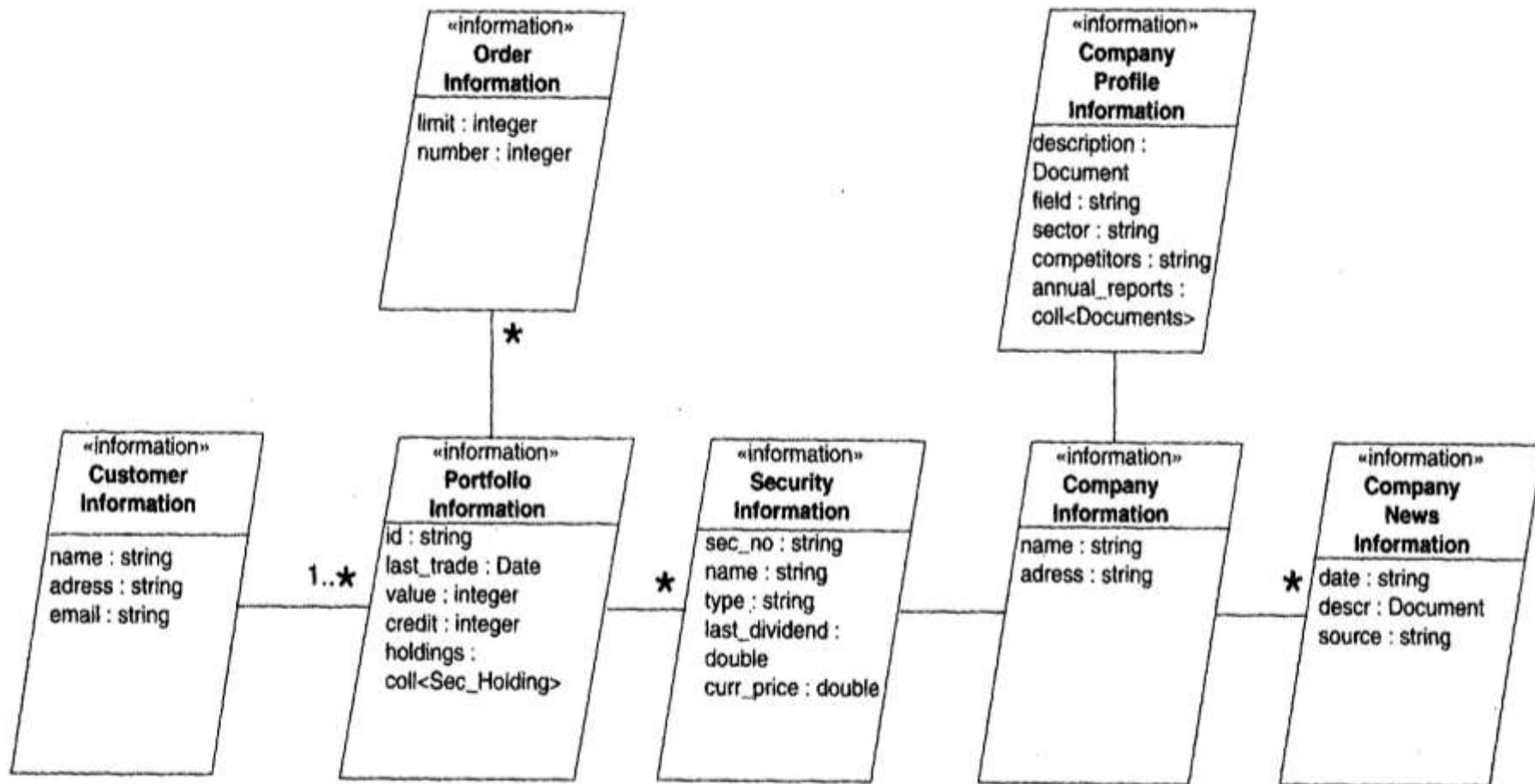
(2) 信息建模

- 信息建模为业务中具有战略意义的重要信息建立相应的模型。尽管信息同样也是资源的一种，但将它从其他资源中分离出来并为其建立独立的模型仍然是非常有意义的。
- 信息在信息系统中对业务进行支持，它对业务具有重要的战略价值。
- 信息建模是定义信息系统中存储信息的早期步骤，尽管一些细节如数据库设计问题是业务模型的一部分，这样的细节信息将在软件信息系统建模过程中加以定义。
- 业务决定了信息需求，但有时一些有效的信息也会反作用于业务，例如，业务可以为自身创造新的商机、业务可以捕捉到更多的客户信息、业务对其产品和服务的改造和配置可以具有更大的灵活性。



信息类图示例

- 类图包含了那些用于描述Sample业务公司最重要的信息资源的类。请注意，业务模型中可以同时包含客户(Customer)以及客户信息(Customer information)所对应的类。





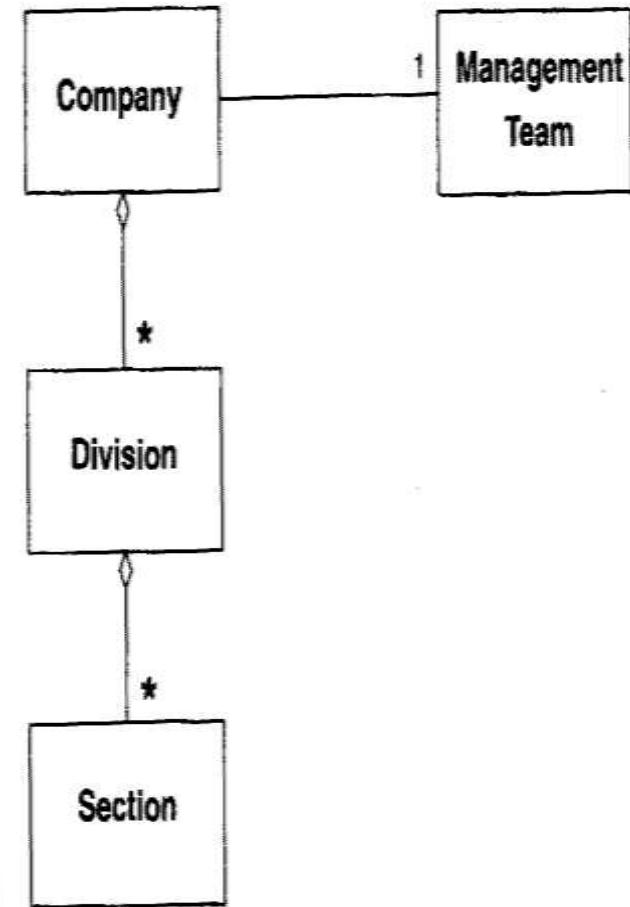
(3) 组织建模

- 组织建模是资源建模的另一个特例，在组织建模中，资源以特定的规则分配到各个相互关联的组织单元。组织中被分配的资源包括雇员、机器和场所。
- 过程同样也可以被分配到组织中去，而组织中的资源也可以被分配到过程。
- 组织应该尽最大努力优化业务中各种资源的使用，同时努力避免内部的次优化，但是，这在许多组织中都是很普遍的现象。
- 组织的基本功能就是展示**资源分配情况、任务分配情况以及公司的管理模式**。
- 组织可以包括数个维度，如组织单元、地理位置以及分属的业务过程。



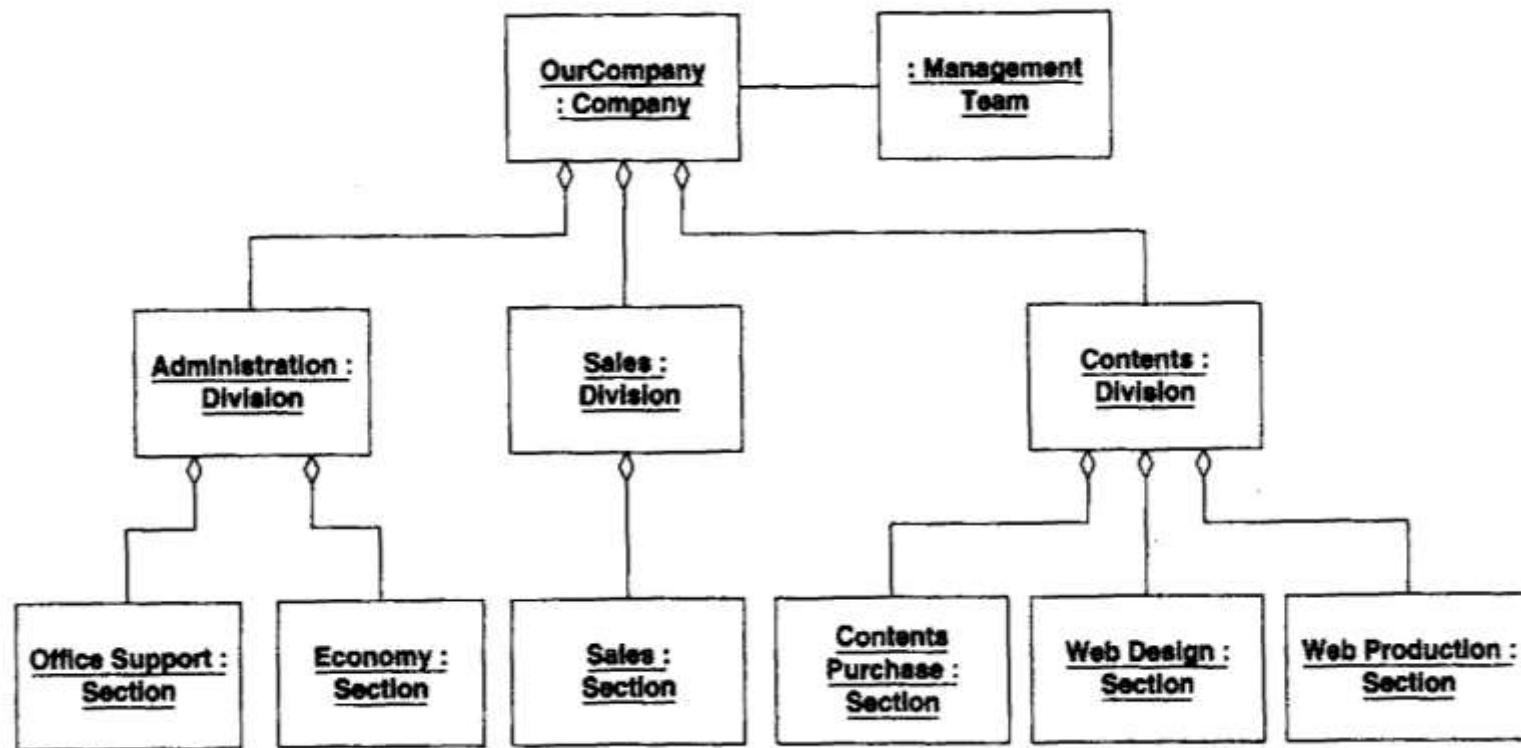
- 组织的全局结构通过类图和对象图来表示。类图指明了组织的基础结构和规则，而对象图展示了当前组织的实际形式。

- 类图设计的越灵活越精良，对组织结构的改动也就越容易，组织的变动可以得以隔离从而避免对业务模型或其支持信息系统产生连带性影响。
- 类图描述了组织单元的名称及其相互关联的业务规则。公司组织所对应的类图，此公司拥有一个被分化为数个部门的管理团队。
- 资源通过对象图进行分配，对象图中的各种资源被链接到不同的组织对象。





- 系统可允许对部门或者小组进行增删；但不允许对实际结构进行变更（例如：为某个组添加次级或者加入新的组织概念）



2.4 业务行为视图

- ④ 业务行为视图展示了业务中相关资源和过程的独立行为，以及资源和过程之间的交互关系。
 - 业务过程视图决定了资源对象的单独的行为，展示了业务运作中的整体控制流程。
 - 业务行为视图更加细致地观察了业务所涉及的各个对象：它们的状态、每个状态下各自的行为和可能的状态变换。
 - 行为视图也展示了不同过程之间的交互作用，如它们以何种方式保持同步。
- ⑤ 通过这种方式，行为视图成为众多活动职责划分以及定义过程中各种资源行为的重要工具。

业务行为视图和业务过程视图的差异：

- ④ **过程视图**通过业务中的资源、目标和规则之间的交互关系展示系统的行为、转变和功能。
- ④ **行为视图**则展示这些活动所涉及的各个动态行为。某些活动在更加细致的层次上得以描述，而同时对过程视图中没有表现出来的交互关系和职责也进行了相应的定义。
- ④ 这两种视图之间自然必须保持一致。
- ④ 业务行为视图通常是由过程建模者在业务架构师的协助下创建，这里业务架构师保证了这些模型与业务过程视图相一致。

- 各个过程和对象的组合状态定义了当前的系统环境。
- 状态通过系统操作而变化，也就是说，通过各个过程的运作发生改变。实际上是由业务中的各种资源完成了业务功能；过程所做的仅仅是驱动或调整众多资源的运作。
- 当过程状态发生变化，相应的事件就会产生以通知其他过程关于状态变化的信息。
- 状态同样决定了可能要发生的事件，当某个状态发生改变时那些动作得以执行，并决定如何使某个对象转移到某个特定状态(必须触发哪些事件以促成对象状态的改变)。
- 因而状态与动作和事件一样，是行为视图中一个非常重要的组成部分。

- 行为视图通过UML动态图表：状态图、序列图、协作图、过程以及装配线图来定义。

Agenda

1. UML业务建模扩展

2. 业务视图

 2.1 业务愿景视图

 2.2 业务过程视图

 2.3 业务结构视图

 2.4 业务行为视图

3. 业务规则

4. 业务模型到软件系统

5. 小结

3 业务规则

- 业务规则是对业务包括业务工具如何运作的一种需求。它们可能是法规或者表达成业务的架构和风格。
- 业务规则的表达：严格、形式化，从而使得它们能够成为自动化的基础。
 - Object Constraint Language (OCL)
 - 其它自然语言
 - » 结合结构化英语，可以确定一组保留表达式用于定义规则。
 - » IF
 - » ONLY IF
 - » WHEN
 - » THEN
 - » ELSE
 - » IT MUST ALWAYS HOLD THAT 必须总是满足
 - » IS CORRECTLY COMPLETED 被完全实现



3.1 业务规则

- 一个业务模型包含了各种业务规则，它们定义了业务过程如何运作的相关约束、条件和策略。
- 业务规则可以影响所有其他的概念；它们可以约束业务的执行过程、资源的行为，以及达成特定目标所采用的方法和手段。
- 规则往往会影响到各个概念之间的关联、定义概念相互关联的途径以及特定时刻保证关联有效的前提条件。

下面是业务规则的一个定义：

- 可以控制或影响业务执行过程以及业务资源结构的语句。此语句指明了必须得到满足的条件，或者此条件决定了下一步哪个活动将得以执行。它可以表达一个业务目标，指明过程执行的方式，可以对一个关联的条件进行具体描述，或者约束某项资源的行为。

④ 三种业务规则：

- **派生** 定义了知识如何从一种形式转化为其他类型的规则，如某些信息是如何从其他信息中推导出来的。推导可以是计算规则，也可以是一个推理规则，此规则意味着如果某个条件为真，那么另外一个推理得到的条件也同样为真。
- **约束** 对对象或者过程可能出现的结构或者行为进行约束的规则。当对象得以创建成其关联发生改变的时候，约束维护了对象的完整性。对象的操作可以用操作预处理和后续处理进行约束。在操作执行之前，预处理操作必须首先执行；当操作执行后，后续处理随后也必须执行。
- **存在性** 定义了事物何时可以出现以及何时应该出现的规则。



业务规则具体类别

- 派生规则：从其它事实推导或计算的条件和策略
 - 推理规则
 - 计算规则
- 约束规则：对对象的结构和行为进行限定的策略和条件。
 - 激励和响应规则：描述何时或者什么条件需要为真，才能触发相应的行为
 - 操作约束规则：确定操作的前提和后续条件以保证操作的正确性
 - 结构约束规则：有关类，对象及其关系的策略和条件。
- 存在性规则：事物何时出现和消失的规则。
 - 存在性规则：一个对象何时创建和销毁。

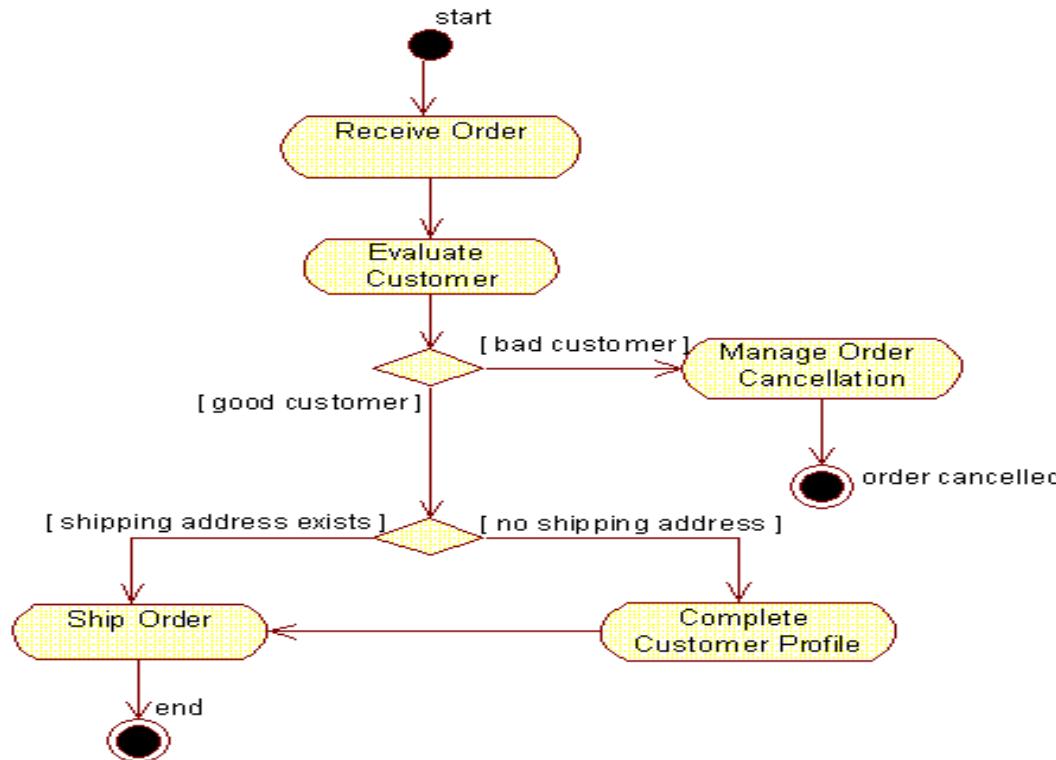
◎ 规则如何体现在模型中

- 业务规则影响了模型。它也影响了在活动图里如何排列你的活动，它甚至影响了业务实体之间的联系。
- 有些规则并不能够很容易地直接翻译成模型的样子，它们可能内含在模型元素中。
- 无论如何，将业务规则作为文本描述总是有用的。

派生规则（1）推理规则

- 推理规则一般看起来像刺激和响应，操作约束或者结构规则。它们的区别在于为了达到结论，需要经过一系列考虑的步骤。
- 规则隐含了工作流中某一活动状态中需要的方法，最终体现为业务工作者或者业务实体的操作。

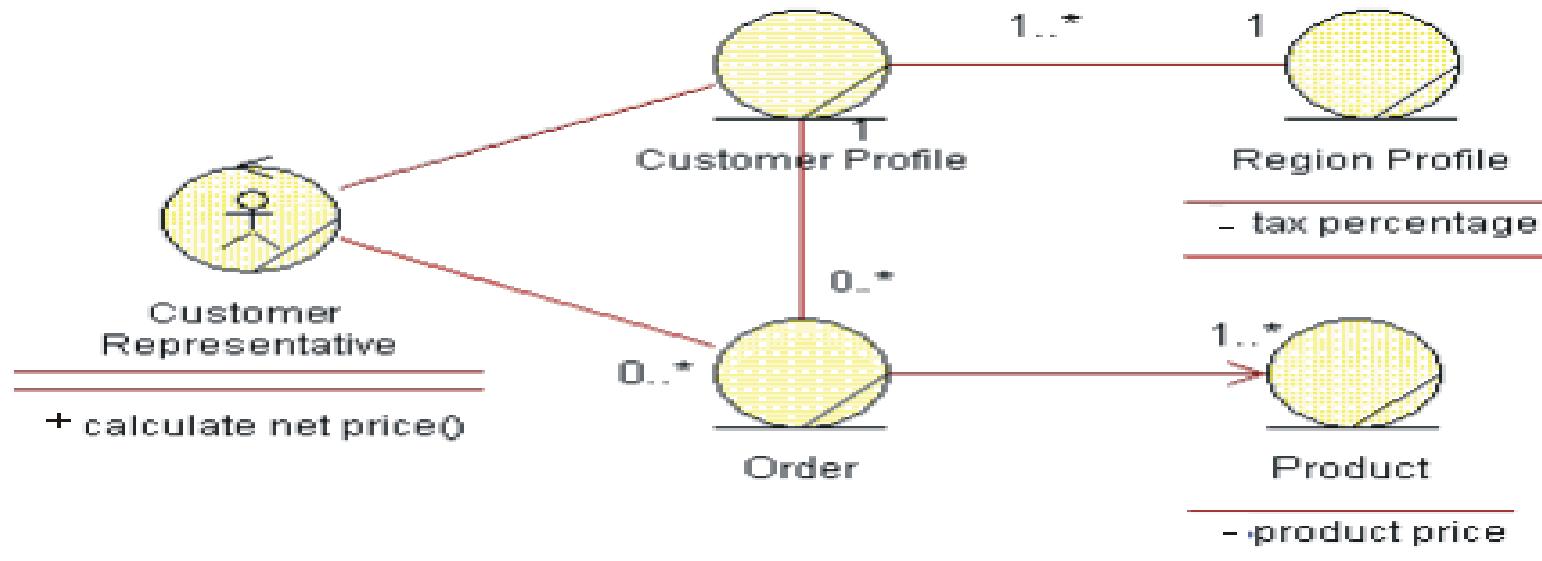
- ④ 你可能建立了下列规则来确定客户的状态:
- ④ A Customer is a Good Customer IF AND ONLY IF
- ④ the unpaid invoices sent to this Customer are less than 30 days old.





派生规则（2）计算规则

- 计算规则与推理规则类似，区别在于前者更正式，看起来像一个算法。计算规则也是可以追溯到工作流中的某一活动中，最终表现为业务工作者和业务实体的操作。
- 计算规则可以描述值的计算：
 - 产品的净价格可以这样计算The net price of a Product IS COMPUTED AS FOLLOWS
 - Product Price * (1+tax percentage/100).
 - 当你为订单生成一份帐单时，计算净价格可以成为发送订单的一部分。在业务对象Customer Representative上实现这个功能

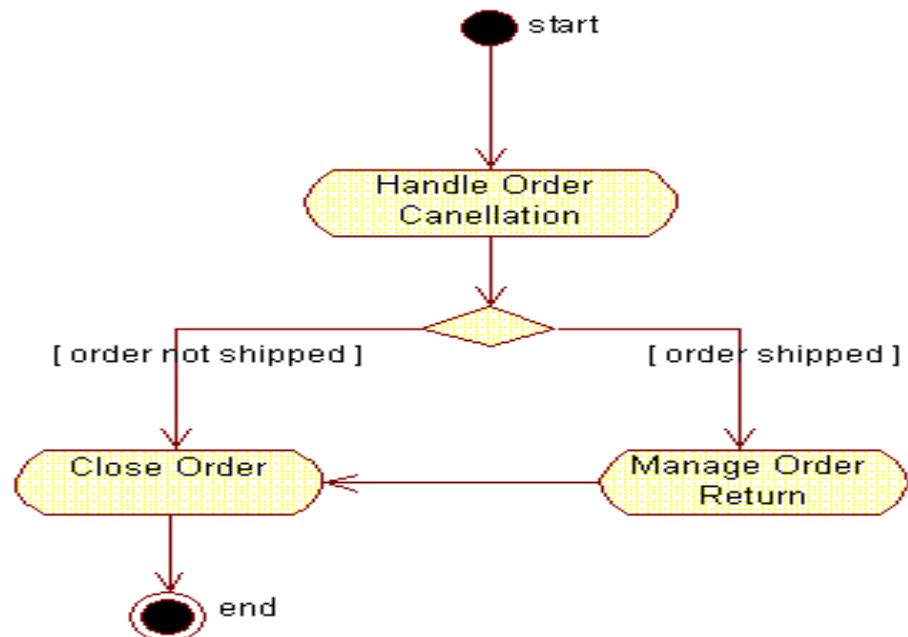




◎ 激励反应规则

- 该类别的规则影响业务用例中的工作流，并可以追溯到它们适用的业务用例中。你可以选择一个条件分支，或者可选分支。如果涉及的动作不明显，你可以将它作为一个活动状态的一部分。
- 在业务对象模型中，该类别的规则可以影响如业务实体的生命周期，或者它可以称为业务工作者操作的描述的一部分。

- ④ 在订单处理中，你可以发现以下规则：
- ⑤ WHEN an Order is cancelled
 - IF Order is not shipped
 - THEN close Order.

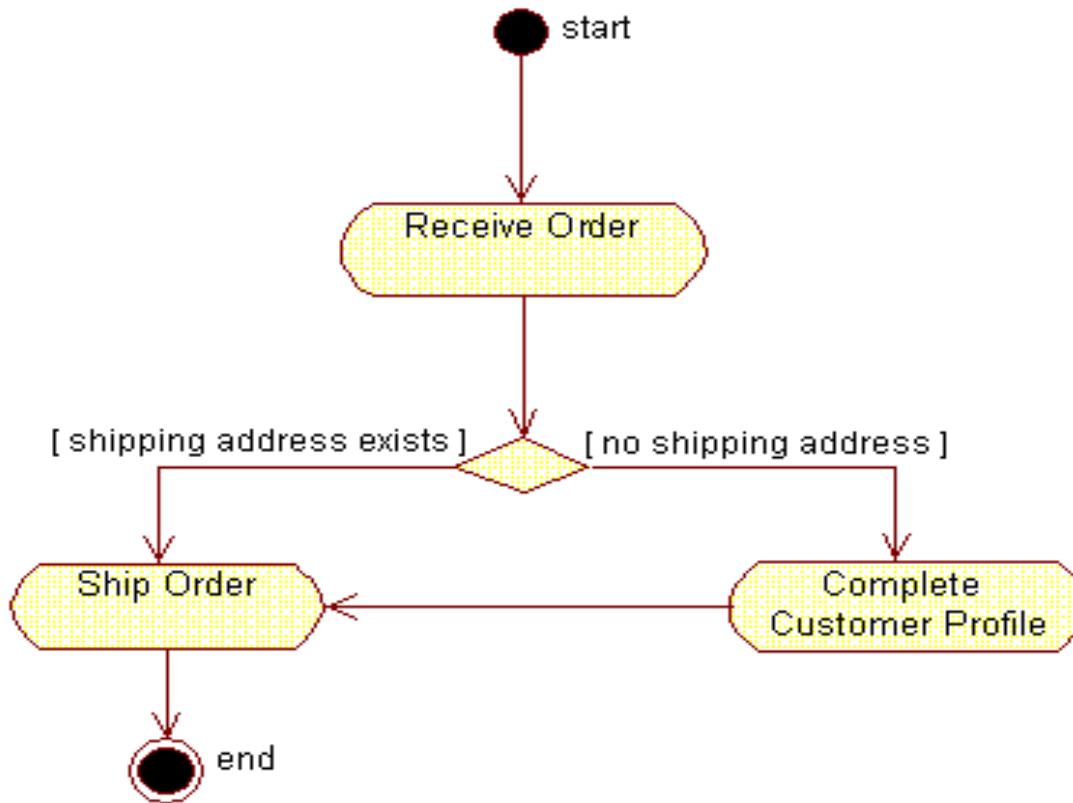




◎ 行为（操作）约束规则

- 此类业务规则通常转化为工作流程的“前置条件”或“后置条件”，或者转化为工作流程中的条件路径或备选路径。
- 它可以是要达到的性能目标或某个应该追踪到应用该规则的业务用例的其他非行为性规则。

- 在订单处理中，你可以发现以下规则：
- Ship Order to Customer
- ONLY IF Customer has a shipping address.





结构约束规则

- 它影响着业务实例之间的关系。它们通过业务实体之间的联系来表达，某些时候是隐藏在联系的多重性上的。

例如：

- 订单处理中，你可以发现以下规则：
- IT MUST ALWAYS HOLD THAT
- an Order refers to at least 1 Product.



存在规则

- 存在规则控制确定的对象什么时候是存在的。
- 这种信息可以是在类模型中固有的（比如聚合对象只有当周围的对象都存在的时候，它才存在）。



3. 2 业务规则的OCL表达

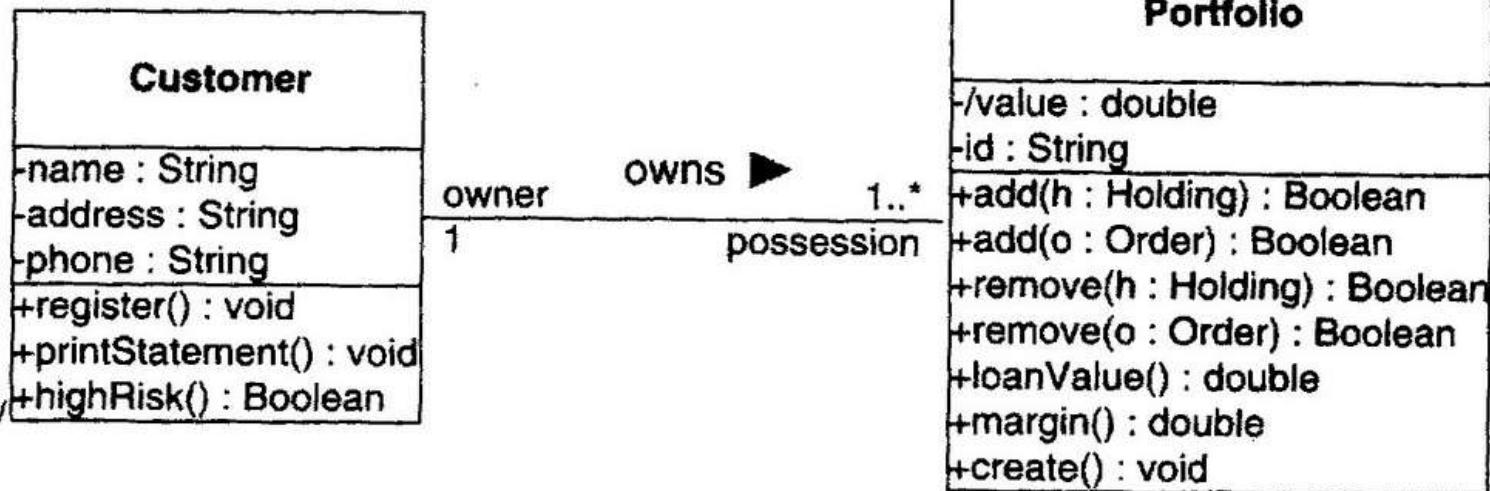
- ④ OCL是在UML中特别用来对约束和规则进行说明的语言。OCL被明确设计成足够简单同时又非常严密，可以定义所有类型的规则和约束。
- ④ OCL的主要特征：
 - OCL是一个查询性的语言：任何OCL的动作都不会对模型本身造成任何的影响或者改变。（如对象的属性），但可以确定变化（某状态的改变将会是一个操作的结果）。
 - OCL是一个强类型的语言：OCL的任何一个元素，都是有类型的，并且任何操作的返回值一定有一个确定的类型。所有运算符都有特定的类型，每个运算符只能应用于特定的操作数类型。

- OCL很强调时间点。任何操作都定义为瞬时完成的，即操作中模型的状态不会改变。
 - 基于实现考虑，这么规定有一定的道理，不然在多线程系统中，OCL约束很有可能失效。另外 precondition和postcondition也明确规定是在方法执行的前后时间点才有约束，时间点不对约束无效。
- OCL是一个宣言式（Declarative）的语言。描述了 what to do，没有描述how to do。
 - 如`self.attribute->select(i | i.name = 'wxb_nudt')`。
 - OCL是基于集合论和谓词逻辑的。这点从它的表达式中可以很轻易的看出来。这是OCL被列为MDA的支柱之一的原因。



推理规则的OCL描述

- 例子：客户考虑HighRisk利润是否会高于投资组合总贷款价值的90%。



«business rule»

```
context Customer::highRisk() : Boolean
post: result =
  if (possession.margin() > 0.90 * possession.loan_value())
    then true
    else false
  endif
```



计算规则的OCL描述

计算投资组合的贷款价值

```
«business rule»  
context Portfolio::loanValue() : Real  
post: result = holdings->select (  
    security.oclType = Bond or security.oclType = Stock )  
->iterate( h: Holding; value : Real = 0;  
if (h.security.oclType = Bond)  
    then value + (number * security->price * 0.90)  
    else value + (number * security-price * 0.60)  
endif  
);
```

Portfolio

```
-value : double  
-id : String  
+add(h : Holding) : Boolean  
+add(o : Order) : Boolean  
+remove(h : Holding) : Boolean  
+remove(o : Order) : Boolean  
+loanValue() : double  
+margin() : double  
+create() : void
```

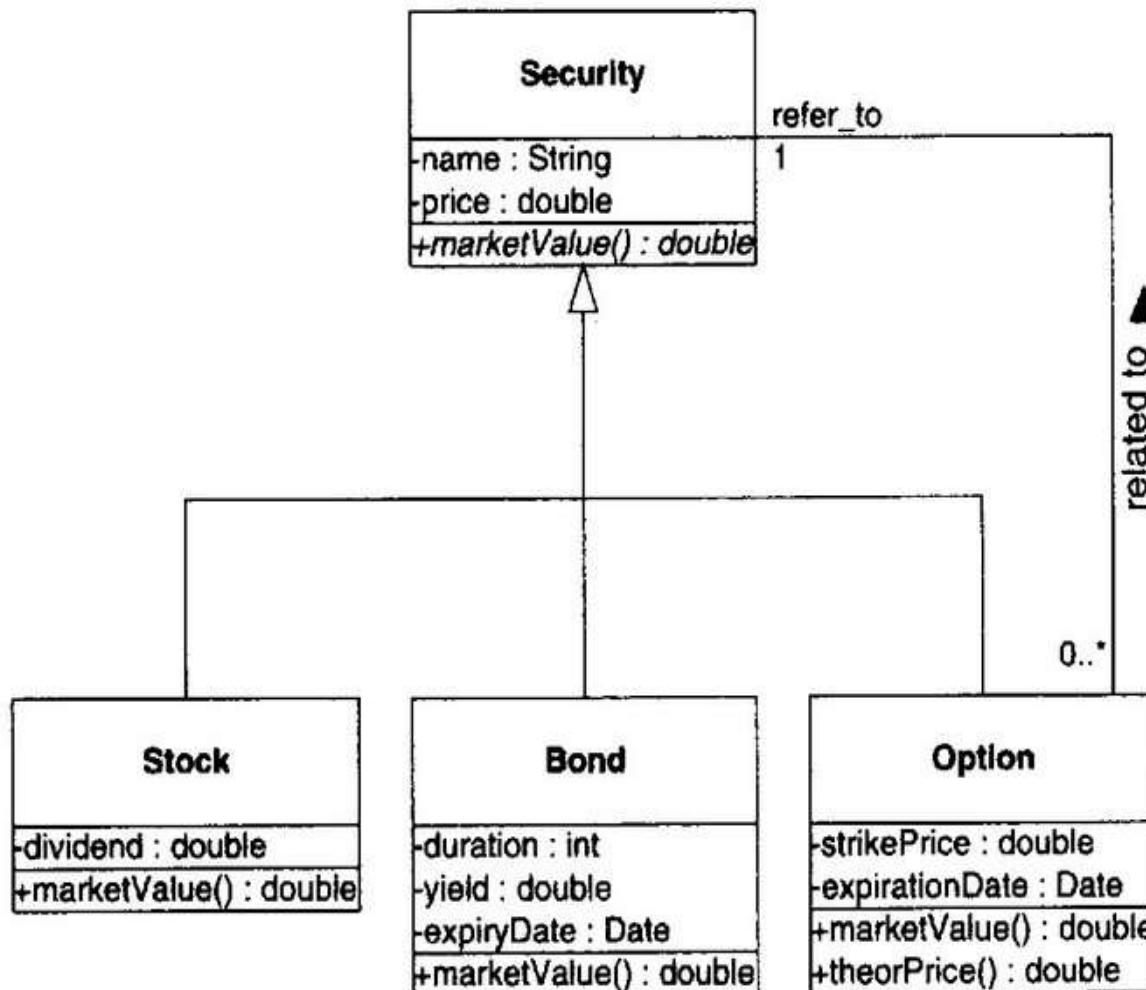
计算投资总和

```
«business rule»  
value = holdings->iterate(h : Holding;  
result : Real = 0;  
result + (h.number * h.security->price));
```



结构约束的OCL描述

- 结构约束可以通过类图中的关联和多重性定义，而其他约束都不能在图中表示，而必须在OCL中指定。



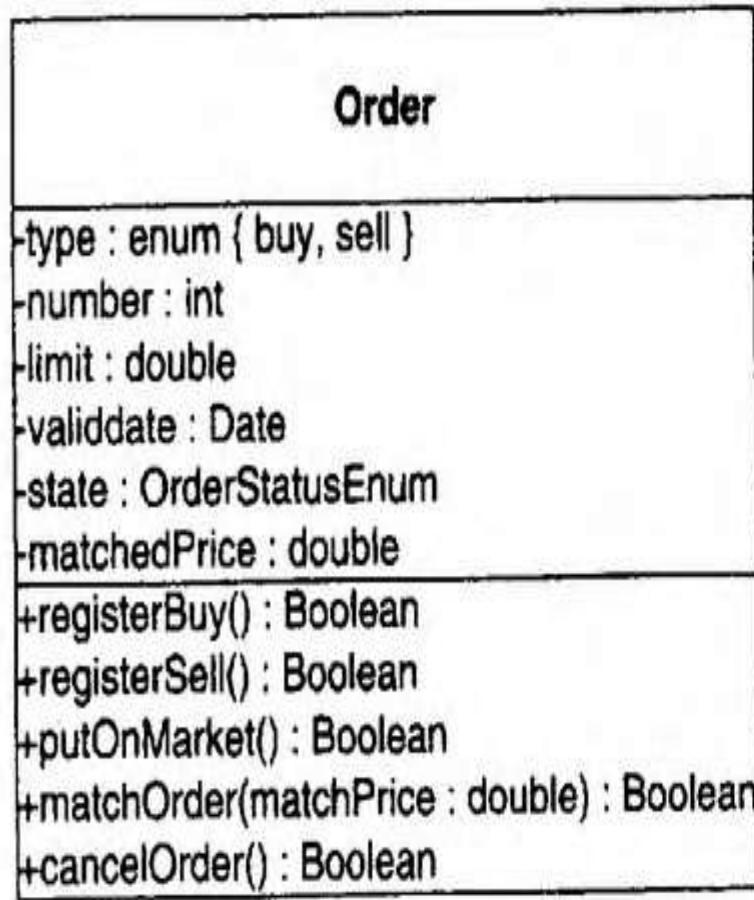
如一个自己不能

自己的期权的表达
Context option inv
Self<>refer_to



行为约束的OCL描述

- 表现了订单处理过程开始前必须为真的条件——前置条件
- 表现了订单处理过程之后必然为真的条件——后置条件



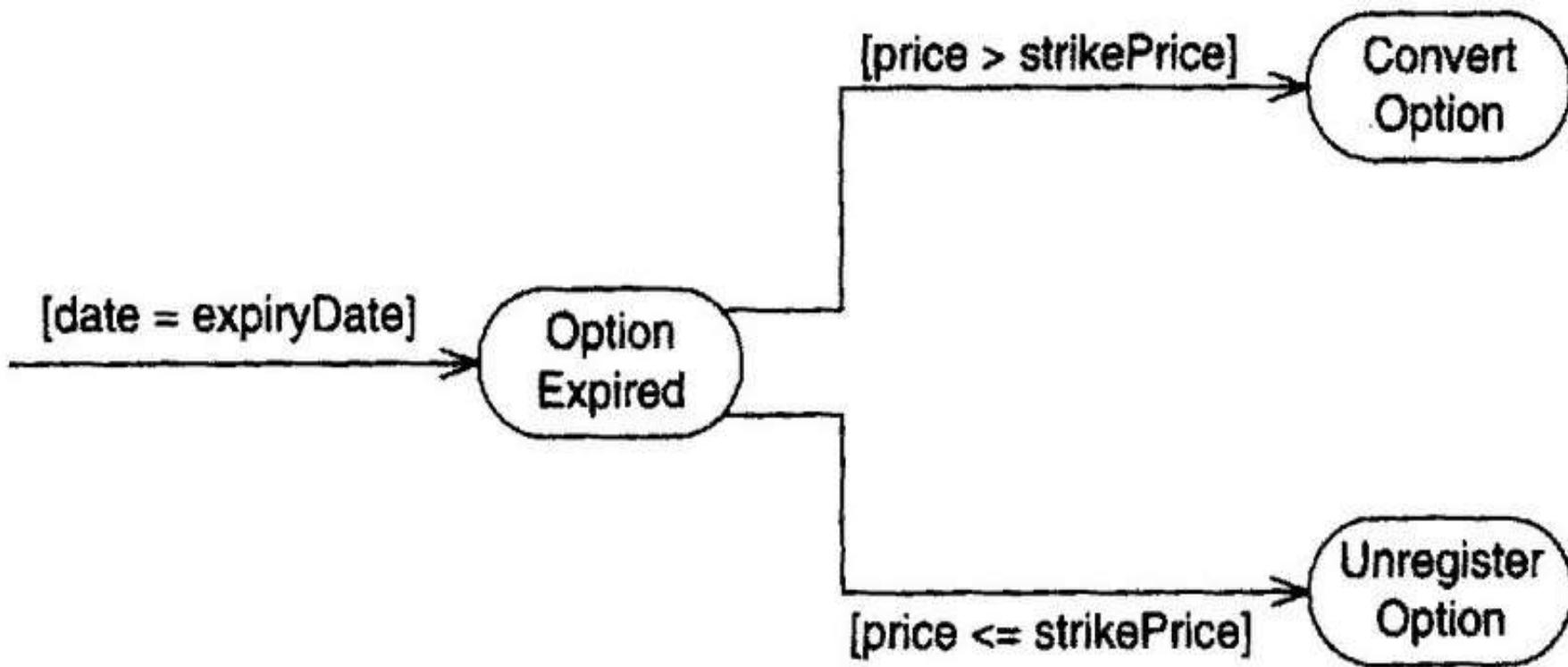
“business rule”

```
context Order::putOnMarket() : Boolean
pre: state = #created
post: state = #onMarket
```



激励反应约束的OCL描述

- 由于OCL是陈述语言，激励反应约束不能全部被定义。激励的事件部分可以在OCL被作为**保护**定义。
- 图中了三个OCL定义的**保护**确定了条件，以把对象移入某状态或使对象发生指定行为。





存在规则

- 如果一个Option对象的有效期比当前日期要老的话，那么就不会存在。

| Option |
|-------------------------|
| -strikePrice : double |
| -expirationDate : Date |
| +marketValue() : double |
| +theorPrice() : double |

Context option inv

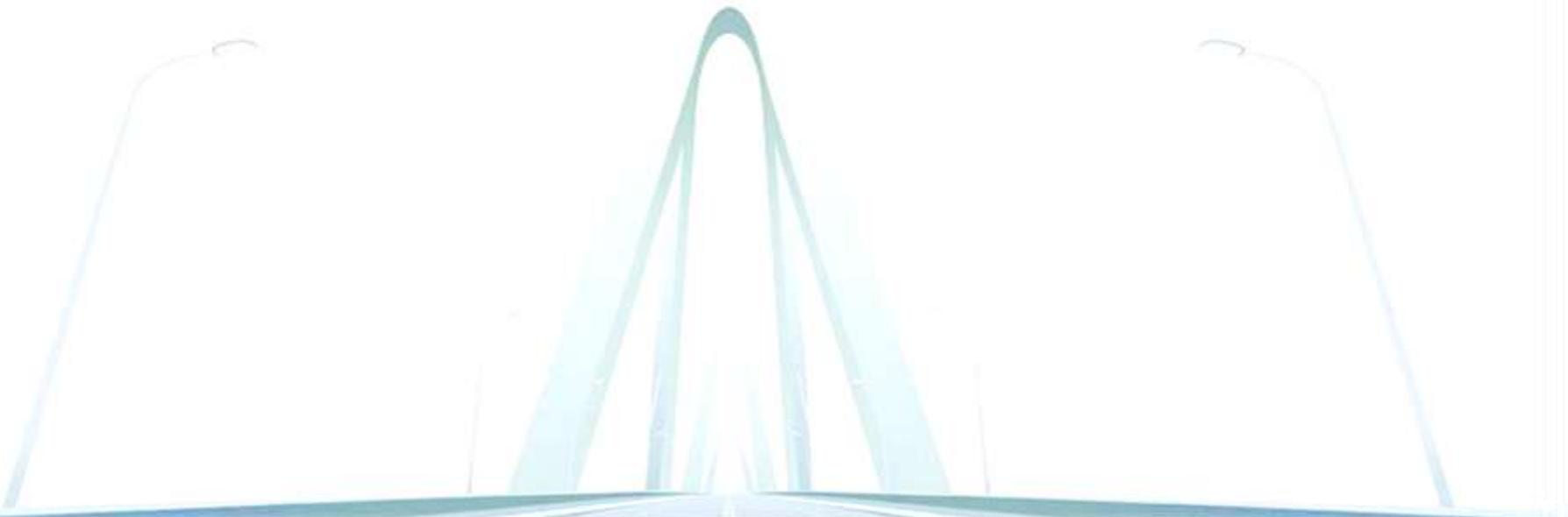
Not date.now.isAfter
(self.expirationData)



3.3 业务规则的其他表达方法

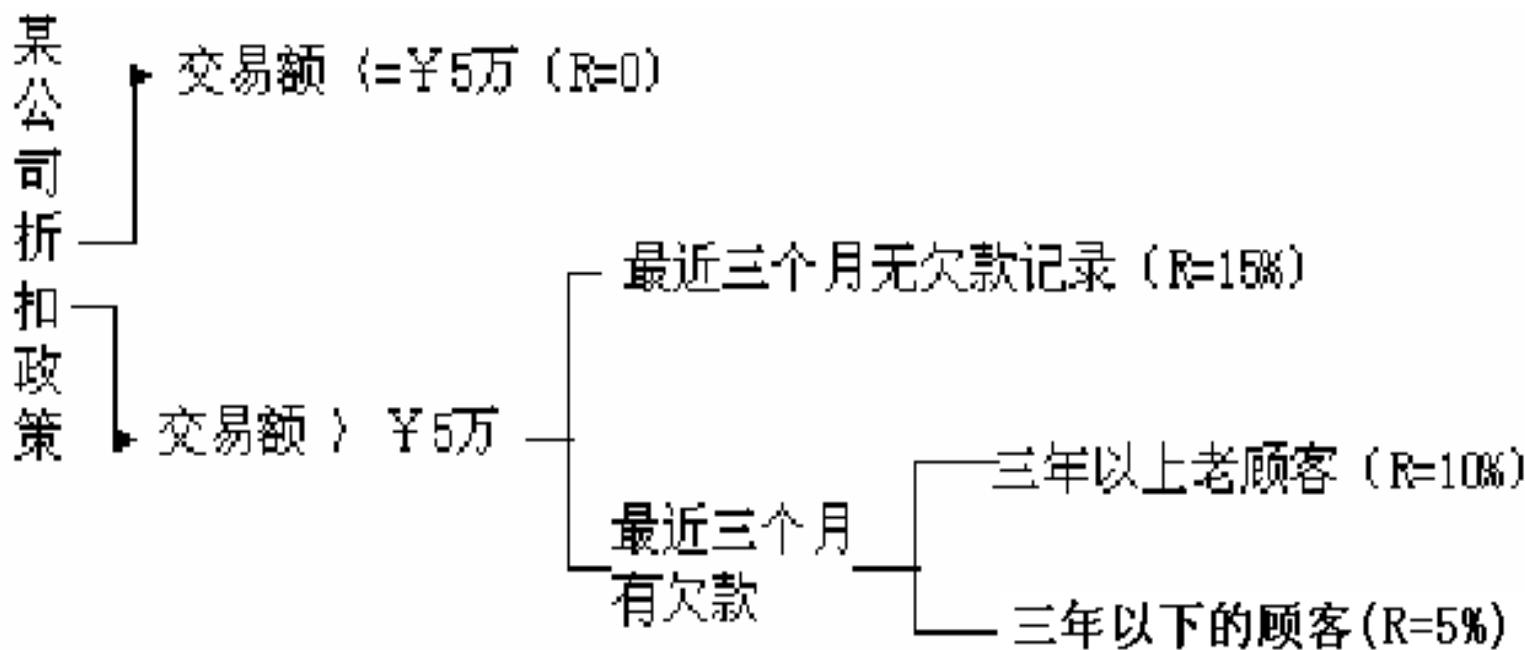
④ 传统方法的业务逻辑描述方法

- 流程图
- 伪码
- 决策树*
- 判断表*



业务逻辑描述1：决策树

- 用图形方式（树型）描述加工逻辑。通常用来描述根据不同条件及其取值来选择的处理过程。
- 例如一个公司的折扣政策：





业务逻辑描述2：判断表

- 变量表

| 条件名称 | 取值 | 含义 |
|------|--------|-------------------------|
| 金额 | M L | >\$50,000 <=\$50,000 |
| 欠账情况 | N Y | 无欠帐记录 有欠帐记录 |
| 时间 | D N | 3 年以上老客户 3 年以下客户 |

- 判断表

| | 可能方案 | 1 | 2 | 3 | 4 |
|----|--------|---|---|---|---|
| 条件 | C1(金额) | M | M | M | L |
| | C2(欠款) | N | Y | Y | / |
| | C3(时间) | / | D | N | / |
| 结果 | R=15% | X | | | |
| | R=10% | | X | | |
| | R=5% | | | X | |
| | R=0 | | | | X |

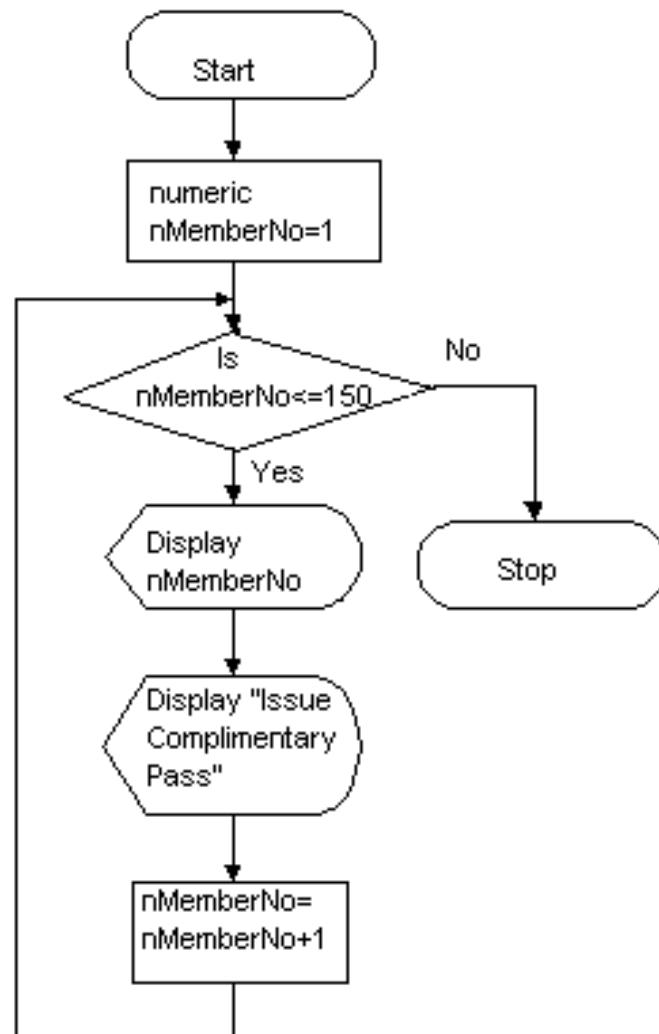


业务逻辑描述3：伪码

```
//-----  
// The part of main program  
// variable declaration  
//      cCertify: The public variable of this program. used    to return the //  
result to main program about    user validation.  
//                  Value is "PASS" or "NOPASS"  
//      cCustomer: The public variable of this program.used  
//                  to save the name of customer who had ordered toy.  
//-----  
begin  
    character cCertify, cCustomer  
  
    cCertify = "NOPASS"  
  
    call Password  
  
    if cCertify = "PASS"  
    begin  
        call Purchase  
    end  
end
```



业务逻辑描述4：流程图



Agenda

1. UML业务建模扩展

2. 业务视图

 2.1 业务愿景视图

 2.2 业务过程视图

 2.3 业务结构视图

 2.4 业务行为视图

3. 业务规则

4. 业务模型到软件系统

5. 小结

A1.信息编码方法

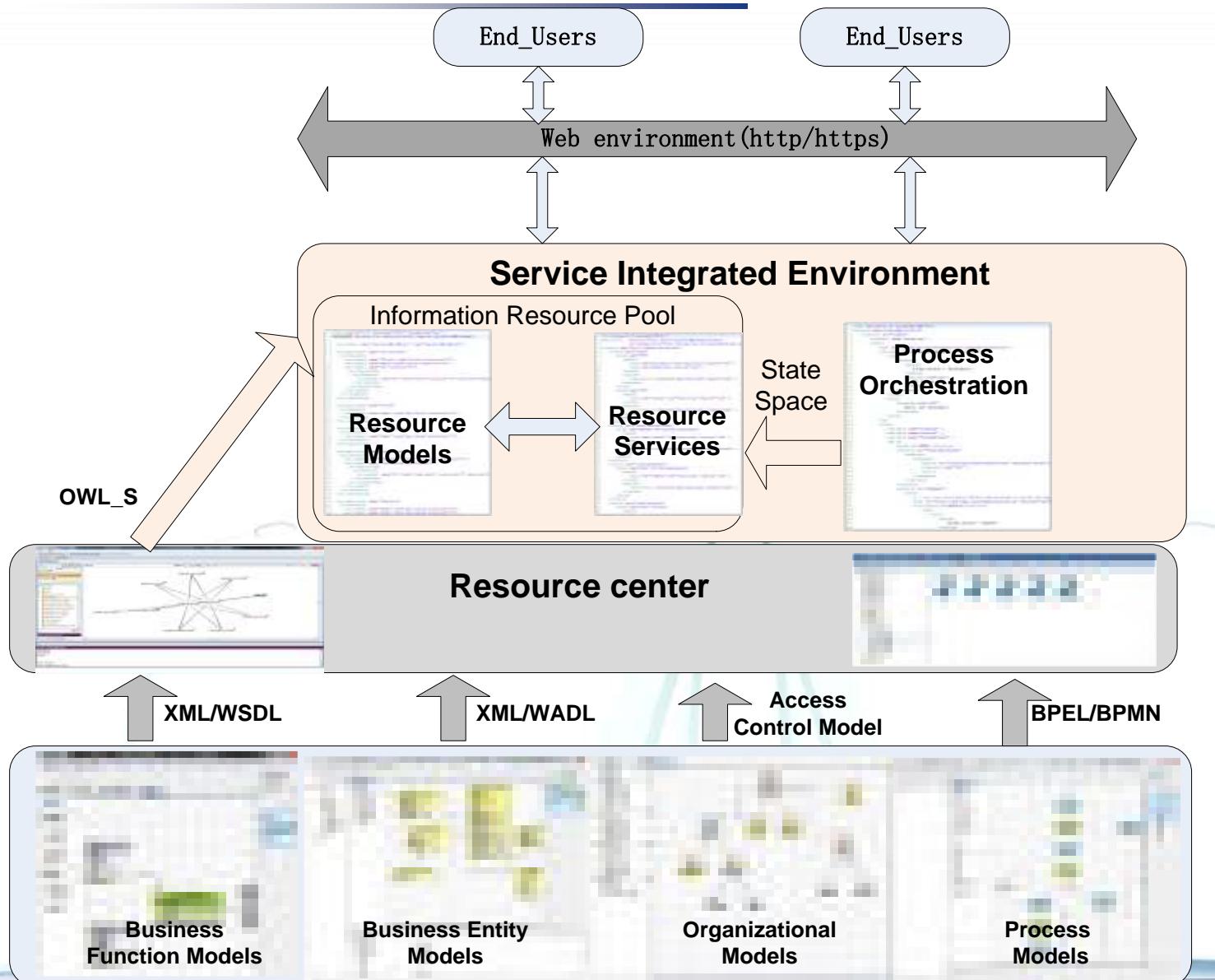


4 业务模型到软件系统

- 面向软件系统，业务模型可作为基础模型，识别一个或多个信息系统的功能和非功能需求。
- 业务模型通过抓换，建立软件系统的组件。重要的一点是要意识到，从业务模型到软件系统不是一个一对一映射。
- 面向服务的业务模型转换是开发实施软件系统的一种重要方法。



业务模型驱动的服务转换软件



Agenda

1. UML业务建模扩展

2. 业务视图

 2.1 业务愿景视图

 2.2 业务过程视图

 2.3 业务结构视图

 2.4 业务行为视图

3. 业务规则

4. 业务模型到软件系统

5. 小结



5 小结

业务架构是业务建模的组成要素，一个架构捕捉了系统中一系列元素，它们之间通过各种关系相互连接。

定义业务架构的概念有：

- **过程** 是在一系列输入对象的基础上为客户创建一系列输出价值的若干个活动的抽象。输入输出对象，以及过程中所使用的其他对象构成了系统中的所有资源。事件由某个过程产生并且由一个或者多个其他过程接收。
- **资源** 业务中所用到的概念或者事物。
- **目标** 一个或多个资源所期待的状态。目标附属于整个业务以及业务中的过程。
- **业务规则** 对业务某些方面进行定义或者约束的声明，同时也展现了业务知识控制了业务如何运作。