

## *Lecture4* 应用架构及建模方法

蔡鸿明

*hmcai@sjtu.edu.cn*

信息系统技术实验室 [ist.sjtu.edu.cn](http://ist.sjtu.edu.cn)

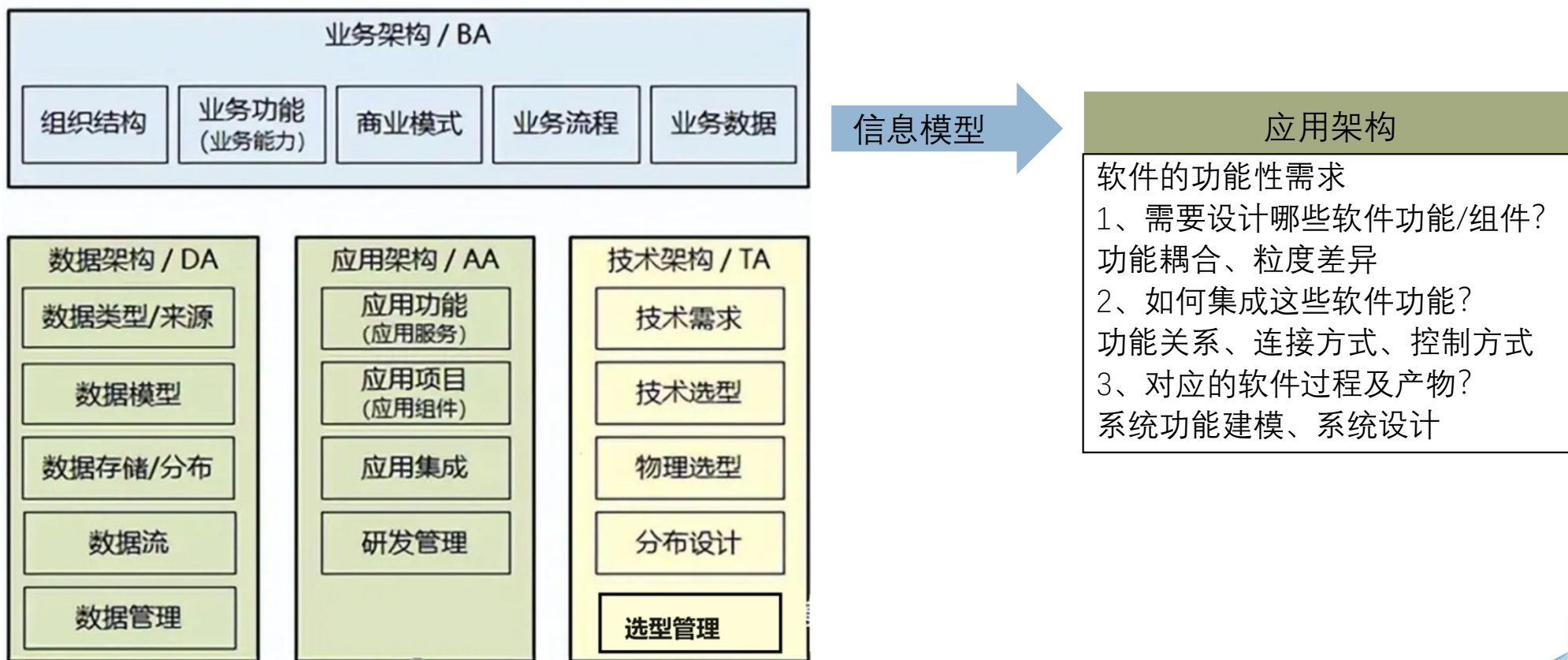


- 1 业务架构驱动的应用架构设计
  - 从模型到软件的转换-业务模型-建模方法发展
- 2 基于活动的过程建模方法
  - 业务任务规划-软件功能设计-活动时序流描述-活动执行控制
- 3 基于数据的过程建模方法
  - 数据分类-数据建模-数据流图-组织建模
- 4 基于状态的过程建模方法
  - DEDS-经典Petri网-高阶Petri网-PNG流程建模-PNG仿真实例
- 5 基于事件的过程建模方法
  - EPC-EPC规则语义-EPC建模规范-企业管理基础-ARIS实施实例
- 6 小结

# 1 业务架构驱动的应用架构设计



- 业务架构作为企业架构的上层指导性架构，描述企业的业务策略及实现，作为信息架构、技术架构等部分创建和分析的基础。
- 因此，业务架构的核心是建立各种模型，作为后续应用架构、以及技术架构等基础。

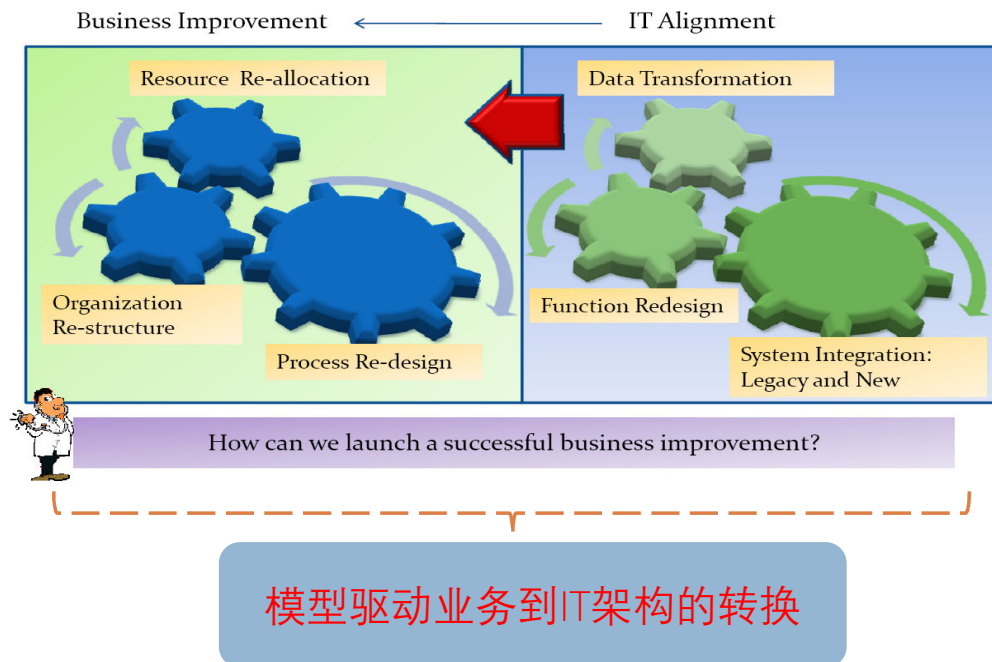


## 1.1 模型转换



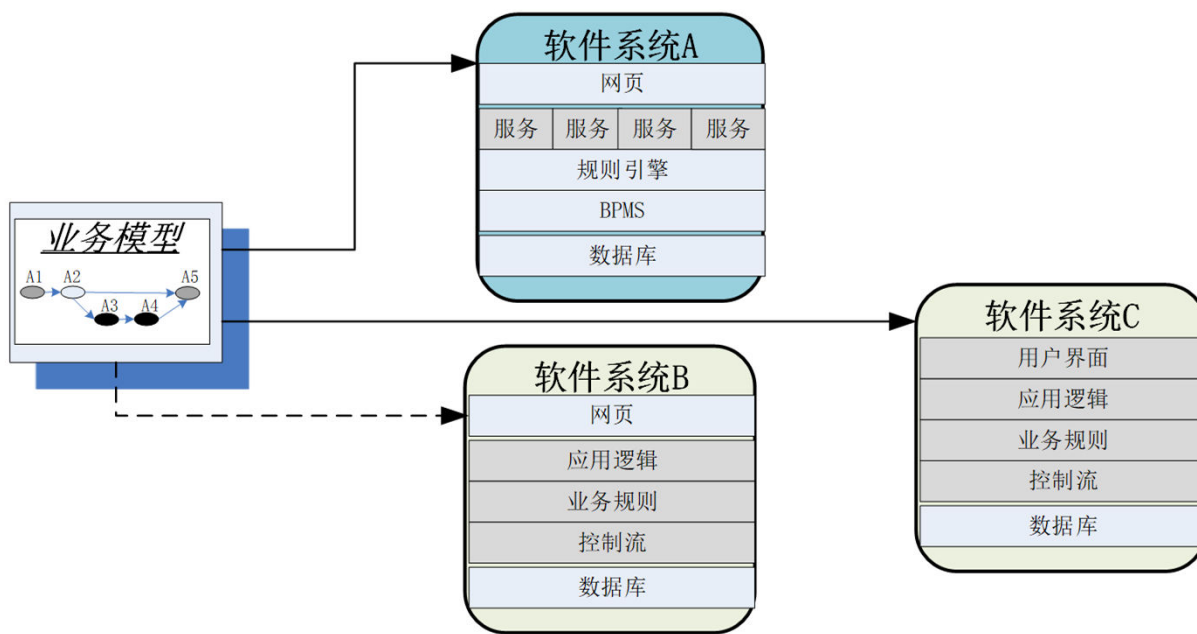
- 软件系统是为了解决企业业务问题的技术系统，软件系统与业务模型密不可分。
  - 从宏观上讲，软件系统本身可以理解为一个模型，软件系统的建立就是计算机信息模型的建立过程。
  - 从微观上讲，软件系统本质上可以看作是由一系列相关模型构成的有序集合。
- 面向复杂软件构造，需要以模型描述并分析企业的业务问题，业务模型是分析、设计、以及实施复杂软件系统的重要步骤，也是软件重用性的体现。
- 因此，构造复杂软件的开始往往就是对涉及到的企业或者组织的业务开展架构以开展建模分析。

- 对于企业来说，将软件系统建立一个业务模型之上可以带来以下好处：
- 业务逻辑可以在系统之间重用；
  - 系统之间更易于整合，便于信息的交换和共享；
  - 软件系统自然成为总体业务的一个内在组成部分，为业务提供足够的支持并提高工作的效率；
  - 系统更容易随着业务模型的变化进行相应的升级及修改，这大大减少了软件系统维护和持续更新的成本。



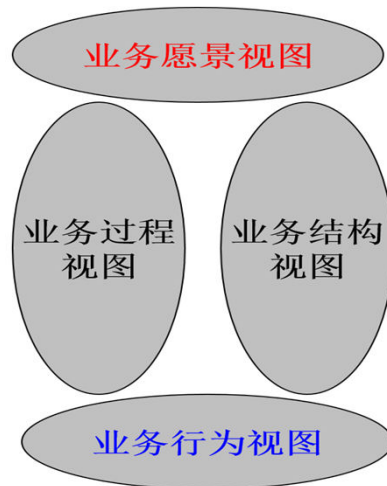


- 业务建模涉及一组具有相关性的活动、方法和工具，它们被用来建立描述企业的业务。
- 业务模型到软件模型的映射
  - 理想情况下，业务模型中的对象可以映射为软件系统中的模型或对象
  - 这并不是一一映射，软件系统中有业务模型中完全没有涉及的对象，反之亦然

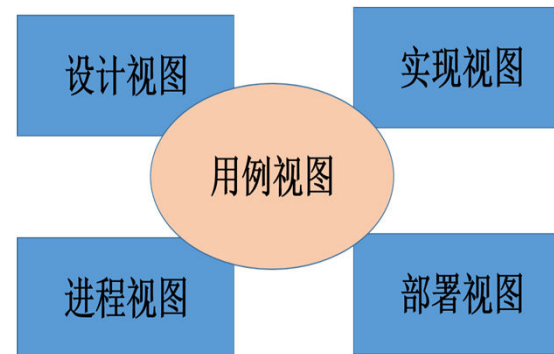




- 业务建模与软件建模处于生命周期的不同阶段；
- 业务模型可以转变为相应的软件系统模型；
- 软件系统和业务系统之间存在许多相似之处，也存在一些差异；
- 业务建模中的常用概念和标准的系统建模是不一样的；



UML业务建模视图



UML软件建模视图



- 业务建模和软件模型的对应转换，是业务域与软件域的跨域双向映射问题，是业务与IT对齐的核心。

模型	业务模型	软件模型
目的	面向企业业务分析	面向软件系统开发
领域	属于业务层面	属于技术层面
模型层次	层次较高：组织、业务概念，流程…	比较底层，数据类、函数、表字段，进程…
支持框架	ARIS, Zachman, UML业务建模框架	SSH, RUP



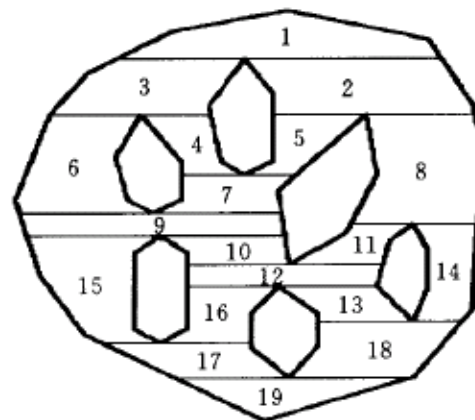


- 业务模型是针对业务功能的抽象描述，是为了应对错综复杂现实世界的简化，而建立的标准而规范的描述手段和统一视图。
- 从建模角度来说，需要定义模型的目的、范围、视角、以及粒度。
  - 从**目的**来说，业务建模是为了建立相应的描述，以更好地理解业务的关键机制。
  - 从**范围**来说，企业本身是具有某种特定目的的复杂系统，业务中的不同要素相互作用以实现这个目标。
  - 从**视角**来说，业务模型是从业务人员的角度来描述问题，而不是如何实现系统来解决问题，或者说软件开发的角度的。
  - 从**粒度**来说，业务模型的当前用途决定了模型的细致程度。
- 简单来说，**标准而规范的描述手段**，就是信息建模。

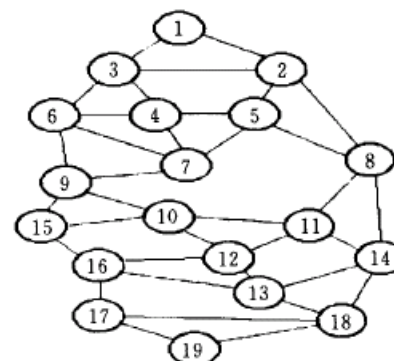
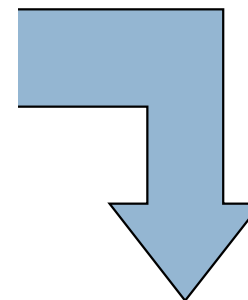
# 一个建模实例（绣花问题的算法建模）



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



## ➤ 在一个企业中，哪些模型能用于描述企业业务并帮助构造软件呢？

- 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：

GPU，设备，路由器，网络设施， ...

- 服务/产品：

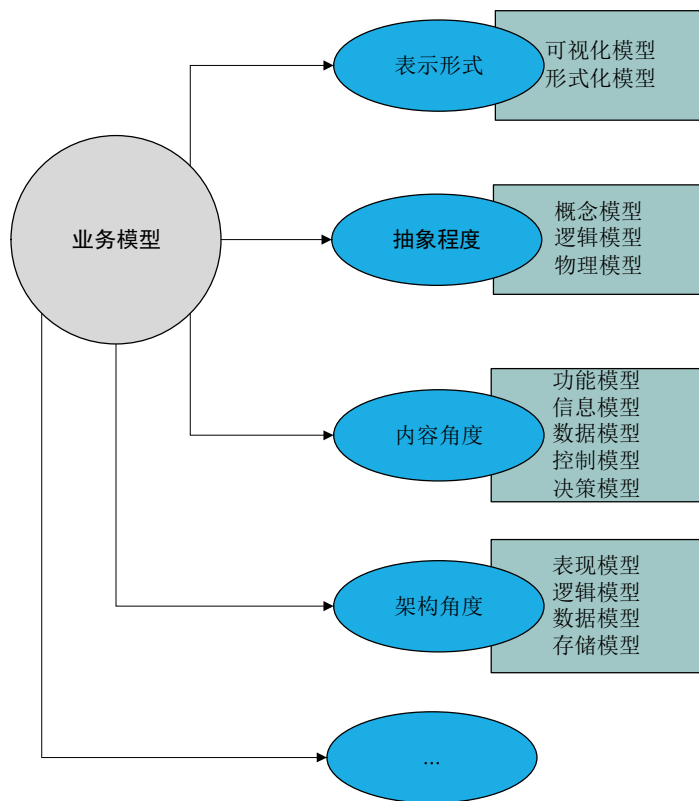
信用查询，身份认证，云存储...



这些不同对象不是孤立的，  
存在而是存在着复杂的关联  
以及交互关系。

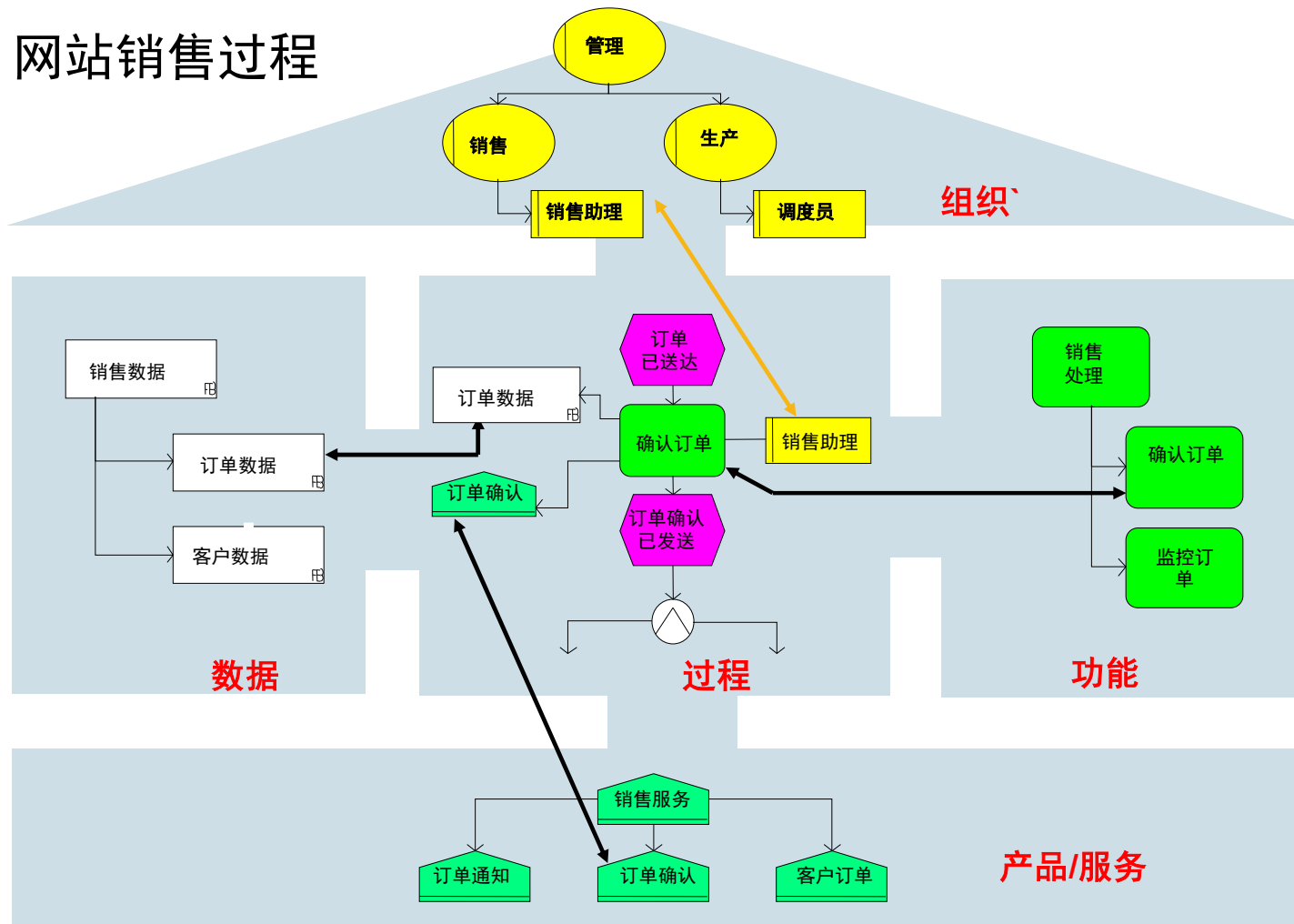


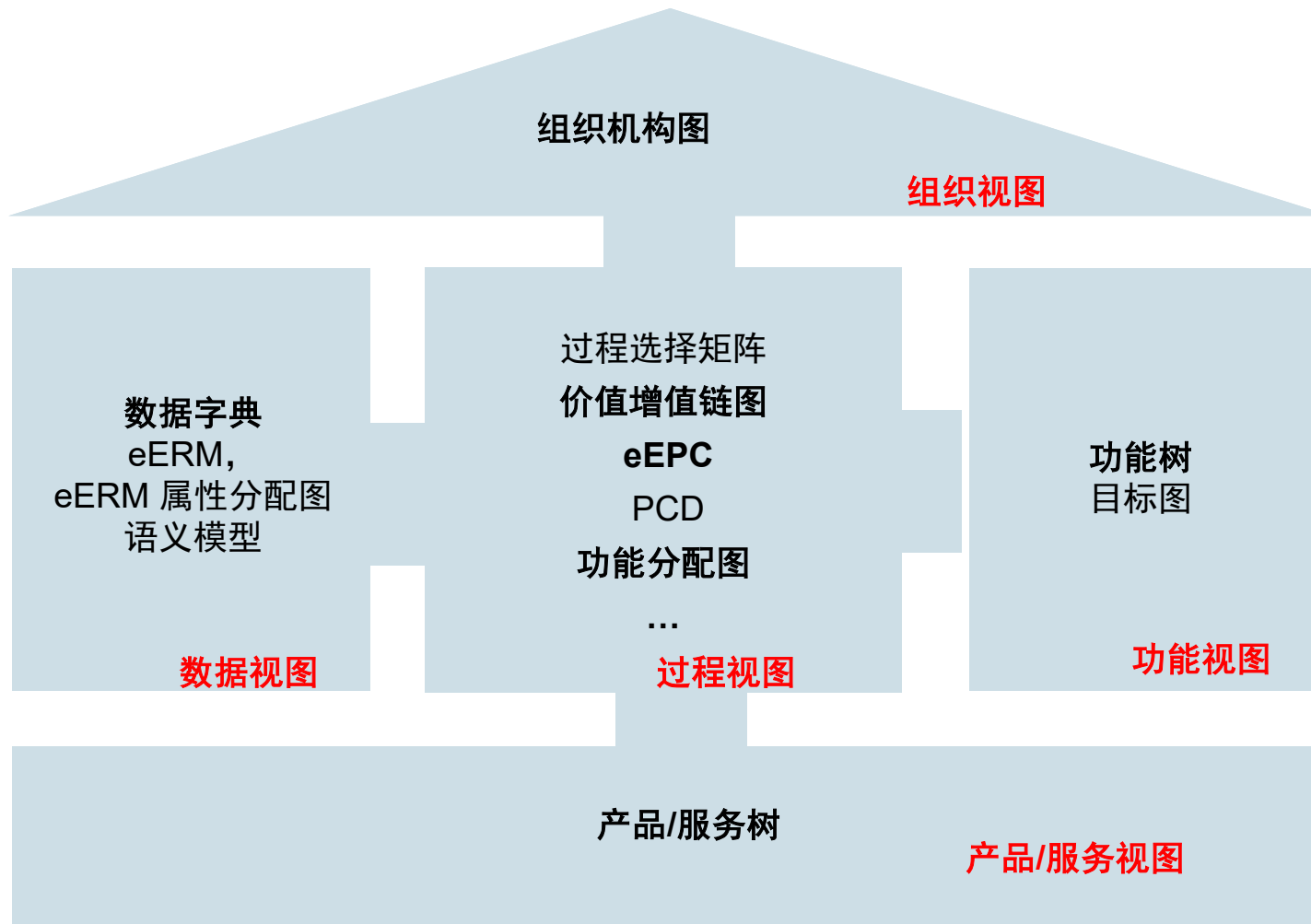
- 建模是我们研究客观世界的途径。
- 建模不是复制，而是模拟，是强调本质而忽略细节的抽象模拟。
- 按照不同的角度，业务模型可以进行不同的分类。



模型具有  
多层次、  
多视角、  
复杂关联、  
时序相关  
等特点！

## 网站销售过程







- 业务模型具有**多视角、多层次、复杂相关、时序依赖**等特点，非常复杂。
- 因此，企业的业务建模要遵循一定原则。
  - 分离：对企业业务的各个侧面进行分离，构建不同视图以逐步进行分析。
  - 分层：通过不同的抽象程度，反映系统的不同层面。
  - 分解：根据总体目标，将功能活动逐步分解成各个组件，进而逐步细化。
  - 一致性：各组件在语义及语法上保持一致。
  - 模块化：构建相对独立的各个模块，以便维护模型，从而适应用各种变化。
  - 通用性：需要提高模型的通用化程度，通过定义构件、接口、协议等方法，将模型等共性要素统一表示。
  - 功能与行为分离：功能考虑做什么，行为关系如何做，区分功能和行为，以保证不至于过早陷入细节泥潭。
  - 活动与资源解耦：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。

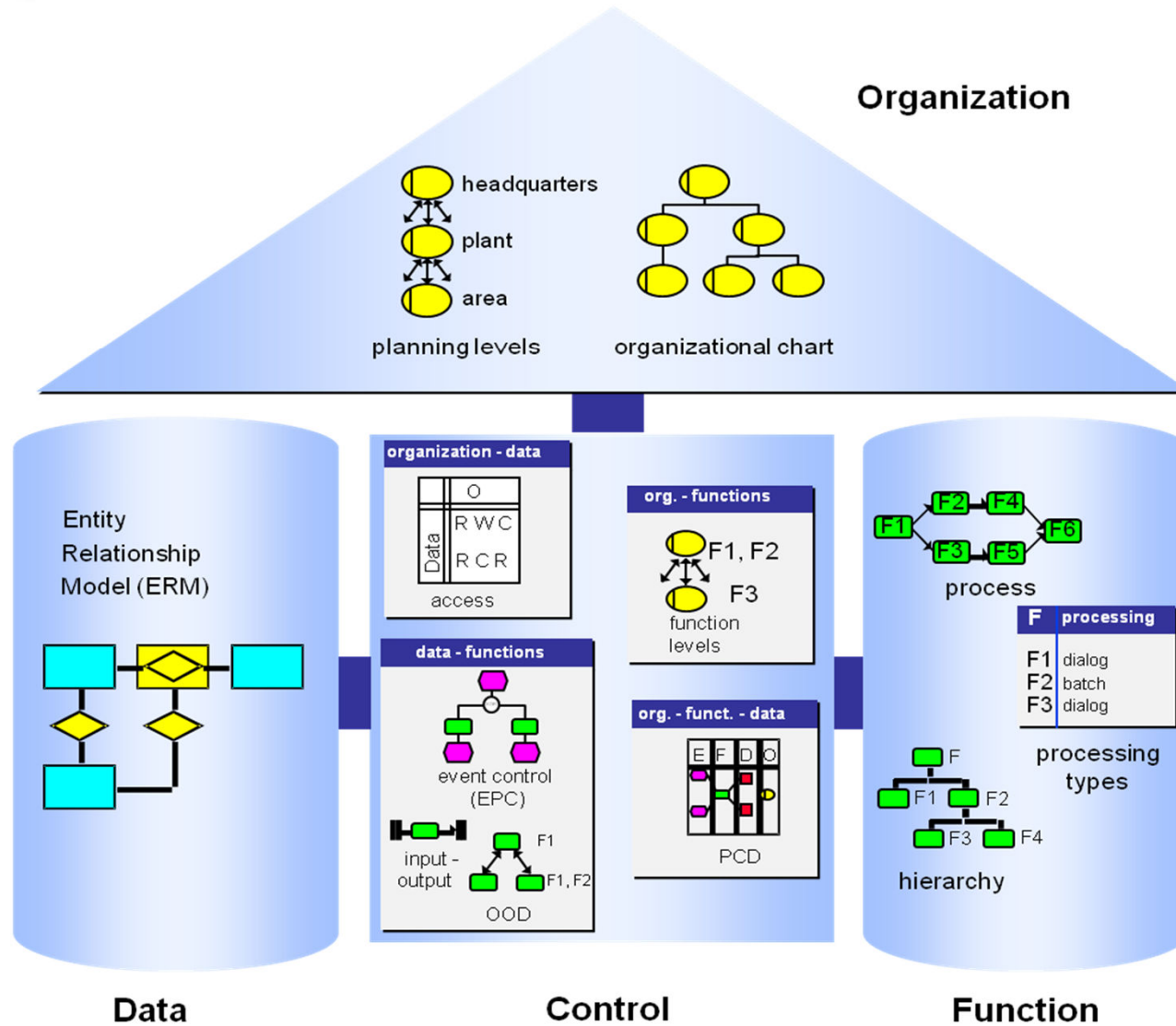
这八大原则可以分为三类，体现是不同的建模要求：

- **分离、分解、分层和一致性原则**描述的是建模的方法，企业业务要素纷繁复杂，如不开展分离层次、分离视图、分解功能，将无法有效建模，一致性则是这个过程中的质量保证；
- **模块化和通用性原则**描述了实现的目标，是工程化的要求；
- 最后两种原则的则是建模中容易出现的问题。**功能与行为分离原则**可以控制复杂性的展开，避免过早陷入细节泥潭；而**活动与资源解耦原则**是为了更好地体现系统的柔性，提高业务执行的效率。

一般情况下，业务模型可以划分为三个层次：

- **视图**：一个业务模型由数个不同的视图表示，其中各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。
- **模型图**：每个视图都由数个模型图组成，每个模型图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不同方面或机制。
- **业务对象及关联**：模型图中有不同的对象及其关系，通过使用不同的对象表征业务概念，并通过关系让模型图中的不同概念相互关联起来。
  - 对象可能是物理存在的，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。
  - 而关联是连接各对象的方式，目前来说，通过过程联系各对象，是当前的主要形式，即通过过程中的活动，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。

# 一个典型的业务模型框架 (ARIS)



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

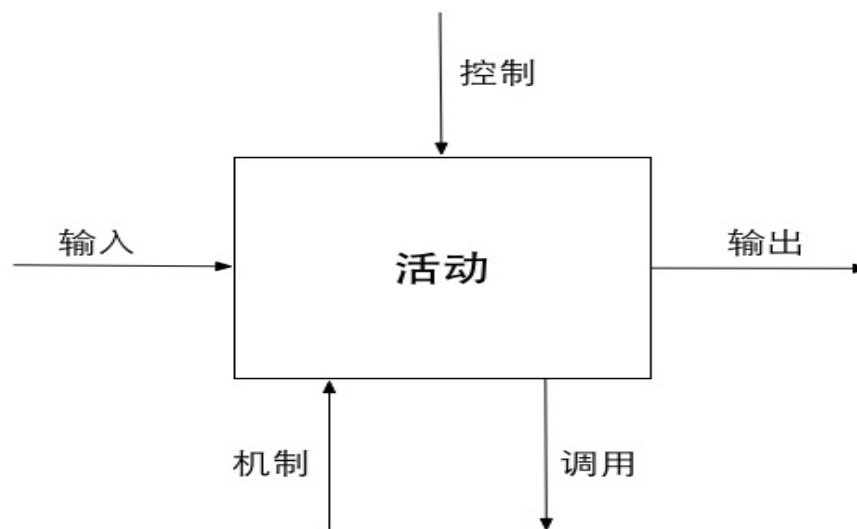


基于核心的原子要素，可以划分以下信息建模方法：

- 基于功能
- 基于流程
  - 集成化建模（多视图建模）
- 基于服务
- 基于大数据



- SADT(结构化分析与设计方法)和IDEF (功能分解法)
- 基于功能的建模方法却存在几个主要问题：由于功能分解法的基本组件仅有一个，因此该方法虽然有很好的通用性，但缺乏丰富的语言描述能力；
- 其次，功能分解法的严格递阶关系很容易导致企业的组织结构之间，尤其是在不同的组织单元之间造成交流障碍，形成自动化的孤岛。







- 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、活动的执行可能需要触发条件。
- 基于过程的建模方法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
    - 以数据流图为代表，将活动按照数据流向连接。
  - (2) 基于**活动**的业务流程建模
    - 以活动图为代表，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为代表，将业务活动通过消息事件连接。
  - (4) 基于**状态**的业务流程建模
    - 以PetriNet为代表，将业务活动通过状态事件连接。

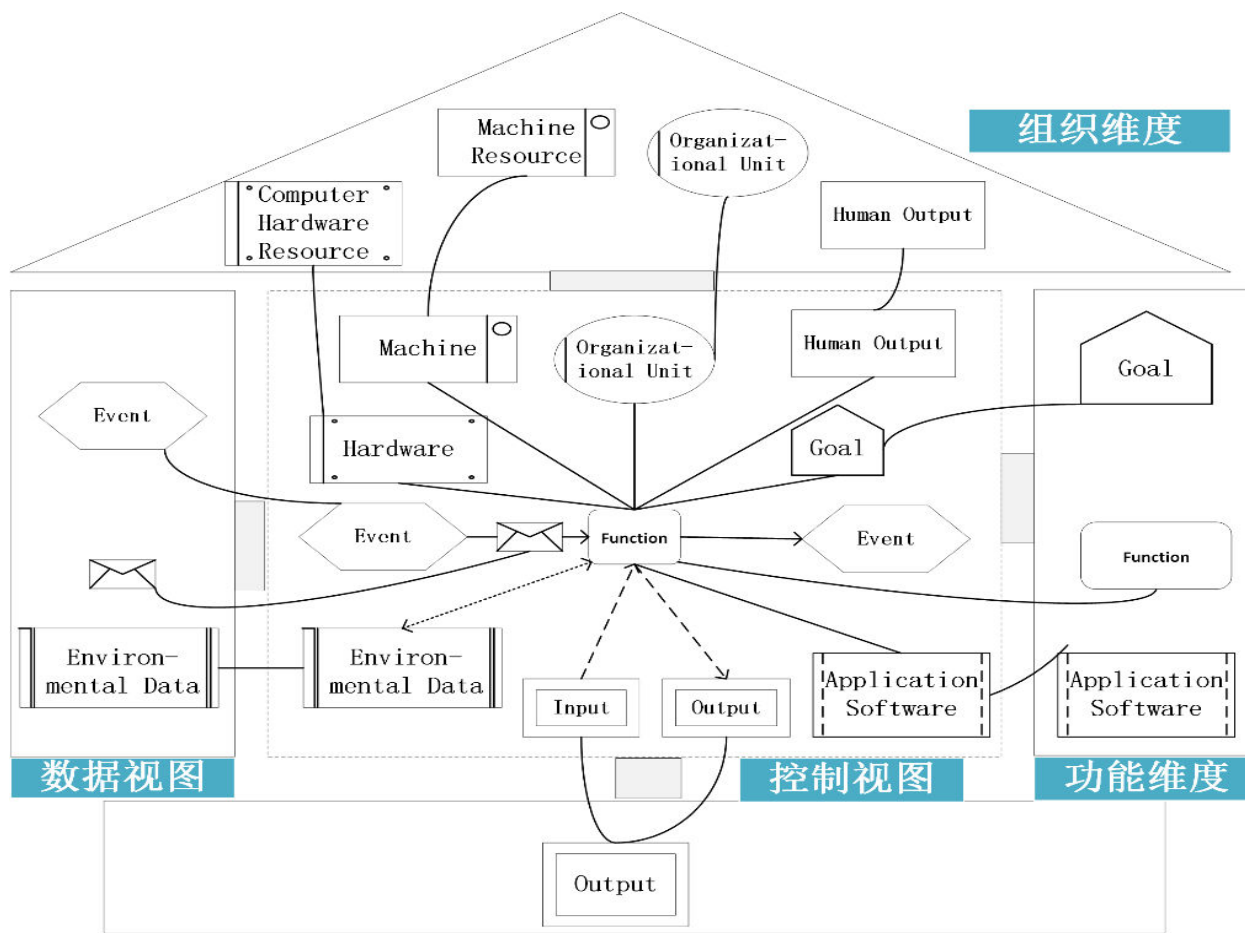
- 基于流程模型，可以实现业务中的控制流、物料流和信息流等有效集成。
- 四种实现过程的不同连接方法的要素比较：

方法	主要组件	连接方式	主要代表模型
基于 <b>数据</b> 的业务流程建模	数据实体、数据活动	数据流	数据流图
基于 <b>活动</b> 的业务流程建模	时序活动	时间系列流	活动图
基于 <b>事件</b> 的业务流程建模	功能活动、事件	事件流	EPC
基于 <b>状态</b> 的业务流程建模	系统状态、状态转移事件	状态变迁流	PetriNet

## 多视图建模方法



- 多视图建模其实是基于过程的业务建模的发展，**以过程模型（ workflow模型）为核心**，其他视图模型（功能视图、信息视图、组织视图、资源视图）为辅助来实现集成化建模。



### (3) 面向服务的建模方法

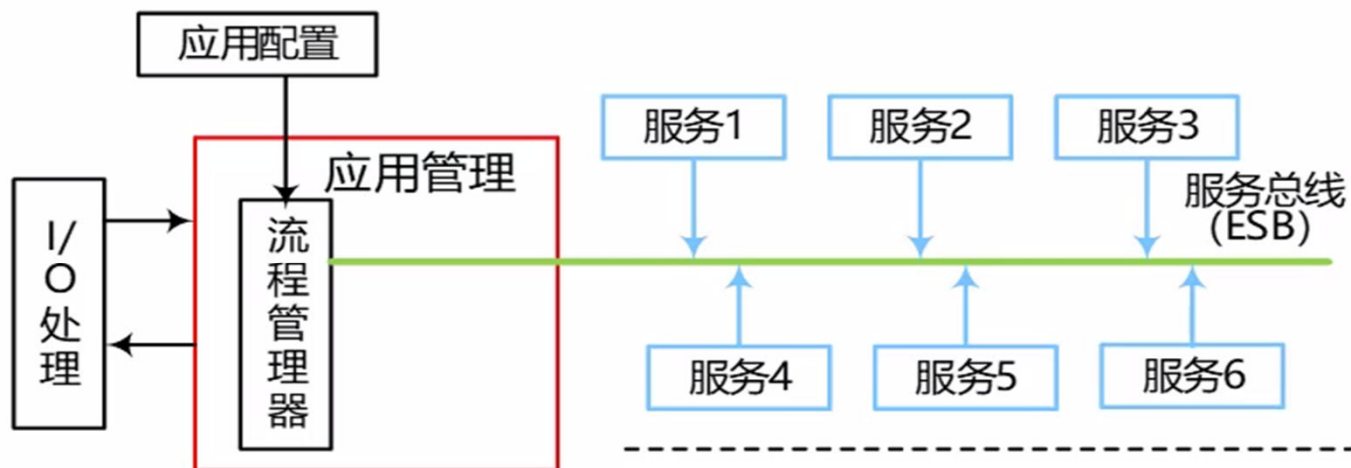


- 面向服务的建模有两个层次，基于面向应用的**服务能力**，或面向**软件实现**开展建模，从服务角度开展业务的分析和设计。
- 基于软件服务构造方面，可以面向服务开展业务建模，业务转换，构造出标准**SOA** 服务或者**RESTful** 服务等方式，实现建模的服务转换及应用。



**正逐渐发展完善成新兴体系!**

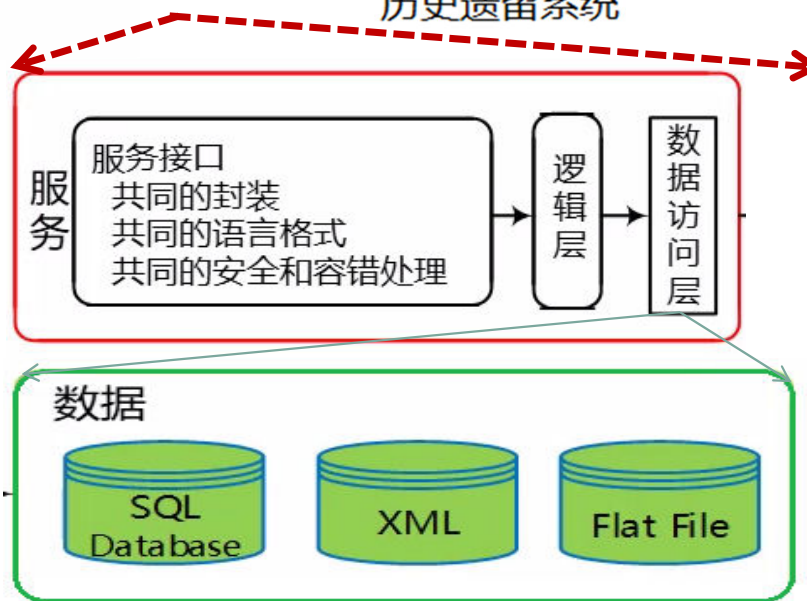
	服务操作	服务提供	服务定义	服务需求
设计	技术信息的发展和交付	法律信息和技术架构的设计和提供	信息的规范化服务协同	用户需求的概念信息
交付	技术信息服务 / 基础设施	系统级功能信息服务供应	信息服务的使用	信息提供者提供服务
评估	服务基础设施和运营绩效	服务提供的有效性及其SLA指标	SLA服务的感知和期望	满足客户需求的信息



历史遗留系统

## 服务设计原则：

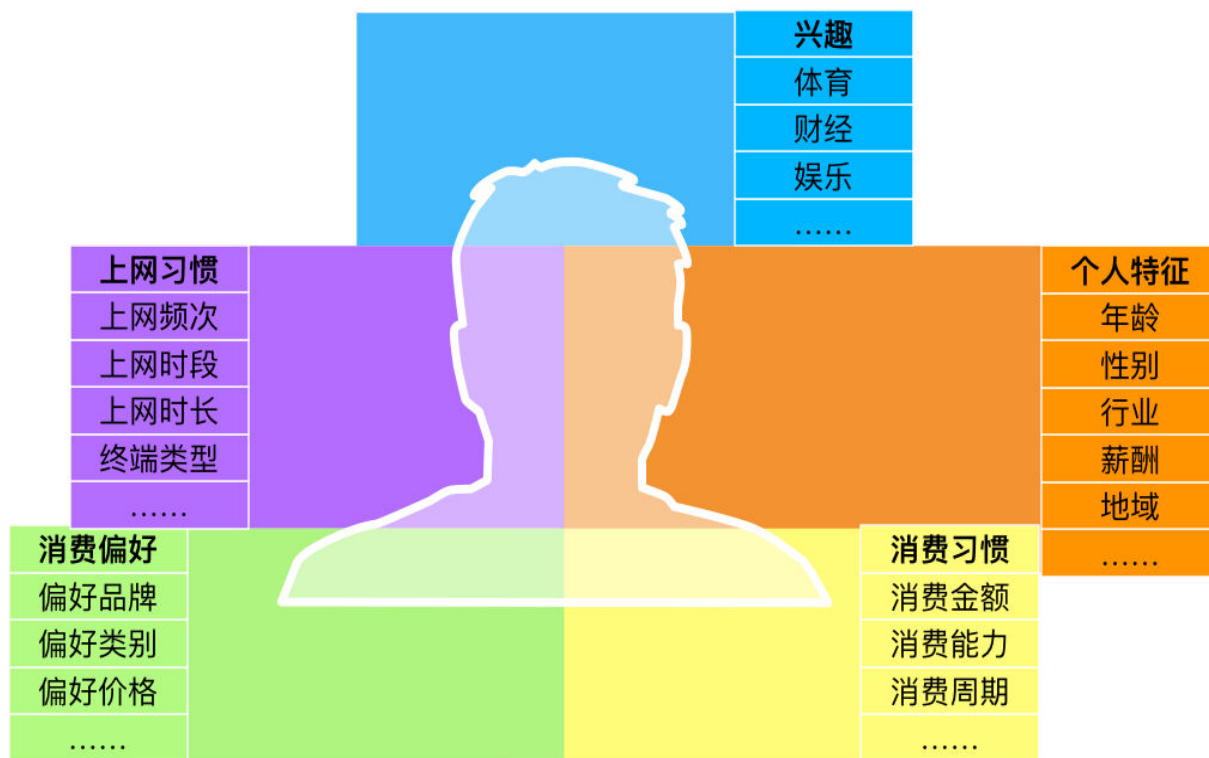
- 明确的接口定义
- 自包含与模块化
- 松耦合
- 粗粒度



## (4) 面向大数据的业务建模方法



- 从目前来说，随着大数据、机器学习技术的兴起，面向大数据的业务建模逐渐成为热点。
- 其本质上大多是基于**核心主体**的多维度**特征**综合分析，例如，基于用户画像的购买行为分析模型，基于地理位置聚类的店铺竞争分析模型等等。



一种面向大数据的信息建模方法是**特征工程**，其核心任务是将原始数据转化为特征，这些特征能够更好地表示预测模型处理的实际问题，提升对未知数据的预测准确性。



- 特征工程是从原始数据出发，开展预处理、转换和选择,以提取出更有价值的特征，并将其转换为适合智能模型处理的格式。
- 好的数据和特征是所有模型和算法发挥到极致的前提。
- 特征工程主要包括特征选择、特征获取、特征处理和特征监控等方面，主要步骤也根据不同场景有所差异。一般来说可以包括：
  - **数据预处理**：包括处理缺失值、离散特征的连续化处理等。例如，可以使用均值或中位数填充连续值的缺失值，使用独热编码处理离散值。
  - **特征选择**：通过统计测试选择性能最佳的特征，以消除噪声影响并加速计算。特征选择技术包括过滤、嵌入和包裹方法。
  - **特征处理**：特征进一步处理及降维等，例如采用降维技术通过减少特征的维度来提高模型的效率和准确性。常用的降维技术包括主成分分析（PCA）和线性判别分析（LDA）。
  - **特征监控**：是对重要特征数据进行监控及优化，防止特征质量下降影响模型效果。

➤ 按照数据**采集**、**治理**、**存储**、**计算**、**应用**（显控）流程构造的大数据平台。

