软件系统分析与设计

第一讲　系统+设计

饶元

社会智能与复杂数据处理实验室

西安交通大学软件学

2021.9

参考文献：

系统分析设计（2012版）， 饶元，西安交通大学软件学

面向对象技术高级课程，邵维忠，北京大学信息科学与技术学院

Object-Oriented Analysis and Design, Christian Kastner， CMU

Christian Kästner Charlie Garrod

目录

- 什么是系统？
- 系统分析之责
- 系统设计之要
- 软件系统分析与设计的主要方法？
- 本课程的主要任务
- 参考书目
- 学习方法与考试要求

系统

关于系统

- 系统是相互联系相互作用的诸元素的综合体。

抽象定义：

如果对象集S满足下列两个条件：

（1）S中至少包含两个不同元素

（2）S中的元素按一定方式相互联系

则称S为一个系统，S的元素为系统的组分。

一是多元性，系统是多样性的统一，差异性的统一；

二是相关性，系统不存在孤立元素组分，所有元素或组分间相互依存、相互作用、相互制约；

三是整体性，系统是所有元素构成的复合统一整体。

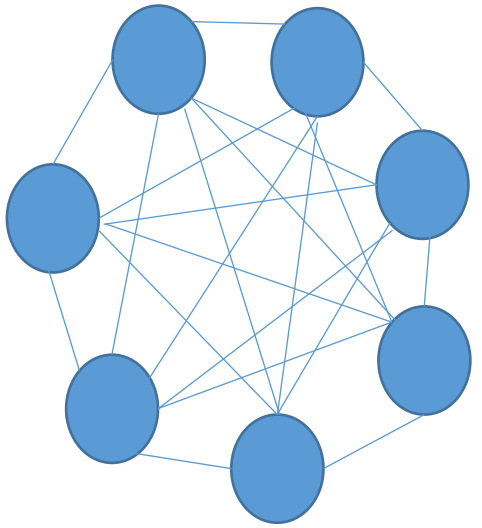
贝塔朗菲（1901～1972），美籍奥地利生物学家，一般系统论和理论生物学创始人，50年代提出抗体系统论以及生物学和物理学中的系统论，并倡导系统、整体和计算机数学建模方法和把生物看作开放系统研究的概念，奠基了生态系统、器官系统等层次的系统生物学研究。

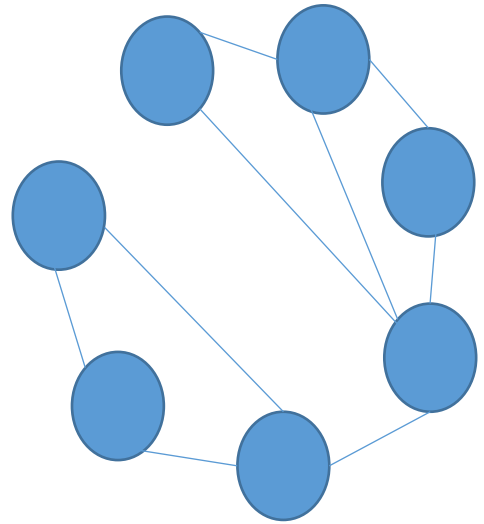
关于系统

- 系统科学是从事物的整体与部分、全局与局部以及层次关系的角度来研究客观世界的。
- 客观世界包括自然、社会和人自身。
- 能反映事物这个特征最基本和最重要的概念就是系统。
- **所谓系统是指由一些相互关联、相互作用、相互影响的组织部分构成并具有某些功能的整体。**
- 系统在客观世界中是普遍存在的。

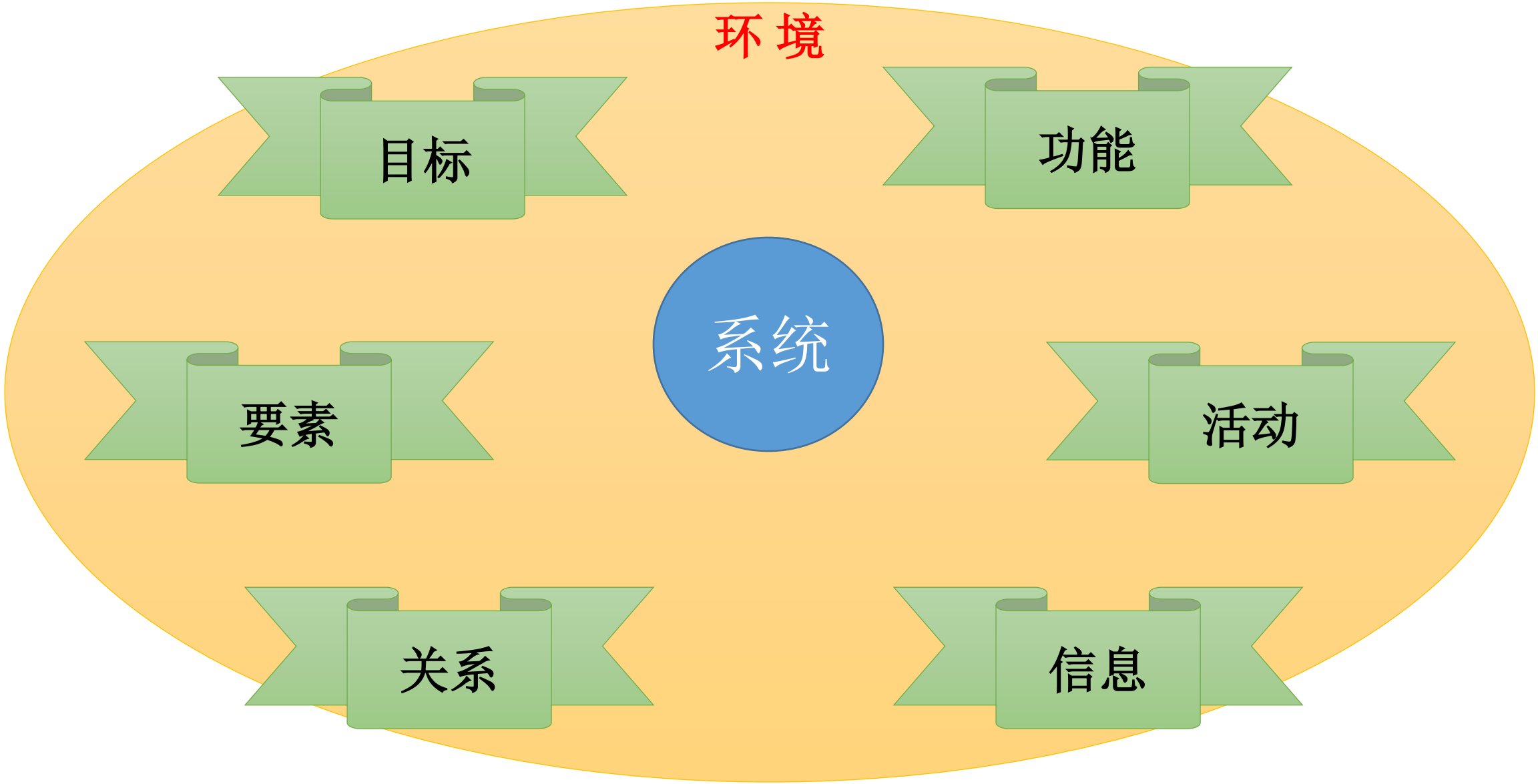
最佳定义：

- 系统是指由相互联系、相互作用的若干组成部分构成的有机整体，这个整体具有其各个组成部分所没有的新性质和功能，并和一定的环境发生交互作用。
- 系统各要素之间、要素与整体之间，以及整体与环境之间，存在着一定的有机**联系**，从而在系统的内部和外部形成一定的**结构和秩序**。
  - 要素是指组成系统的基本成分，是系统形成的基础。要素和系统的关系，是部分与整体的关系，它们互相联系，互相作用。
  - 功能是指系统与外部环境在相互联系和作用的过程中所产生的效能。
  - 活动是系统形成、发展、变化的动态过程，这个过程通过系统内部诸要素之间、要素与系统之间以及系统与环境之间相互影响、相互作用而完成的。
  - 信息是指事物存在的方式或运动状态以及这些方式、状态的直接或间接的传播与表述。
  - 环境是指处于系统边界之外并和系统进行着物质、能量和信息交换的所有事物。





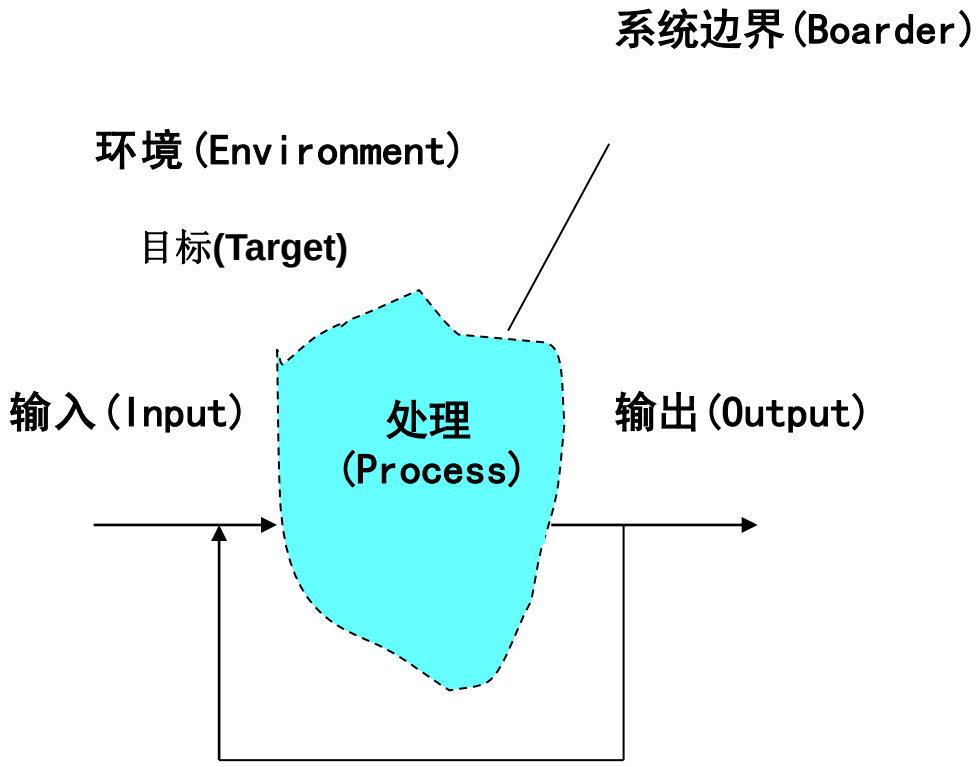
系统的特征



软件系统核心要素

•软件系统： 指在一定的软件开发与应用环境下，为了达到某一目的而相互联系、相互作用的若干个软件要素所组成的有机整体。

- 软件系统要素
  - 系统环境(Environment)
  - 目标(Target)
  - 边界(Boundaries)、
  - 系统架构(System Architecture)
  - 子系统(Subsystem)
  - 输入/输出(Input/Output)
  - 接口(Interface)
  - 组成实体/对象(Element)
  - 实体/对象关系(Relationship)



关注微观、不断细分科学对象：分子、夸克、DNA……

1 + 1 < 2

还原论

整体论

关注宏观、交叉融合、整体特性

1 + 1 =1

两种科学方法论

简单巨系统

简单系统

复杂系统

系统分类

特殊复杂巨系统

复杂巨系统

系统科学

人脑思维

系统论（钱学森）

宏观与微观的结合

1 + 1 >>2

以人为本

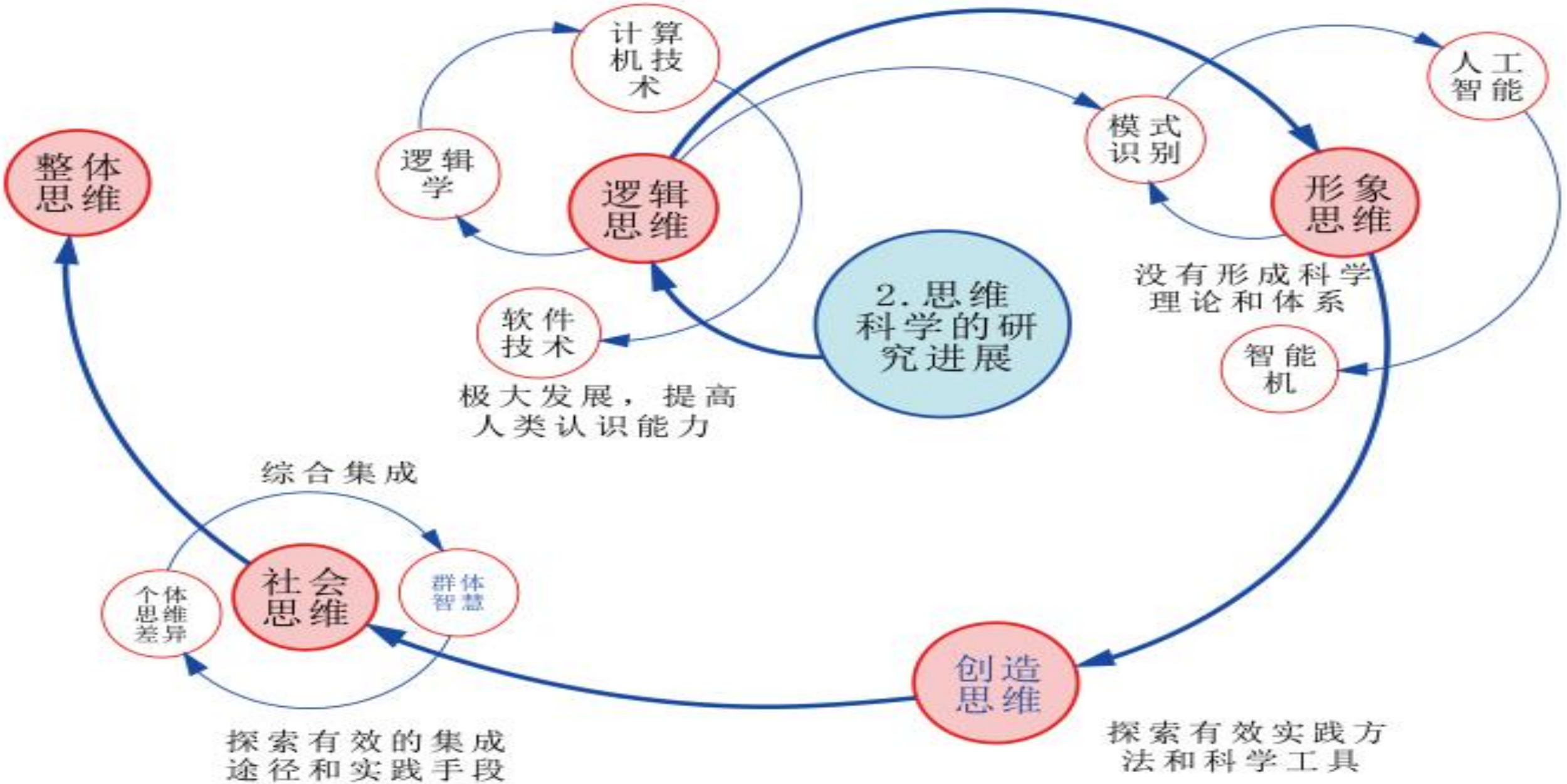
人机结合

综合集成方法（钱学森）

从定性到定量

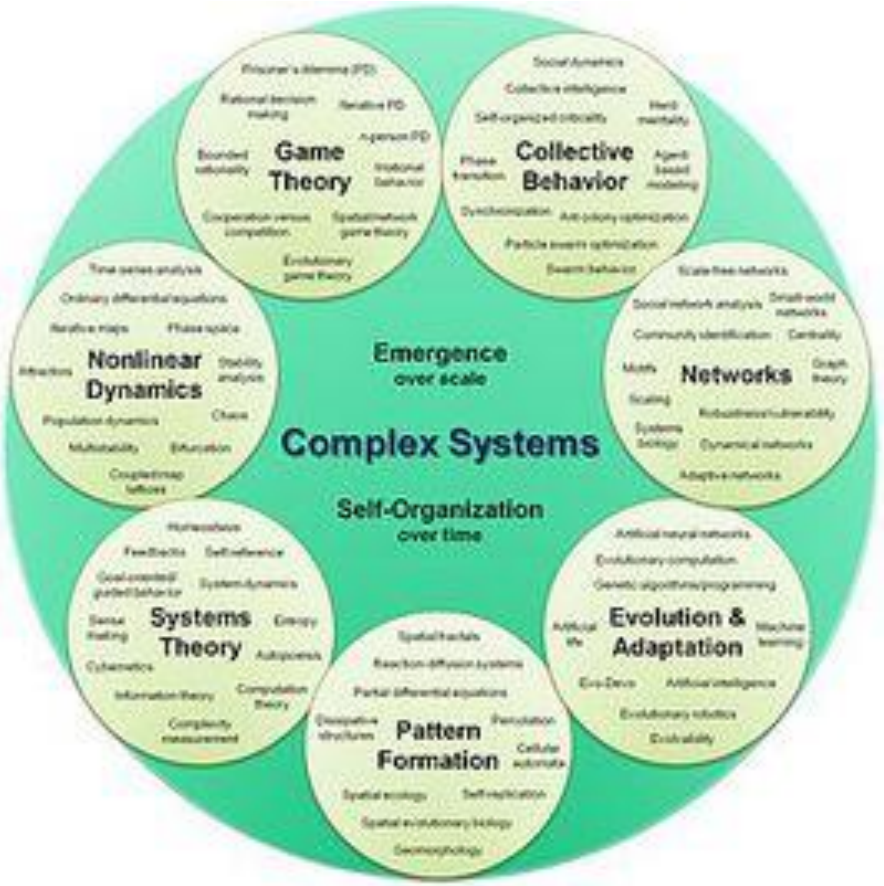
1+1>2 涌现

- 群体的需求是一种涌现现象（EmergenceTheory），即整体大于部分之和，这种高层次具有的属性、特征、行为和功能还原到低层次就不复存在。
- 自然界中，涌现现象无处不在。蜜蜂个体的力极低，但蜂群会涌现出极高的工程智慧和生存策略。大雁列阵飞行并不是因为他们故意为之，是因为前面大雁拍打翅膀造成的尾波乱流，能让后面的大雁获得更多的升力，于是大雁群体自然涌现出了一字型雁阵。类似的还有鱼群、鸟群、羊群等。
- 到现实中，对应的就是人群的需求。品牌的消费者个体只想买到自己需求的商品，消费者群体却会涌现出对价值观、文化、品味的需求。参与组织协作的个体，只想要明白自己要做的事和目标，而整个协作体系却需要共识和方向。

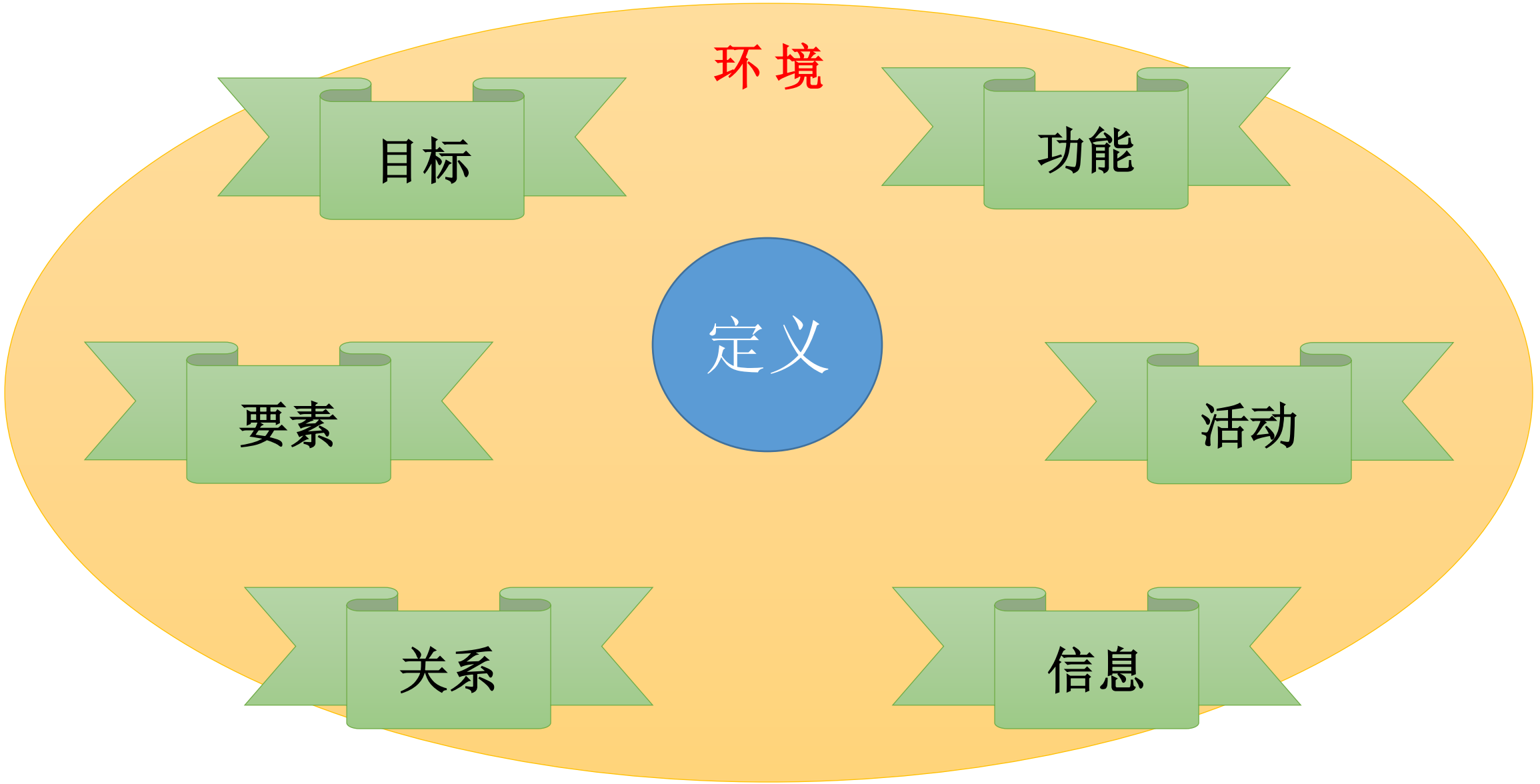

计算机技术
逻辑学
逻辑思维
模式识别
人工智能
形象思维
整体思维
2.思维科学的研究进展
软件技术
极大发展，提高人类认识能力
没有形成科学理论和体系
智能机
综合集成
社会思维
个体维异思差
群体智慧
创造思维
探索有效的集成途径和实践手段
探索有效实践方法和科学工具

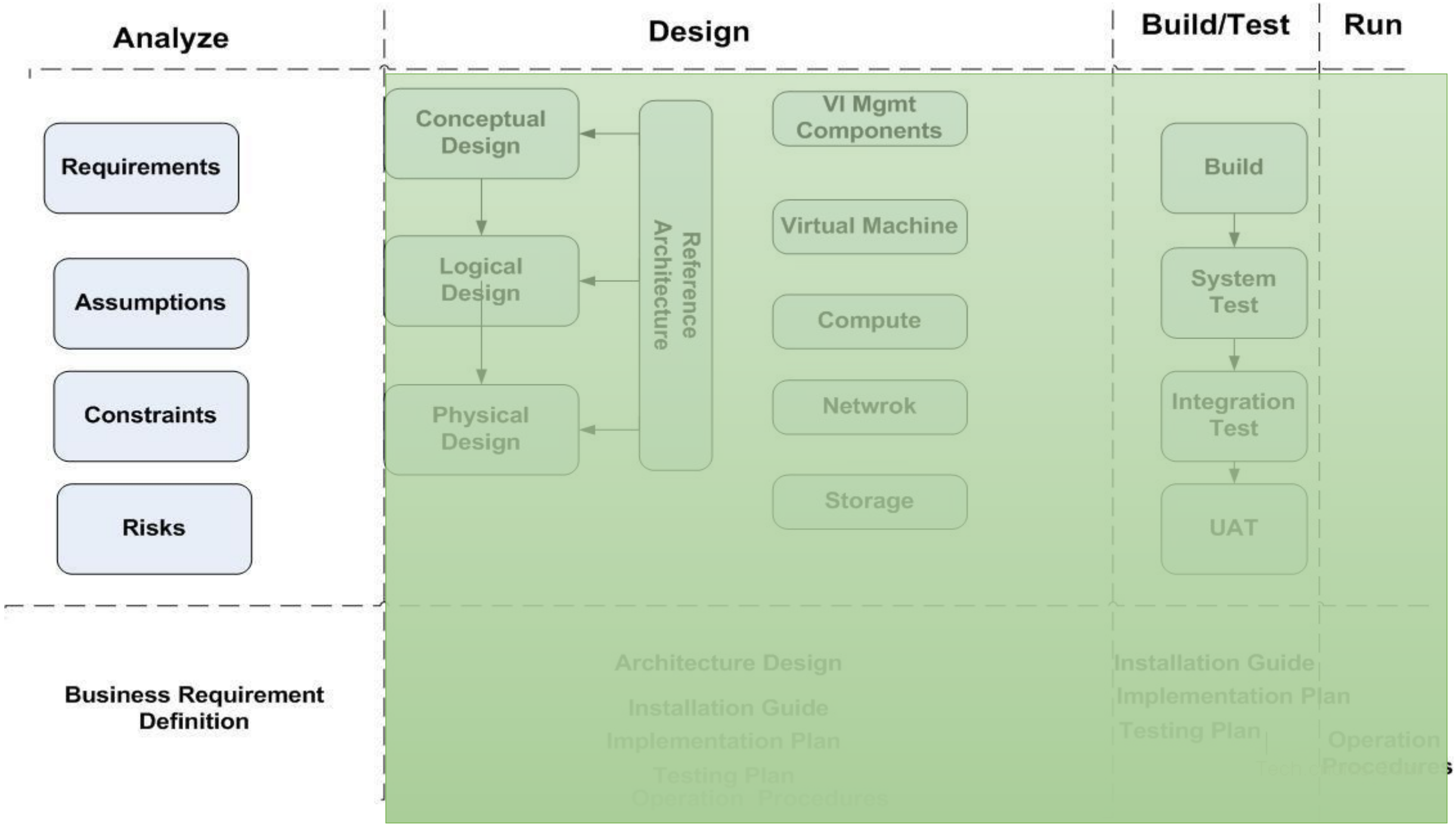
目录

- 什么是系统?
- 分析之责
- 设计之要
- 软件系统分析与设计的主要方法?
- 本课程的主要任务
- 参考书目
- 学习方法与考试要求



系统分析任务：




Analyze
Design
Build/Test
Run
Requirements
Assumptions
Constraints
Risks
Conceptual Design
Logical Design
Physical Design
Reference Architecture
VI Mgmt Components
Virtual Machine
Compute
Netwrok
Storage
Build
System Test
Integration Test
UAT
Business Requirement Definition
Architecture Design
Installation Guide
Implementation Plan
Testing Plan
Operation Procedures
Installation Guide
Implementation Plan
Testing Plan
Operation Procedures

系统分析

- **客户需求**
- 在需求分析阶段，需要挖掘出客户**真正在乎的需求**，最好对需求进行分优先级，不能眉毛胡子一把抓。
- 而且需求**并不是一成不变**，项目过程中增减需求是平常的事，但由此造成的影响要评估并更新文档。
- 有时项目组需要和客户协商撤销或者推后某些需求。
- 可能的原因有：造成整个方案成本大幅上升；与其它关键需求冲突；可能造成项目延迟等等。

系统分析

- **假设的条件**
- 有时在项目执行中会因为某些需要客户或第三方完成的事情不具备，而造成项目延迟。
- 这就需要在合同中就对这些假设特别说明，以避免后面的责任不清。
  - 比如假设客户网络环境是可以支持你的设计的，可实施时才发现上行网络的防火墙的带宽或端口限制会大大影响你的方案的性能。
  - 比如假设你需要使用客户已有的数据库，却发现版本和你的方案不兼容。

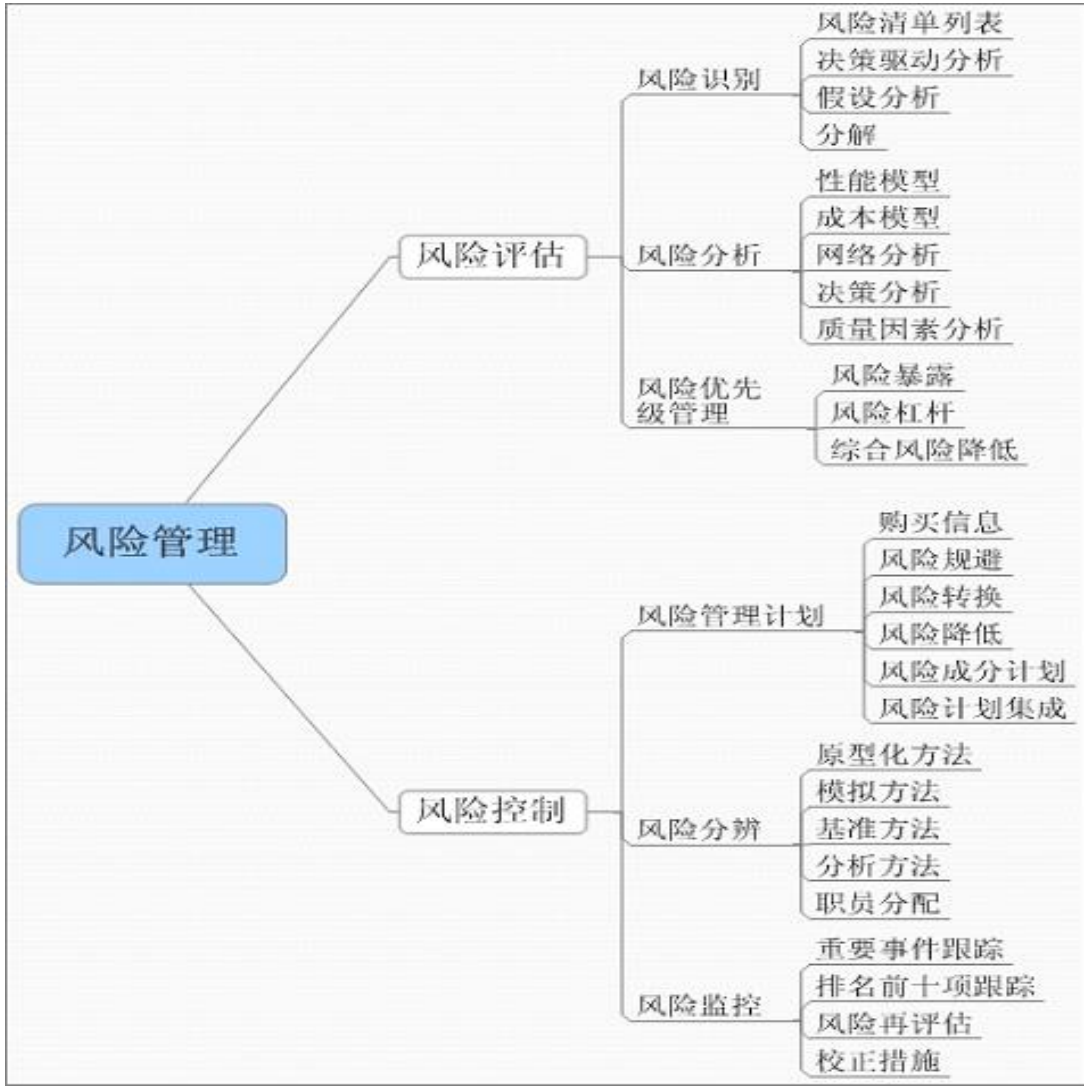
系统分析

- **环境的限制**
- 这点尤其重要，却常常在分析阶段容易被忽略。
- 尽可能挖掘限制条件，会避免后面阶段很多的问题:
  - 比如客户已经在使用NFS，并且现有维护人员有能力维护该系统。你在推荐SAN的时候就要考虑带来的影响；
  - 若客户与某大供应商有协议，你是否可以考虑其它厂商？
  - 若客户有较严格的安全性策略，设计共享时要考虑哪些部分是不可以共享的，是否需要虚拟层的防火墙等等。


12306_百度搜索
铁路客户服务中心
模因论_百度百科
(2268封未读) 网易邮箱6
无法访问 http://12306.cr
www.12306.cn/mormhweb/
应用
百度
百度一下
打开新的标签页
车辆相关
algorithm trading
其他书签
客服热线：12306
中国铁路客户服务中心
www.12306.cn是中国铁路客户服务中心唯一网站。截止目前，没有授权其他网站开展类似服务内容，敬请广大用户注意
2017年3月13日 星期一
首 页
客运服务
货运服务
行包服务
旅游超市
车站引导
铁路常识
站车风采
客户信箱
最新动态
为保障您顺畅购票，请下载安装根证书。
哈铁国旅
12306平常一天的PV（page views）值大约是在 2500万到 3000万左右， 在2015年春运高峰日的PV值是297亿，流量增加1000倍
这样海量的请求，假如不能在短时间内动态调整网络带宽或增加服务器数量，就会造成网络阻塞或是服务器性能无法满足要求，甚至使整个系统不稳定。
我的保险
退票
余票查询
旅客列车时刻表查询
出发地
目的地
3月13日
3月14日
3月15日
3月16日
北京
哈尔滨
5713
9524
9560
9301
北京
齐齐哈尔
2266
4034
4283
5188
北京
佳木斯
1350
1926
2127
2156
主要营业站受理服务电话
货运运费查询
货物追踪
相关链接

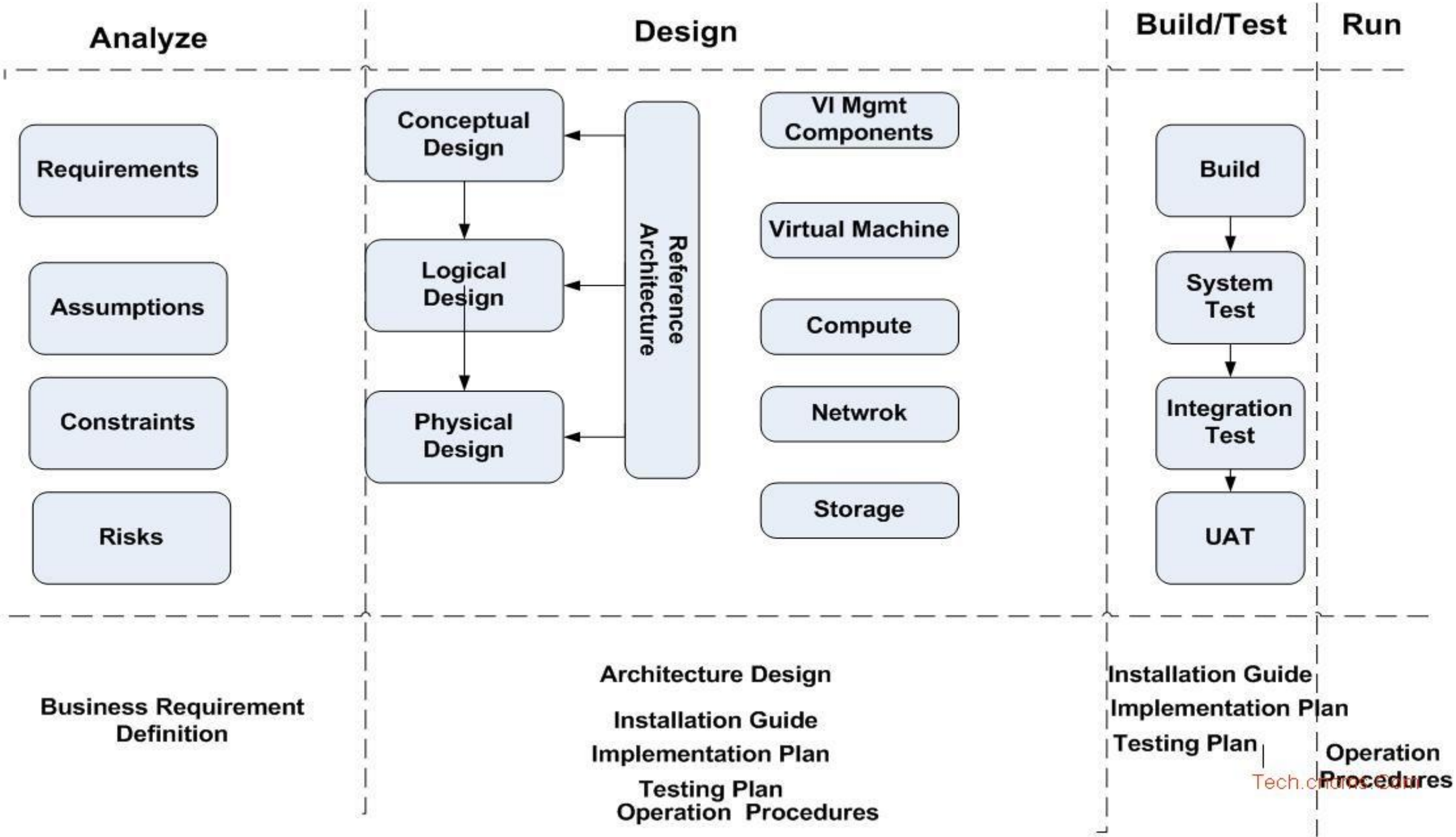
系统分析

- 风险
- 在开发新的软件系统过程中，由于存在许多不确定因素，软件开发失败的风险是客观存在的。
- 风险分析对于软件项目管理是决定性的。
- 风险分析实际上就是贯穿在软件工程过程中的一系列风险管理步骤，其中包括：风险识别、风险估计、风险管理策略、风险解决和风险监督等。

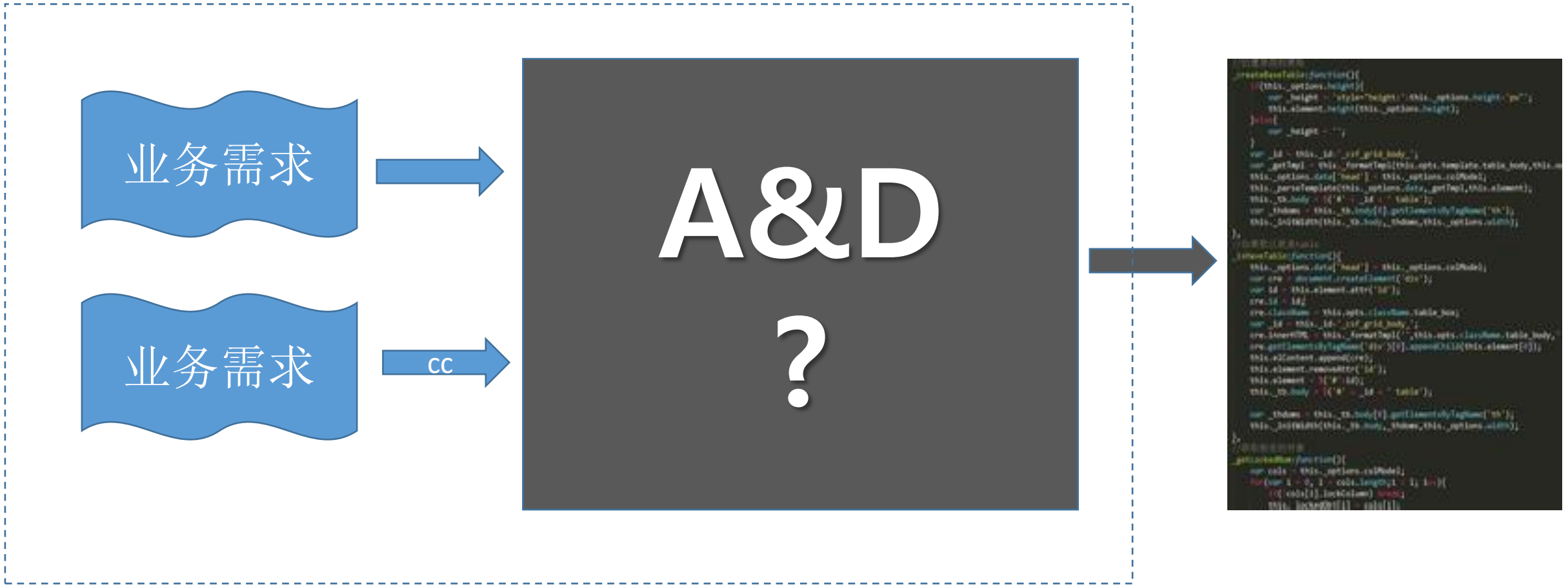


目录

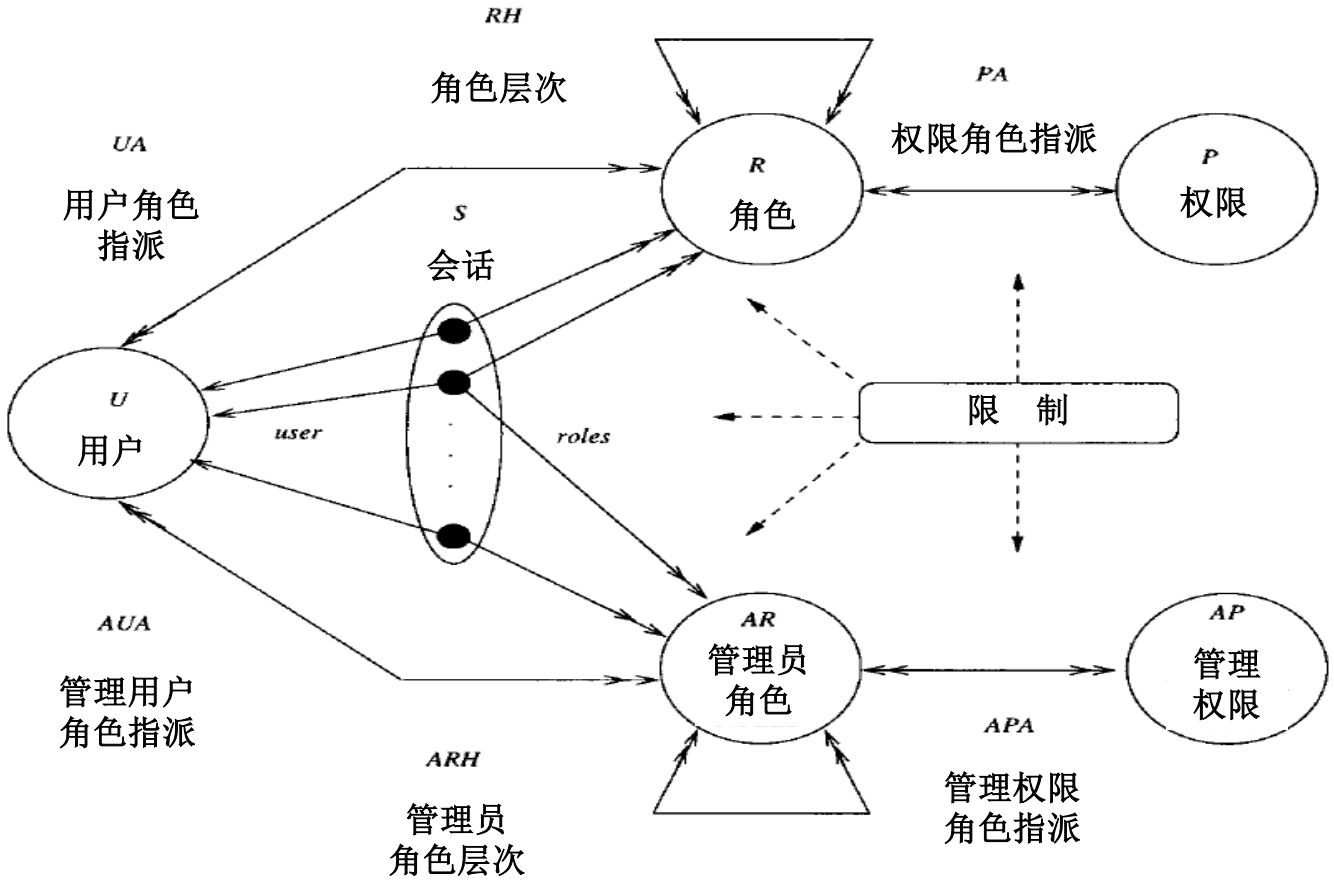
- 什么是系统?
- 分析之责
- **设计之要**
- 软件系统分析与设计的主要方法?
- 本课程的主要任务
- 参考书目
- 学习方法与考试要求

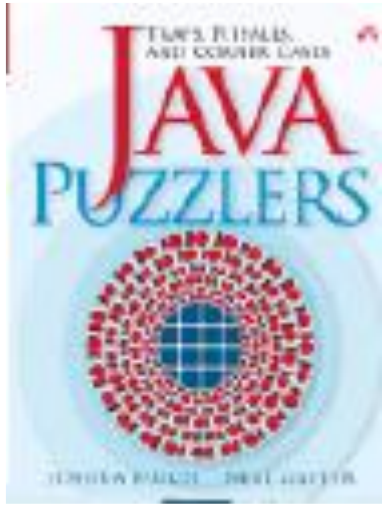


设计之要：模型驱动 ⟶ 灵活且可扩展



用户关注点分离


RH
角色层次
PA
权限角色指派
UA
用户角色
指派
R
角色
P
权限
S
会话
U
用户
user
roles
限 制
AUA
管理用户
角色指派
AR
管理员
角色
AP
管理
权限
APA
管理权限
角色指派
ARH
管理员
角色层次




```
int a = 010 + 3;
System.out.println("A" + a);
```

Sorting with configurable order, variant A

```
void sort(int[] list, String order) {
    …
  boolean mustswap;
  if (order.equals("up")) {
    mustswap = list[i] < list[j];
  } else if (order.equals("down")) {
    mustswap = list[i] > list[j];
  }
  …
}
```

Sorting with configurable order, variant B

```
void sort(int[] list, Comparator cmp) {
    …
   boolean mustswap;
   mustswap = cmp.compare(list[i], list[j]);
   …
}
interface Comparator {
   boolean compare(int i, int j);
}
class UpComparator implements Comparator {
   boolean compare(int I, int j) { return i<j; }}

class DownComparator implements Comparator {
   boolean compare(int I, int j) { return i>j; }}
```

Sorting with configurable order, variant A

```
void sort(int[] list, String order) {
   …
  boolean mustswap;
  if (order.equals("up")) {
    mustswap = list[i] < list[j];
  } else if (order.equals("down")) {
    mustswap = list[i] > list[j];
  }
  …
}
```

设计中的约束

如何选择设计

Sorting with configurable order, variant B

```
void sort(int[] list, Comparator cmp) {
   …
  boolean mustswap;
  mustswap = cmp.compare(list[i], list[j]);
  …
}
interface Comparator {
  boolean compare(int i, int j);
}
class UpComparator implements Comparator {
  boolean compare(int I, int j) { return i<j; }}

class DownComparator implements Comparator {
  boolean compare(int I, int j) { return i>j; }}
```

```
class Program
    {
        static void Main(string[] args)
        {
        }
    }

class Alerter
    {
        public Alerter(Car car)
        {
Car.Notify(N
        }

        public void NotEnoughPetrol(int value)
        {
            Console.ForegroundColor =
Console
value.ToString() + " gallon petrol left!");
            Console.ResetColor();
        }
    }
```

```
class Car
    {
        public delegate void Notify(int value);
        public event Notify notifier;
{
    petrol = value;
    if (petrol < 10)  //当petrol的值小于10时，出发警报
    {
        if (notifier != null)
        {
            notifier.Invoke(Petrol);
        }
    Petrol = petrol;
}
public void Run(int speed)
{
    int distance = 0;
    Petrol--;
    distance += speed;
    Console.WriteLine("Car is running... Distance is "
                      + distance.ToString());
}
}
```

问题1：为什么不在public int Petrol中直接调用Alerter.NotEnoughPetrol呢？

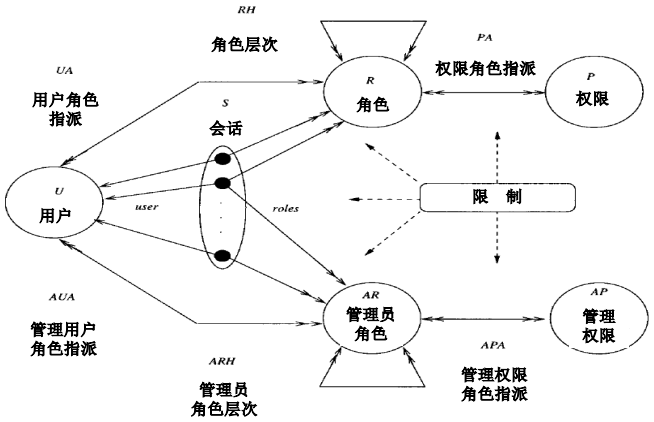
问题2：关于事件？

声明事件：若要在类内声明事件，首先必须声明该事件的委托类型。

问题3：观察者模式（观察者模式又称Source/Listener模式）的实现

设计原则：

- 为不同的用户而设计
- 为变化而设计
- 为重用而设计

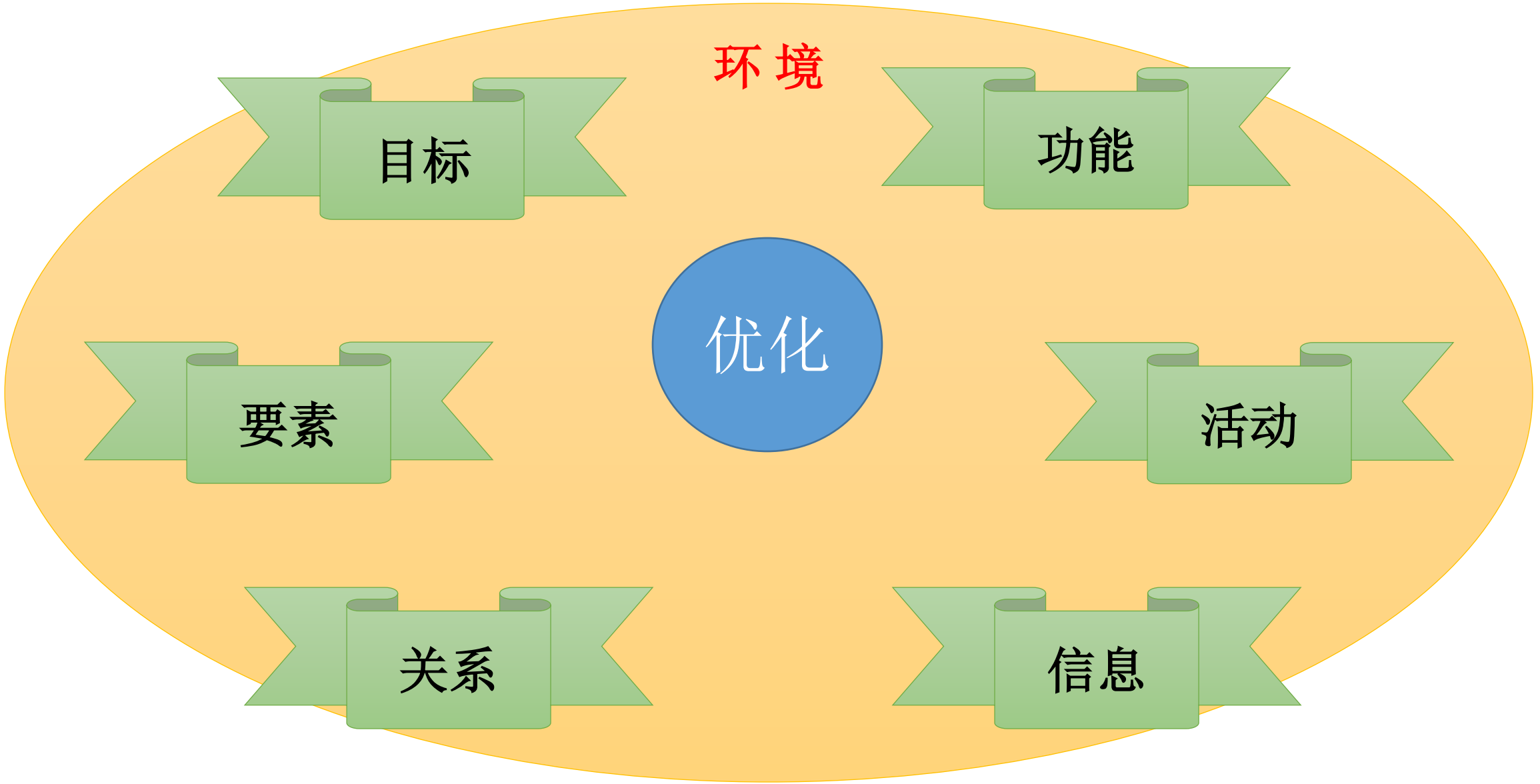


清晰的边界
明确的接口
信息隐藏
隐藏可能的变化

- Design Goals
  - Design for Change
  - Design for Division of Labor
- Design Principles
  - Explicit Interfaces (clear boundaries)
  - Information Hiding (hide likely changes)
- Design Patterns
  - Strategy Design Pattern
  - Composite Design Pattern
- Supporting Language Features
  - Subtype Polymorphism
  - Encapuslation

设计模式
语言的特征
多态（subtype Polymorphism)
封装：（Encapuslation）

系统设计任务：



案例

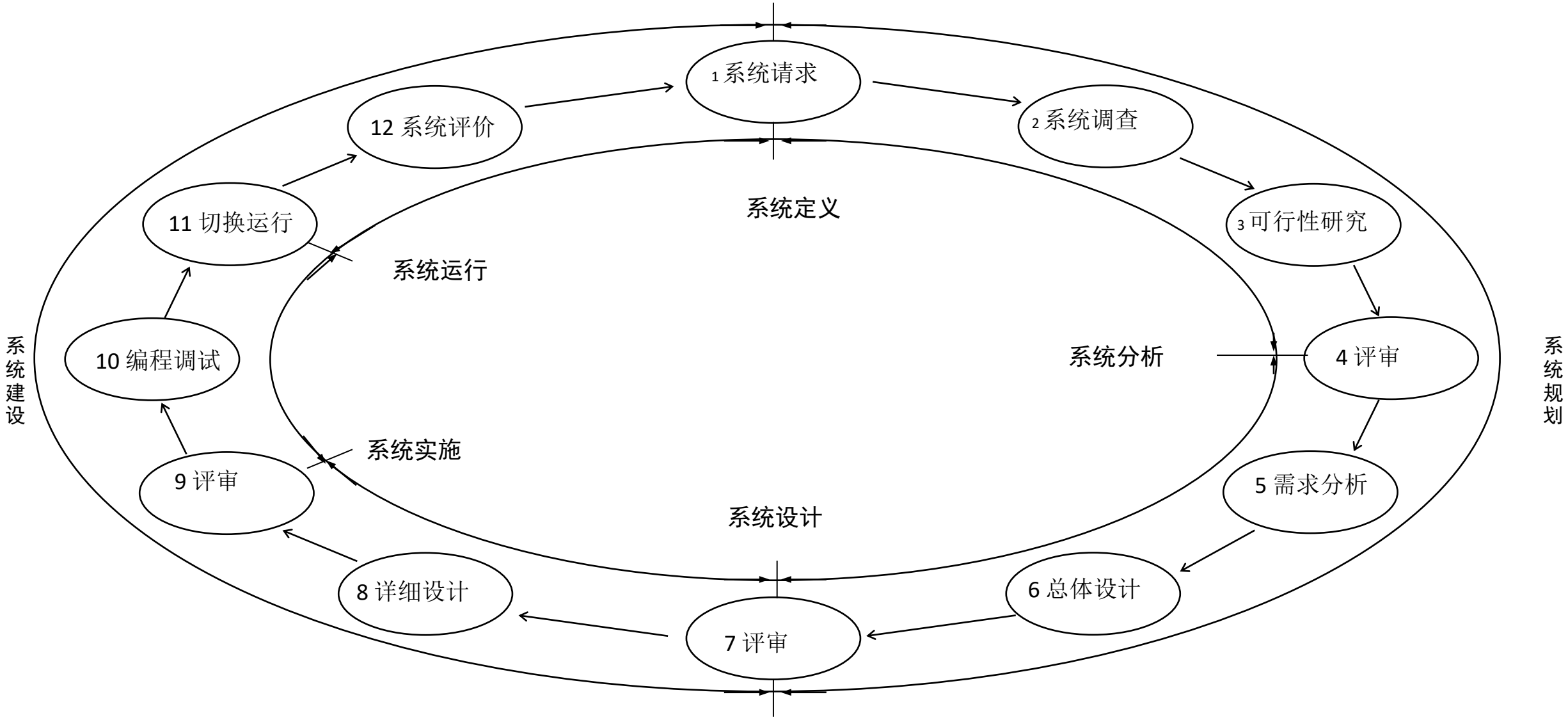

网格化管理系统
一图总览
12:53:01
兴华街道办事处
10/13
2
2
58
重点服务对象
三维地图
平面地图
周边查询


地
人
组织
事件
物
社会管理创新
实有人口
特殊人口
重点人群
常住人口
租（借）人口
办公商用
公共设施
道路交通设施
房屋土地
基础设施
相关企事业单位
基层综治组织
社团组织
“两薪”组织
群防群治力量
社会治安防控
特殊人群服务
重点人群管理
网络事件处理
突发事件处理
社会矛盾调解

目录

- 什么是系统?
- 分析之责
- 设计之要
- **软件系统分析与设计的主要方法**
- 本课程的主要任务
- 参考书目
- 学习方法与考试要求

系统开发生命周期


1 系统请求
2 系统调查
3 可行性研究
4 评审
5 需求分析
6 总体设计
7 评审
8 详细设计
9 评审
10 编程调试
11 切换运行
12 系统评价
系统定义
系统分析
系统设计
系统实施
系统运行
系统规划
系统建设

基本的软件分析与设计方法

1 功能分解法（function decomposition）

以系统需要提供的功能为中心来组织系统。

首先定义各种功能，然后把功能分解为子功能

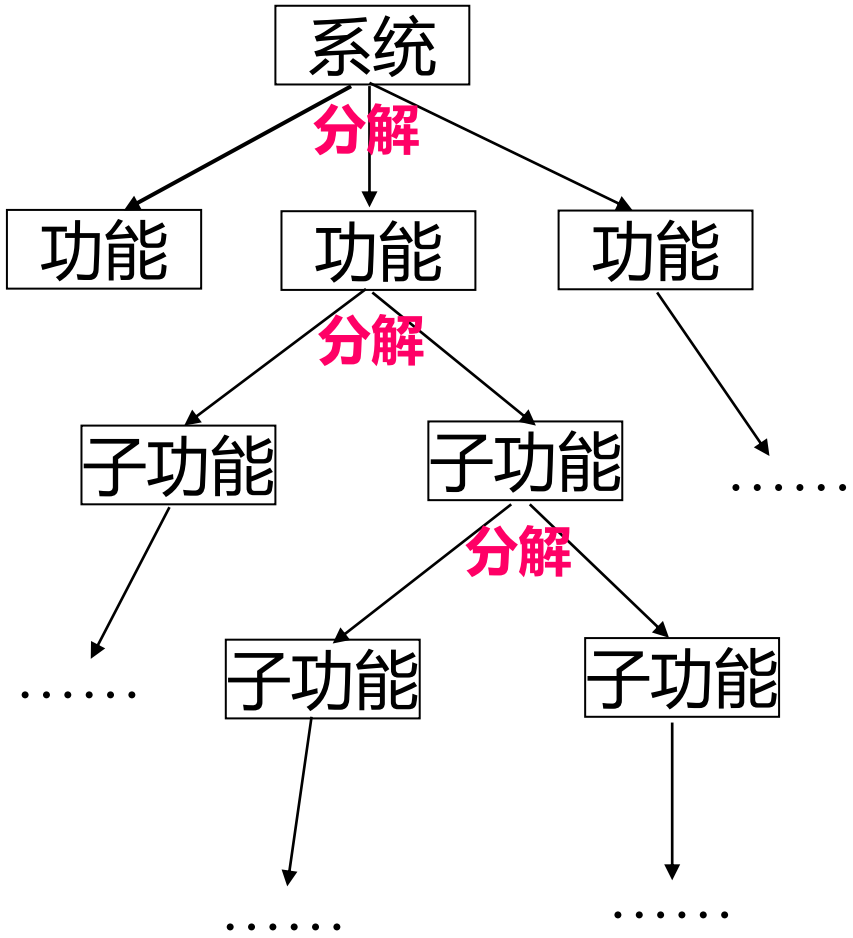
对较大的子功能进一步分解，直到可给出明确的定义

设计功能 / 子功能所需要的数据结构

定义功能/子功能之间的接口。

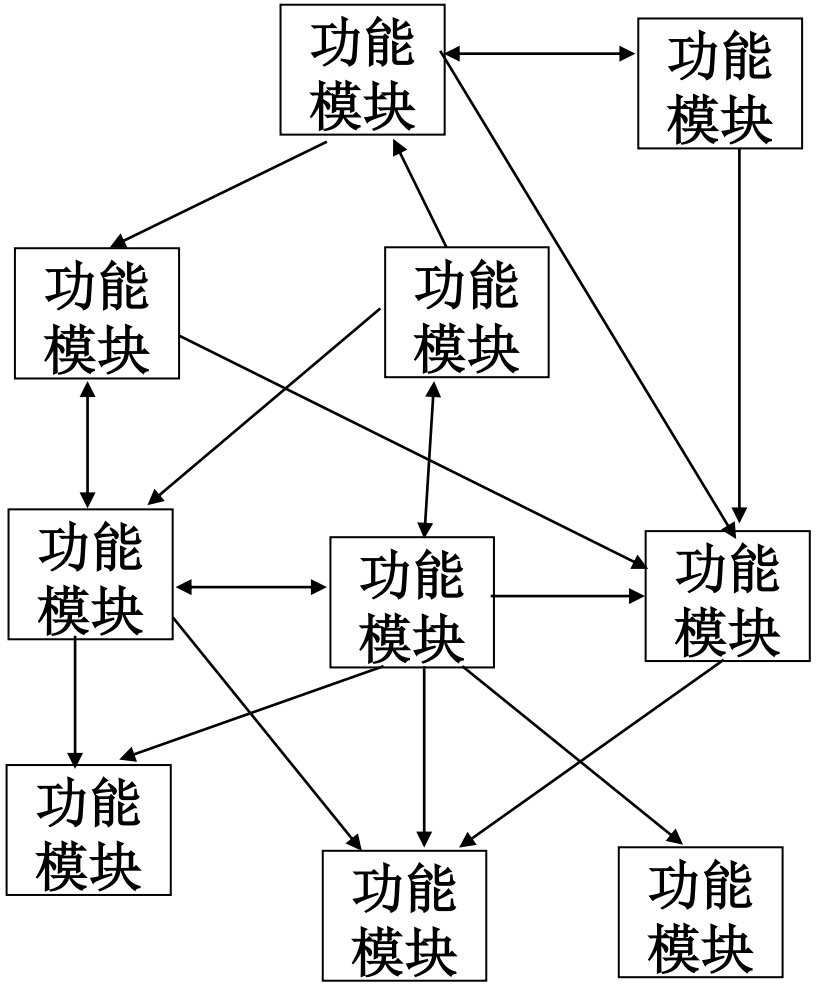
作为一种早期的建模方法，没有明确地区分分析与设计

建模过程：
层层进行功能分解



功能分解法得到的系统模型

由模块及其接口构成



容易产生错误与偏差

优点与缺点：

直接地反映用户的需求

所以工作很容易开始

不能直接地映射问题域

很难检验结果的正确性

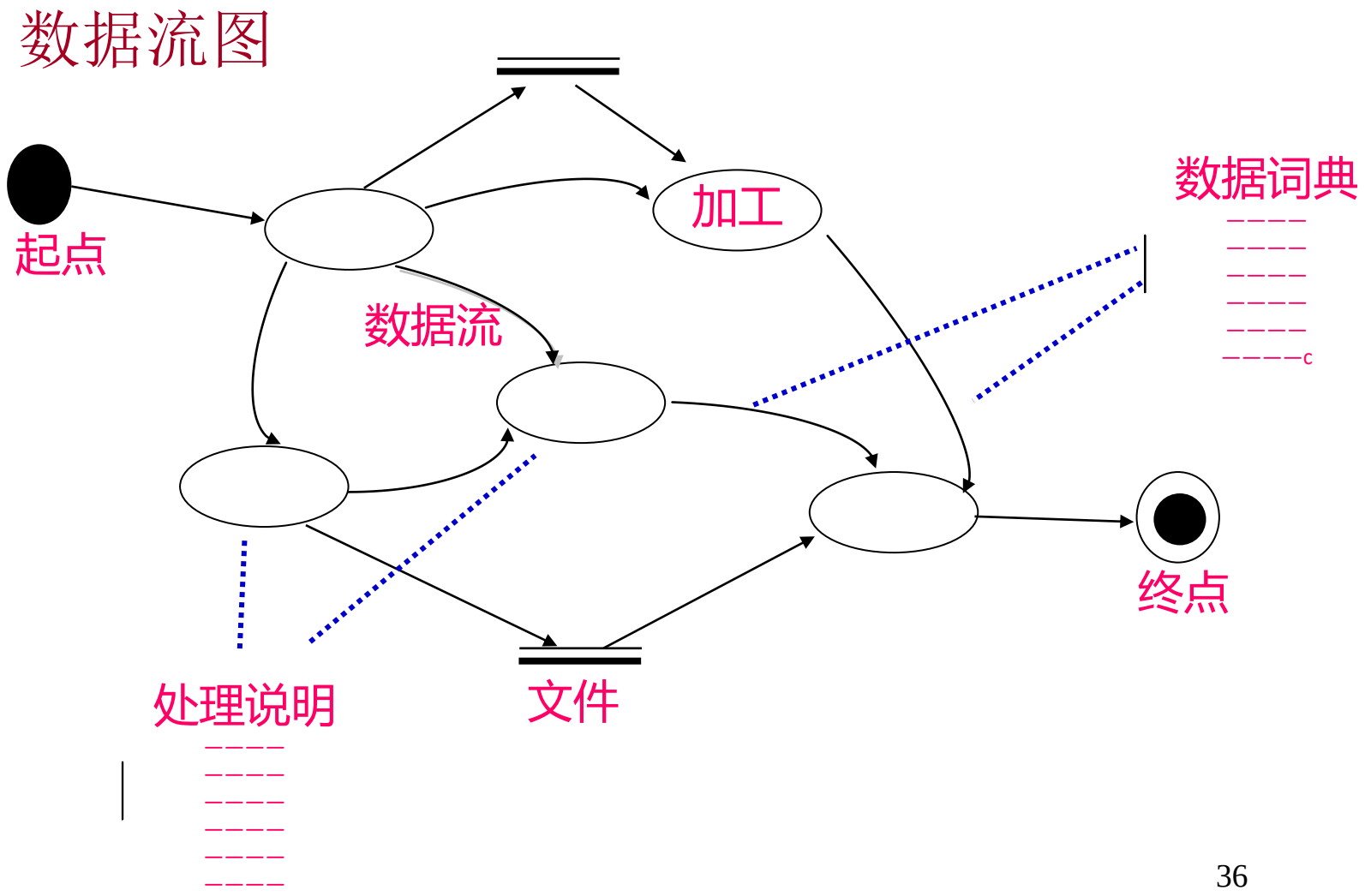
对需求变化的适应能力很差

局部的错误和修改很容易产生全局性的影响。

2 结构化方法

结构化分析（structured analysis，SA）
结构化设计（structured design，SD）

结构化分析又称数据流法，其基本策略是跟踪数据流，即研究问题域中数据如何流动，以及在各个环节上进行何种处理，从而发现数据流和加工。得到的分析模型是数据流图（DFD），主要模型元素是数据流、加工、外部实体及存储，外加处理说明和数据字典。



结构化设计与功能分解法基本相同，基于模块的概念建立设计模型，分为概要设计和详细设计。

概要设计：确定系统中包含哪些模块以及模块之间的调用关系，得到模块结构图（MSD）。

详细设计：描述每个模块内部的数据结构和操作流程。

结构化方法的优缺点

优点：

强调研究问题域，并且有严格的法则。

缺点：

仍然是间接映射问题域；

分析与设计的概念不一致，从分析到设计的过渡比较困难；

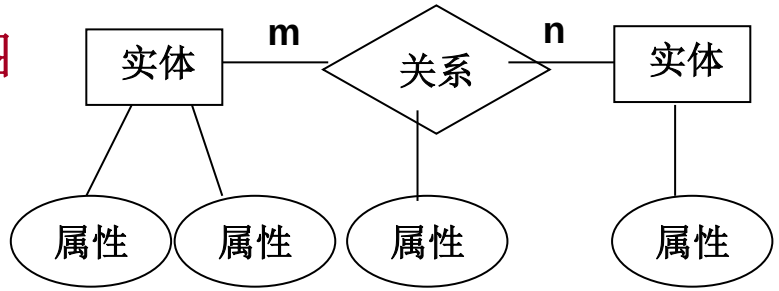
数据流和加工的数量太多，引起分析文档的膨胀。

3 信息建模法（information modeling）

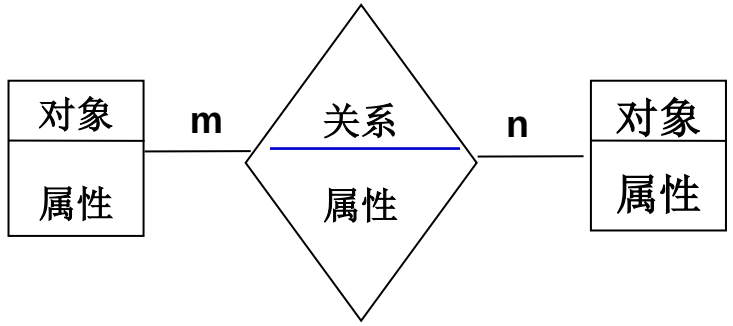
核心概念是实体和关系。实体描述问题域中的事物，关系描述事物之间在数据方面的联系，都可以带有属性。

发展之后的方法也把实体称作对象，并使用了类型和子类型的概念，作为实体（对象）的抽象描述。

E-R 图



信息模型



信息建模法已经很接近面向对象方法，因此有的文献也把它称为一种面向对象方法，但有以下差别：

1. 强调的重点是信息建模和状态建模，而不是对象建模
2. 实体中只有属性没有操作
3. 只有属性的继承，不支持操作的继承
4. 没有采用消息通讯

4 面向对象方法

面向对象的分析（OOA）+面向对象的设计（OOD）

运用对象、类、继承、封装、聚合、关联、消息、多态性等概念来构造系统。

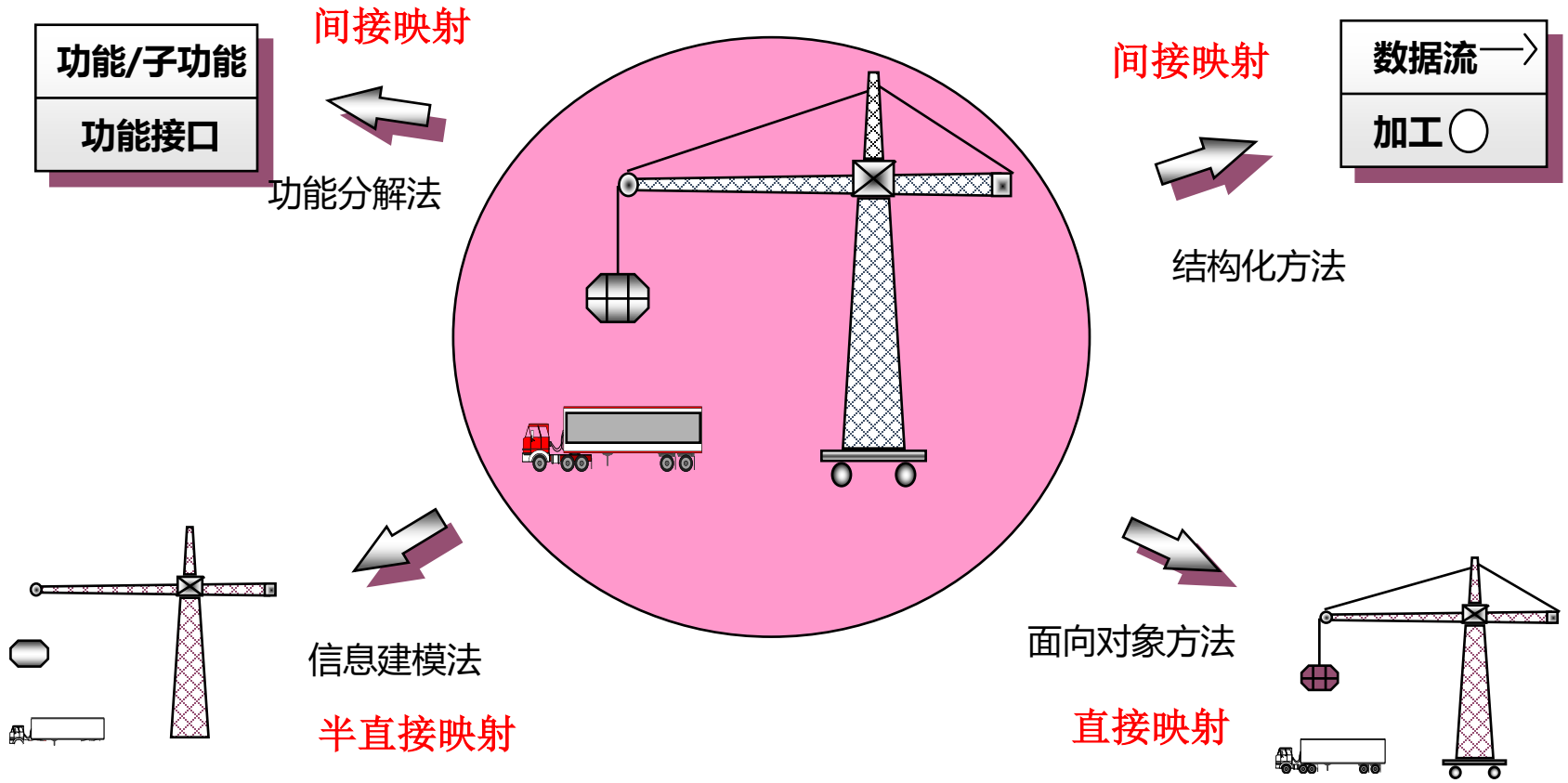
把问题域中的事物抽象为对象，作为系统的基本构成单位，其属性和操作刻画了事物的静态特征和动态特征——完整地刻画了问题域中事物。

用类作为对象的抽象描述，建立它们之间的继承、聚合、关联、消息等关系——如实地表达了问题域中事物之间的各种关系。

封装、继承、聚合、关联、消息通讯等原则符合人类的日常思维——使系统的复杂性得到控制。

不同的建模方法 体现于

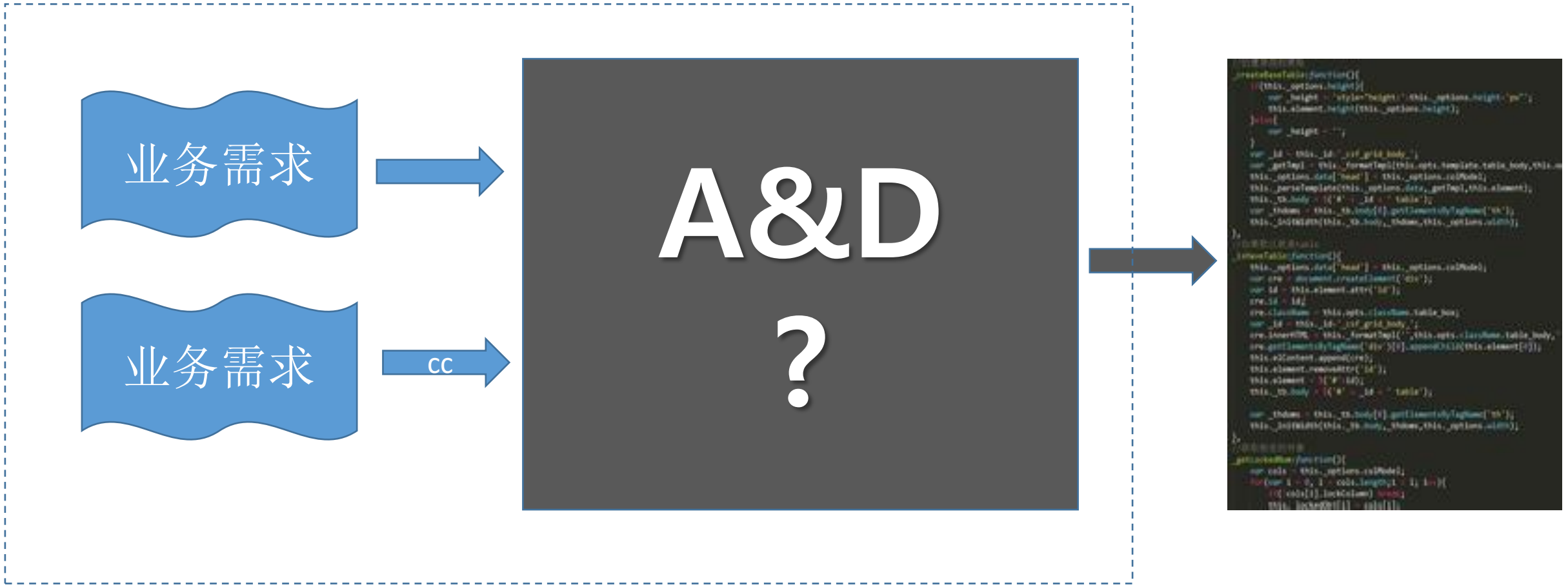
从不同的概念出发来认识问题域
用不同的概念进行系统构造
系统对现实世界的映射方式不同



目录

- 什么是系统?
- 分析之责
- 设计之要
- 软件系统分析与设计的主要方法?
- **本课程的主要任务**
- 参考书目
- 学习方法与考试要求

课程目标框架



课程的前期基础要求与目标

- 基本的编程技巧（C++，C#, java,…）
- 数据库以及基于事务的编程
- 基本的计算机形式化与逻辑分析基础
- 基本的数据结构与算法

- 独立分析与设计一个中小型的软件系统
  - 设计的目标
  - 设计的原则
  - 设计的方法与模式
- 理解OO的概念以及利用它完成设计决策
  - 多态，封装，迭代，对象的识别
- 理解模型设计方法与技巧
  - UML
  - ER
  - DFD
- 实践能力

目录

- 什么是系统?
- 分析之责
- 设计之要
- 软件系统分析与设计的主要方法?
- 本课程的主要任务
- **参考书目**
- 学习方法与考试要求

主要参考书目

(4小时出库)

系统分析与设计方法(原书第7版)

SYSTEMS ANALYSIS AND DESIGN METHODS SEVENTH EDITION

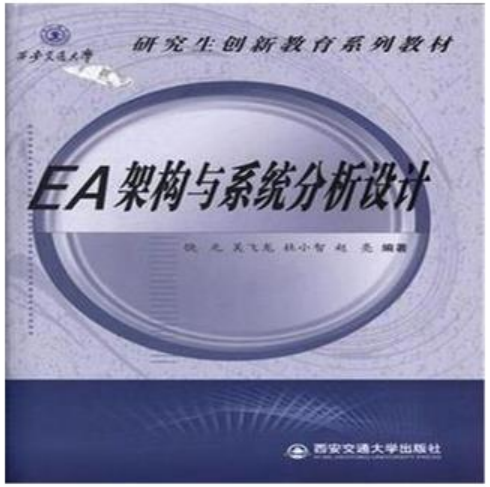
作　　者：(美)JEFFREY L. WHITTEN；LONNIE D. BENTLEY

译　　者：肖刚；孙慧

出版日期：2007-8-1

出 版 社：机械工业出版社　　ISBN：9787111205517

市 场 价：￥59　　4-5星会员：￥41.3



EA架构与系统分析设计

作者:锐元　出版社:西安交通大学出版社　出版时间:2015年12月

0条评论

当当价

¥41.30 7.8折 定价¥53.00

配 送 至　北京 至 北京市东城区　缺货 运费5元起

服　　务　由"北京图书大厦旗舰店"发货，并提供售后服务。

1　卖光了

分享　送积分413　收藏商品

(4小时出库)

系统分析与设计

SYSTEMS ANALYSIS AND DESIGN IN A CHANGING WORLD

评　　价：★★★★★

作　　者：JOHN W. SATZINGER ROBERT B. JACKSON STEPHEN D. BUR

译　　者：朱群雄 汪晓男 等

出版日期：2002-9-1

出 版 社：机械工业出版社　　ISBN：7111108485

市 场 价：￥65　　4-5星会员：￥45.5

(4小时出库)

面向对象的系统分析与设计（UML版）

OBJECT-ORIENTED SYSTEMS ANALYSIS AND DESIGN WITH UML

评　　价：★★★★

作　　者：(美) ROBERT V. STUMPF，LAVETTE C. TEAGUE

译　　者：梁金昆

出版日期：2005-8-1

出 版 社：清华大学出版社　　ISBN：7302113114

市 场 价：￥54　　4-5星会员：￥40.5

目录

- 什么是系统?
- 分析之责
- 设计之要
- 软件系统分析与设计的主要方法?
- 本课程的主要任务
- 参考书目
- **学习方法与考试要求**

教学方式

	教育1.0	教育2.0
模式	教师向学生的一个单向知识灌输渠道 单向传输	师生共同创建的一个建构知识、分享知识、促进交流、共同发展的空间 互动探究
基本构成单元	教材	学生认识水平、规化指导
教学形态	讲授、习题、作业	问题、思考、交流、反馈、推广
知识结构	教材	教材、资料、教学用具、观察实践、信息媒体、他人等
内容创建者	教师	教师组织引导学生共同创建
教育主导者	教师	教育过程中全体参与者
师生关系	教师权威	师生平等的伙伴关系
学生地位	单纯的学习者	学习的主体、研究团队中的个体
教师地位	单纯的传授者	管理者、帮促者，研究团队中的个体

主要授课方式

- 我讲你听 + 你讲我听 + 你做我评 + 我评你论
- 以专题讲授为主, 参考教材为辅;
- 作业与团队项目并行 + 知识与实践并行

- 多分析,多实践,多应用, 少背

考试方式

- 闭卷考试: 占 60%
- 课堂参与+平常作业+大作业：占30%
- 平时考勤：10%

诚信规则：
如果发现有作业或者考试雷同，无需解释，相关人均为 0分。

- 联系方式：
  - Tel: 82663000转8002
    13572238819
  - Email: yuanrao@163.com
    raoyuan@mail.xjtu.edu.cn

微信： iroyals

谢谢大家