软件架构设计

Software Architecture Design

陈长清

博士，副教授

电子邮件：ccqcLczy@163.com
ccqcL@sina.com

华中科技大学软件学院

前言

体系结构简称架构或构架。

构：本义架木造屋，引申为构造

构架：建筑的结构

结：用绳、线、皮条等绾成的疙瘩；关键点；被联结状态

结构：组成整体的各部分的搭配和安排

对住房的功能需求：能够居住。

对住房的质量需求：

安全性：能避免地震、台风、暴雨等各种自然灾害。

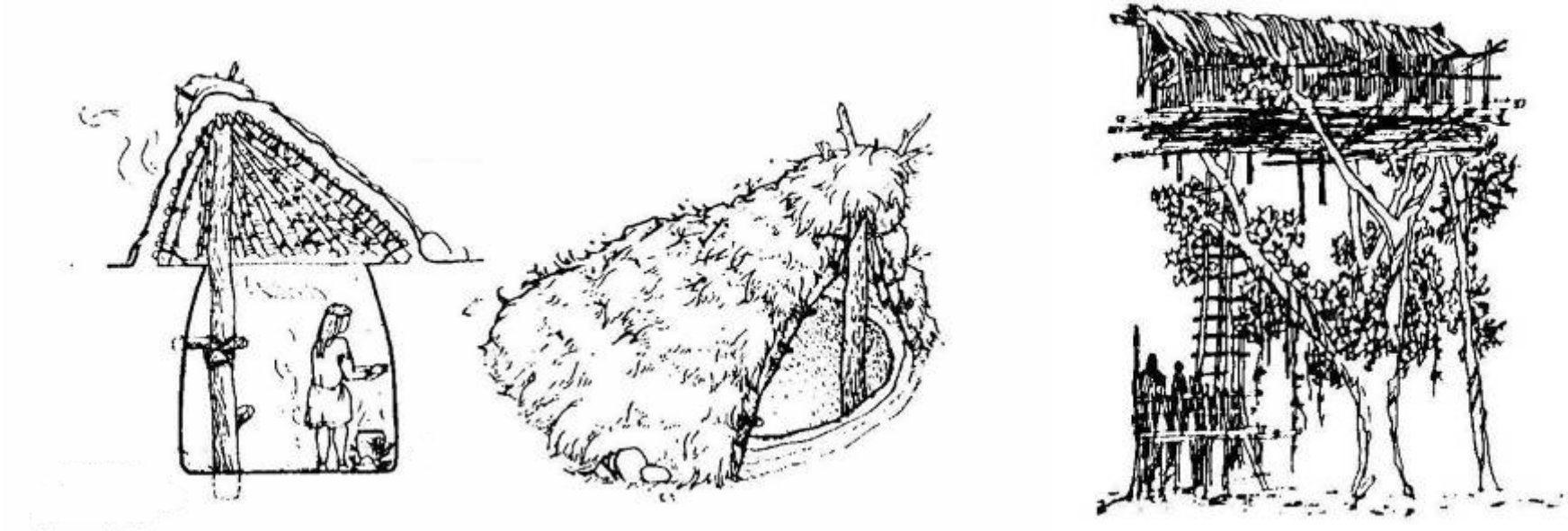
居住者在建筑内的健康性，舒适性。

美观性：有亲和感，社会文化的体现。

不同住房具有相同的功能，但其架构不同，所满足的居住质量也不相同。

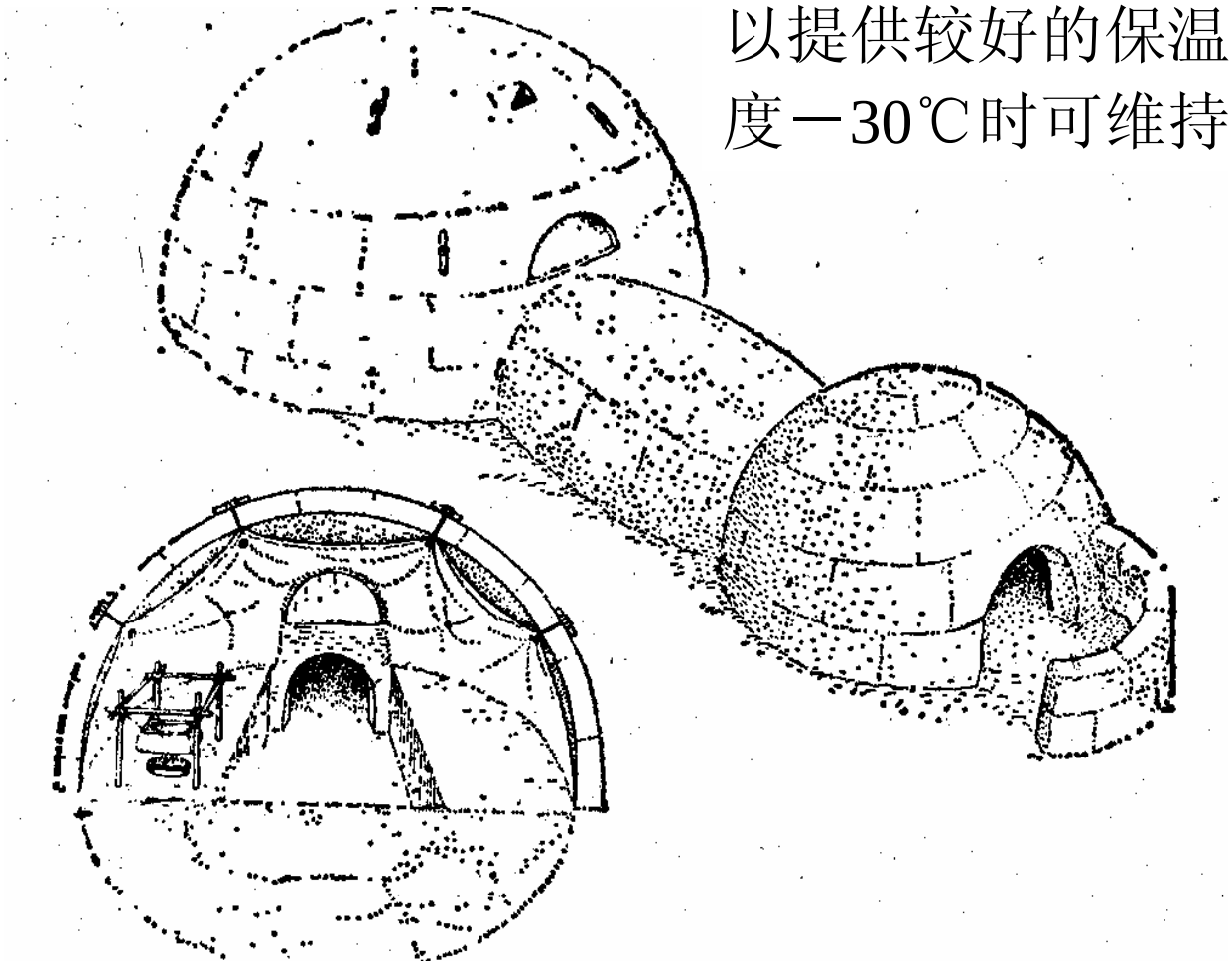
人类最早的居住方式：巢居和穴居

- 炎热或高海拔地区的穴居方式，可获得相对稳定的室内热环境，顶部的天窗既可采光又可排烟。



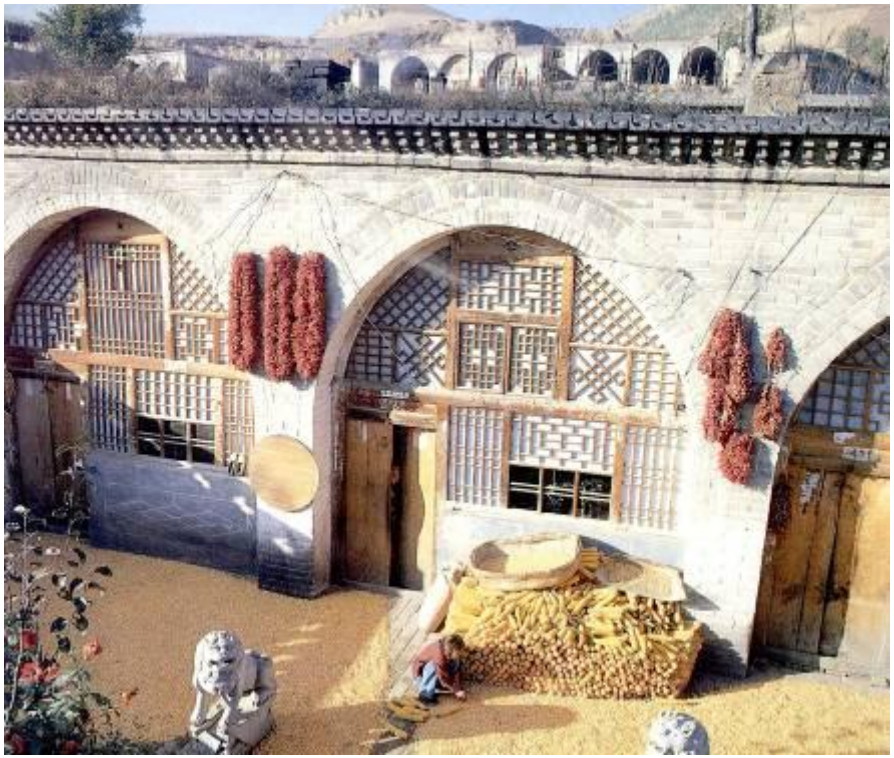
爱斯基摩雪屋的外观和室内布置

用干雪沏成，厚度500ｍｍ的墙体可以提供较好的保温性能。当室外平均温度－30℃时可维持室内温度－5℃以上。



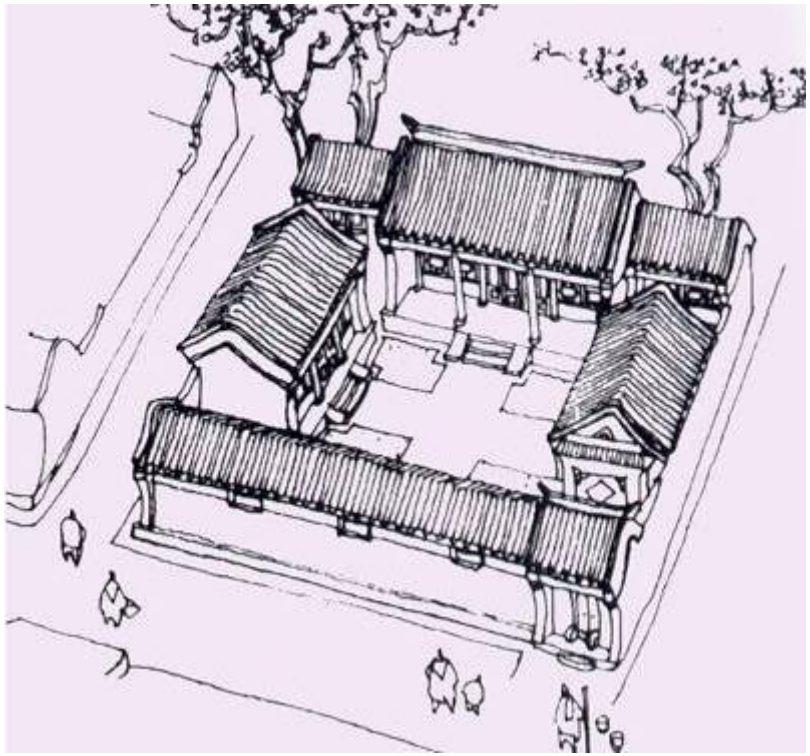
大陆气候的中国民居

土窑洞借助土壤大热惯性，达到冬暖夏凉的目的。



中国四合院：座北朝南的典范

- 利用太阳高度角的特点，仅在北方出现。
- 四合院建筑冬季有效地利用了太阳能采暖和抵御北风侵袭，屋顶设计避免了夏季室内过热。



湿热地区的中国民居

云南干阑竹楼：防雨，防湿和防热。

地球环境还可持续发展吗？

空调的普及使人们不再关心建筑的冷暖，但高能耗又使环境受到影响。

Ü 遍布全球的玻璃和钢筋盒子建筑

为什么研究软件架构?

思想有多远，我们就能走多远

高度决定思路，思路决定出路

系统的建立是为了满足组织的需求（包括功能和质量），质量需求决定了系统必须达到的特征, 包括性能, 可靠性, 互操作性以及生命周期等。随着软件系统的日益复杂，涉众对软件的要求已不局限于功能上的满足，而是更加注重质量。

很少有人注意到组织（开发组织、客户等）在系统设计和系统成败上扮演的角色。

系统的质量特征受到软件架构的限制，或者说构架设计的选择受到要达到的质量特征的影响。

本课程的目的：对软件架构的产生、演化做通俗介绍，减少对架构认识的神秘感，对软件架构设计提供实用的指导。

阅读指南

第1部分: 软件构架的基础

第2部分: 构架的创建

第3部分: 构架的分析与评审

案例分析的组织

- 案例的简要说明
- 功能需求和质量属性
- 构架解决方案
- 总结

第1部分

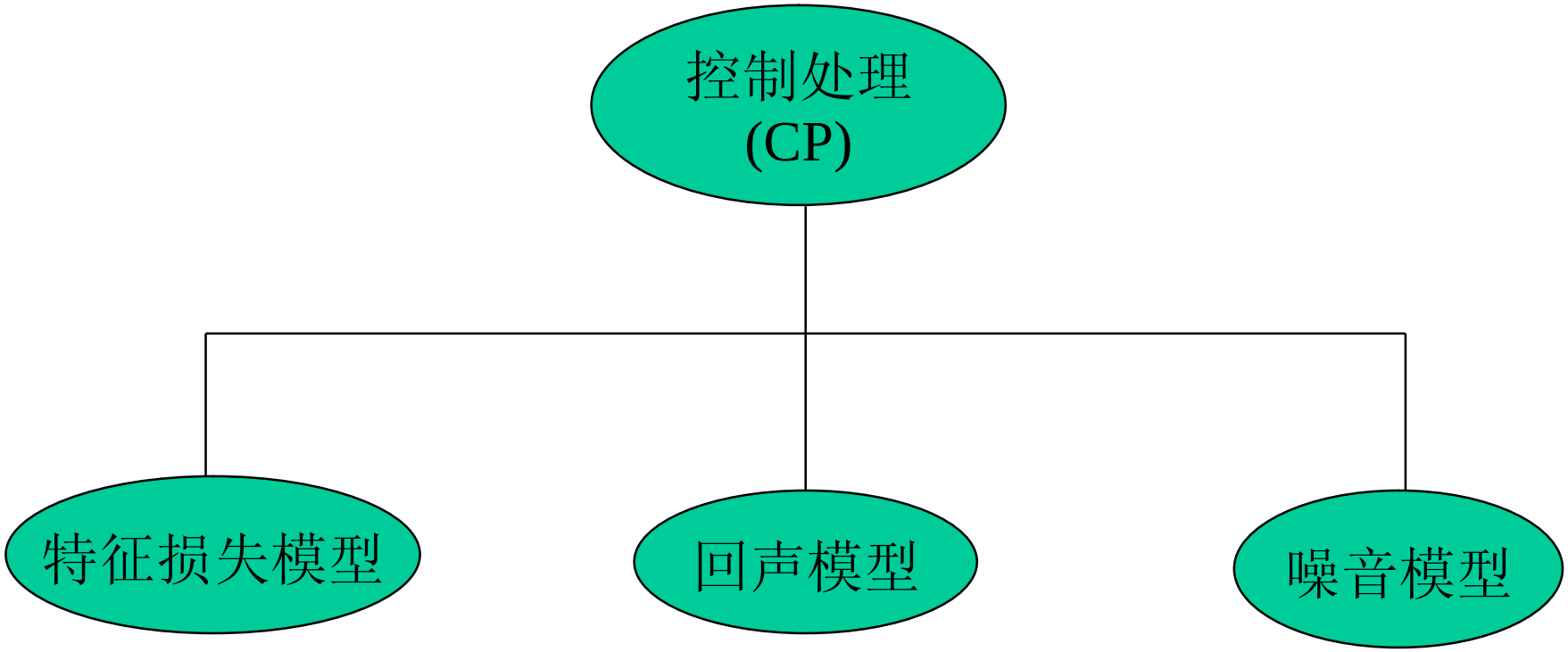
软件架构基础

第 1 章

认识软件架构

- 1.1 软件架构的概念
- 1.2 软件架构的多个结构
- 1.3 软件架构的产生
- 1.4 软件的架构不是静止的
- 1.5 软件架构的重要性
- 1.6 小结
- 1.7 讨论

1.1 软件架构的概念



你从这个图中可以知道什么？

1.1.1 软件架构的定义

软件架构—在一定的设计原则基础上，从不同角度对组成系统的各部分进行搭配和安排，形成系统的多个结构而组成架构，它包括该系统的各个组件、组件的外部可见属性及相互关系。

外部可见属性—指其它组件可对该组件所做的假设，如该组件提供的服务、具备的性能特征、错误处理、共享资源的使用。

为什么设计原则是架构的一部分？

做事先做人

很多人不懂或不遵守走路的规则，结果丢掉性命

很多人不遵守开车的规则，结果造成严重交通事故

同样，不遵循架构设计的原则，架构也容易失败。

架构定义可以从下面六个方面来理解：

- 架构应建立在一定的设计原则之上，否则很容易失败。
- 系统可能由多个结构组成，其中任何一个结构都不能与构架等同。
- 每个软件系统都有自己的架构。
- 软件架构决定了各个组件。
- 只要某个组件的行为可以从其它组件的角度观察到或区别开，这样的行为就是软件架构的内容。
- 软件架构是抽象的，它不考虑实现、算法和数据表示的细节，而集中研究“黑盒”组件的行为和交互，是设计第一步。

1.1.2 其它观点

观点1 软件架构是高层次的设计

观点2 软件架构是软件系统的总体结构

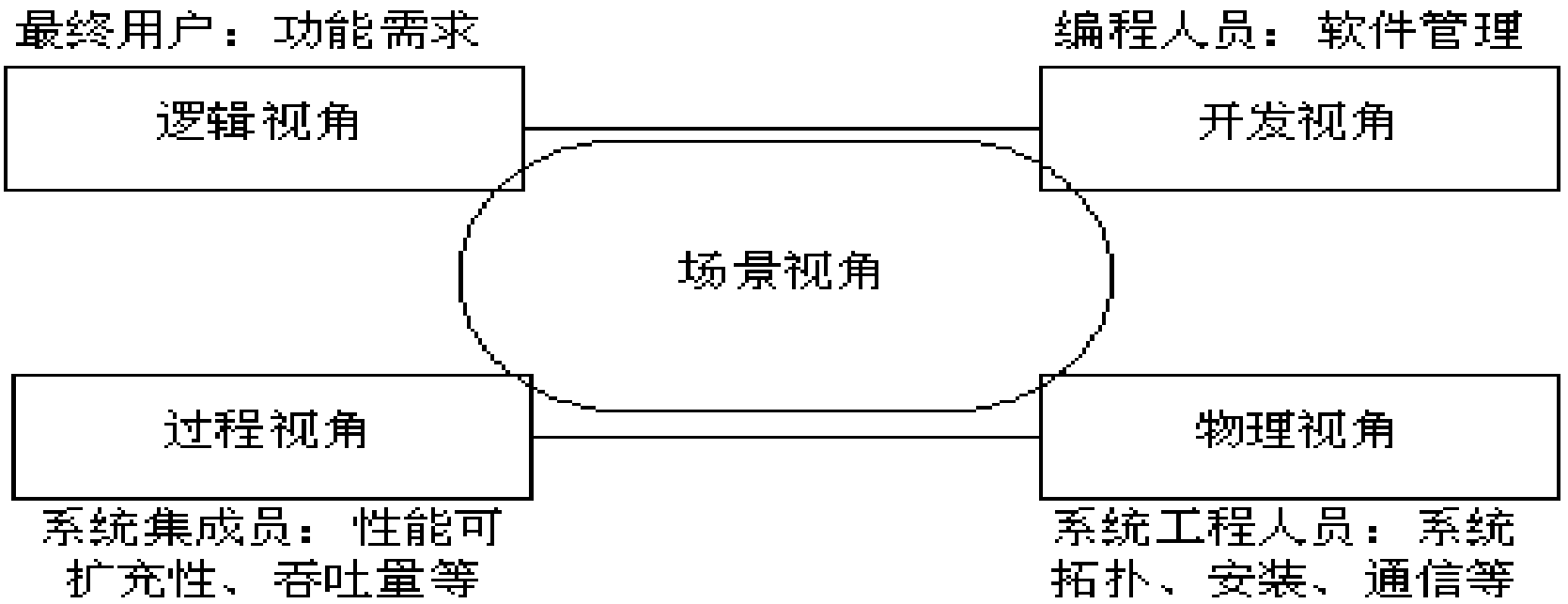
观点3 IEEE的定义：软件架构是一个程序或系统的组件结构、组件之间的相互联系及支配组件设计和进化的指导原则

观点4 Perry和Wo1f提出：软件架构是具有一定形式的结构化元素，包括处理元素、数据元素和连接元素。处理元素负责对数据进行加工，数据元素是被加工的信息，连接元素把架构的不同部分组合连接起来。

观点5 Mary Shaw和David Garlan认为软件体系结构是软件设计过程中的一个层次，这一层次超越计算过程中的算法设计和数据结构设计。体系结构问题包括总体组织和全局控制、通讯协议、同步、数据存取，给设计元素分配特定功能，设计元素的组织，规模和性能，在各设计方案间进行选择等。软件体系结构处理算法与数据结构之上关于整体系统结构设计和描述方面的一些问题，如全局组织和全局控制结构、关于通讯、同步与数据存取的协议，设计构件功能定义，物理分布与合成，设计方案的选择、评估与实现等。

观点6 Kruchten指出，软件体系结构有四个角度，它们从不同方面对系统进行描述：概念角度描述系统的主要构件及它们之间的关系；模块角度包含功能分解与层次结构；运行角度描述了一个系统的动态结构；代码角度描述了各种代码和库函数在开发环境中的组织。

这些角度形成了一个“4+1”的视角模型。“4+1”模型从5个不同的视角包括逻辑视角、过程视角、物理视角、开发视角和场景视角来描述软件体系结构。每一个视角只关心系统的一个侧面，5个视角结合在一起才能够反映系统的软件体系结构的全部内容。“4+1”模型如图所示：



观点7 Hayes Roth则认为软件架构是一个抽象的系统规范，主要包括用其行为来描述的功能构件和构件之间的相互连接、接口和关系 。

观点8 Barry Boehm提出，软件架构包括系统构件，互联及约束的集合；系统需求说明的集合；一个基本原理用以说明这一构件，互联和约束能够满足系统需求。

观点9 软件架构为软件系统提供了一个结构、行为和属性的高级抽象，由构成系统的元素的描述、这些元素的相互作用、指导元素集成的模式以及这些模式的约束组成。软件架构不仅指定了系统的组织结构和拓扑结构，并且显示了系统需求和构成系统的元素之间的对应关系，提供了一些设计决策的基本原理。

1.2 软件架构的多个结构

静态的角度：

1. 模块结构—体现了任务的划分，每个模块有其接口描述、代码和测试计划等，各模块通过父子关系联系起来，在开发和维护阶段用于分配任务和资源。
2. 概念（或逻辑）结构—系统功能需求的抽象，功能图。
3. 类结构—对象之间的继承或实例关系。

动态的角度：

4. 进程结构—运行系统的动态特征，包括进程间的同步关系、缺少不能运行、存在不能运行、先后等关系，与模块结构、概念结构成垂直正交关系。
5. 数据流—模块之间可能发送数据的关系，最适合用于系统需求的追踪

6. 控制流—程序、模块或系统状态之间的“之后激活”的关系，适合于对系统功能行为和时序关系的验证。
7. 使用结构—描述过程或模块之间的联系，这种联系是“假设正确存在”的关系，用于设计可轻松扩展的系统。

   如果过程A的运行必须以过程B的正确运行为前提，则说过程A使用过程B。
8. 调用结构--（子）过程之间调用和被调用的关系，可用来跟踪系统的执行过程。
9. 层次结构—是一种特殊的使用结构，层就是相关功能的一致集合，在严格的分层结构中，第n层仅能使用第n-1层提供的服务。

部署的角度：

10. 物理结构—软件与硬件之间的映射关系，在分布式或并行系统中有重要意义。

各种结构间的相互关系

1. 各个结构都是从不同角度考察系统，但它们并不完全独立，它们之间的联系是多对多的。
2. 每个项目在开发时一般是注重一个结构，按照这一主要结构来考虑和运用其它结构。
3. 经验表明，系统规模越大，这些结构之间的差异越明显。
4. 使用结构与调用结构的区别：

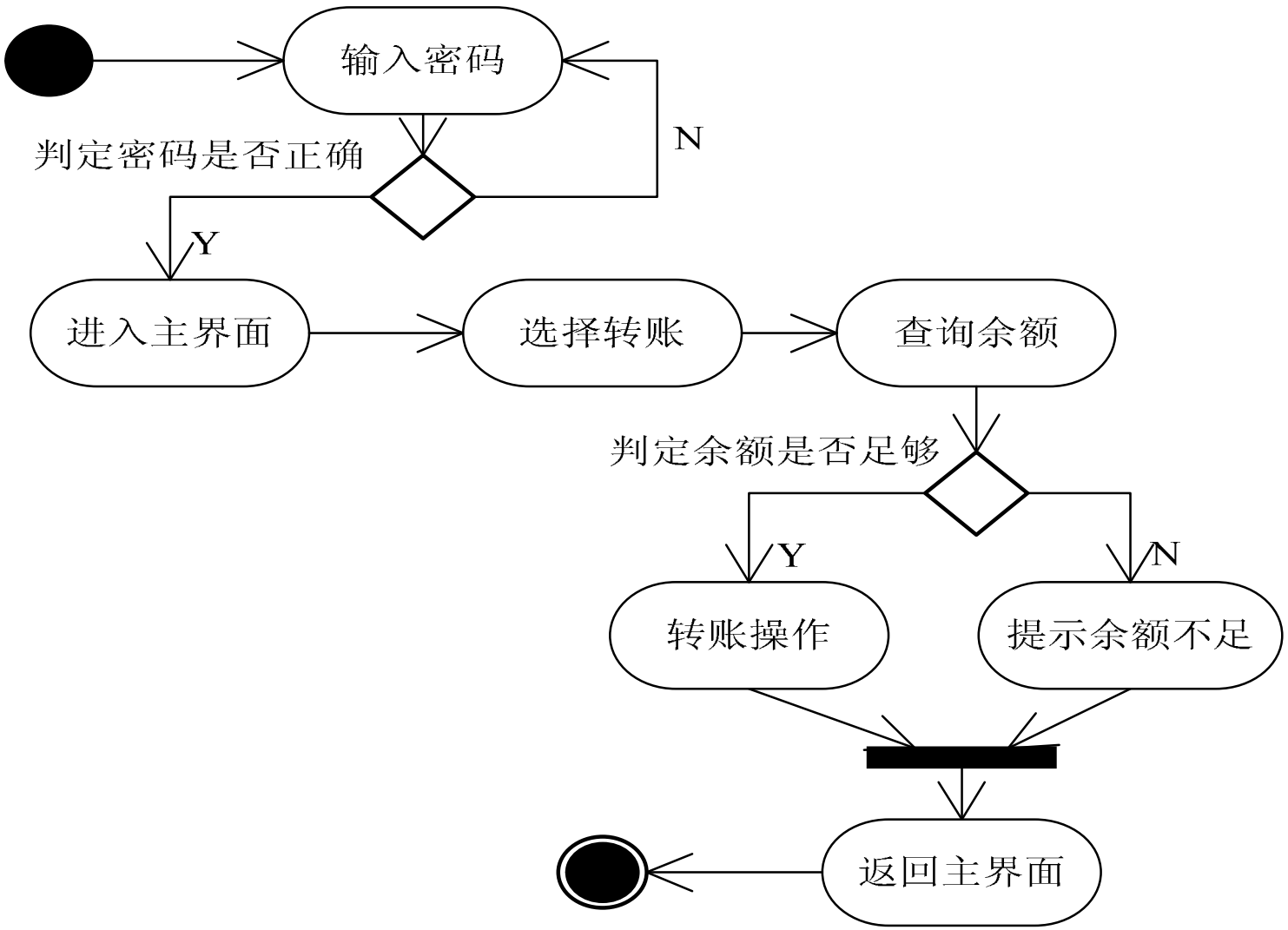
   A使用B的结果，但A不调用B；A调用B，但A不使用B的结果，例如显示网页的各个部分。

调用结构的例子

暂时不能提供的服务是“我的阅读”功能，但这不影响其它显示。

使用结构的例子

2007年1月28日晚，在宁波务工的湖南人唐风军在鄞州古林镇一个银行ATM机转账时，机器出现故障，唐风军疯狂转账225万，后伙同其弟唐风光，跑了10多家银行，取走58万元现金。


输入密码
判定密码是否正确
N
Y
进入主界面
选择转账
查询余额
判定余额是否足够
Y
N
转账操作
提示余额不足
返回主界面

1.3 软件架构的产生?

设计仅是系统功能需求分析的产物?

功能需求 **à** 设计 **à** 系统开发?

否也！！！

911中是世界贸易大厦的倒塌。

2007年1月18日，招商证券的客户发现，在公司进行网络交易时，公司网站竟然无法登陆。一部分客户转向电话服务寻求帮助，却发现电话线路出现同样的阻塞。行情转瞬即逝。拥堵是由于交易量激增，原有系统容量不足引起的 。

奥运门票销售系统的失败

2009年1月3号广州火车站16万张票同时出闸致系统瘫痪。

1.3.1 架构受系统风险承担者的影响

风险承担者—对构建软件系统感兴趣的人或组织，包括合同中的客户、系统最终用户、开发人员、开发组织、系统维护人员等，他们所关注的问题各不相同，但都要求系统在他们所关注的方面提供保证或优化。

事物有主要矛盾和次要矛盾之分。

开发系统时，首先要确定其软件构架。借助于构架，设计师可以分析众多风险承担者所提出的各种要求的优先级，并将这些要求转化为系统的各个特性，再针对它们在系统结构上做折衷，从而得到和谐的架构。

开发组织所关心的问题不同于客户，它对软件构架的影响分为3类：

· 直接影响

如希望向产品线发展

· 长远影响

如行业布局

· 组织结构的影响

如软件外包

· 开发团队的经验对设计师的影响

1.3.2 架构受设计师的素质和经验的影响

1 熟悉.NET的设计师在设计时会考虑.NET的框架和技术。

2 熟悉J2EE的设计师在设计时会考虑J2EE的框架和技术。

3 设计师具有数据库方向的背景，系统会被认为是数据库的应用。

4 设计师具有网络安全的背景，系统的安全会被放在很突出的位置。

1.3.3 构架受技术环境的影响

现在B/S样式很流行，设计师在设计时往往首先考虑系统能否在互联网环境下运行。

1.3.4 设计师的沟通能力

设计师的沟通能力从下面三点体现：

1 多看别人的长处，这样才能屈身理解涉众要求。

案例1：一位项目经理的成长过程

第一年，我是一个新手，我的年终总结是我从周围同事学到许多东西；

第二年，公司经营遇到困难，我的年终总结是我明年会做得更好；

第三年，我的年终总结是我和同事合作愉快，他们有很多优点值得我去学习。

春节刚过，我就被提拔为项目经理。

设计师必须明白对业务需求的理解程度往往不如涉众。

2 姿态放低一点

案例2：逗狗师的故事

富人喜欢养很名贵的狗，但狗往往不听主人的话，只好请逗狗师帮忙训练狗并参加比赛。逗狗师可以让一条陌生的狗在几分钟内听他的话，询问原因，乃逗狗师看狗的眼光会让狗觉得他比它低点。

我们都喜欢平易近人的领导，愿意和他们沟通。

有些领导担心姿态放低了，别人不把他当领导，其实这样的人是很少的。就像逗狗师一样，姿态虽然低点，但最终的结果是狗听他的话。

3 设计师还要会讲故事

设计结果往往是抽象的，因此设计师要给客户讲故事，把抽象的设计变成一个个故事（场景）。给领导讲、给用户讲、给开发团队讲，给所有的利益相关者讲。故事讲好了，所有利益相关者都听懂了，听了都兴奋，利益相关者的需求就更容易表达、理解和验证。这个设计师就是成功的设计师，他所设计的软件就容易取得成功。

设计师讲故事当然不是脱离实际的胡编乱造，其本真有四个基础：意义、产品功能、产品质量和技术。

设计师要能用一般生活常识，把抽象的设计讲解透彻。不从概念出发，不从已只的模式出发，不从自身经验出发，不闭目塞听，不自说自话，而应讲究以心比心，换位思考，敢于直面令人不快的真实，以此捕获利益相关者的心。在讲故事的互动过程中，发现利益相关者的真实需求。

企业家作为企业的设计者早就发现了这一方法。

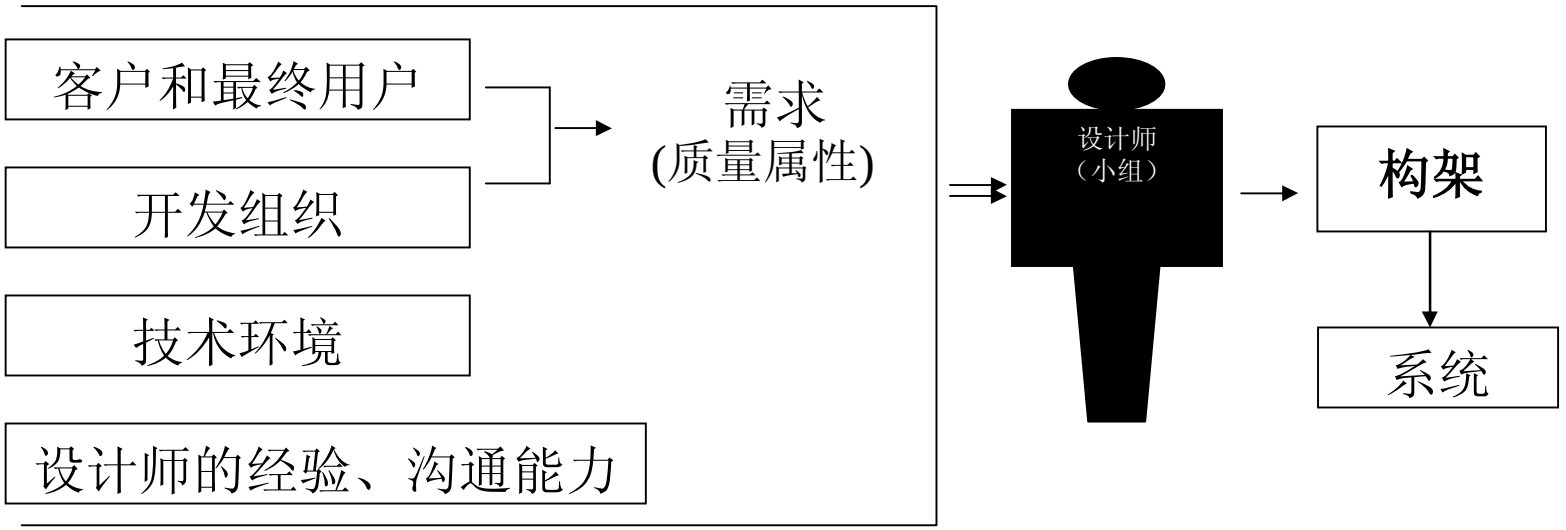
任正非至今依然活跃在客户一线，他把华为塑造成运营商整体解决方案的提供商，在使所有运营商无风险赚钱的同时，自己也成为运营商不可或缺的一个有机部分。当年，任正飞凭华为创新体系的故事，一举从银行拿到最便宜的资金。

马云打着让天底下生意更好做的旗帜，把阿里巴巴的意义渗透到每一个角落。当年，马云凭借一个设想，会面15分钟，就从孙正义那里募得2000万美元。

史玉柱则醒悟说一切高规格的应酬都是浪费时间，故而在公司上市后依然一天干十小时客服，把求索客户心智当成头等大事。

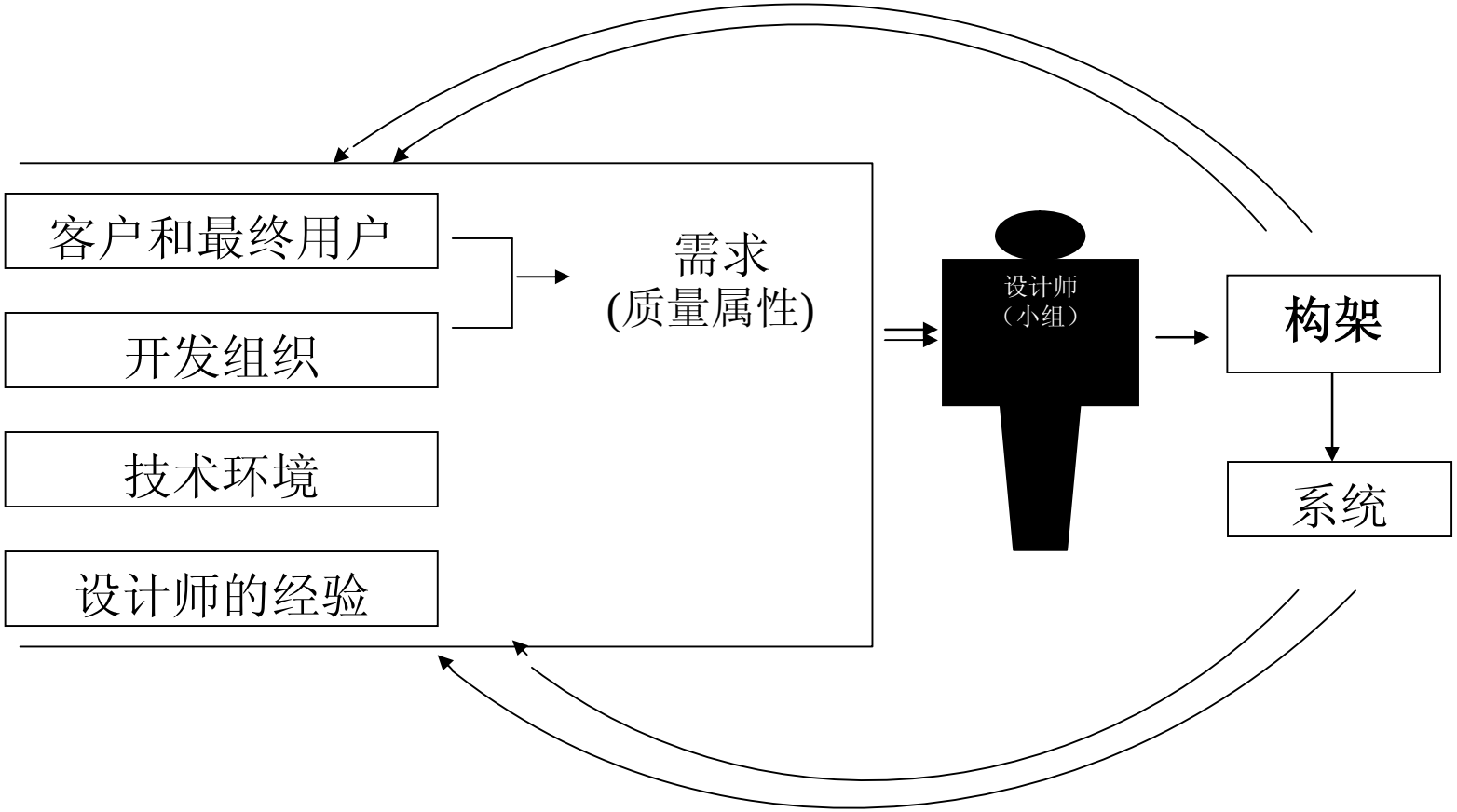
我们现在谈论SOA，研究SOA，其实就是IBM给我们讲了一个故事。

IBM提出随需而变，也是向用户讲了一个故事。



架构所受的影响

1.4 软件的架构不是静止的



1. 软件在开发过程中或交付使用后，都可能会发生修改，这些修改往往涉及到架构的变更。因此软件版本的演进也是软件架构的演进。
2. 软件架构影响设计师的经验。
3. 软件架构影响开发组织的内部结构和经营目标。
4. 软件架构可能会影响客户对下个系统的需求
5. 有些系统甚至会影响并实际改变软件工程的发展，以及开发人员学习和实践的技术环境，如互联网、嵌入式、手机等。

架构商业周期—架构是软件开发的必经之路和必要手段，它受到来自客户和开发组织的影响，也受到设计师的素质和经验以及技术环境的影响；反过来，构架也影响着被开发的系统，对客户、开发组织、构架和技术环境也都有影响，还影响着客户及其开发组织的未来目标。围绕着构架的这些影响和反馈循环构成构架商业周期。

事物是静止的，也是运动、发展和变化的，软件架构也不例外。软件的架构是运动、发展和变化的。

1.5 软件架构的重要性

1. 风险承担者之间的交流平台
2. 早期设计决策的体现
3. 有助于实现构架级重用

1.6 小结

构架不只是功能需求的结果。

1.7 讨论

谈软件架构与建筑结构的联系与区别