

Software Architecture

Reference books

- 软件构架实践（影印版）（第2版），
Software Architecture in Practice (Second Edition), Len Bass, Paul Clements, Rick Kazman, 清华大学出版社
- 软件体系结构, Shaw, Mary、Garlan, David, (牛振东、江鹏、金福生翻译), 清华大学出版社
- 软件体系结构理论与实践, 作者: 马冲、江贺、冯静芳, 出版社: 人民邮电出版社

Other Resources

- www.sei.cmu.edu/architecture
- www.gaudisite.nl
- www.bredemeyer.com
- www.booch.com/architecture/index.jsp
- http://www.iturls.com/softwareresource/SW_303.asp

Content

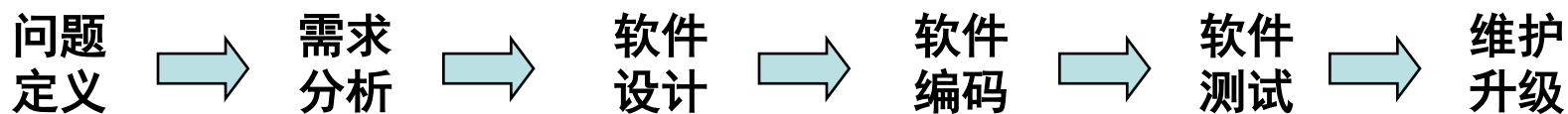
- 软件体系结构的概念
- 软件体系结构的基本原理
- 现代主流软件的架构分析

得分

- 平时: 30%
- 期末: 70%

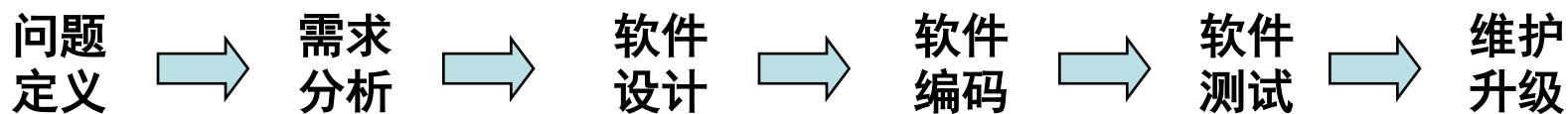
缘起...

- 软件开发过程



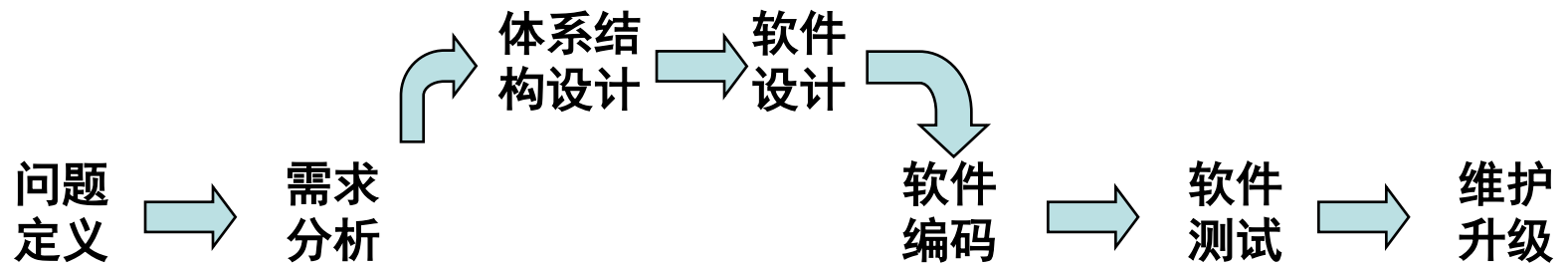
缘起...

- 软件开发过程



软件设计

缘起...



软件体系结构定义

- 存在至少数十种的软件体系结构定义，尚未形成公认的统一定义，其原因在于：
 - 软件体系结构形成历史较短，尚不完善。
 - 不同的研究者对于软件体系结构有不同的理解。
 - 定义一旦被接受，其影响很大！

软件体系结构定义

- 定义1: Booch & Rumbaugh & Jacobson 定义
 - 软件体系结构 = {组织, 元素, 子系统, 风格}

软件体系结构定义

- 定义2: Bass 定义
 - 软件体系结构是一个或多个结构，包括软件构件（component）、构件的外部可视属性（property）和构件之间的关系（relationship）。

软件体系结构定义

- 定义3: Shaw 定义

四种分类法

- 结构模型 体系结构由构件、构件间连接及其他一些方面组成。
- 框架模型 与结构模型类似，强调整个系统的结构。
- 动态模型 强调系统的行为质量。
- 过程模型 体系结构是一系列过程的结果。

软件体系结构定义

- 定义4: Garlan & Shaw 定义

体系结构={构件, 连接件, 约束}

软件体系结构定义

- 定义5: Perry & Wolf 定义

体系结构 = {元素, 形式, 准则}

元素:

- 处理元素: 负责完成数据加工
- 数据元素: 作为被加工的信息对象
- 连接元素: 用于将体系结构的不同部分组合连接起来

形式:

- 专有特性: 用于限制体系结构元素的选择
- 关系: 用于限制体系结构元素组合的拓扑结构

软件体系结构定义

- 定义6: Garlan & Perry 定义

体系结构: 是一个程序/系统各构件的结构、它们的相互关系, 以及进行设计的原则和指导方针。

软件体系结构定义

- 定义7: Soni & Nord & Hofmeister 定义

体系结构包括 至少 4 种具体形态:

- 概念体系结构
- 模块互连体系结构
- 执行体系结构
- 代码体系结构

软件体系结构定义

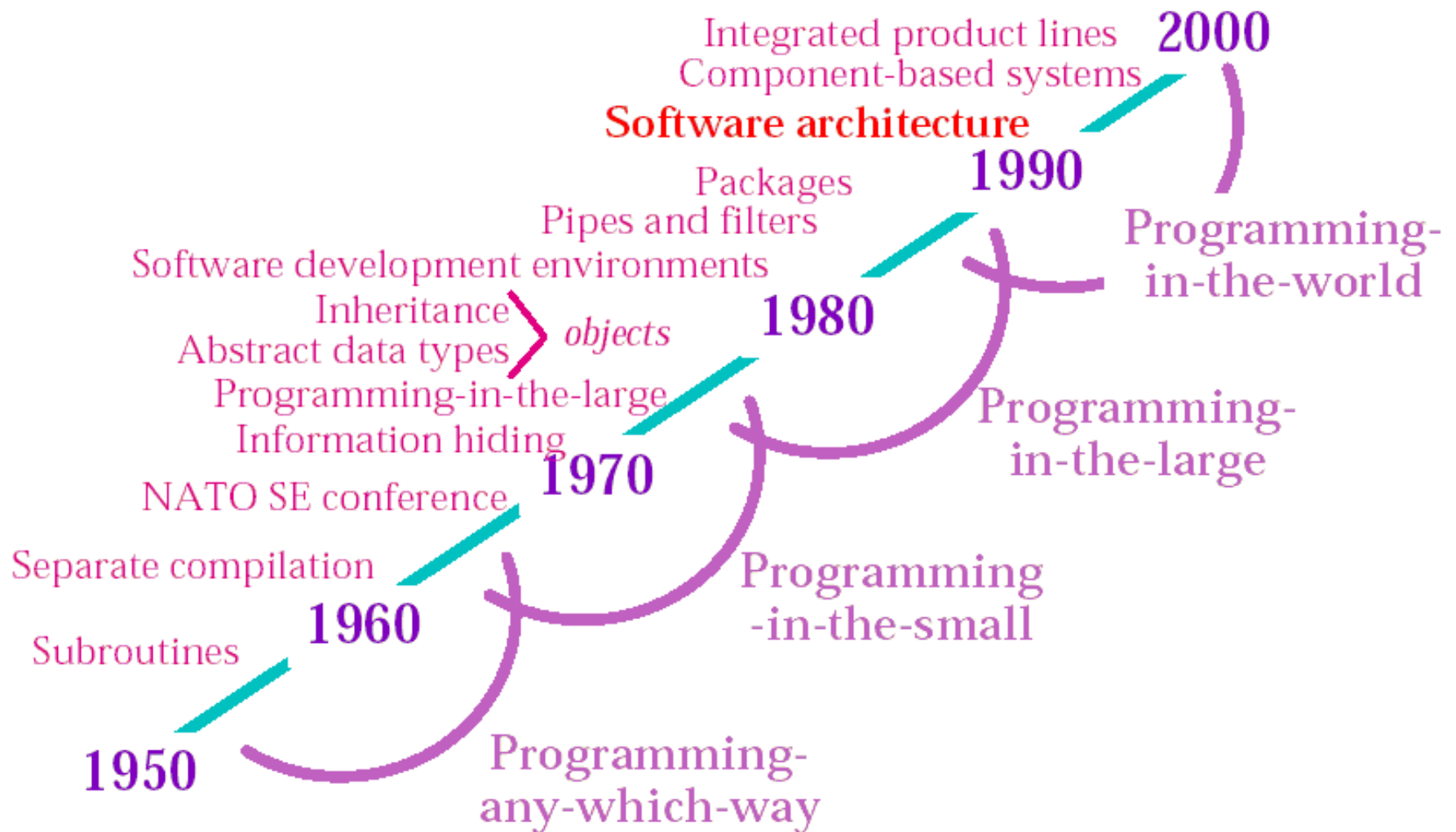
- 定义8: Boehm 模型

体系结构 = {构件, 连接件, 约束, 不同人员的需求, 准则}

软件体系结构的发展阶段

- “无体系结构”设计阶段
- 萌芽阶段
- 初级阶段
- 高级阶段

§ 1.1 软件体系结构历史回顾



软件体系结构的发展阶段

- “无体系结构”设计阶段（**1946年ENIAC计算机问世——ALGOL68诞生前**）
 - 软件开发以汇编语言为主，编程是少数“科学家”的游戏
 - 软件应用以军用和科学计算为主
 - 软件规模小，很少明确考虑系统结构

1	vmulpd	L2E,	%xmm0,	%xmm2
2	vroundpd	\$0xffffffffffffffffffe,	%xmm2,	%xmm2
3	vcvtttpd2dq	%xmm2,	%xmm3	
4	vpaddw	B,	%xmm3,	%xmm3
5	vpsllq	\$0x14,	%xmm3,	%xmm3
6	vpshufd	\$0x3,	%xmm3,	%xmm3
7	vmulpd	C1,	%xmm2,	%xmm1
8	vaddpd	%xmm1,	%xmm0,	%xmm1
9	vmovapd	T1,	%xmm0	
10	vlddqu	T8,	%xmm2	
11	vmulpd	%xmm1,	%xmm2,	%xmm2
12	vaddpd	T7,	%xmm2,	%xmm2
13	vmulpd	%xmm1,	%xmm2,	%xmm2
14	vaddpd	T6,	%xmm2,	%xmm2
15	vmulsd	%xmm1,	%xmm2,	%xmm2
16	vaddpd	T5,	%xmm2,	%xmm2
17	vmulpd	%xmm1,	%xmm2,	%xmm2
18	vaddpd	T4,	%xmm2,	%xmm2
19	vmulpd	%xmm1,	%xmm2,	%xmm2
20	vaddpd	T3,	%xmm2,	%xmm2
21	vmulsd	%xmm1,	%xmm2,	%xmm2
22	vaddpd	T2,	%xmm2,	%xmm2
23	vmulsd	%xmm1,	%xmm2,	%xmm2
24	vaddsd	%xmm0,	%xmm2,	%xmm2
25	vmulpd	%xmm1,	%xmm2,	%xmm1
26	vaddsd	%xmm0,	%xmm1,	%xmm0
27	vmulpd	%xmm3,	%xmm0,	%xmm0
28	retq			

软件体系结构的发展阶段

- 萌芽阶段（1970——1980年）
 - 提出了结构化编程的思想
 - “软件工程” 的思想被广泛应用
 - 出现了一批结构化程序设计语言，包括 PASCAL、Ada 语言等
 - 软件体系结构已经成为系统开发中的一个明确概念

软件体系结构的发展阶段

- 萌芽阶段

优点：

- 软件应用走向商用、民用
- 软件规模达到数万行、数十万行
- 软件开发成为一种职业，诞生了程序员这一群体

软件体系结构的发展阶段

- 萌芽阶段

困境：

- 硬件像一匹越来越快的马，而软件则是一个越来越胖的人，硬件与软件之间的鸿沟依然存在，且愈发严重
- 软件规模扩大导致软件质量的下降
- 大量的重复劳动，软件重用任重道远

软件体系结构的发展阶段

- 初级阶段（80年代初——90年代初）
 - 面向对象技术的提出，出现数十种支持软件开发的面向对象方法，如Booch, Coad/Yourdon, OMT, Jacobson等
 - 各种面向对象语言如C++的出现。
 - 软件以对象作为基本元素，系统由离散的对象构成
 - 统一建模语言UML提出

软件体系结构的发展阶段

- 初级阶段

优点：

- 易于表述概念、易于相互交流
- 作为描述、分析和建立文档的手段，提高了软件的易读性、可维护性和可重用性
- 其中的继承、封装和多态性等机制，为软件重用提供了一种可行的手段

软件体系结构的发展阶段

- 初级阶段

困境：

- 基于对象的重用属于代码重用，需要对类的内部设计、实现有清晰的认识。
- 软件的升级、维护等需要重新编译、调试，无法实现动态升级。
- 难以实现设计重用，缺乏描述体系结构的语言。

软件体系结构的发展阶段

- 高级阶段（90年代初——现在）
 - 软件开发强调构件化技术和体系结构技术。
 - COM、CORBA、EJB、WebService的提出
 - 软件 = 构件 + 基于体系结构的构件组装
 - 体系结构作为开发文档和中间产品，开始出现在软件开发过程中。

软件体系结构的发展阶段

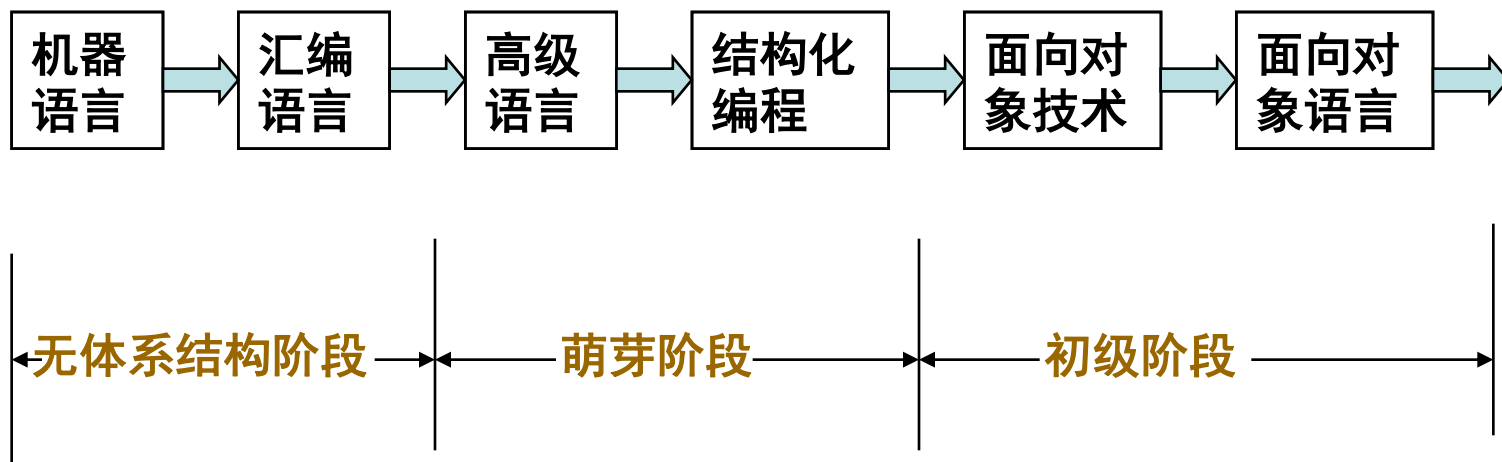
- 高级阶段

优点：

- 构件实现了可执行二进制代码的重用
- 构件的实现与实现语言无关
- 构件可以单独开发、单独编译、单独调试
- 基于构件的软件系统可以实现动态升级和维护
- 体系结构提供了设计重用的可能性，设计重用是比构件重用更加抽象、更加高级的重用。

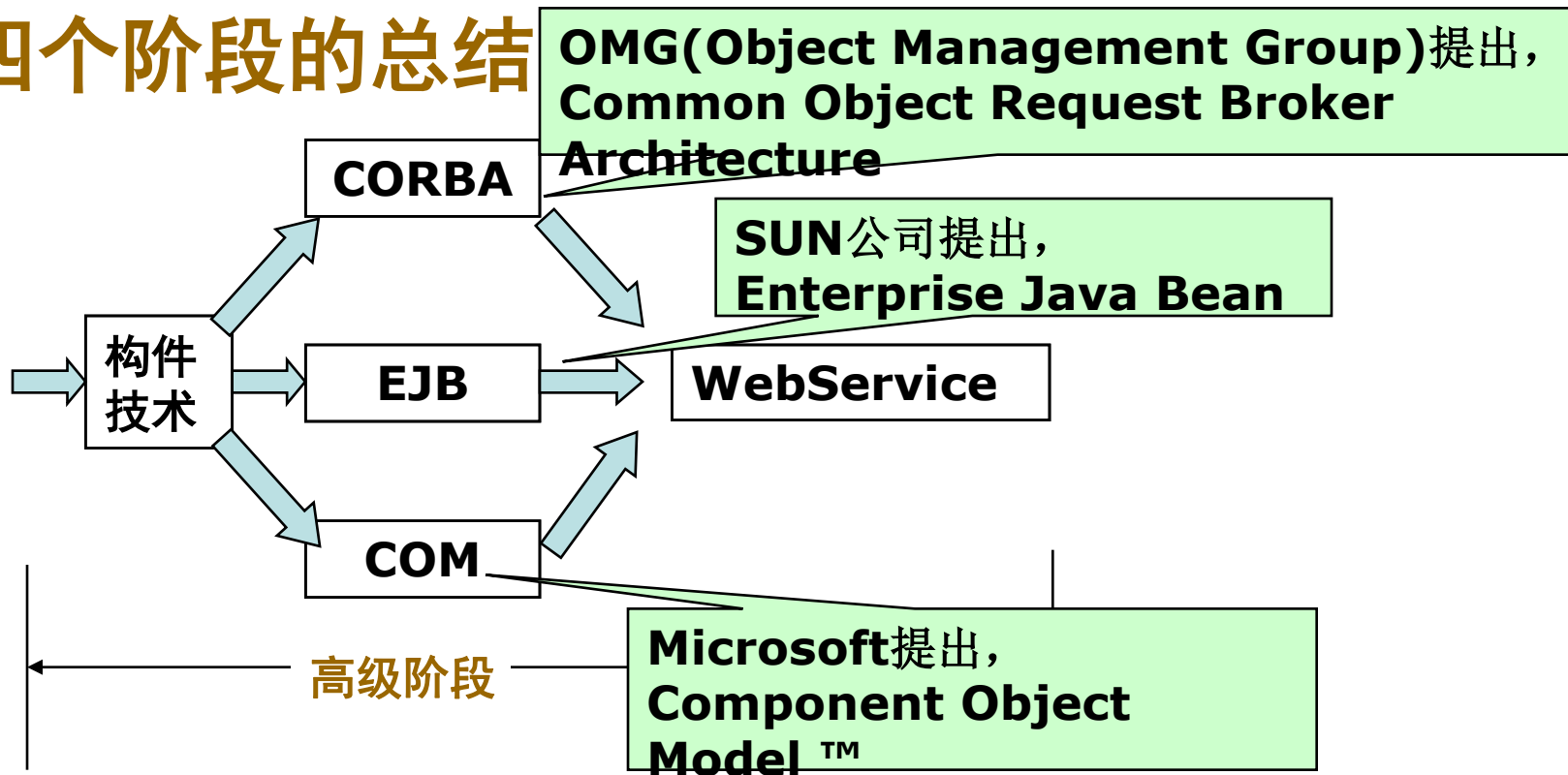
软件体系结构的发展阶段

- 四个阶段的总结



软件体系结构的发展阶段

- 四个阶段的总结



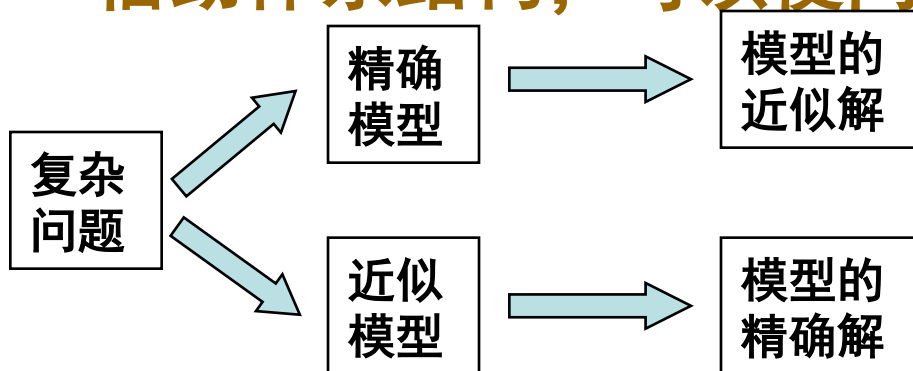
软件体系结构的意义

- 有利于系统分析
- 有利于软件开发
- 有利于软件复用
- 有利于软件系统的演化

软件体系结构的意义

- 有利于系统分析

- 借助体系结构，可以使问题进一步抽象



- 体系结构是软件系统相关各方交流的平台

软件体系结构的意义

- 有利于系统分析
 - 必须始终明确：用户的需求往往很难100%满足。一个简单的例子：



软件体系结构的意义

- 有利于软件开发
 - 体系结构是系统实现的基本约束
 - 体系结构决定了开发和维护项目的组织结构
 - 体系结构有利于软件质量的控制

软件体系结构的意义

- 有利于软件复用
 - 构件复用
 - 软件子系统复用
 - 设计复用

软件体系结构的意义

- 有利于软件系统的演化

- 局部的
- 非局部的
- 体系结构级的

软件设计的目标

- 便于维护和升级，因而应该是模块化的
- 设计应该是便于移植的（移植比重新设计花费要小的多）
- 设计应该具有适用性
- 设计过程应该收到理性的控制 Intellectual Control
- 设计应该表现出概念的完整性 Conceptual Integrity
 - 内在结构
 - 外在表现