

软件体系结构



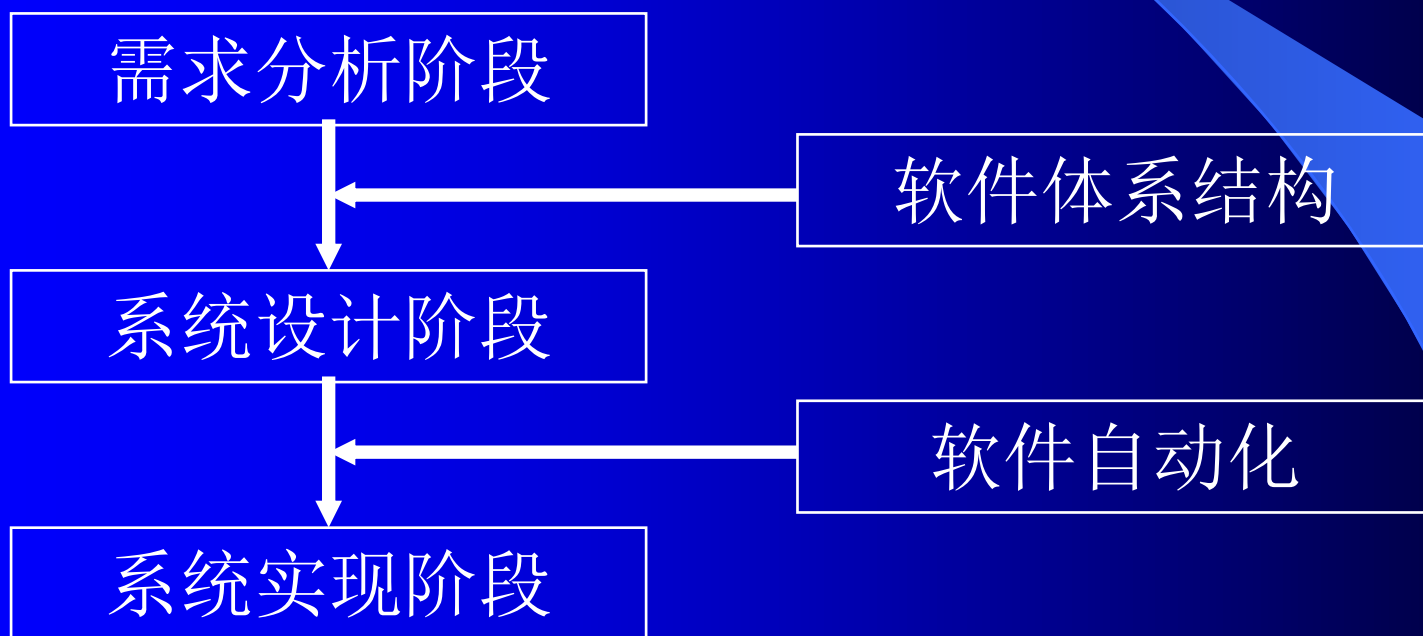
课程名称：软件体系结构

教材：张友生等，软件体系结构（第二版），
清华大学出版社

主讲：周晓峰

zhouxf@hhu.edu.cn

传统软件工程的流程



第一章 软件体系结构概论

1.1 软件危机

1.2 发展历史

1.3 定义

1.4 研究内容

1.5 意义

1.1 软件危机

定义：落后的软件生产方式无法满足迅速增长的计算机软件需求，从而导致软件开发与维护过程中出现一系列严重问题的现象。

1.1 软件危机

产生背景:

20 世纪60年代以前，计算机刚刚投入实际使用，软件设计往往只是为了一个特定的应用而在指定的计算机上设计和编制，采用密切依赖于计算机的机器代码或汇编语言，软件的规模比较小，文档资料通常也不存在，很少使用系统化的开发方法，设计软件往往等同于编制程序，基本上是个人的设计、个人的使用、个人的操作、自给自足的私人化的软件生产方式。

1.1 软件危机

产生背景:

60年代中期，大容量、高速度计算机的出现，使计算机的应用范围迅速扩大，软件开发急剧增长。高级语言开始出现；操作系统的发展引起了计算机应用方式的变化；大量数据处理导致第一代数据库管理系统的诞生。

软件系统的规模越来越大，复杂程度越来越高，软件可靠性问题也越来越突出。原来的个人设计、个人使用的方式不再能满足要求，迫切需要改变软件生产方式，提高软件生产率，软件危机开始爆发。

1.1 软件危机

产生背景:

1968 年北大西洋公约组织的计算机科学家在联邦德国召开国际会议，第一次讨论软件危机问题，并正式提出“软件工程”一词，从此一门新兴的工程学科“软件工程学”为研究和克服软件危机应运而生。

1.1 软件危机

主要表现

- 软件成本日益增长
- 开发进度难以控制
- 软件质量差
- 软件维护困难

1.1 软件危机

20世纪60年代的软件危机使得人们开始重视软件工程的研究。

起初，人们把软件设计的重点放在数据结构和算法的选择上，随着软件系统规模越来越大、越来越复杂，整个系统的结构和规格说明显得越来越重要。

1.1 软件危机

软件危机的程度日益加剧，现有的软件工程方法对此显得力不从心。对于大规模的复杂软件系统来说，对总体的系统结构设计和规格说明比起对计算的算法和数据结构的选择已经变得明显重要得多。在此种背景下，人们认识到软件体系结构的重要性，并认为对软件体系结构的系统、深入的研究将会成为提高软件生产率和解决软件维护问题的新的最有希望的途径。

1.2 发展历史

自从软件系统首次被分成许多模块，模块之间有相互作用，组合起来有整体的属性，就具有了体系结构。

好的开发者常常会使用一些体系结构模式作为软件系统结构设计策略，但他们并没有规范地、明确地表达出来，这样就无法将他们的知识与别人交流。

软件体系结构是设计抽象的进一步发展，满足了更好地理解软件系统，更方便地开发更大、更复杂的软件系统的需要。

一、发展历程

软件体系结构这个术语第一次出现在1969年由NATO组织的一个关于软件工程技术会议上[1]。

从那时候一直到20世纪80年代末期，出现了一些与软件体系结构相关的研究，其中典型的研究成果包括：

[1] Kruchten P, Obbink H, Stafford J. The past, present, and future of software architecture. IEEE Software, 2006,23(2):22–30.

一、发展历程

Edsger Dijkstra首先提出了软件结构的重要性，他认为“软件体系结构的研究应该注重软件系统的分割及其结构，而不应该仅仅局限于实现一个正确的软件系统”，提出了结构化设计的思想，并且指出良好的软件结构体现出来的概念完整性将使软件的开发和维护大大受益[2]。

一、发展历程

Fred Brooks提出了概念结构的思想，认为系统的体系结构是完整、详细的用户接口说明，提出了在系统结构设计中，概念完整性是最重要的考虑因素，并进一步指出体系结构设计必须和实现区分开来，他指出“体系结构陈述的是发生了什么，而实现描述的是如何实现” [3]。

一、发展历程

David Parnas发展了软件结构研究，先后提出了信息隐蔽模块、软件结构和程序家族等概念[4-7]。

总体来说，该阶段的研究还没有形成完整的技术体系，并且在概念上也比较混乱，如软件结构、软件系统结构、软件系统组织等。

一、发展历程

软件体系结构作为一个单独的学科并对其开展系统的研究始于20世纪90年代初。

1991年，在Winston W. Royce和Walker Royce父子的一篇论文中，软件体系结构第一次同时出现在论文标题和论文内容中[8]。

一、发展历程

在1992年，Dewayne Perry和Alexander Wolf发表了对以后发展有巨大影响的文章“Foundations for the Study of Software Architecture”，在这篇文章中提出了关于什么是软件体系结构的著名公式[9]:

$\{\text{elements, forms, rationale}\} = \text{software architecture}$

这篇文章也被许多学者看作是软件体系结构正式成为软件工程的一个研究方向的标志。

一、发展历程

软件体系结构得到学术界和工业界广泛认同并进入全面发展始于1995年。

1995年第一届软件体系结构国际研讨会 IWSA-1 召开，IEEE Software 和 IEEE Transactions on Software Engineering 软件体系结构专刊出版。

1996年《Software Architecture : Perspectives on an Emerging Discipline》专著出版。标志着软件体系结构作为软件工程的一个研究分支正式提出。

1.2 发展历史

此后十年内，软件体系结构领域得到了蓬勃发展。越来越多的研究者关注并参与到软件体系结构的研究中来，与软件体系结构相关的会议、期刊、书籍等逐步增多，越来越多的知名国际会议将软件体系结构列入主要议题，并举行了大量直接以软件体系结构为主题的研讨会或国际会议（如软件体系结构国际研讨会ISAW, WICSA 等）。

1.2 发展历史

许多知名国际期刊中，与软件体系结构相关的研究成果逐渐增多，并出版了大量软件体系结构方面的书籍（如 SEI 软件工程系列丛书）。软件体系结构的研究还得到了工业界的广泛关注与认同，如 UML2 标准中引入了软件体系结构领域中连接子的概念，在实际软件开发过程(如统一软件开发过程)中也引入软件体系结构的概念和原则。2006 年出版的 IEEE Software 软件体系结构专刊总结了这十年间的软件体系结构研究与实践。

1.2 发展历史

随着软件系统的规模变得越来越大，复杂程度变得越来越高，软件体系结构也由最初模糊的概念发展到一个渐趋成熟的理论和技术。软件体系结构的发展大致经历了4个阶段：

- 无体系结构阶段
- 萌芽阶段
- 初级阶段
- 高级阶段

1.2 发展历史

无体系结构阶段(20世纪70年代以前)。
以汇编语言进行小规模应用程序开发为特征，这个阶段软件规模较小，很少明确考虑软件结构的问题。

1.2 发展历史

萌芽阶段(20世纪70年代中后期)。由于结构化开发方法的出现与广泛应用，软件开发中出现了概要设计与详细设计，而且主要任务是数据流设计与控制流设计。此时，软件结构已作为一个明确的概念出现在系统的开发中。

1.2 发展历史

初级阶段(20世纪80年代初至90年代中期)。面向对象开发方法逐渐兴起与成熟。因为对象是数据与基于数据之上操作的封装，因此在面向对象开发方法下，数据流设计与控制流设计则统一为对象建模。这个阶段出现了从不同侧面描述系统的结构模型，以UML(Unified Modeling Language, 统一建模语言)为典型代表。

1.2 发展历史

高级阶段(20世纪90年代中期以后)。以 philippe Kruehten提出的“4+1”模型为标志，以描述系统的高层抽象结构为中心，不关心具体的建模细节，划分了体系结构模型与传统软件结构的界限。

在高级阶段，软件体系结构已经作为一个明确的文档和中间产品存在于软件开发过程中，同时，软件体系结构作为一门学科逐渐得到人们的重视，并成为软件工程领域的研究热点，因而 Dewayne Perry和 Alexander Wolf认为，“未来的年代将是研究软件体系结构的时代”。

1.3 定义

虽然软件体系结构已经在软件工程领域中有着广泛的应用，但迄今为止还没有一个软件体系结构被大家所公认的定义。许多专家学者从不同角度和不同侧面对软件体系结构进行了刻画。

长期以来，CMU-SEI（卡内基梅隆大学软件工程研究所）在其网站上公开征集SA的定义，至今已有百余种。

1.3 定义

<http://www.sei.cmu.edu/architecture/start/glossary/index.cfm>

- Classic Software Architecture Definitions

经典的软件体系结构定义

- Modern Software Architecture Definitions

现代的软件体系结构定义

- Bibliographic Software Architecture Definitions

著作中的软件体系结构定义

- Published Software Architecture Definitions

出版物中的软件体系结构定义

- Community Software Architecture Definitions

机构的软件体系结构定义

1.3 定义

Perry & Wolf 1992

$SA = \{\text{elements}, \text{form}, \text{rational}\}$ 。软件体系结构是由一组元素(elements)、软件体系结构形式(form)和准则(rational)组成。

1.3 定义

Perry & Wolf 1992

$SA = \{\text{elements}, \text{form}, \text{rational}\}$ 。软件体系结构是由一组元素(elements)、软件体系结构形式(form)和准则(rational)组成。

元素(elements) 包括处理元素(processing elements)、数据元素(data elements)和连接元素(connecting elements)共3类，处理元素负责对数据进行加工，数据元素是被加工的信息，连接元素把体系结构的不同部分组合连接起来。

1.3 定义

Perry & Wolf 1992

$SA = \{\text{elements}, \text{form}, \text{rational}\}$ 。软件体系结构是由一组元素(elements)、软件体系结构形式(form)和准则(rational)组成。

软件体系结构形式(form)是由专有特性(properties)和关系(relationship)组成，专有特性用于限制软件体系结构元素的选择，关系用于限制软件体系结构元素组合的拓扑结构。

1.3 定义

Perry & Wolf 1992

$SA = \{\text{elements}, \text{form}, \text{rational}\}$ 。软件体系结构是由一组元素(elements)、软件体系结构形式(form)和准则(rational)组成。

而在多个体系结构方案中选择合适的体系结构方案往往基于一组准则(rational)。

1.3 定义

Garlan & Shaw 1993

软件体系结构是设计过程的一个层次，它处理算法和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，如大组织结构和全局控制结构；关于通讯、同步和数据存取的协议；设计元素的功能定义、物理分布和合成；设计方案的选择、评估和实现等。

1.3 定义

Bass等1994

一个系统的体系结构设计至少从以下三个方面进行描述：该系统应用领域的功能分割、系统结构和结构的领域功能分配。

1.3 定义

Hayes-Roth 1994

软件体系结构是一个由功能构件组成的抽象系统的说明，按照功能构件的行为、界面和构件之间的相互作用进行描述。

1.3 定义

Garlan & Perry 1995

软件体系结构包括一个程序/系统的构件的结构、构件的相互关系、以及控制构件设计演化的原则和指导三个方面。

1.3 定义

Boehm等1995

一个软件体系结构由以下三部分组成：一组软件系统的成分、连接和约束；一组系统仓库管理员提出的需求；一个理论，它证明用构件、连接和约束定义的一个系统在实现后能满足系统仓库管理员需求。

1.3 定义

Soni, Nord, Hofmeister 1995

软件体系结构至少由四个不同的实例化结构，这些结构从不同的方面描述了系统：

- 概念体系结构按照主要设计元素和这些元素之间的关系描述系统；
- 模块连接体系结构包含两个正交的结构：功能分解和分层；
- 执行体系结构描述了一个系统的动态结构；
- 代码结构描述了在开发环境中源代码、二进制和库是如何组织的。

1.3 定义

Show 1995

将软件体系结构定义分类为多个模型：

结构模型认为软件体系结构是由构件、构件间的连接加上一些其他方面组成的，包括：配置，风格；条件，语义；分析，属性；理论，需求说明、仓库管理员需求。

框架模型类似于结构观点，但主要强调整个系统的内在结构（常常是单一的），而不是它的组成。

动态模型重在系统的行为质量，“动态”可以是整个系统的配置变化、通讯和交互通道的设置或取消，也可以是计算过程中的动态性，如数值改变等。

过程模型重在体系结构的构造，构造的步骤和过程。按照这个观点，体系结构是实施过程描述的结果。

1.3 定义

ANSI/IEEE标准1471-2000

SA={component, relationship, environment, principle}。软件体系结构是一个系统的基本组织结构，表现为构件、构件之间的相互关系、构件与环境之间的相互关系，以及指导其设计和演化的原理。

1.3 定义

Bass, Clements, Kazman 2003

一个程序或计算系统的软件体系结构是该系统的一个或多个结构，它们由软件元素、元素的外部可见属性以及这些元素之间的关系组成。其中，“外部可见”属性是指其他元素可以了解一个元素的前提，包括提供的服务、性能特性、错误处理、共享资源使用等。

1.3 定义

Clements等 2010

软件体系结构是关于系统的一组结构，它们由软件元素、软件元素之间的关系和两者的属性组成。

1.3 定义

张效祥，计算机科学技术百科全书，清华大学出版社，p628，2005

软件体系结构software architecture是软件总体结构的抽象表示，或以此为研究对象的学科。

- 规定性含义
- 描述性含义
- 多视面含义

1.3 定义

规定性含义

软件体系结构=（结构元集，结构形，结构理）

结构元集为一组构成软件的结构元。结构元有三类，即处理元、信息元和连接元。处理元为对信息元施行处理的构件，信息元为处理元的处理对象，连结元负责构件间的连接。

结构形包括特性、联系以及权重。特性用以约束结构元的选取，联系则约束结构元间的交互与组织，权重表示特性及联系的重要程度。

结构理刻画体系结构人员选取体系结构风格、结构元、结构形的动因与根据。

体系结构风格是各种特定体系结构中结构元与结构形的抽象，它不如特定体系结构约束严格，也不如特定体系结构完备。

1.3 定义

描述性含义

软件体系结构=（构件集，连件集，模式，约束集）

构件集表示构成软件的一组组成元素，连件集为一组连件，用以刻画各构件间的交互，模式为软件设计风格的描述，反映由构件及连件构成软件的构成原则，约束集中的约束表示对模式所加的限制条件。

1.3 定义

多视面含义

软件体系结构为软件的一个或多个结构，每一个结构反应一种视面。

软件体系结构=结构集

结构=（构件集，外部可见特性集，联系集）

构件集表示构成软件的一组组成元素，外部可见特性反映为其他构件可利用该构件所作的假定，联系用以沟通相关构件。

1.4 研究内容

- 软件体系结构建模
- 软件体系结构描述方法
- 软件体系结构分析、设计与验证
- 软件体系结构发现、演化与重用
- 基于体系结构的软件开发方法
- 特定领域的体系结构框架
- 软件体系结构评估方法

1.4 研究内容

1、软件体系结构建模

研究软件体系结构的首要问题是如何表示软件体系结构，即如何对软件体系结构建模。根据建模的侧重点不同，可以将软件体系结构的模型分为5种，分别是结构模型、框架模型、动态模型、过程模型和功能模型。在这5个模型中，最常用的是结构模型和动态模型。

1.4 研究内容

1、软件体系结构建模

上述5种模型各有所长，将5种模型有机地统一在一起，形成一个完整的模型来刻画软件体系结构更合适。Phillppe Kruchien在1995年提出了一个“4+1”的视图模型。

“4+1”视图模型从5个不同的视角包括逻辑视图、进程视图、物理视图、开发视图和场景视图来描述软件体系结构。每一个视图只关心系统的一个侧面，5个视图结合在一起才能反映软件体系结构的全部内容。

1.4 研究内容

2、软件体系结构描述方法

- 图形表达工具
- 模块内连接语言
- 基于软构件的系统描述语言
- 软件体系结构描述语言
- 软件体系结构描述框架标准

1.4 研究内容

2、软件体系结构描述方法

➤图形表达工具

用矩形框代表抽象构件，框内标注的文字为抽象构件的名称，用有向线段代表辅助各构件进行通讯、控制或关联的连接件。

1.4 研究内容

2、软件体系结构描述方法

➤模块内连接语言

模块内连接语言 (Module Interconnection Language, MIL)将一种或几种传统程序设计语言的模块连接起来。由于程序设计语言和模块内连接语言具有严格的语义基础，因此能支持对较大的软件单元进行描述。

1.4 研究内容

2、软件体系结构描述方法

➤基于软构件的系统描述语言

基于软构件的系统描述语言将软件系统描述成一种是由许多以特定形式相互作用的特殊软件实体组成的系统。

1.4 研究内容

2、软件体系结构描述方法

➤软件体系结构描述语言

软件体系结构描述语言 (Architecture Description Language, ADL)是参照传统程序设计语言的设计和开发经验, 针对软件体系结构特点, 重新设计、开发的描述方式。

1.4 研究内容

2、软件体系结构描述方法

➤软件体系结构描述框架标准

鉴于体系结构描述的概念与实践的不统一，IEEE于1995年8月成立了体系结构工作组，综合体系结构描述研究成果，并参考业界的体系结构描述的实践，负责起草了体系结构描述框架标准即IEEE P1471，并于2000年9月21日通过IEEE-SA标准委员会评审。

IEEE P1471仅仅提供了体系结构描述的概念框架、体系结构描述应该遵循的规范，但如何描述以及具体的描述技术等方面缺乏更进一步的指导。

1.4 研究内容

3、软件体系结构分析、设计与验证

软件体系结构分析的内容可分为结构分析、功能分析和非功能分析。

1.4 研究内容

3、软件体系结构分析、设计与验证

体系结构设计是生成一个满足软件需求的体系结构的过程，其本质在于将系统分解成相应的组成成分，并将这些成分重新组装成一个系统。目前，体系结构设计主要有两大类方法，分别是过程驱动方法和问题列表驱动方法。

体系结构设计研究的重点内容之一就是体系结构风格/模式，体系结构风格在本质上反映了一些特定的元素、按照特定的方式组成一个特定的结构，该结构应有利于上下文环境下的特定问题的解决。体系结构风格可分为两大类，分别为固定术语和参考模型。

1.4 研究内容

3、软件体系结构分析、设计与验证

体系结构验证是对系统结构进行测试和验证，体系结构验证有两大类方法：体系结构测试和体系结构形式化验证。

1.4 研究内容

4、软件体系结构发现、演化与重用

体系结构发现解决如何从已经存在的系统中提取软件体系结构的问题，属于逆向工程范畴。

1.4 研究内容

4、软件体系结构发现、演化与重用

体系结构演化是指由于系统需求、技术、环境、分布等因素的变化而最终导致软件体系结构变动的过程。软件在运行时刻的体系结构变化称为体系结构的动态性，而将体系结构的静态修改称为体系结构扩展。

体系结构的动态性分为有约束的和无约束的以及结构动态性和语义动态性。

1.4 研究内容

4、软件体系结构发现、演化与重用

体系结构重用属于设计重用，比代码重用更抽象。由于软件体系结构是系统的高层抽象，反映了系统的主要组成元素及其交互关系，因而较算法更稳定，更适合于重用。

1.4 研究内容

5、基于体系结构的软件开发方法

引入了体系结构之后，软件开发的过程变为“问题定义—>软件需求—>软件体系结构设计—>软件设计—>软件实现”，可以认为软件体系结构架起了软件需求与软件设计之间的一座桥梁。而在由软件体系结构到实现的过程中，借助中间件技术与软件总线技术，软件体系结构将易于映射成相应的实现。

1.4 研究内容

6、特定领域的体系结构框架

DSSA就是在一个特定应用领域中为一组应用提供组织结构参考的标准软件体系结构。对DSSA研究的角度、关心的问题不同导致了对DSSA的不同定义。

1.4 研究内容

7、软件体系结构评估方法

软件体系结构的设计是整个软件开发过程中关键的一步，但是，怎样才能知道为软件系统所选用的体系结构是恰当的呢？如何确保按照所选用的体系结构能顺利地开发出成功的软件产品呢？要回答这些问题并不容易，因为它受到很多因素的影响，需要专门的方法来对其进行评估。

体系结构评估可以只针对一个体系结构，也可以针对一组体系结构。从目前已有的软件体系结构评估技术来看，可以归纳为三类主要的评估方式，分别是基于调查问卷或检查表的方式、基于场景的方式和基于度量的方式。

1.5 意义

- 风险承担者进行交流的手段
- 早期设计决策的体现
- 可传递和可重用的模型

1.5 意义

➤ 风险承担者进行交流的手段

构建软件体系结构的过程是一种对软件系统的抽象过程，它有选择性地突出和隐蔽某些系统特征，完成对系统的高层建模活动，从而更有利于人们对复杂软件的系统理解：其一，有效增强系统相关人员的信息交流；其二，有效改进软件系统和软件过程的理解。

一个良好的软件体系结构是一张融合了诸多需求角度的多维视图，它提供了一种通用语言，使不同的需求得以表达、交流、协商和达成共识。良好的体系结构不仅有效改善了人员间的交流，而且使得基于体系结构描述的系统理解取得了巨大成功。

1.5 意义

➤早期设计决策的体现

软件体系结构体现了系统的最早的一组设计决策，这些早期的约束比起后期的工作重要的多，对系统生命周期的影响也大得多。

- (1) 明确了对系统实现的约束条件
- (2) 决定了开发和维护组织的结构
- (3) 制约着系统的质量属性
- (4) 使推理和控制变更更简单
- (5) 有助于循序尽进的原型设计

1.5 意义

➤可传递和可重用的模型

软件体系结构级的重用意味着体系结构的决策能在具有相似需求的多个系统中发生影响，这比代码级的重用有更大的好处。

谢谢大家！

再见！

