

卡内基梅隆大学软件研究所关于软件架构的定义[3] PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/141/2021_2022__E5_8D_A1_E5_86_85_E5_9F_BA_E6_c29_141467.htm 系统架构的设计要求架构师具备软件和硬件的功能和性能的过硬知识，这一工作无疑是架构设计工作中最为困难的工作。此外，从每一个角度上看，都可以看到架构的两要素：元件划分和设计决定。首先，一个软件系统中的元件首先是逻辑元件。这些逻辑元件如何放到硬件上，以及这些元件如何为整个系统的可扩展性、可靠性、强壮性、灵活性、性能等做出贡献，是非常重要的信息。其次，进行软件设计需要做出的决定中，必然会包括逻辑结构、物理结构，以及它们如何影响到系统的所有非功能性特征。这些决定中会有很多是一旦作出，就很难更改的。来源：www.examda.com 源：www.examda.com 根据作者的经验和经验，一个基于数据库的系统架构，有多少个数据表，就会有多少页的架构设计文档。比如一个中等的数据库应用系统通常含有一百个左右的数据表，这样的系统设计通常需要有一百页左右的架构设计文档。构架描述 为了讨论和分析软件构架，必须首先定义构架表示方式，即描述构架重要方面的方式。在 Rational Unified Process 中，软件构架文档记录有这种描述。构架视图 我们决定以多种构架视图来表示软件构架。每种构架视图针对于开发流程中的涉众（例如最终用户、设计人员、管理人员、系统工程师、维护人员等）所关注的特定方面。构架视图显示了软件构架如何分解为构件，以及构件如何由连接器连接来产生有用的形式 [PW92]，由此记录主要的结构设计决策。这些设计决策必须基于需

求以及功能、补充和其他方面的约束。而这些决策又会在较低层次上为需求和将来的设计决策施加进一步的约束。典型的构架视图集 构架由许多不同的构架视图来表示，这些视图本质上是以图形方式来摘要说明“在构架方面具有重要意义”的模型元素。在 Rational Unified Process 中，您将从一个典型的视图集开始，该视图集称为“4 1 视图模型” [KRU95]。

它包括：

- 用例视图：包括用例和场景，这些用例和场景包括在构架方面具有重要意义的行为、类或技术风险。它是用例模型的子集。
- 逻辑视图：包括最重要的设计类、从这些设计类到包和子系统的组织形式，以及从这些包和子系统到层的组织形式。它还包括一些用例实现。它是设计模型的子集。
- 实施视图：包括实施模型及其从模块到包和层的组织形式的概览。同时还描述了将逻辑视图中的包和类向实施视图中的包和模块分配的情况。它是实施模型的子集。
- 进程视图：包括所涉及任务（进程和线程）的描述，它们的交互和配置，以及将设计对象和类向任务的分配情况。只有在系统具有很高程度的并行时，才需要该视图。在 Rational Unified Process 中，它是设计模型的子集。
- 配置视图：包括对最典型的平台配置的各种物理节点的描述以及将任务（来自进程视图）向物理节点分配的情况。只有在分布式系统中才需要该视图。它是部署模型的一个子集。

构架视图记录在软件构架文档中。您可以构建其他视图来表达需要特别关注的不同方面：用户界面视图、安全视图、数据视图等等。对于简单系统，可以省略 4 1 视图模型中的一些视图。构架重点 虽然以上视图可以表示系统的整体设计，但构架只同以下几个具体方面相关：模型的结构，即组织模式，例如分层。基本元素，即关

键用例、主类、常用机制等，它们与模型中的各元素相对。几个关键场景，它们表示了整个系统的主要控制流程。记录模块度、可选特征、产品线状况的服务。构架视图在本质上是整体设计的抽象或简化，它们通过舍弃具体细节来突出重要的特征。在考虑以下方面时，这些特征非常重要：系统演进，即进入下一个开发周期。在产品线环境下复用构架或构架的一部分。评估补充质量，例如性能、可用性、可移植性和安全性。向团队或分包商分配开发工作。决定是否包括市售构件。插入范围更广的系统。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com