

软件体系结构 作业03

22920212204392 黄勛

1 Reading [Software Architecture4+1](#) ，试给出SA中4+1视图的描述

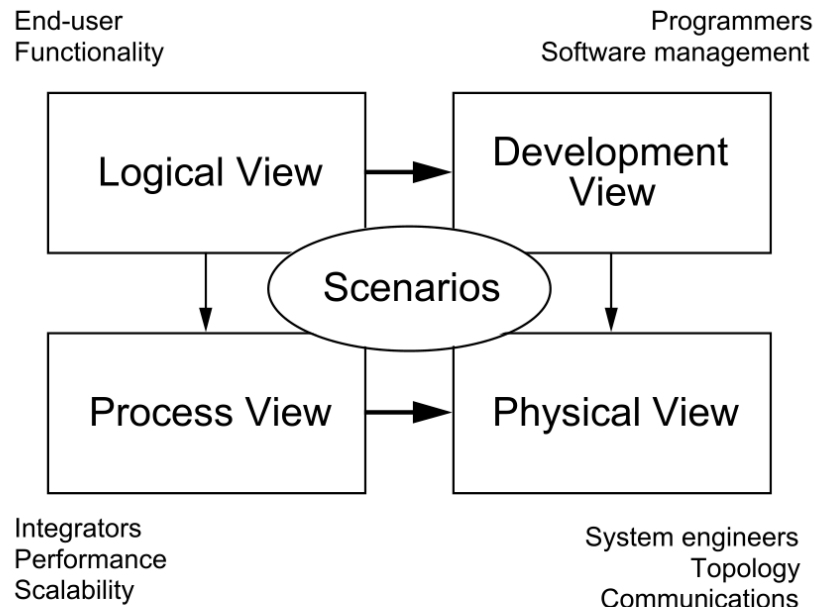


Figure 1 — The “4+1” view model

SA中的4+1视图中的“4”分别为逻辑视图、过程视图、物理视图、开发视图，“1”为场景视图。

逻辑体系结构面向对象的分解逻辑体系结构主要支持系统在为用户提供服务方面应提供的功能需求。系统被分解为一组关键抽象，这些抽象取自问题域，以对象或对象类的形式。它们利用了抽象、封装和继承的原则。这种分解不仅用于功能分析，还用于识别系统各个部分的常见机制和设计元素。

过程体系结构过程分解过程体系结构考虑了一些非功能性需求，如性能和可用性。它解决了并发性和分布性、系统的完整性、容错性以及逻辑视图中的主要抽象如何适应过程体系结构的问题，其中控制线程是对实际执行的对象的操作。

开发体系结构子系统分解开发体系结构侧重于软件开发环境中的实际软件模块组织。该软件被打包成小块程序库或子系统，可以由一个或少数开发人员开发。子系统按层次结构组织，每一层都为其上面的层提供一个狭窄且定义明确的接口。

物理体系结构将软件映射到硬件物理体系结构主要考虑系统的非功能性要求，如可用性、可靠性、性能和可扩展性。该软件在计算机网络或处理节点上执行。所识别的各种元素——网络、过程、任务和对象——需要映射到各种节点上。我们预计将使用几种不同的物理配置：一些用于开发和测试，另一些用于为不同的站点或不同的客户部署系统。因此，软件到节点的映射需要高度灵活，并且对源代码本身的影响最小。

场景将所有这些放在一起四个视图中的元素通过使用一小组重要场景无缝地协同工作。从某种意义上说，这些场景是对最重要需求的抽象。他们的设计是使用对象场景图和对象交互图来表达的。该视图与其他视图是多余的，但它有两个主要目的：作为在架构设计过程中发现架构元素的驱动因素；作为该架构设计完成后的验证和说明角色。