

软件体系结构和质量属性的关系

性能 可用性 可靠性 安全性 易用性 可更改性 可移植 可重用性 可集成性 可测试性（可为内部指标）其他为外部指标

在纯数据流系统中，处理之间除了数据交换，没有任何其他的交互

数据流风格软件系统被看作 由一些系列作用于数据集合上的“变换”组件组成 这些组件相互独立，可重复使用，可并发执行 不存在全局控制组件的行为

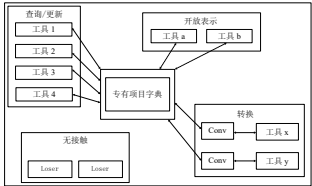
软件工程师关注点 数据如何流动 数据的可用性、变换和延迟



- (1) 由于每个组件行为不受其他组件的影响，整个系统的行为易于理解。
- (2) 支持功能模块的重用，任意两个过滤器只要相互间所传输的数据格式上达到一致，就可以连接在一起。
- (3) 系统易于维护和扩展。新的过滤器容易加入到系统中，旧的过滤器也可被改进的过滤器替换。
- (4) 支持特殊的分析，如吞吐量分析、死锁分析。

特征	仓库风格	黑板风格
数据控制	被动存储，组件主动读写	主动驱动，知识源响应数据变化
问题类型	确定性、数据密集型问题	非确定性、高推理的复杂问题
协作机制	组件间无直接交互	知识源通过黑板动态协作
流程	预定义、顺序化	动态、迭代、异步或协调式
典型控制方式	人集中控制，组件自主操作	可能引入控制模块调整知识源执行顺序

- (5) 支持并发执行，每个过滤器既可以独立运行，也可以和其它过滤器并发执行。



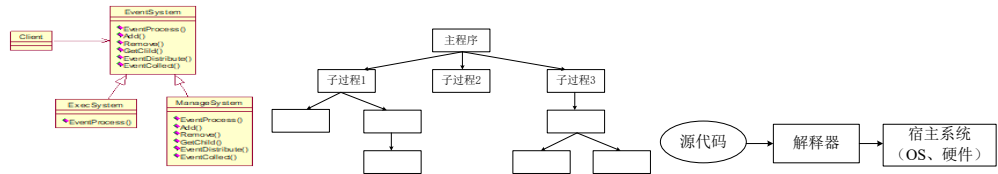
优点 便于模块间的数据共享，方便模块的添加、更新和删除，避免了知识源的不必要的重复存储等。缺点对于各个模块，需要一定的同步/加锁机制保证数据结构的完整性和一致性等。

- (1) 便于多客户共享大量数据，他们不关心数据何时有的、谁提供的、怎样提供的。(2) 既便于添加新的作为知识源代理的应用程序，也便于扩展共享的黑板数据结构。3) 知识源可重用。(4) 支持容错性和健壮性。

不同的知识源代理对于共享数据结构要达成一致，而且，这也造成对黑板数据结构的修改较为困难——要考虑到各个代理的调用 需要一定的同步/加锁机制保证数据结构的完整性和一致性，增大了系统复杂度。不能保证有好的解决方案 测试困难 低效，开发成本高

类结构简明 定义了包含执行子系统和和管理子系统的类层次结构。简化客户代码 使整个系统的设计更具有一般化。

- 构件放弃了对系统计算的控制 一个构件触发一个事件时，不能确定其它构件是否会响应它。即使它知道事件注册了哪些构件的构成，它也不能保证这些过程被调用的顺序。有时数据可被一个事件传递，但另一些情况下，基于事件的系统必须依靠一个共享的仓库进行交互。这也使全局性能和资源管理便成了问题。正确性的推理存在问题 这是由于过程的语义必须依赖于被触发事件的上下文约束。



面向对象组织结构有两个重要特点：(1)对象负责维护其表示的完整性 (2)对象的表示对其他对象而言是隐藏的。

优点 (1) 对象隐藏了其实现细节，所以可以在不影响其它对象的情况下改变对象的实现，不仅使得对象的使用变得简单、方便，而且具有很高的安全性和可靠性。(2) 设计者可将一些数据存取操作的问题分解成一些交互的代理程序的集合。

缺点 当一个对象和其它对象通过过程调用等方式进行交互时，必须知道其它对象的标识。无论何时改变对象的标识，都必须修改所有显式调用它的其它对象，并消除由此带来的一些副作用。

层次化结构基本思想

在分层系统（Layered System）中，系统被组织成若干个层次，每个层次由一系列组件组成；层次之间存在接口，通过接口形成 call/return 的关系——下层组件向上层组件提供服务，上层组件被看作是下层组件的客户端。

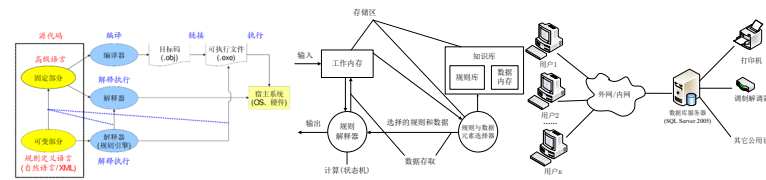
优点 支持基于抽象程度递增的系统设计，使设计者可以把一个复杂系统按递增的步骤进行分解

支持功能增强，因为每一层至多和相邻的上下层交互，因此功能的改变最多影响相邻的上下层。支持重用。只要提供的服务接口

定义不变，同一层的不同实现可以交换使用。这样就可以定义一组标准的接口，而允许各种不同的实现方法。

缺点 并不是每个系统都可以很容易的划分为分层的模式，甚至即使一个系统的逻辑结构是层次化，出于对系统性能考虑，系统设计者不得不把一些低级或高级的功能综合起来。很难找到一个合适的、正确的层次抽象方法

优点 (1) 它有利于实现程序的可移植性和语言的跨平台能力；(2) 可以对未来的硬件进行模拟和仿真，能够降低测试所带来的复杂性和昂贵花费 缺点 额外的间接层次导致了系统性能的下降，



C/Sf 风格优点：(1) 客户机组件和服务组件分别运行在不同的计算机上，有利于分布式数据的组织和处理。(2) 组件之间的位置是相互透明的(3) 客户机程序和服务程序可运行在不同的操作系统上，便于实现异构环境和多种不同开发技术的融合(4) 软件环境和硬件环境的配置具有极大的灵活性，易于系统功能的扩展(5) 将大规模的业务逻辑分布到多个通过网络连接的低成本的计算机上，降低了系统的整体开销

缺点：(1) 开发成本较高(2) 客户机程序的设计复杂度大，客户机负荷重。(3) 信息内容和形式单一。(4) C/S 体系结构升级需要开发人员到现场更新客户机程序，对运行环境进行重新配置，增加了维护费用。(5) 两层 C/S 结构采用了单一的服务器，同时以局域网为中心，难以扩展到 Intranet 和 Internet。(6) 数据安全性不高。

B/S“ (1) 客户端只需要安装浏览器，操作简单，能够发布动态信息和静态信息。(2) 运用 HTTP 标准协议和统一客户端软件，能够实现跨平台通信。(3) 开发成本比较低，只需要维护 Web 服务器程序和中心数据库。客户端升级可以通过升级浏览器来实现。

P2P：整体稳定，不会因为一点的错误影响全局资源共享，任务分摊架构 1 两点之间可能不通某些情况可以从其他点获得同样的资源 架构 2 两点之间可以寻求多种路径互通每点还要承担一定量(很可能大量)的中转负荷 容易产生泛洪现象，浪费带宽