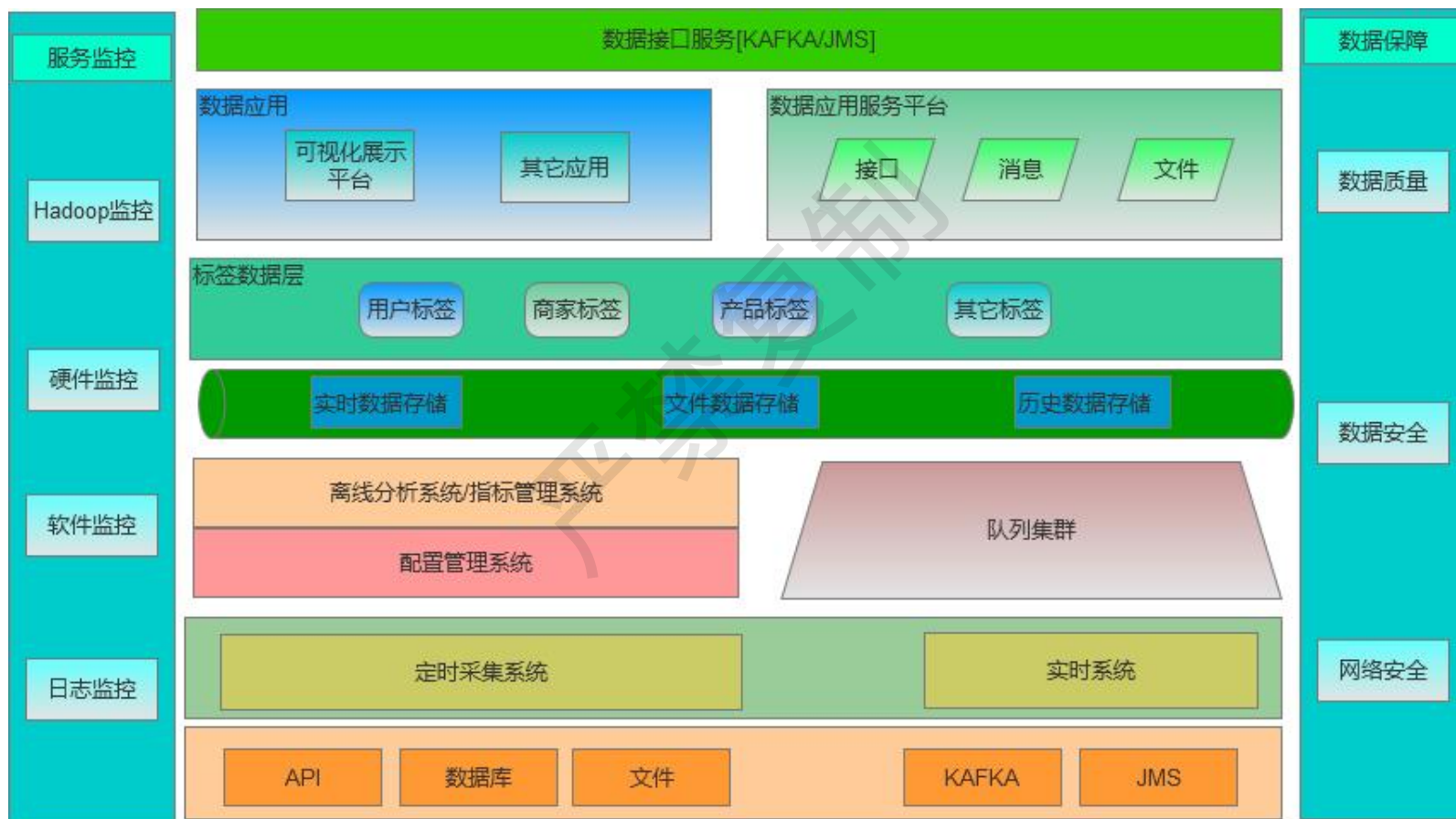


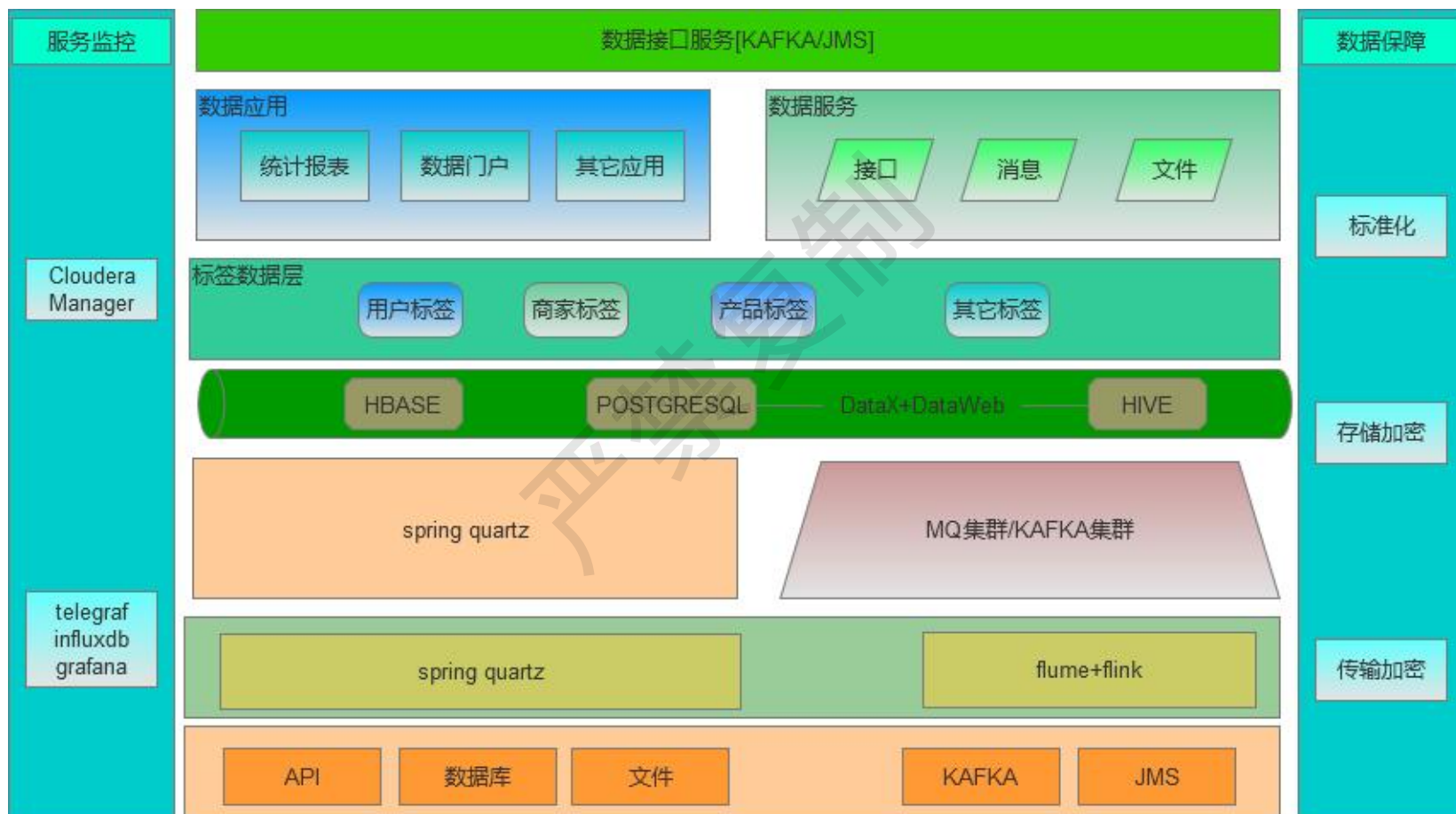


# 软件体系结构原理、方法与实践

## ❁ 软件体系结构描述-引入



## ❁ 软件体系结构描述-引入

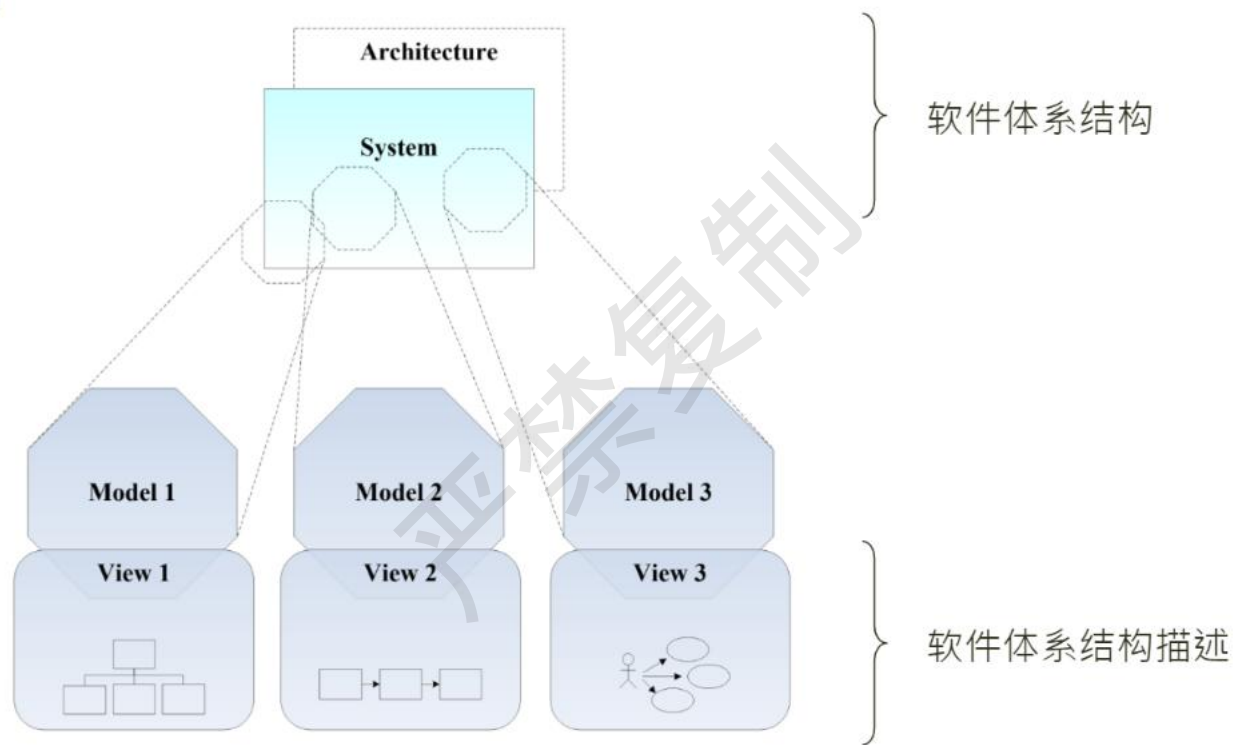


### ❁ 软件体系结构描述-与软件体系结构的区别

软件体系结构是隶属于系统之中的，只要有软件系统就有体系结构。如：一个石头的形状和重量。

软件体系结构描述是将软件体系结构可视化的手段和产物  
如：描述一个石头的形状和重量

### ❁ 软件体系结构描述-与软件体系结构的区别



思考:怎么描述一块石头的形状重量?

### ❁ 描述方法的分类

- ◎ 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 (ADL)



### 描述方法的分类-图形表达工具

**矩形框:**代表抽象构件，框内标注的文字为抽象构件的名称

**有向线段:**代表辅助各构件进行通信、控制和关联的连接件

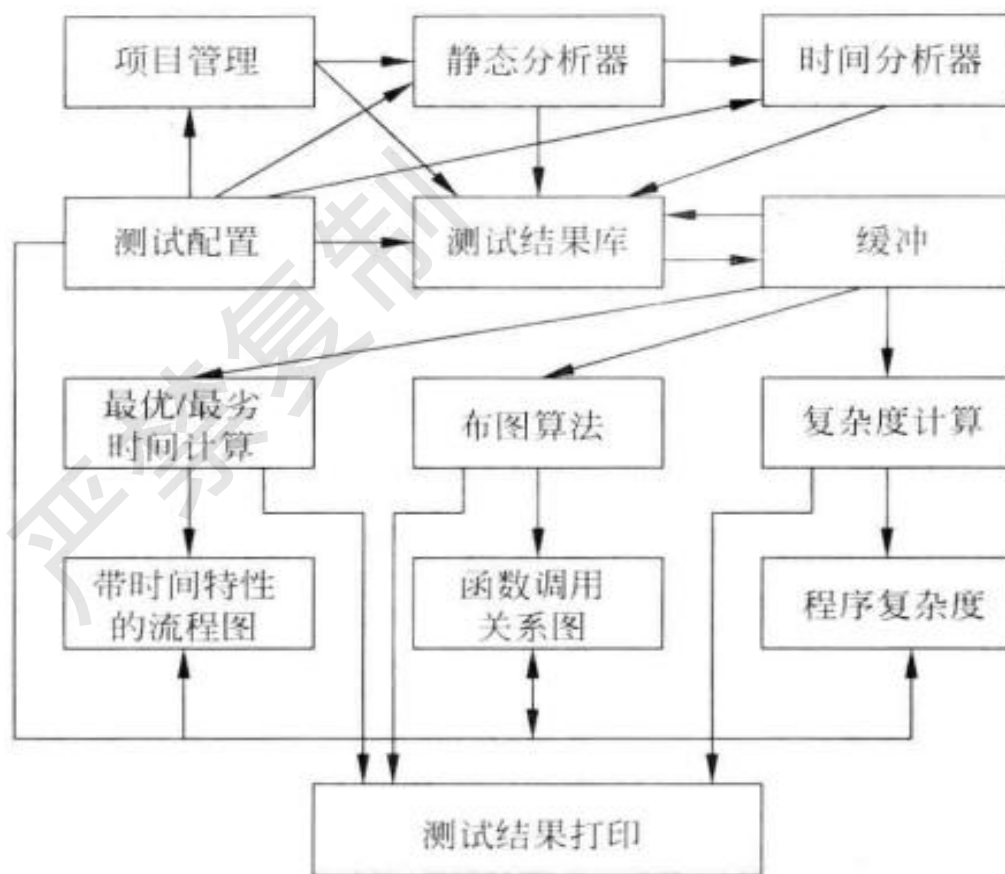


图 4-1 某软件辅助理解测试工具部分体系结构描述



### ❁ 描述方法的分类-图形表达工具

应用现状：

- 在软件设计中占主导地位
- 在实际开发和设计中被广泛使用
- 为工作人员传递了大量的体系结构思想

缺点：

术语和表达语义上存在一些不规范和不精确使得图形表达方式在不同的系统和不同的文档之间有许多不一致甚至矛盾的地方。



### ❁ 描述方法的分类-模块内连接语言

- 采用将一种或几种传统程序设计语言的模块连接起来的描述软件体系结构的方法。如Ada语言，Pascal语言。一般来说MIL与实现的语言无关，只关心构件的对外表现协议及构件之间的通讯关系。
- 特点：
  - 语义比较丰富，局限在实现级别，层次比较低。
  - 语义精确，有编译器做保证
  - 没有或极少有形式化基础

### ❁ 描述方法的分类-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构造组成的组织或系统。
- 例如，一种多变配置语言(PCL)就可以用来在一个较高的抽象层次上对系统的体系结构建模，Darwin(分布式系统配置语言)最初用作设计和构造复杂分布式系统的配置说明语言，因具有动态特性，也可用来描述动态体系结构。
- 这种表达和描述方式虽然也是较好的一种以构件为单位的软件系统描述方法，但是他们所面向和针对的系统元素仍然是一些**层次较低**的以程序设计为基础的通信协作软件实体单元，而且这些语言所描述和表达的系统一般而言都是面向**特定应用**的特殊系统，这些特性使得基于软构件的系统描述仍然不是十分适合软件体系结构的描述和表达。

### 描述方法的分类-基于软构件的系统描述语言

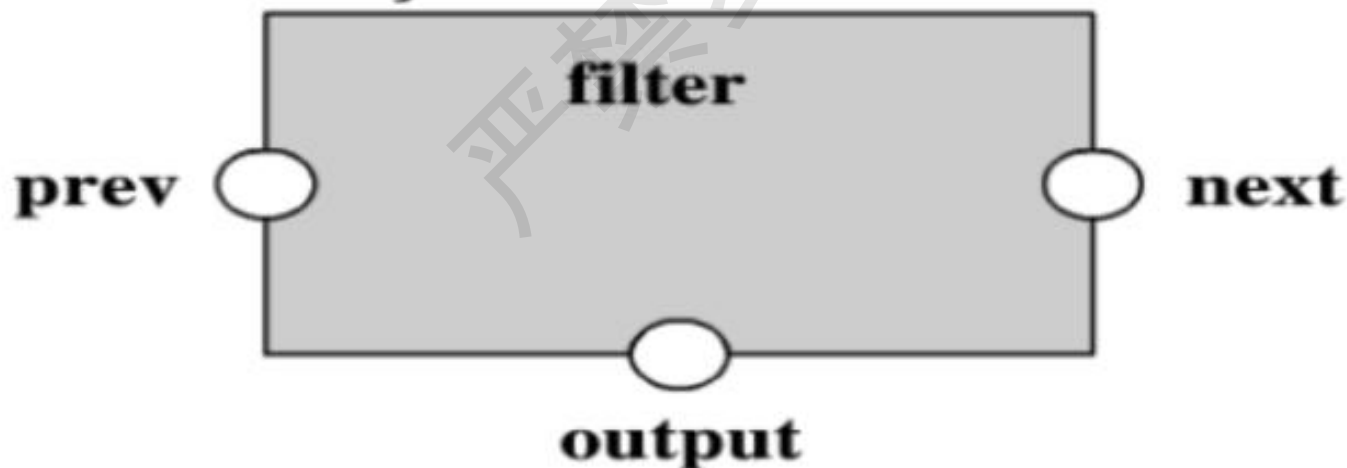
例：Darwin 对 Filter 构件的图形化和文本化描述：

**Component{**

**provide prev <port, int>;**

**require next <port, int>;**

**}**



### ❁ 描述方法的分类-软件体系结构描述语言(ADL)

- ADL是参照传统程序设计语言的设计和开发经验，重新设计、开发和使用针对软件体系结构特点的专门的软件体系结构描述语言。
- 由于ADL是在吸收了传统程序设计中的语义严格精确的特点基础上，针对软件体系结构的整体性和抽象性特点，定义和确定适合于软件体系结构表达与描述的有关抽象元素，因此，ADL是当前软件开发和设计方法学中一种发展很快的软件体系结构描述方法，目前，已经有几十种常见的ADL。

### ❁ 描述方法的分类-软件体系结构描述语言(ADL)

每种ADL强调的侧面不同:

Aesop: 支持体系结构风格的应用

MetaH: 为设计师提供了关于实时电子控制软件系统的设计指导

C2: 支持基于消息传递风格的用户界面系统的描述

Rapide: 支持体系结构设计的模拟并提供了分析模拟结果的工具

SADL: 提供了关于体系结构加细的形式化基础

Unicon: 支持异构的构件和连接类型并提供了关于体系结构的  
高层编译器

Wright: 支持体系结构构件之间交互的说明和分析。

### 描述方法的分类-软件体系结构描述语言(ADL)

#### Wright-概述

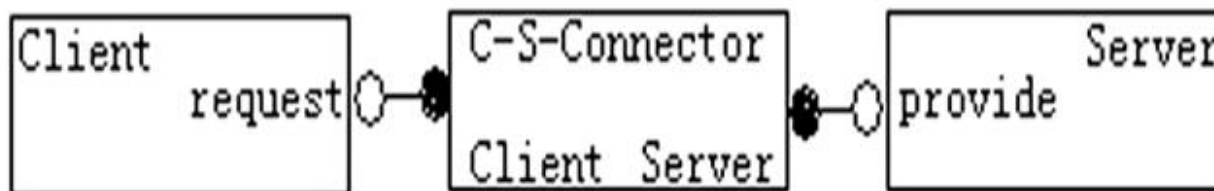
- 特点：使用显式的、独立的连接件类型作为交互模式。
  - 构件定义由3部分组成：构件标识符、端口集合、表示构件计算行为的CSP进程。
  - 构件通过端口与其它构件进行交互。连接件也有类似于构件的端口。粘接规约描述参与交互各方的行为如何协调、组合。
- 符号定义如下：
  - $e?x$ 表示 $x$ 是事件 $e$ 的输入值。在通道 $e$ 上输入 $x$ 。
  - $e!y$ 表示 $y$ 是事件 $e$ 的输出值。在通道 $e$ 上输出 $y$ 。
  - $e \rightarrow P$ 前缀操作符，表示一进程执行事件 $e$ ，然后成为进程 $P$ 。
  - $P \sqcap Q$ 表示一进程不确定的选择进程 $P$ 或 $Q$ 执行。
  - $P \square Q$ 表示一进程根据与其发生交互的其它进程从进程 $P$ 和进程 $Q$ 中选择一个执行。
  - $\surd$ 终止操作符，表示进程可以终止。



### ❁ 描述方法的分类-软件体系结构描述语言(ADL)

#### Wright-例子

- 下图是一个简单的客户机/服务器模型。
  - 该模型由一个连接件和一个系统配置组成。  
客户的request端口和服务器的provide端口分别连接到连接器的Client和Server角色上。





### ❁ 描述方法的分类-软件体系结构描述语言(ADL)

#### Wright-例子

- 一个简单的客户服务器结构的体系结构

```
System SimpleExample
  Component Server =
    port provide [provide protocol]
    spec [Server specification]
  Component Client =
    port request [request protocol]
    spec [client specification]
  Connector C-S-connector =
    role client [client protocol]
    role server [server protocol]
    glue [glue protocol]
  Instances
    s: Server
    c: Client
    cs: C-S-connector
  Attachments:
    s.provide as cs.server
    c.request as cs.client
end SimpleExample
```

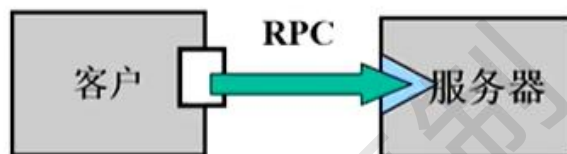
### ❁ 描述方法的分类-软件体系结构描述语言(ADL)

#### ACME-概述

- ACME是一种软件体系结构交换语言
  - 目的：解决目前出现的众多软件体系结构描述语言之间的不兼容问题。
  - 可以作为体系结构设计工具中的通用交换格式，也可以作为开发新的体系结构设计和分析工具的基础。
- ACME项目开始于1995年，由Carnegie Mellon大学的David Garlan等人开发。
  - 当时的目标是在不同的体系结构开发工具之间提供一种能够相互交换体系结构描述的工具。
  - 目前，ACME语言和工具开发库（Acmelib）为软件体系结构的描述、表示、生成和分析，提供了一种通用的、可扩展的基础设施。
- 尽管从严格意义上讲，ACME并不是一种体系结构描述语言，但是它包含了大量ADL特点，实用性很高。我们也作简要介绍。

### ❁ 描述方法的分类-软件体系结构描述语言(ADL)

#### ACME-例子



- 以C/S模型为例。图中包含两个构件和连接件。  
ACME描述如下：

```
System simple_CS={  
    Component Client = { Port send-request }  
    Component Server = { Port receive-request }  
    Connector RPC = { Roles (Caller, Callee) }  
    Attachment: { Client. send-request to RPC.Caller  
                  Server. receive-request to RPC.Callee  
    }  
}
```

### ❁ ADL与其他语言的比较-ADL语言特点

- 构造能力：ADL能够使用较小的独立体系结构元素来建造大型软件系统；
- 抽象能力：ADL使得软件体系结构中的构件和连接件描述可以只关注它们的抽象特性，而不管其具体的实现细节；
- 重用能力：ADL使得组成软件系统的构件、连接件甚至是软件体系结构都成为软件系统开发和设计的可重用部件；
- 组合能力：ADL使得其描述的每一系统元素都有其自己的局部结构，这种描述局部结构的特点使得ADL支持软件系统的动态变化组合；
- 异构能力：ADL允许多个不同的体系结构描述关联存在；
- 分析和推理能力：ADL允许对其描述的体系结构进行多种不同的性能和功能上的多种推理分析。

### ❁ 软件体系结构描述框架标准 – IEEE P1471

- IEEE P1471于2000年9月21日通过IEEE-SA标准委员会评审。
- IEEE P1471适用于软件密集的系统，其目标在于：便于体系结构的表达与交流，并通过体系结构要素及其实践标准化，奠定质量与成本的基础。
- IEEE P1471详细介绍了一套体系结构描述的概念框架，并给出建立框架的思路。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。



### ❁ 软件体系结构描述框架标准 – Rational

- Rational起草了可重用的软件资产规格说明，专门讨论了体系结构描述的规格说明，提出了一套易于重用的体系结构描述规范。该建议草案已经提交OMG。
- 基于RUP (Rational United Process)、采用UML模型描述软件的体系结构，认为体系结构描述的关键是定义视点、视图以及建模元素之间的映射关系。
- 与IEEE P1471相比，该建议标准的体系结构描述方案涉及面比较窄，所注重的层次比较低，因而更具体。由于将体系结构的描述限于UML和RUP，具有一定的局限性，但该建议标准结合了业界已经广泛采用的建模语言和开发过程，因而易于推广，可以有效实现在跨组织之间重用体系结构描述结果。