

软件体系结构原理、方法与实践

❁ 软件体系结构的动态性-引入



❁ 软件体系结构的动态性-背景

- 传统的结构化开发方法、面向对象开发方法、基于软件体系结构、构件的开发方法已经逐渐成为当前软件开发的主流，软件开发的基本单位已从传统的代码行，对象类转变为各种粒度的构件、构件之间的拓扑结构所形成的软件体系结构。
- 这给软件开发带来灵活性：通过构件重用和替换，实现构件“即插即用”
- 当前软件体系结构的研究主要集中在静态体系结构上，即在软件系统运行时，系统的体系结构不能发生改变。
- 静态体系结构缺乏表示动态更新的机制，很难用它描述在动态运行过程中体系结构变化的软件系统，而有些系统需要在运行过程中动态改变体系结构。
- 在软件体系结构层次上实现动态性会给大型软件系统的开发提供可扩展性，用户自定义和演化性。

新的开始，新的起点，让我们一起为梦想而努力。

❁ 软件体系结构的动态性-概述

- ◆ 由于系统需求、技术、环境，分布等因素的变化而导致软件体系结构的变动，称之为软件体系结构的演化。
- ◆ 软件系统在运行时的体系结构称之为体系结构的动态性(动态扩展)；软件体系结构的静态修改称为体系结构的扩展(静态扩展)。
- ◆ 软件体系结构的扩展和动态性都是软件体系结构的演化和适应(进化)。
- ◆ 软件体系结构的动态性主要分三类：
 - ◎ 交互式动态性。 如：允许在复合构件的固定连接中改变数据
 - ◎ 结构化动态性。 如：允许对系统添加或删除构件或连接件
 - ◎ 体系结构动态性。如：允许构件的整个配置发生改变

❁ 软件体系结构的动态性-概述

- ◆ 允许在系统运行时发生更新的软件体系结构称为**动态软件体系结构**，动态软件体系结构允许系统(包括软件和硬件部分)在被创建以后可以被动态更新
- ◆ 对系统的动态更新，动态维护工作很复杂，很难很好的解决。
- ◆ 目前对动态体系结构的研究还不成熟，还处于摸索阶段。
- ◆ 但有一些动态更新机制得到了一些应用，如动态链接库，EJB的构件组装机制。

关于软件体系结构特性描述正确的是？

- ☒ A 静态体系结构可通过构件重用和替换，实现构件“即插即用”
- ☒ B 静态体系结构在软件系统运行时，系统的体系结构不能发生改变
- ☒ C 软件系统在运行时的体系结构称之为体系结构的动态性（动态扩展）；
- ☒ D 软件体系结构的动态性分为交互式动态性，结构化动态性，体系结构动态性

提交

在不停服务的前提下给办公自动化系统增加一个请销假功能，以下说法正确的是()

- ☒ A 属于结构化动态性
- ☐ B 属于交互式动态性
- ☐ C 属于体系结构动态性
- ☐ D 属于软件体系结构的扩展

提交

对某个数据采集系统的配置信息进行重构，使其支持更多数据类型的解析；请问这属于哪种体系结构动态性()

- ☐ A 属于结构化动态性
- ☐ B 属于交互式动态性
- ☒ C 属于体系结构动态性

提交

❁ 动态软件体系结构的研究-模拟和描述体系结构动态更新

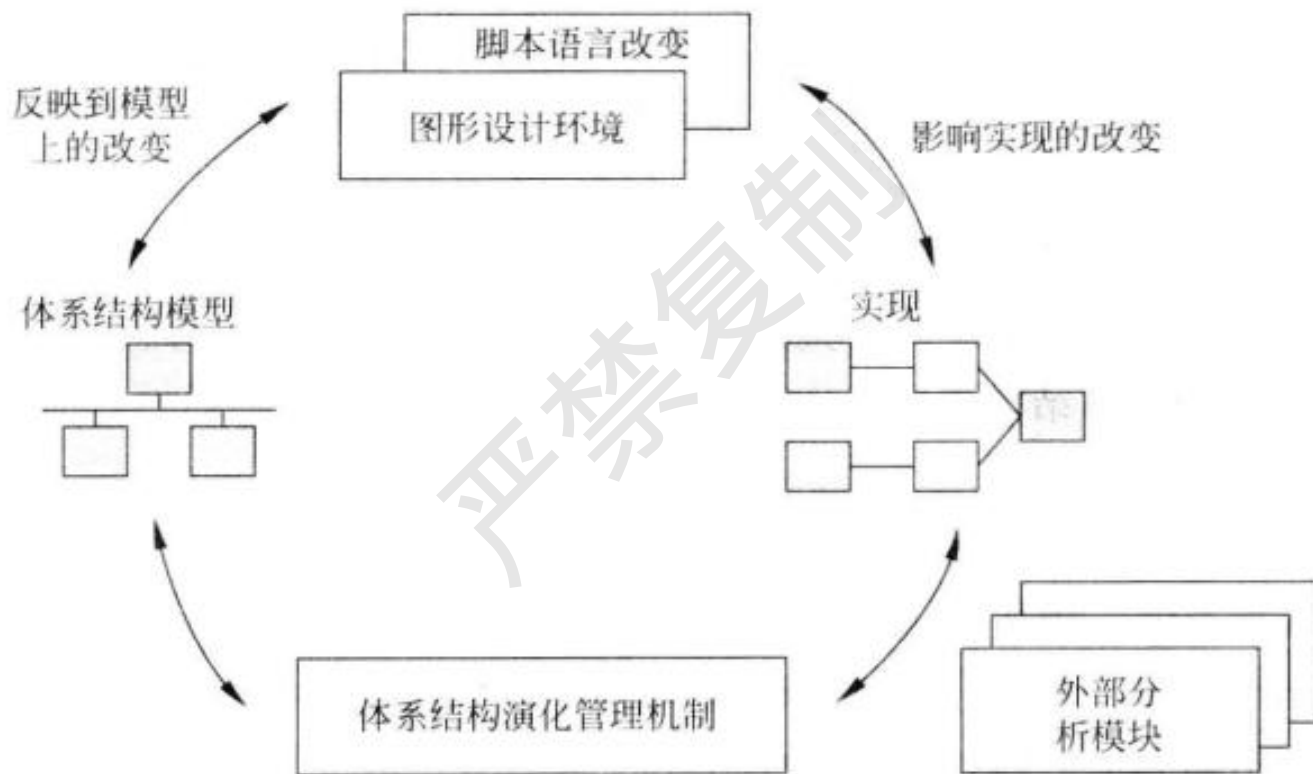
- ◆ ADL提供了一种形式化机制来描述软件体系结构，这种形式化机制主要通过提供语法和语义描述来模拟构件、连接件和配置。但是大多数ADL只描述系统的静态结构，不支持对体系结构的动态性描述。
- ◆ 近些年对支持软件体系结构动态性的ADL研究主要集中在对已有的一些ADL的扩展，以支持体系结构的动态性，主要描述语言有：C2，Darwin，Unicon，Wright等。

❁ 动态软件体系结构的研究-体系结构动态更新的执行

支持动态体系结构机制的工具：

- ◆ ArchStudio工具集：支持交互式图形化描述和C2风格描述的体系结构的动态修改。
- ◆ SAA (Software Architecture Assistant) 软件体系结构助理，是一种交互式图形工具，可以用来描述、分析、建立动态体系结构；不支持运行系统的监控和操作。

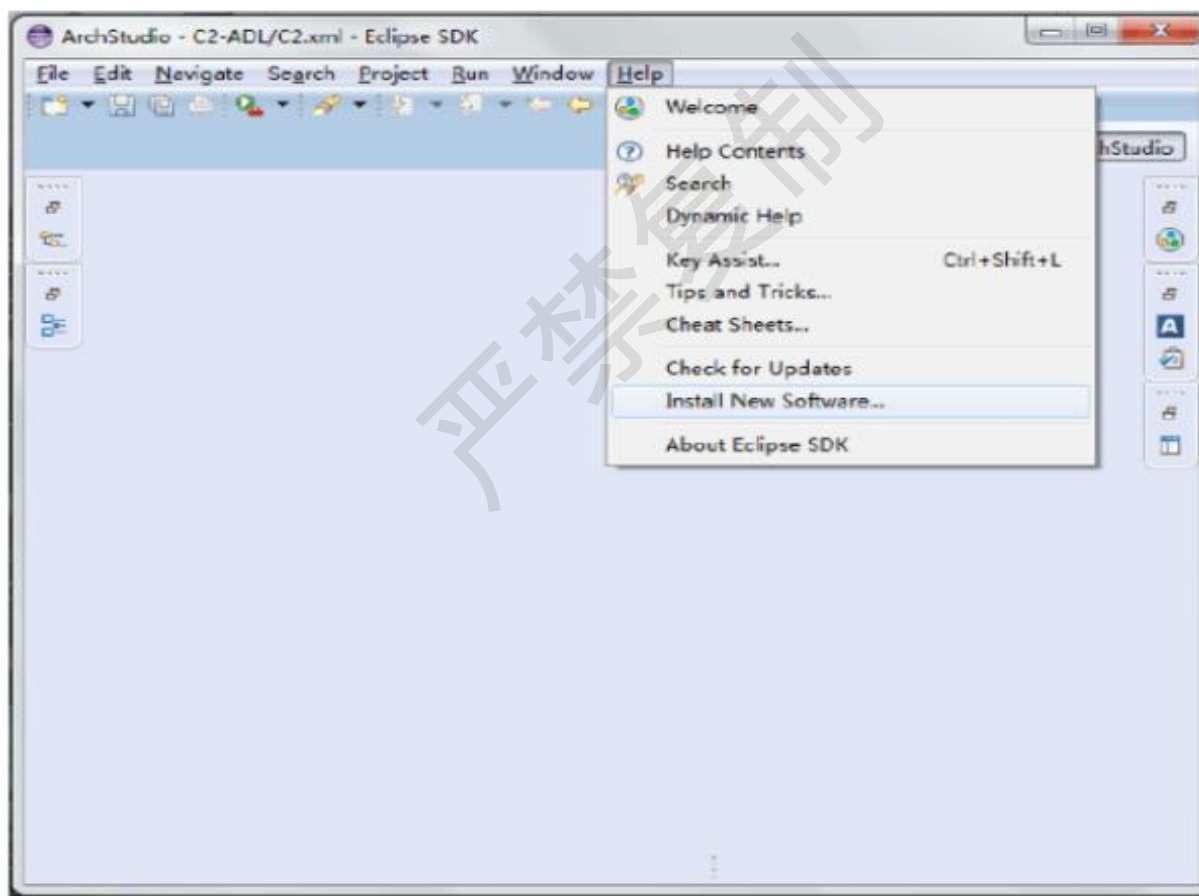
❁ 动态软件体系结构的研究-体系结构动态更新的执行



ArchStudio概念模型

❁ 动态软件体系结构的研究-ArchStudio应用(例)

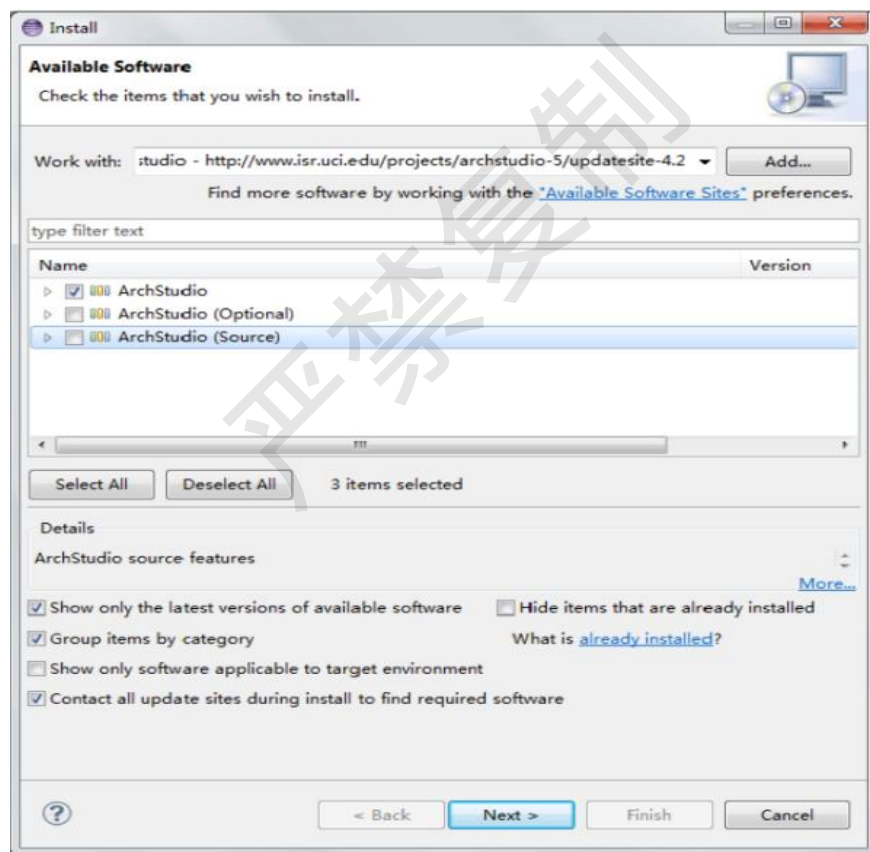
第一步:Eclipse安装archStudio插件



新的开始，新的起点，让我们一起为梦想而努力。

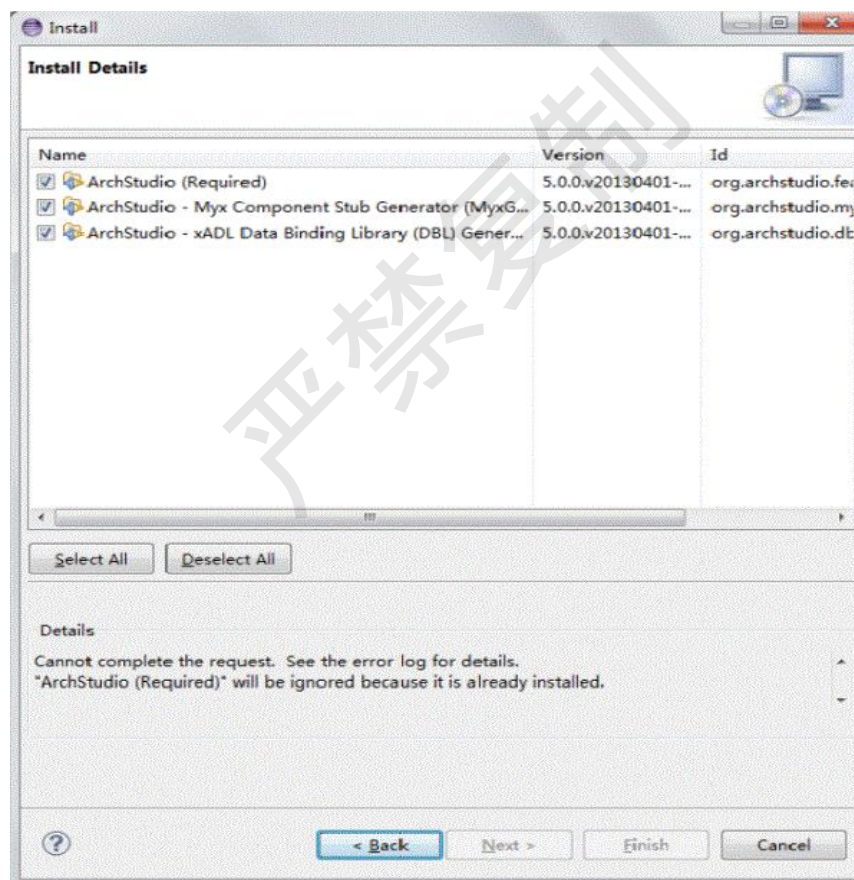
❁ 动态软件体系结构的研究-ArchStudio应用(例)

第一步: Eclipse安装archStudio插件



❁ 动态软件体系结构的研究-ArchStudio应用(例)

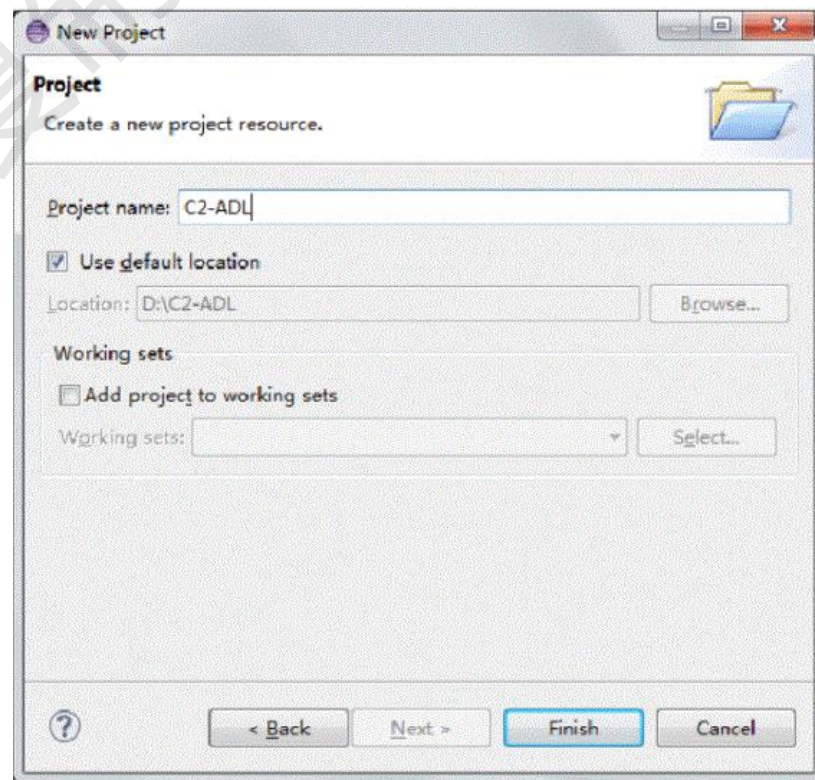
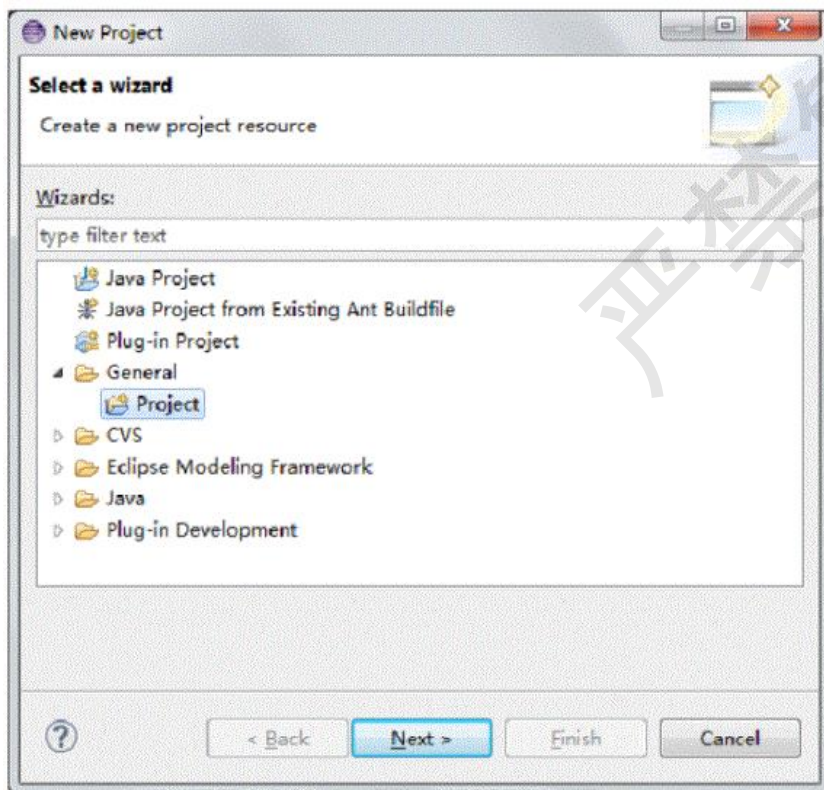
第一步: Eclipse安装archStudio插件



动态软件体系结构的研究-ArchStudio应用(例)

第二步:新建工程

(1) 在Navigatior中新建project。选中General的Project。点击Next输入Proect名字。点击Finish

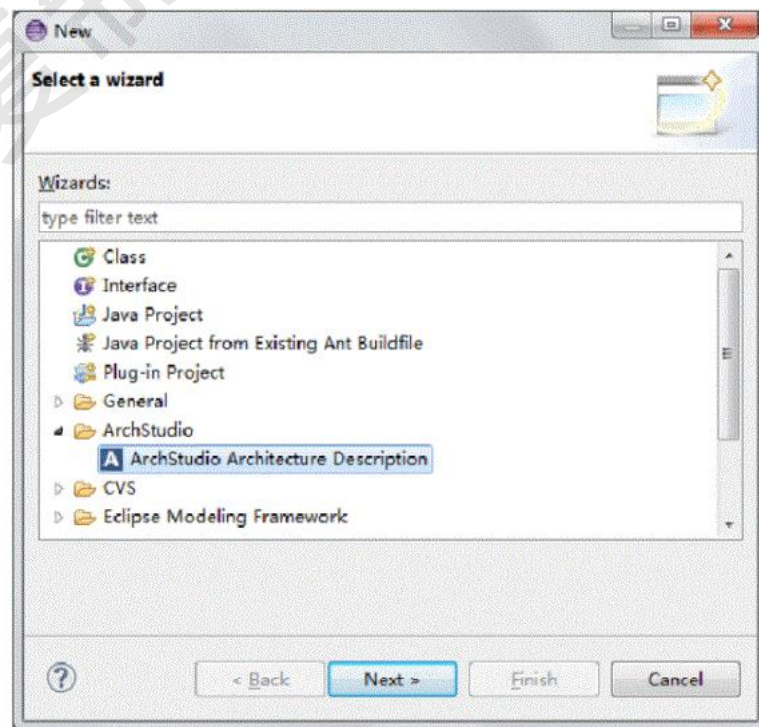
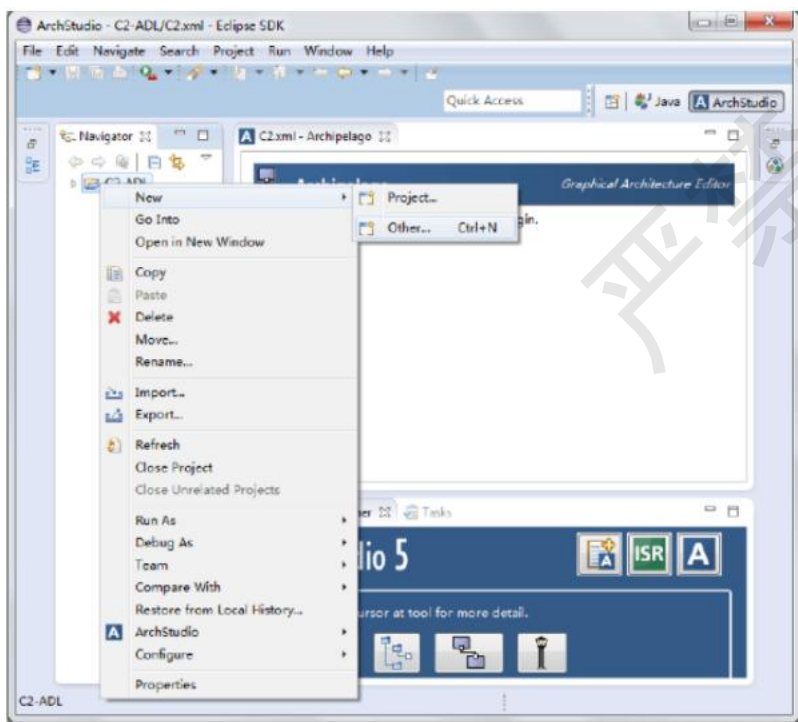


新的开始，新的起点，让我们共同为梦想而努力。

动态软件体系结构的研究-ArchStudio应用(例)

第二步:新建工程

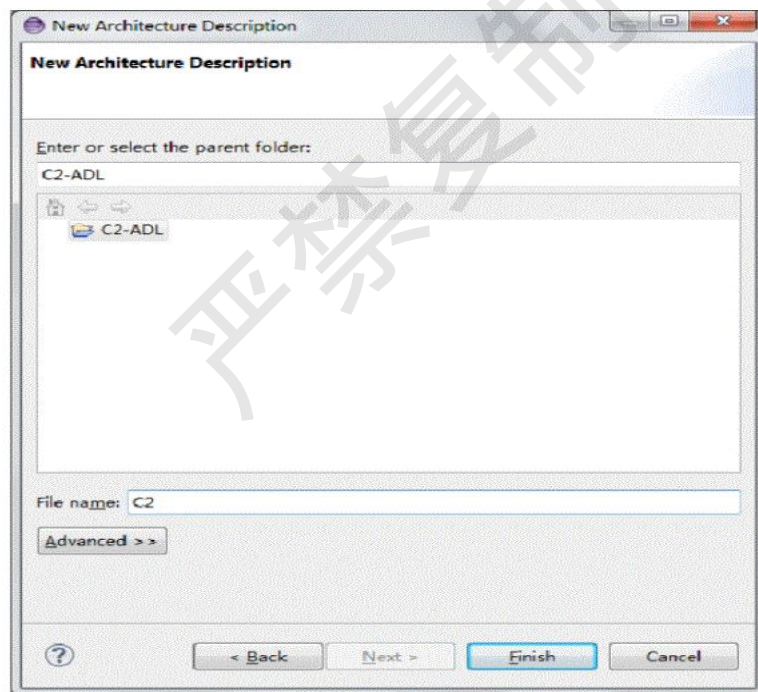
(2) 右键点击在 Navigator新建好的工程 New-->Project。选中ArchStudio的rchStudio Architecture Description。点Next



❁ 动态软件体系结构的研究-ArchStudio应用(例)

第二步:新建工程

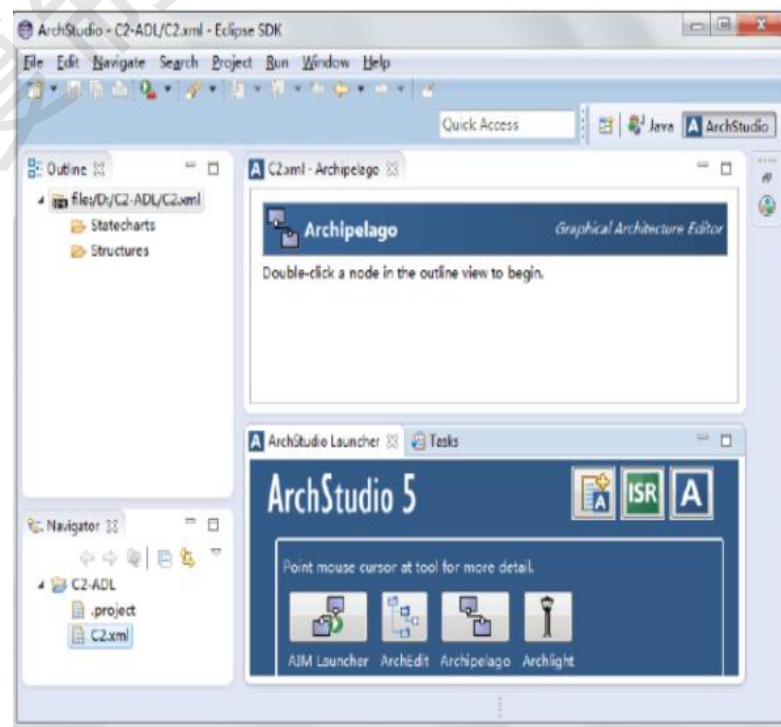
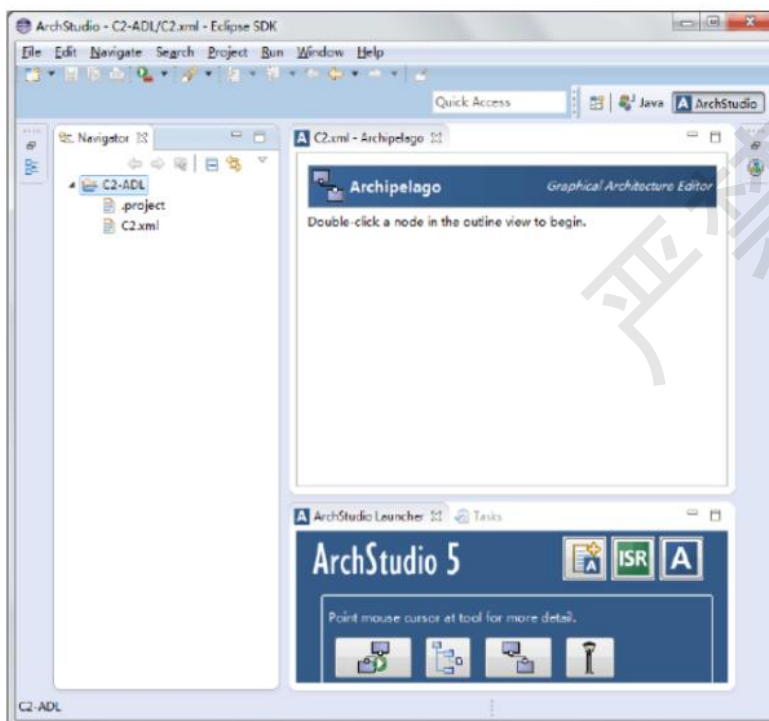
(3) 输入新创建的架构描述名称-点击finish



动态软件体系结构的研究-ArchStudio应用(例)

第二步:新建工程

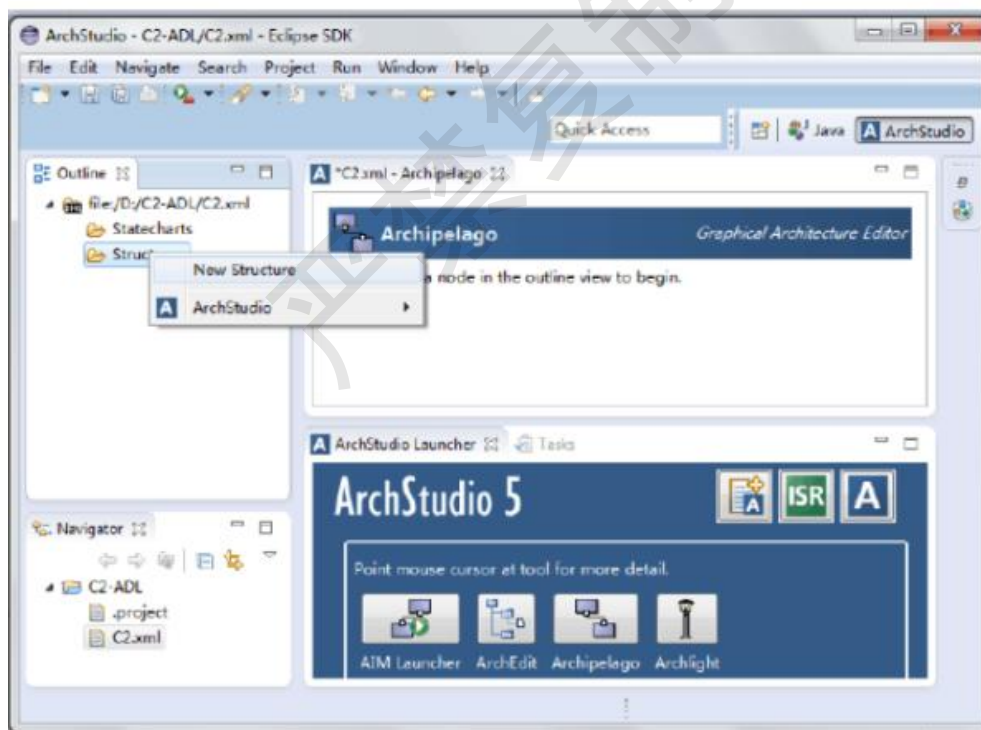
(4) 将C2.xml拖入ArchStudio Launcher窗口中的archipelago。则在outline显示新创建的C2.xml。右键点击 structures 点击 New Structure。



❁ 动态软件体系结构的研究-ArchStudio应用(例)

第二步:新建工程

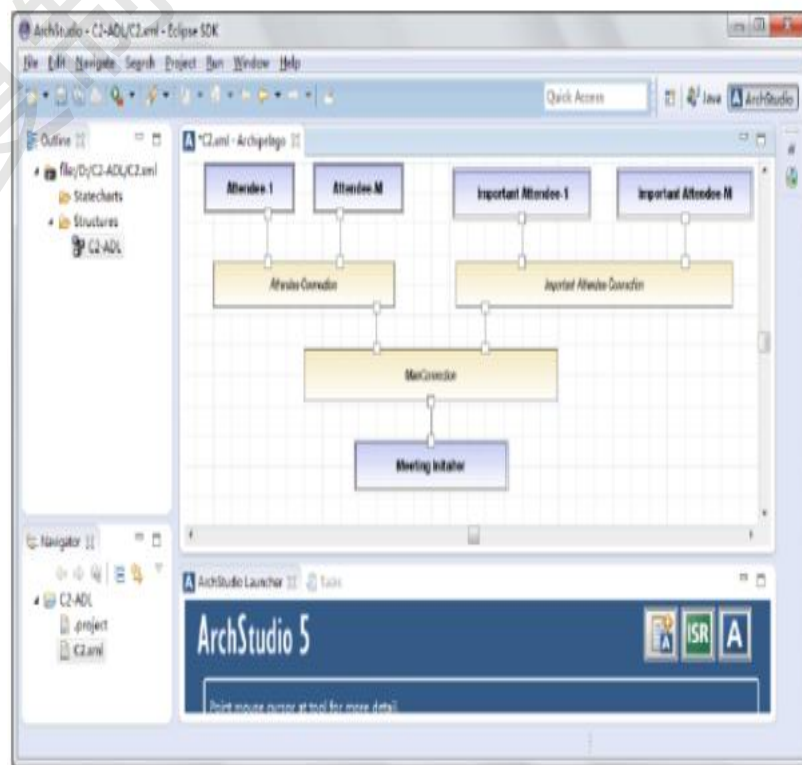
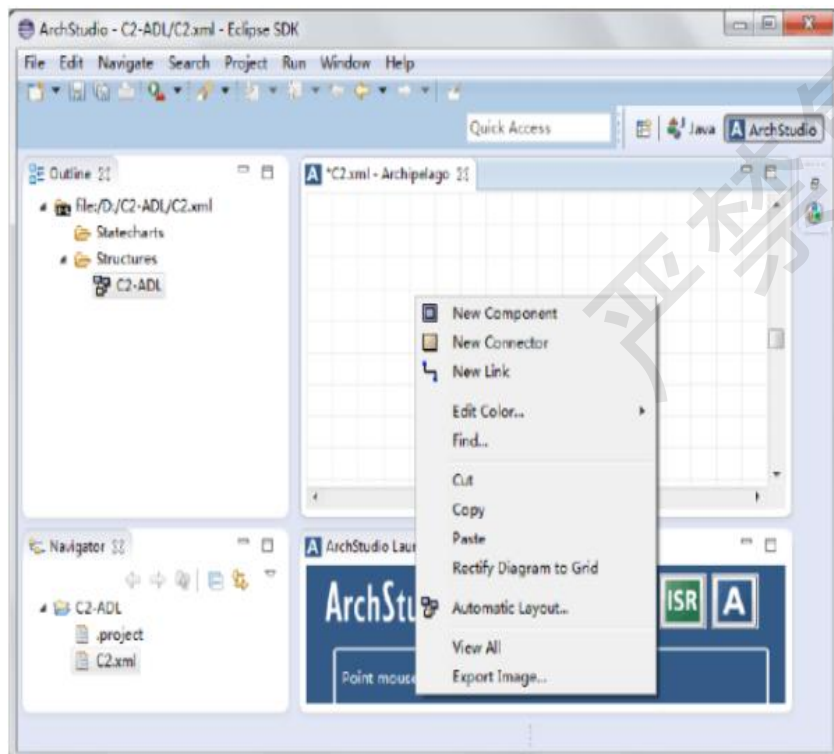
(4) 将C2.xml拖入ArchStudio Launcher窗口中的archipelago。则在outline显示新创建的C2.xml。右键点击 structures 点击 New Structure.



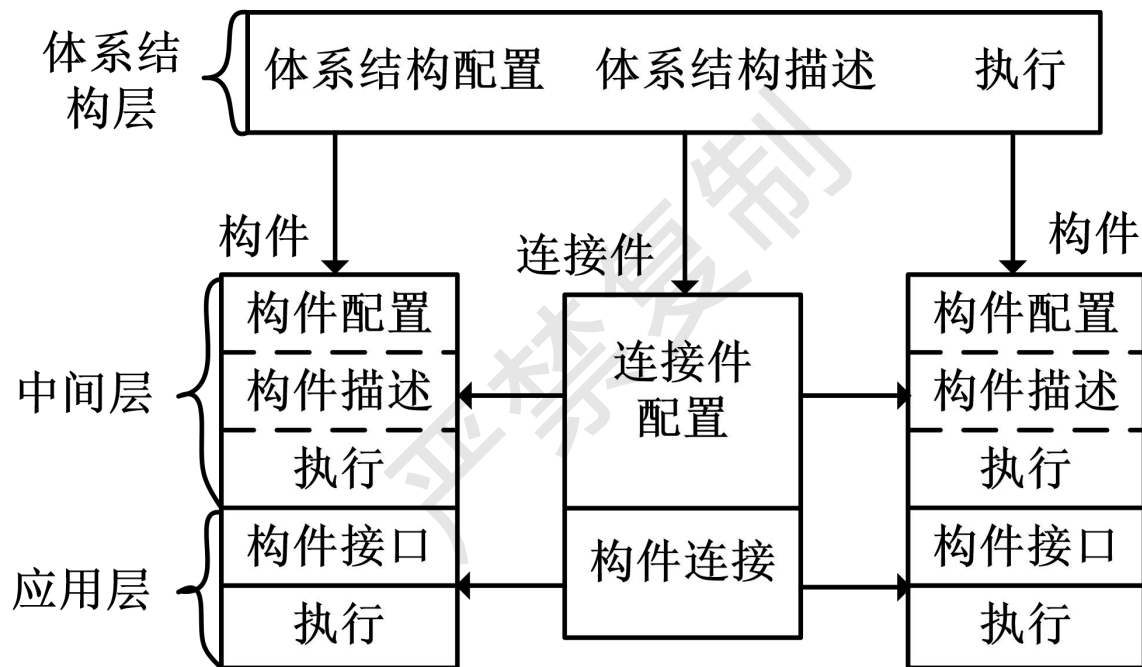
动态软件体系结构的研究-ArchStudio应用(例)

第二步:新建工程

(5) 在C2.xml-Archipelago窗口中--新建连接件、构件、对架构图进行设计。



基于构件的动态系统结构模型 – 概述



[CBDSAM]

✿ 基于构件的动态系统结构模型 – 层次分析

- 应用层:最底层, 包括构件连接, 构件接口和执行
 - 构件连接定义连接件如何与构件连接。构件接口定义构件所提供的服务
 - 本层次可添加、删除、更新构件。
- 中间层:包括连接件配置、构件配置、构件描述和执行
 - 连接件配置管理连接件及其接口的通信配置; 构件配置管理构件行为; 构件描述对构件内部接口、行为、功能和版本等信息加以描述。
 - 本层可添加版本控制机制和不同构件装载的方法

❁ 基于构件的动态系统结构模型 – 层次分析

- 体系结构层:最上层, 控制和管理整个体系结构, 包括体系结构的配置, 描述和执行。
 - 体系结构描述主要描述构件及相联系连接件的数据; 体系结构的配置控制整个分布式系统的执行, 并管理配置层; 体系结构执行对体系结构的行为进行描述。
 - 本层具有更新和扩展更新机制, 更改系统的拓扑结构, 以及更新构件到处理元素之间的映射。

❁ 基于构件的动态系统结构模型 – 更新请求描述

- ◇ 更新类型:包括添加、删除或修改构件。
- ◇ 更新对象列表: 需要更新对象类的ID号
- ◇ 对象的新版本说明: 对象的新版本执行情况
- ◇ 对象更新方法: 更替、动态、静态
- ◇ 更新函数:用来更新一个执行对象进程的状态转换函数。
- ◇ 更新限制: 描述更新及他们之间的约束关系。

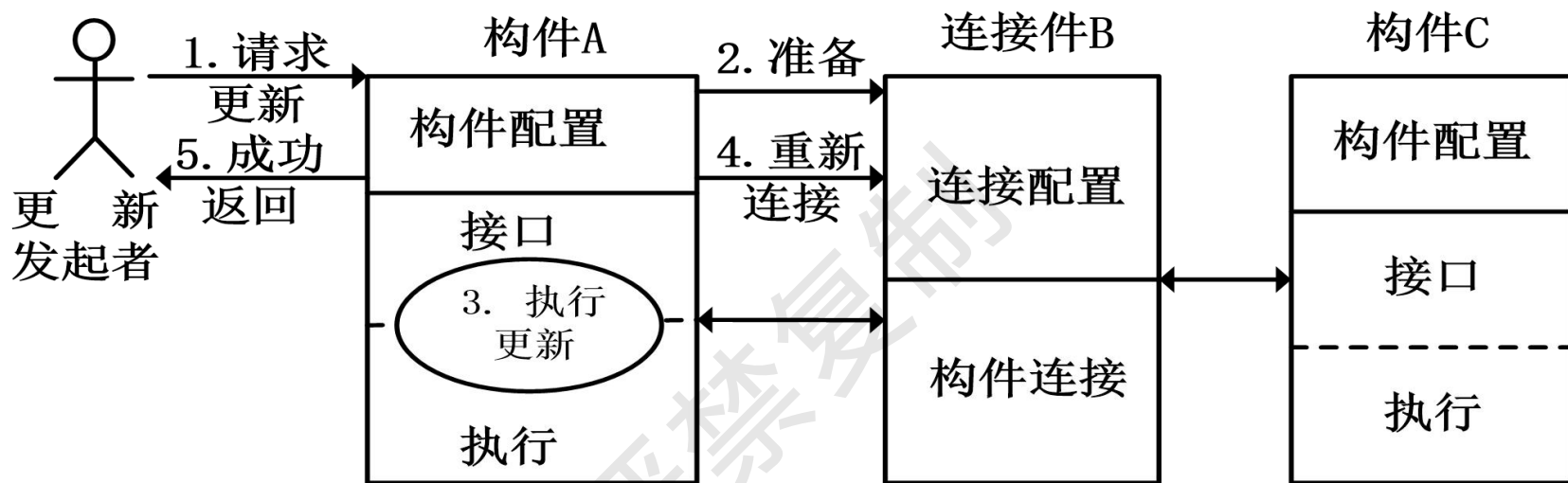
❁ 基于构件的动态系统结构模型 – 更新描述实例

```
<update_descriptor>
  <add_obj to="server01//comp1">
    <object name="C">
      <implementation>...</>
    </object>
  </add>
  <remove_obj from="server01//comp1">
    <object name="D">
      </object>
    </remove>
  <update_obj in="server01//comp1">
    <object name="A" method="replace">
      <old_version>1.0</>
      <new_version>1.1</>
      <implementation>...</>
    </object>
  </update>
```

❁ 基于构件的动态系统结构模型 – 更新执行步骤

- ◇ 检测更新的范围:全局更新还是局部更新
- ◇ 更新准备工作: 应用层需要等待参与进程发出可安全更新的信号; 配置层需要等待连接件及其它构件已完成更新。
- ◇ 执行更新: 告知发起者更新的结果
- ◇ 存储更新: 将所作的更新存储到构件或体系结构描述中

❁ 基于构件的动态系统结构模型 – 实例分析 – 局部更新



第一步,更新发起者发出一个更新请求,这个请求被送到构件 A 的配置器中,构件配置器将分析更新的类型,从而判断它是对象的局部更新。

第二步,由于更新为局部更新,构件 A 的配置器发出一个信号给连接件以隔离构件 A 的通信,准备执行更新。

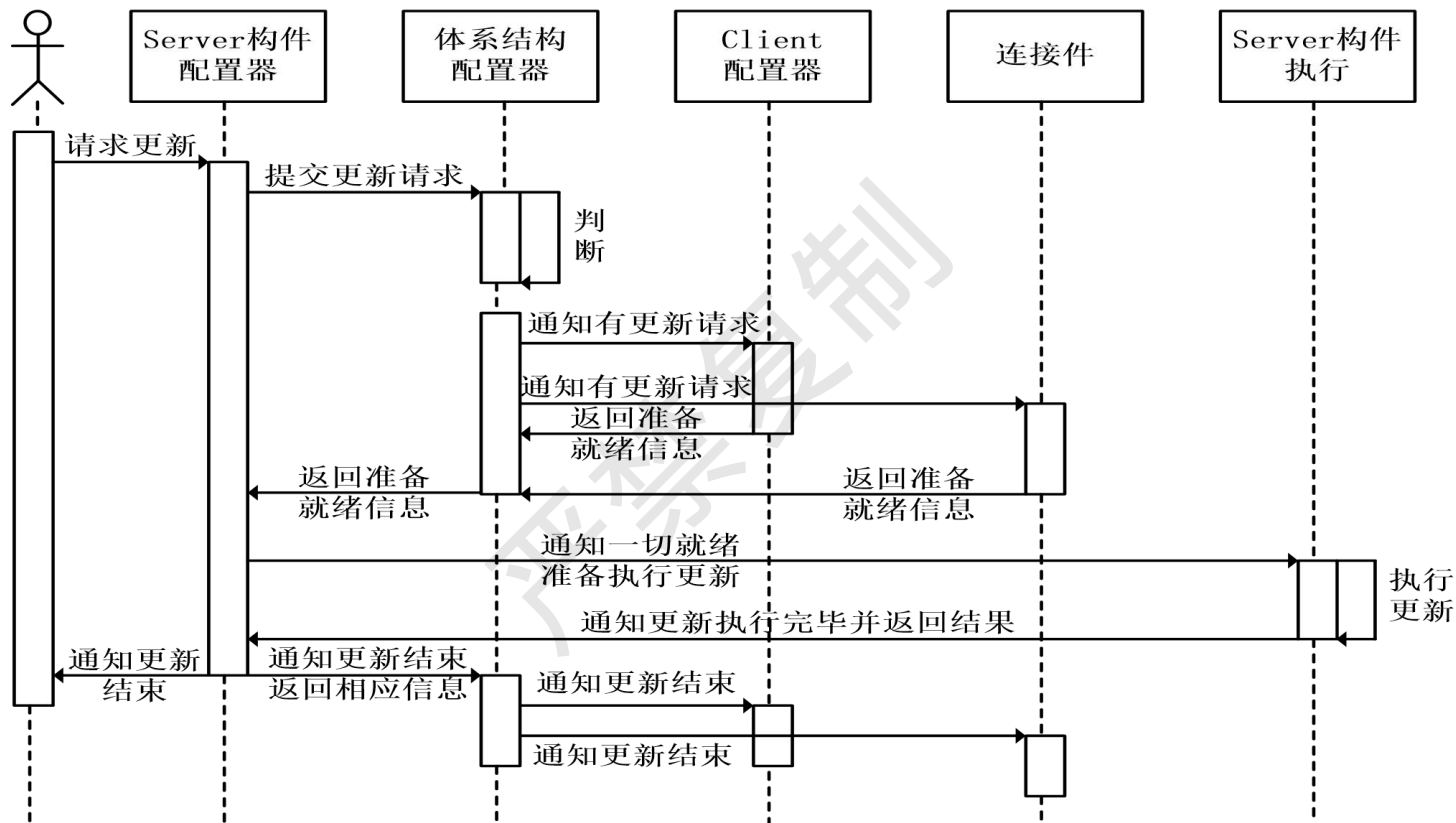
第三步,构件 A 的配置器开始执行更新。

第四步,更新执行完毕后,构件 A 的构件描述被更新,并且构件 A 发送一个消息给连接件 B,两者间的连接被重新存储起来。

第五步,将更新结果返回给更新发起者。

由上述分析可知,在整个更新过程中,构件 C 都没有受到影响,这说明按照 CBDSAM 的方法,不会影响系统的其他部分运行。

基于构件的动态系统结构模型 – 实例分析 – 全局更新



基于构件的动态系统结构模型 – 实例分析 – 全局更新

第一步,Server 构件配置器接收到更新发起者提出的更新请求后,向体系结构配置器提出更新请求。

第二步,体系结构配置器对更新请求的类型进行分析,判断是否在更新请求限制(属于全局更新还是局部更新)范围内,不在更新范围内的更新不予执行;如果在更新限制范围内,体系结构配置器对更新所涉及的连接件和构件(本例中为 Client 构件和连接件)发出消息,要求它们做好更新准备工作。

第三步,准备工作完成后,Client 构件配置器和连接件向体系结构配置器返回就绪信息。

第四步,一切准备就绪后,体系结构配置器通知 Server 构件进行更新。

第五步,更新执行完毕后,向 Server 构件配置器、体系结构配置器和更新发起者通知更新执行完毕并返回更新结果;同时,体系结构配置器通知 Client 构件和连接件更新结束,可继续正常运行。

这样,在没有影响系统运行的情况下,按照更新发起者的要求对系统进行了更新,并维护了系统的一致性。

基于构件的动态体系结构模型更新步骤如下[(1)更新准备工作 (2)检测更新的范围 (3)存储更新 (4)执行更新]，以下排序正确的是 ()

- ☐ A 1,2,3,4
- ☐ B 2,1,3,4
- ☒ C 2,1,4,3
- ☐ D 1,2,4,3

提交

判断题：静态体系结构在系统运行时不能发生改变，缺乏表示动态更新机制，不能运用到一些需要长期运行的重要系统中()

- ☒ A 真
- ☐ B 假

提交

判断题：动态体系结构主要研究由于特殊需要必须在连续运行情况下体系结构变化与支持的平台()

☒ A 对

☐ B 错

提交