文章题目：软件组件中的数据封装

摘要：数据封装是面向对象编程中的常见属性。数据封装不仅有助于在现实世界中对事物进行建模，而且它也通过创建同一个类的多个实例来促进重用 ，每个实例都有自己的标识和私有数据。

对于CBSE（Component-based software engineering，基于组件的软件工程）来说，这种重用显然是关键的需求之一。

但是它必须和组件结合一起实现，这是CBSE的核心。

在本文中，我们通过扩展我们之前定义的组件模型，展示了数据封装如何与组件结合。

1.介绍

在面向对象编程中数据封装是对象的常见属性，它不仅可以用于对现实世界中的事物进行建模，而且还可以通过创建同一个类的多个实例来促进重用，每个实例都有自己的标识和私有数据。

对于CBSE，这种重用显示也是关键需求之一，因为组件被认为是多个组件实例的可重用模板。

但是，因为CBSE的核心是组合，所以问题是如何设计组合机制或者运算符，使组合的每个级别都可以进行数据封装，也就是说，如何确保组合创建的每个组合组件都封装自己的数据。

在本文中，我们认为数据封装和组合的结合在当前的组件模型中是不可能的；然后表明它可以用过扩展我们之前定义的组件模型来实现。

当前的组件模型可以在很大程度上分为两类：（1）组件是对象的模型，比如在面向对象编程中；（2）组件是架构单元的模型，比如在软件架构中。

这两类的示例EJB（Enterprise JavaBeans）和ADL（architecture description languages）。

在组件是对象的模型中，组件通过方法调用进行组装。然而这不是（代数）组合，因为通过调用对象2的方法与对象2组装的对象将导致两个对象，而不是一个（复合）对象。

因此，即使分别在对象1和对象2中进行数据封装，也没有可以正确组成对象1和对象2的组合机制，更不用说保留数据封装了。

在组件是架构单元的组件模式中，端口连接提供组合机制，并且可以提供组合。但是，数据封装并不总是可能被定义。事实上，数据在架构单元中的作用非常不明显。

这些单元既可以表示计算和数据，也可以只表示数据，并且通常数据封装不被视为组合的一部分。在架构单元具有数据端口的地方，可以认为这些端口代表数据封装。

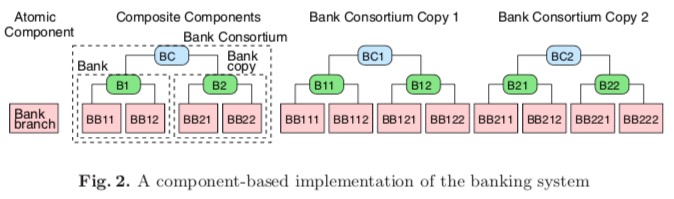
然而，即使在这种情况下，也无法确定数据封装在每个组件级别上是否可行。

在本文中，我们描述了一种组合方法，它允许在组合的每个级别进行数据封装。我们的方法基于一个组件模型，其中组合运算符是第一级的组件（first-class citizens），并且他们还使每个组件实例

（特别是复合组件实例）能够封装其自己的数据。

2.具有数据封装的组合

组件用于组合，因此它们应该是组合的，即组合物。即如果C1和C2是组件，则C1和C2的组合成的C3也必须是组件



3.我们的组建模型

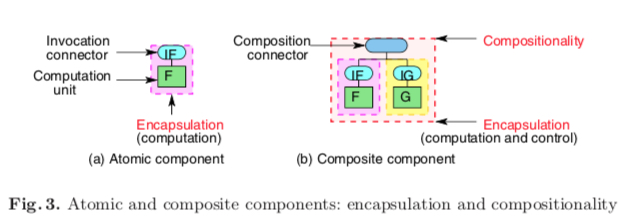
在我们讨论如何扩展组建模型以启用数据封装之前，在本节中我们将简要概述我们在[6]中提出的模型。

在我们的模型中，我们有两种基本实体：（i）计算单元，和（ii）连接器。计算单元U封装计算，提供了一组方法（或服务）。封装意味着U的方法不会调用其他计算单元中的方法，同时当被调用时，它们的所有计算都发生在U中。因此，U可以被认为是一个不调用其他类中的方法的类。

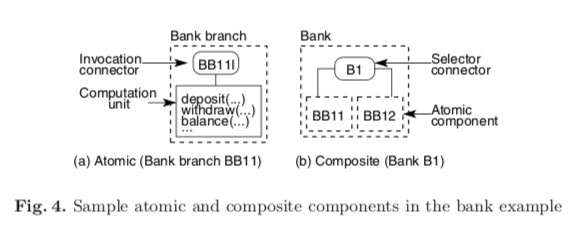
连接器有两种：（i）调用，和（ii）组合。调用连接器连接到计算单元U，以便提供对U的方法的访问。

组合连接器封装控件。它用于定义和协调一组组件（原子或复合）的控制。例如，组成组件C1,…,Cn的顺序控制程序连接器可以按同样的顺序调用C1,…,Cn的方法，另一个例子是选择器连接器，它选择（根据某些指定的条件）它组成的一个组件，并调用它的方法。

组件根据计算单元和连接器定义。有两种组成成分：（i）原子,和（ii）复合（见图3）。原子组件具有调用连接器的计算单元组成，该调用连接器提供组件的接口。复合组件是一组由组合连接器组成的组件（原子或复合组件）组成的。合成连接器提供复合的接口。



举例说明，在示例一的银行系统（见图2）中，原子组件银行分支BB11，可以如图4（a）所示定义，具有调用连接器IBB11，以及具有该分支连接器的计算单元，有存款，取款，还贷等方法。复合组件银行B1，如图4（b）所示定义，作为使用选择器连接器的原子部件BB11和BB12的组成（为方便起见，也可以）。图2中的银行联合（BC）组成部分也可以使用选择器连接器（来自银行）组成，因为联盟必须选择具有客户账户所属分支的银行。



在我们的模型中，调用和组合连接器形成层次结构[8]。这意味着组合以分层方式完成。此外，每种组合物都保留了封装。这种组合性是我们组件模型的显著特征。一个原子组件封装了计算（图3（a）），即计算过程被他的计算单元封装。复合组件封装了计算和控制（图3（b）），它封装的计算是封装在其子组件中的；它封装的控件是由其组件连接器封装的。在复合中，子组件中的封装被保留。实际上，连接器的分层特性意味着组合组件与其子组件自相似；这个属性为分层组合提供了基础。

在下一节中，我们将展示如何扩展我们的模型以包含数据封装。

4．数据封装