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