

MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy

讲师：高焕堂（台湾）

<http://www.microoh.com>

F08_b

观摩：WebView的 PhoneGap小框架设计(b)

By 高煥堂

2、从WebView说起

WebView

- Android的View类别体系里，定义了各式各样的子类别，包括TextView、Button、SurfaceView和WebView等。
- 其中的WebView是给浏览器(如WebKit)用来呈现网页画面的。而画面布局(Layout)是由HTML(含HTML5)来规范的。

```
classDiagram
    class WebKit
    class WebView {
        +addJavascriptInterface()
        +loadURL()
    }
    WebKit --|> WebView
```

WebKit

WebView

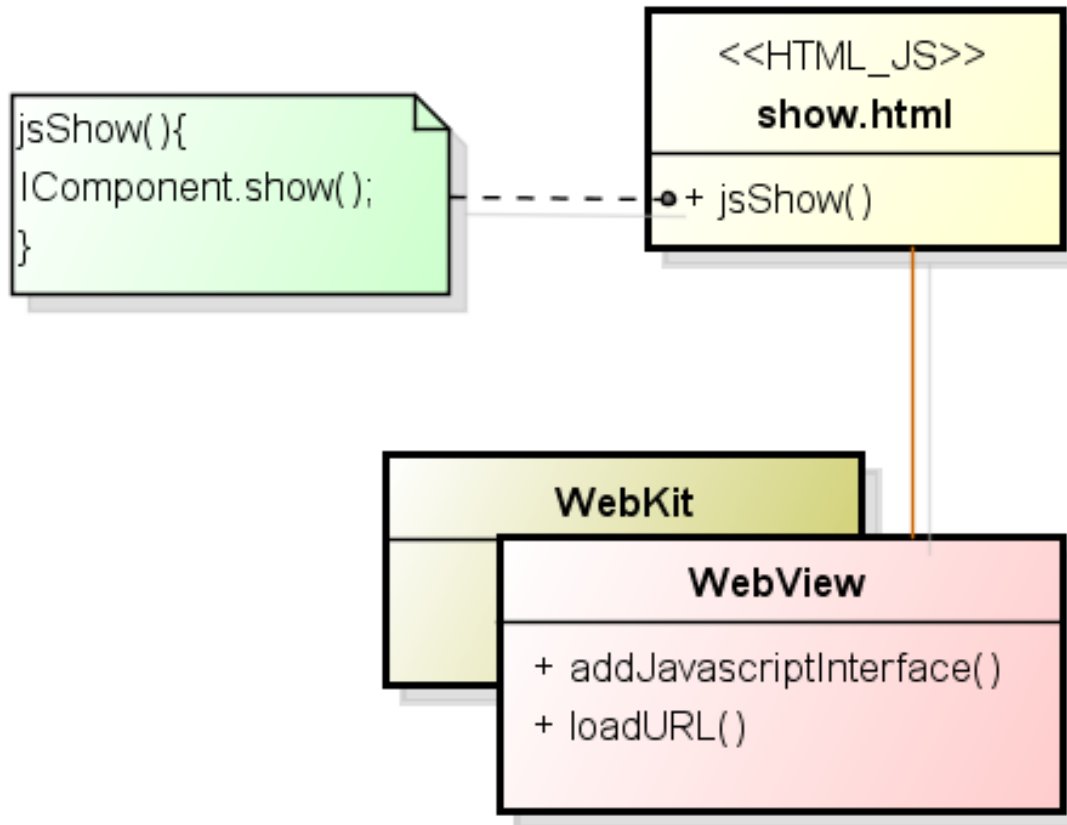
+ addJavascriptInterface()
+ loadURL()

先看大家熟悉的代码

例如，一般的语言表达Android的WebView的一段JavaScript语言：

// show.html (JavaScript代碼)

```
<input type="button" value="Say hello"
      onClick="jsShow('Hello Android!')" />
<script type="text/javascript">
    function jsShow(str) {
        IComponent.show(str);
    }
</script>
```



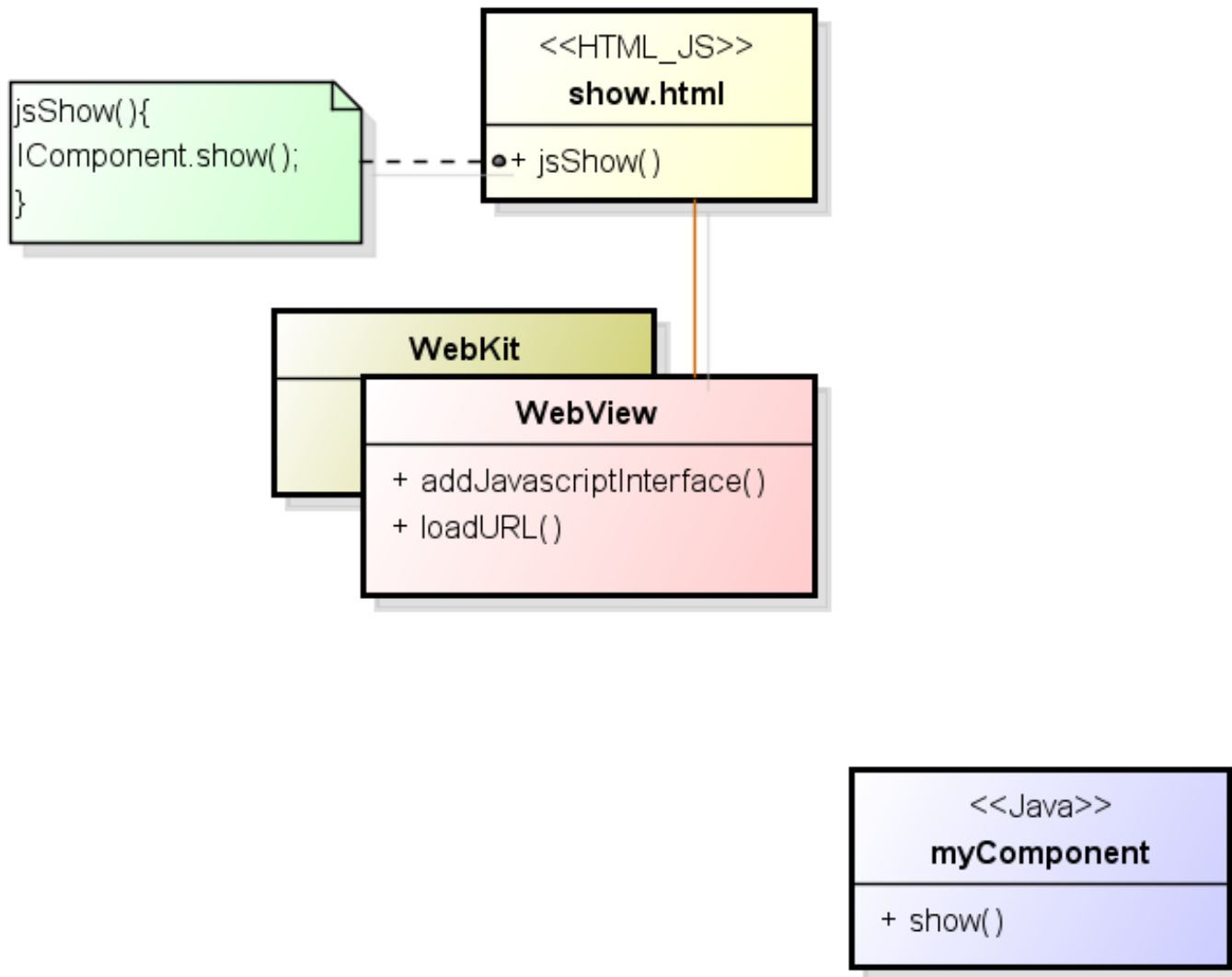
- 当WebView(含幕后的Webkit)执行到jsShow()的指令：

`IComponent.show(str);`

- 就会去调用Java插件(类)里的函数代码。

// myComponent.java

```
public class myComponent{  
    Context mContext;  
    myComponent(Context c) {  
        mContext = c;  
    }  
    public void show(String str) {  
        Toast.makeText(mContext, str,  
                        Toast.LENGTH_SHORT).show();  
    }  
}
```



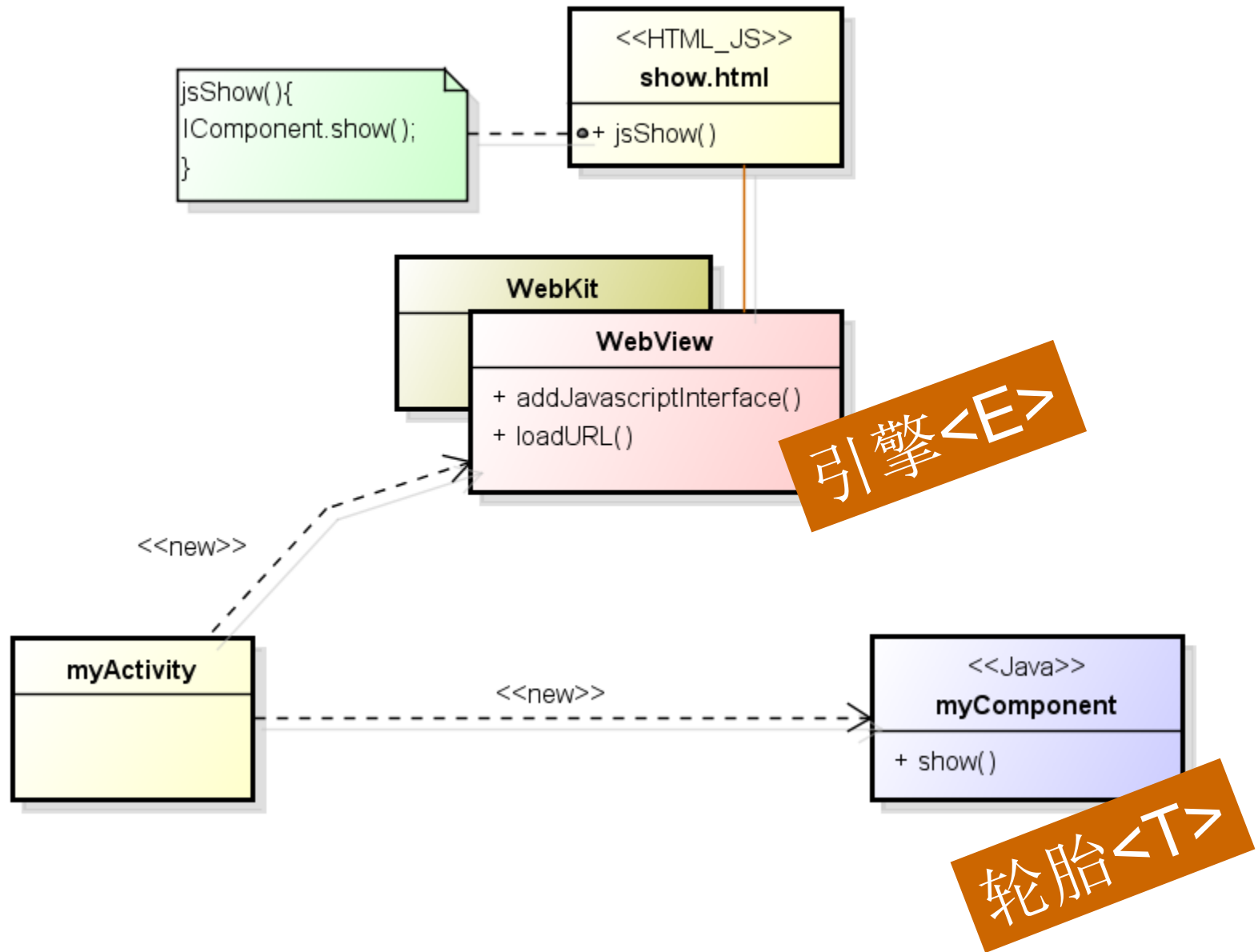
- 于是，WebView执行上述JS代码时，就调用myComponent里的show()函数了。
- 那么，这两段代码之间，又由谁来建立其连结关系呢？那就依赖另一段代码了，就是：

// myActivity.java

```
public class myActivity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        WebView appView = new WebView(this);  
        appView.loadUrl( "file:///android_asset/www/index.html" );  
        appView.getSettings().setJavaScriptEnabled(true);\br/>        myComponent obj = new myComponent(this);  
        appView.addJavascriptInterface(obj, "IComponent" );  
        appView.loadUrl( "file:///android_asset/show.html" );  
        setContentView(appView);  
    }  
}
```

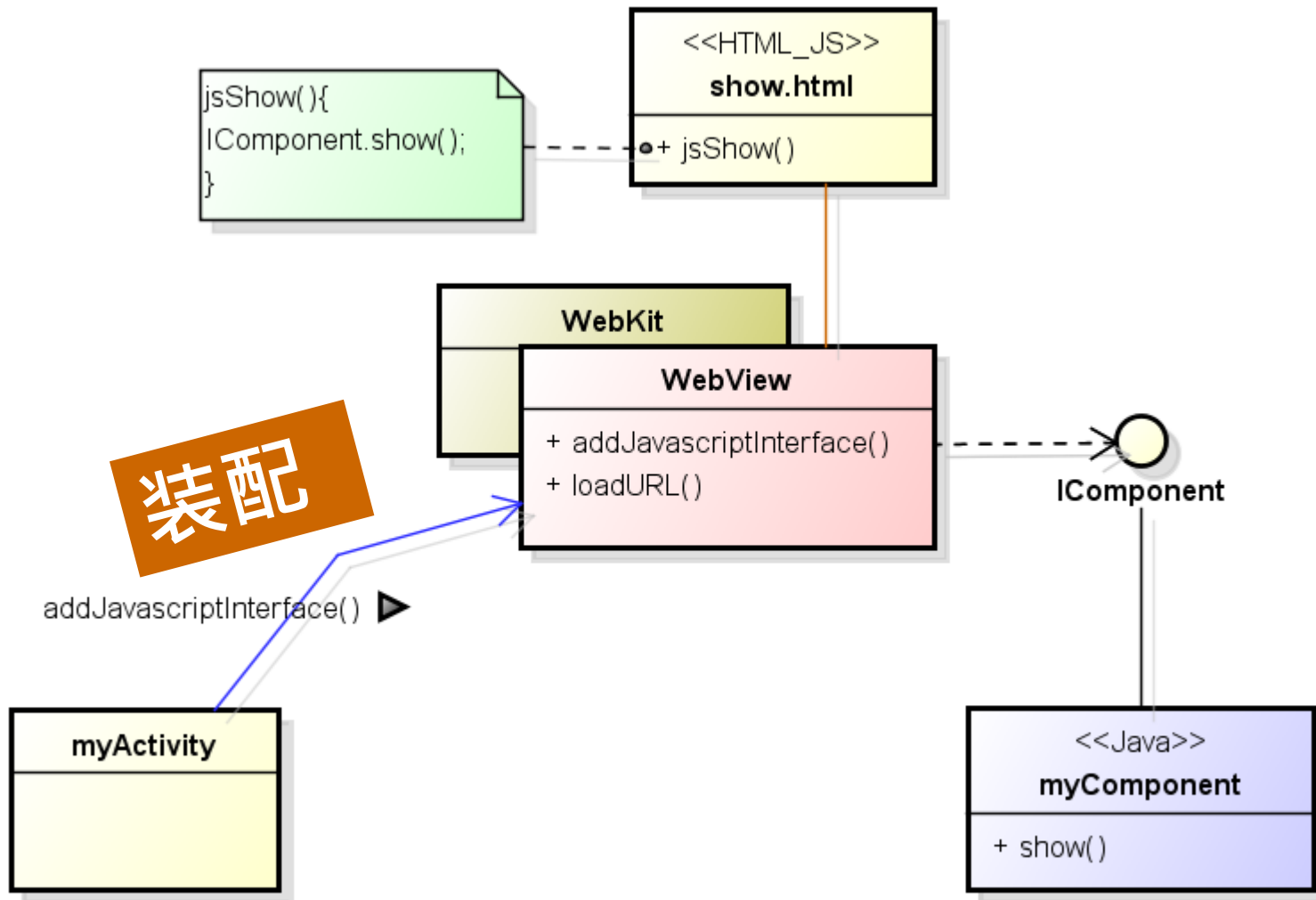
```
WebView appView = new WebView(this);  
// .....  
myComponent obj = new myComponent(this);  
// .....
```

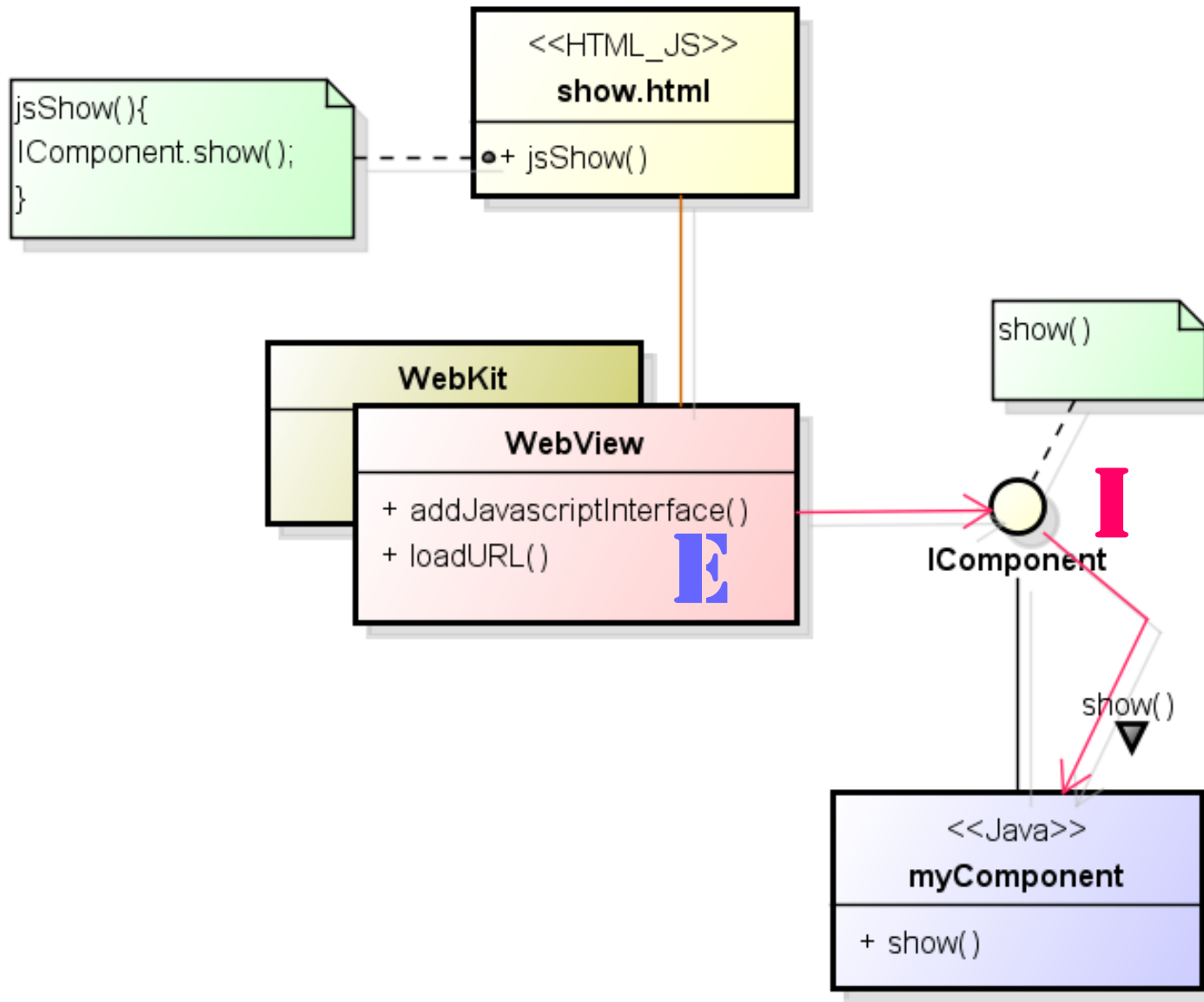
创建了引擎<E>和轮胎<T>。



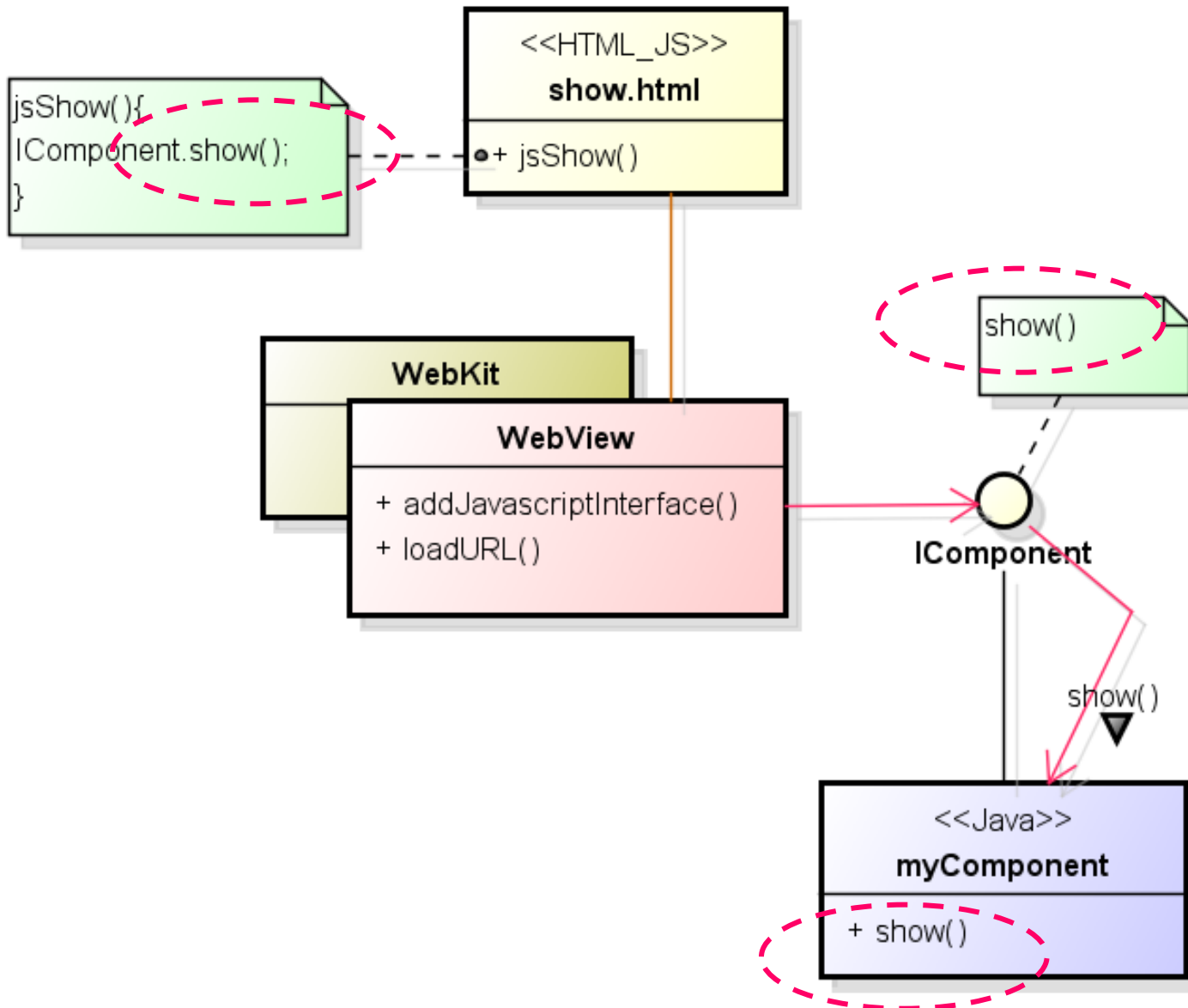
```
// .....  
appView.addJavascriptInterface(obj, "IComponent" );  
// .....
```

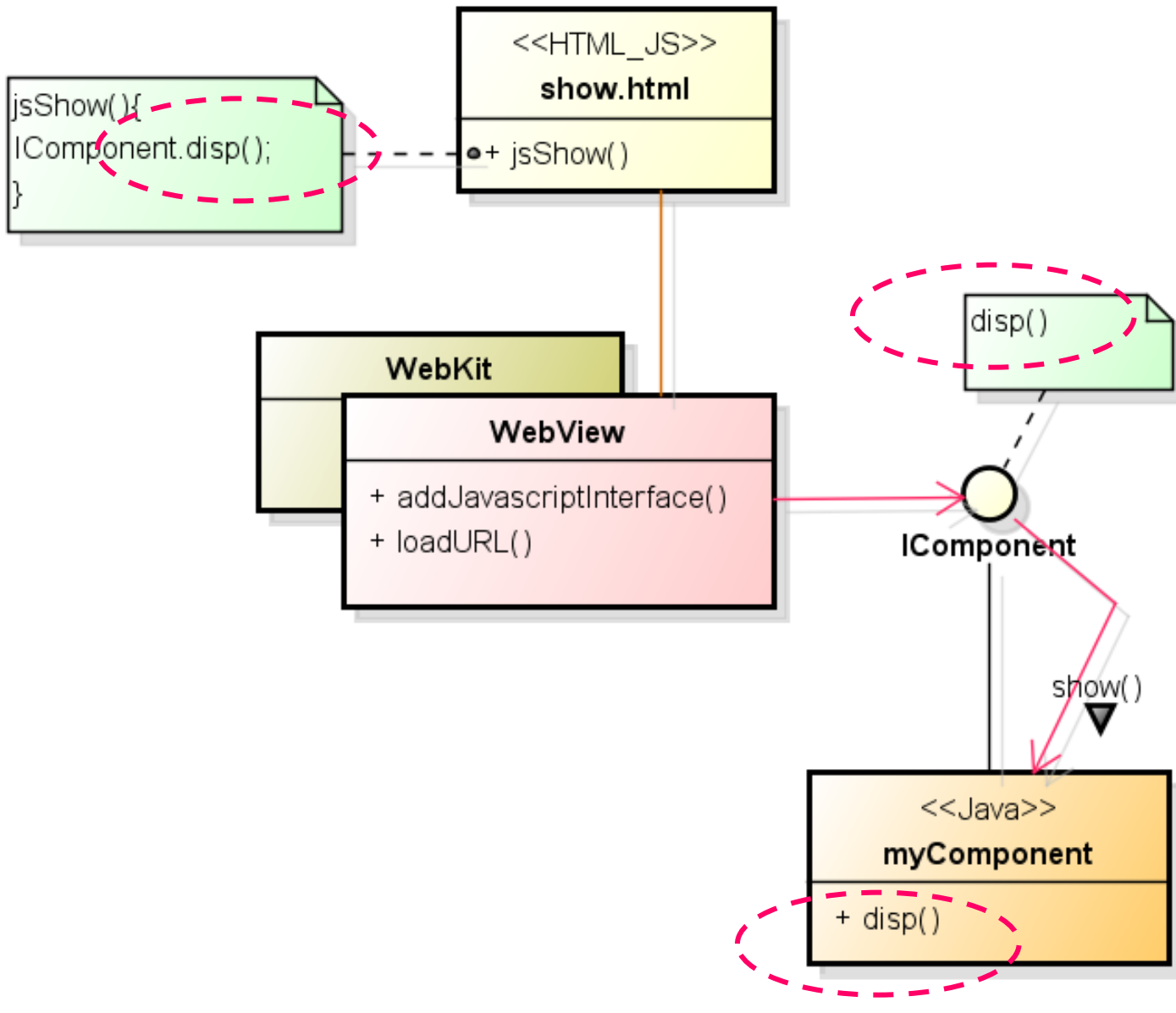
将轮胎<T>装配到引擎<E>





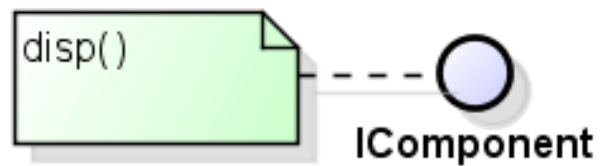
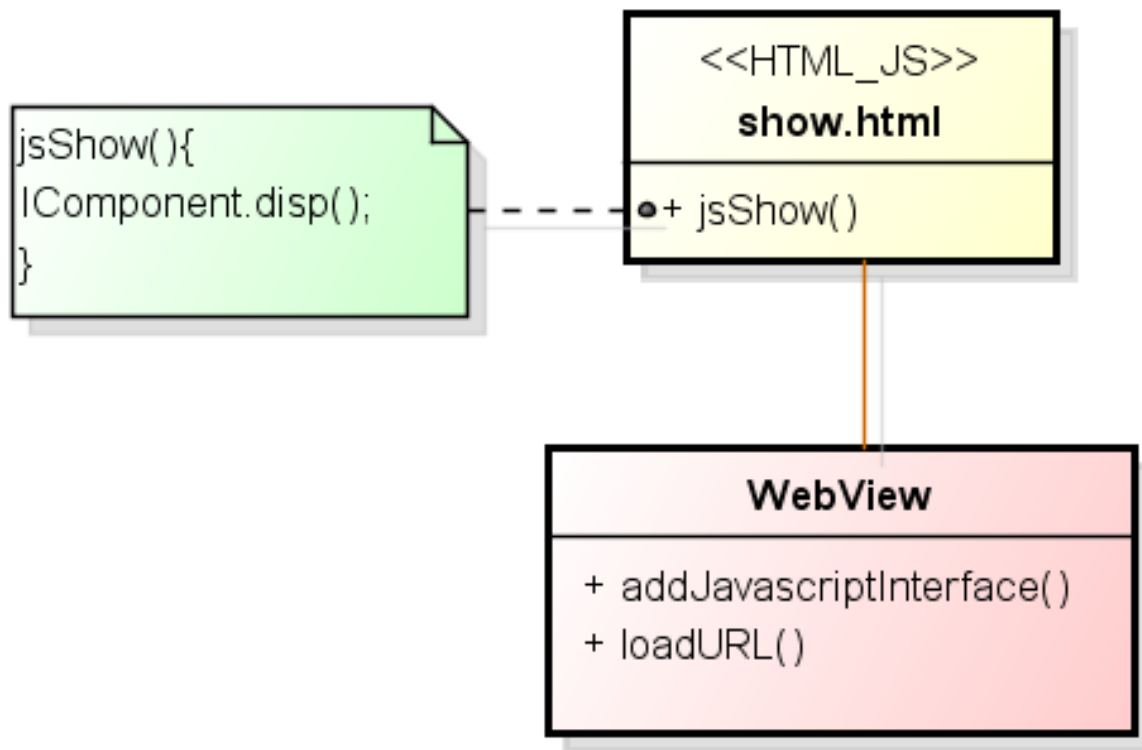
- IComponent是特殊性接口，其变动成本很高；因为牵一发动全身。





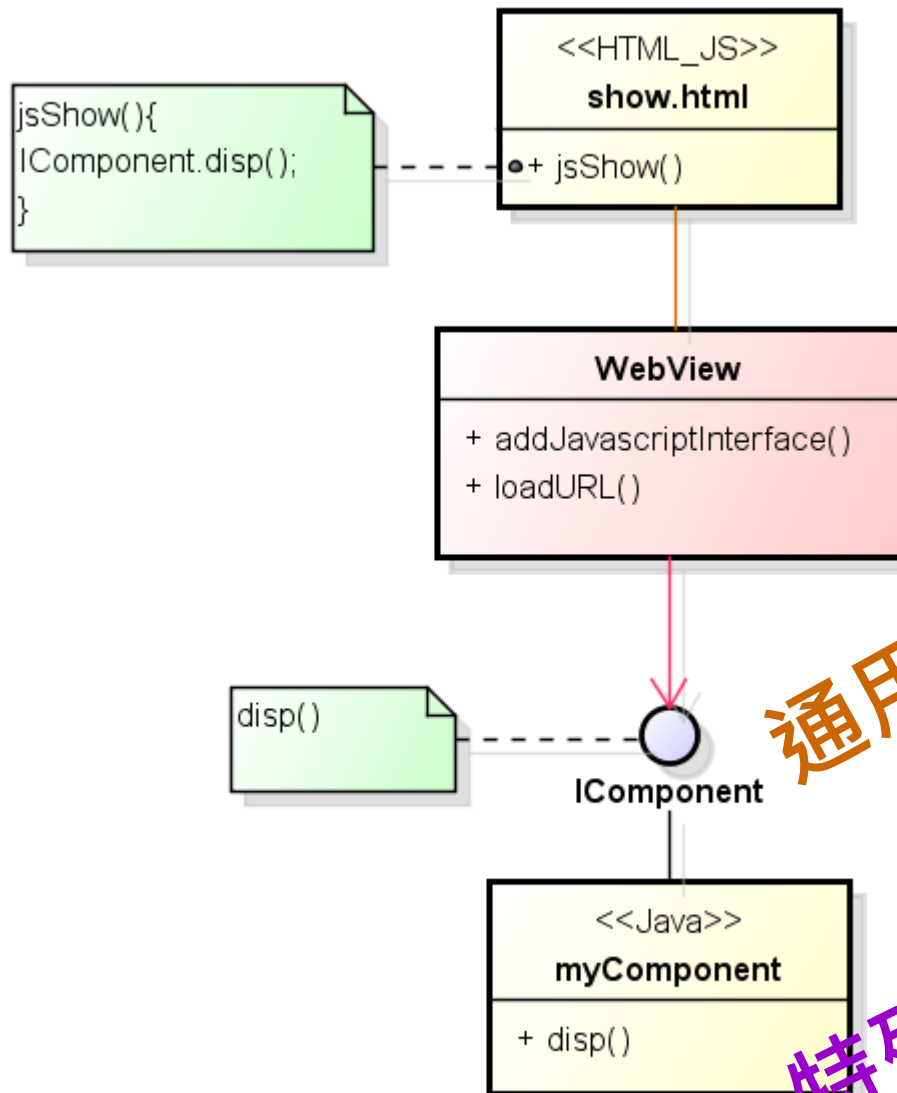
设计通用性接口

- 以上的IComponent接口属于特殊性接口。你知道为什么它是一个特殊性接口吗？
- 如果将IComponent接口设计成为通用性接口，应该由谁来定义这项接口呢？



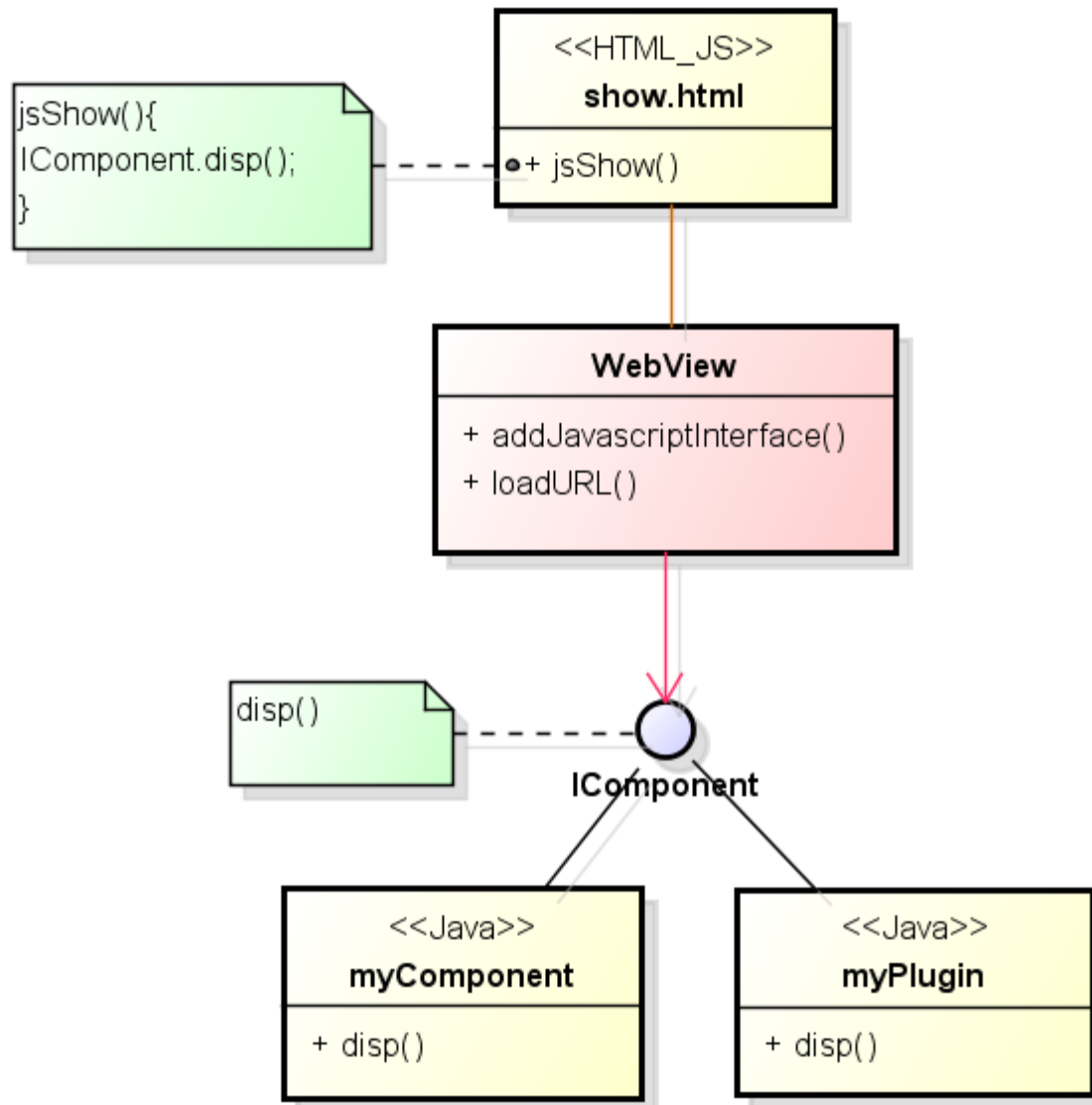
通用性接口

- Java插件实现IComponent通用性接口；然后WebView可以透过IComponent接口而调用Java插件的disp()函数。



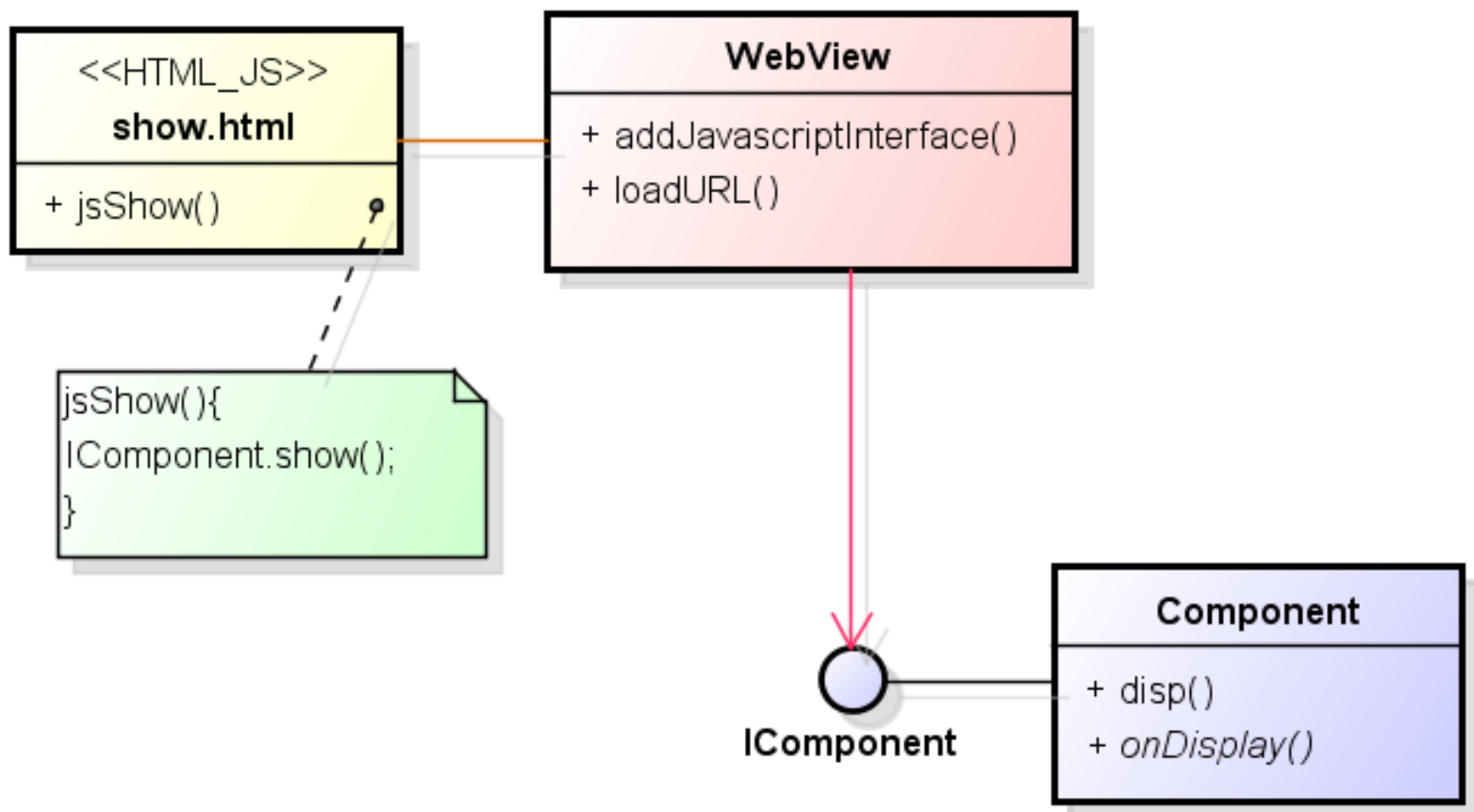
通用性接口

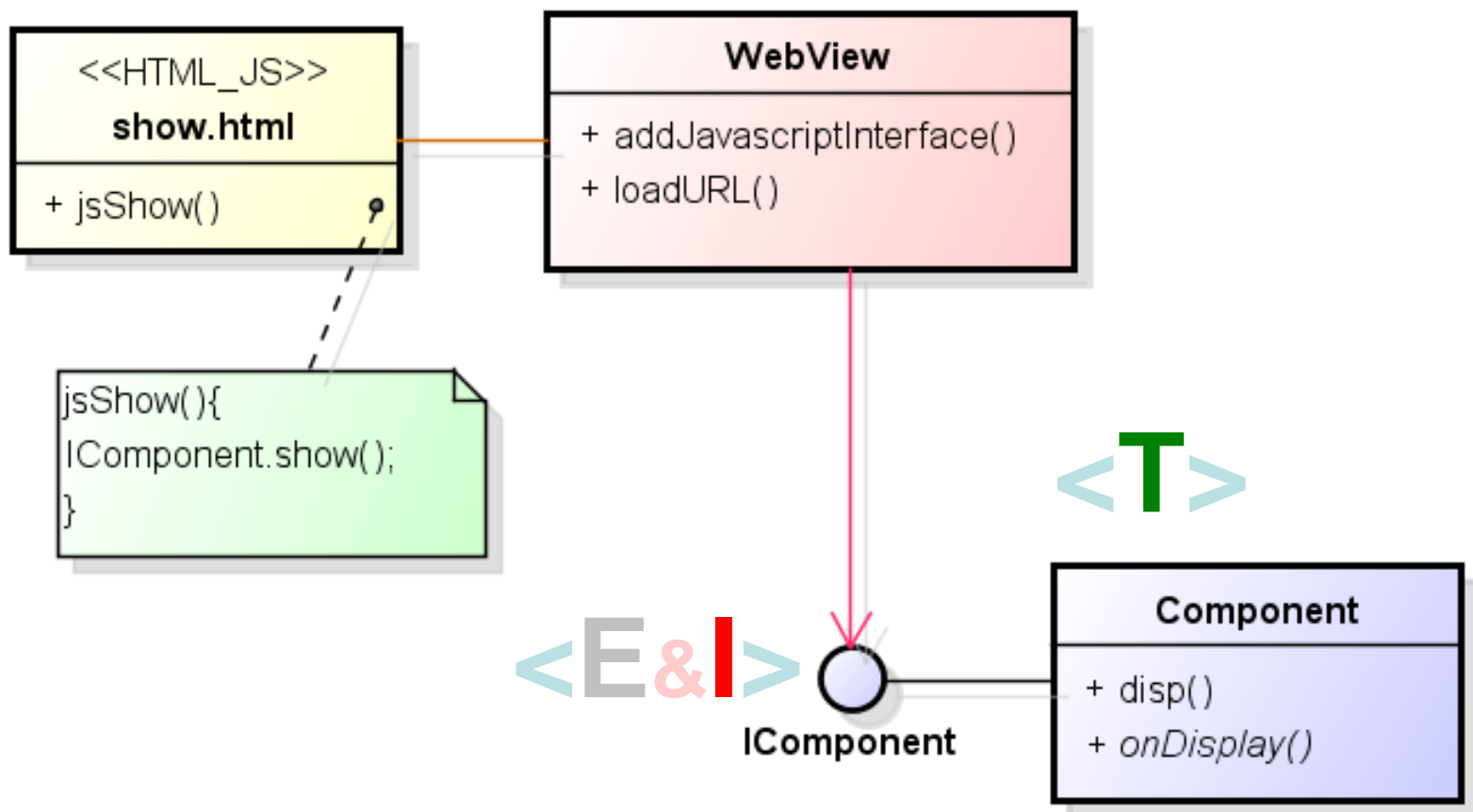
特殊性实现

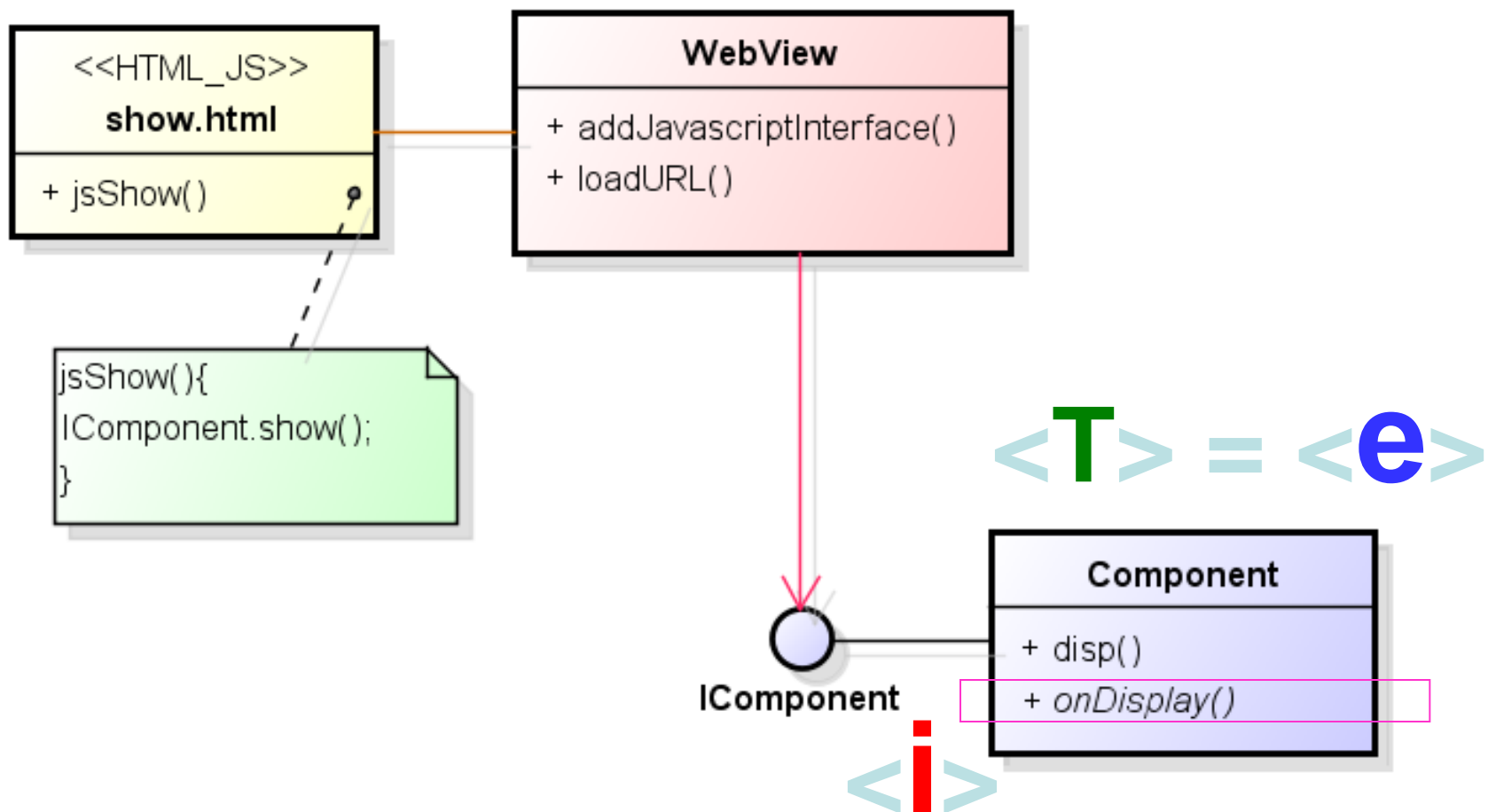


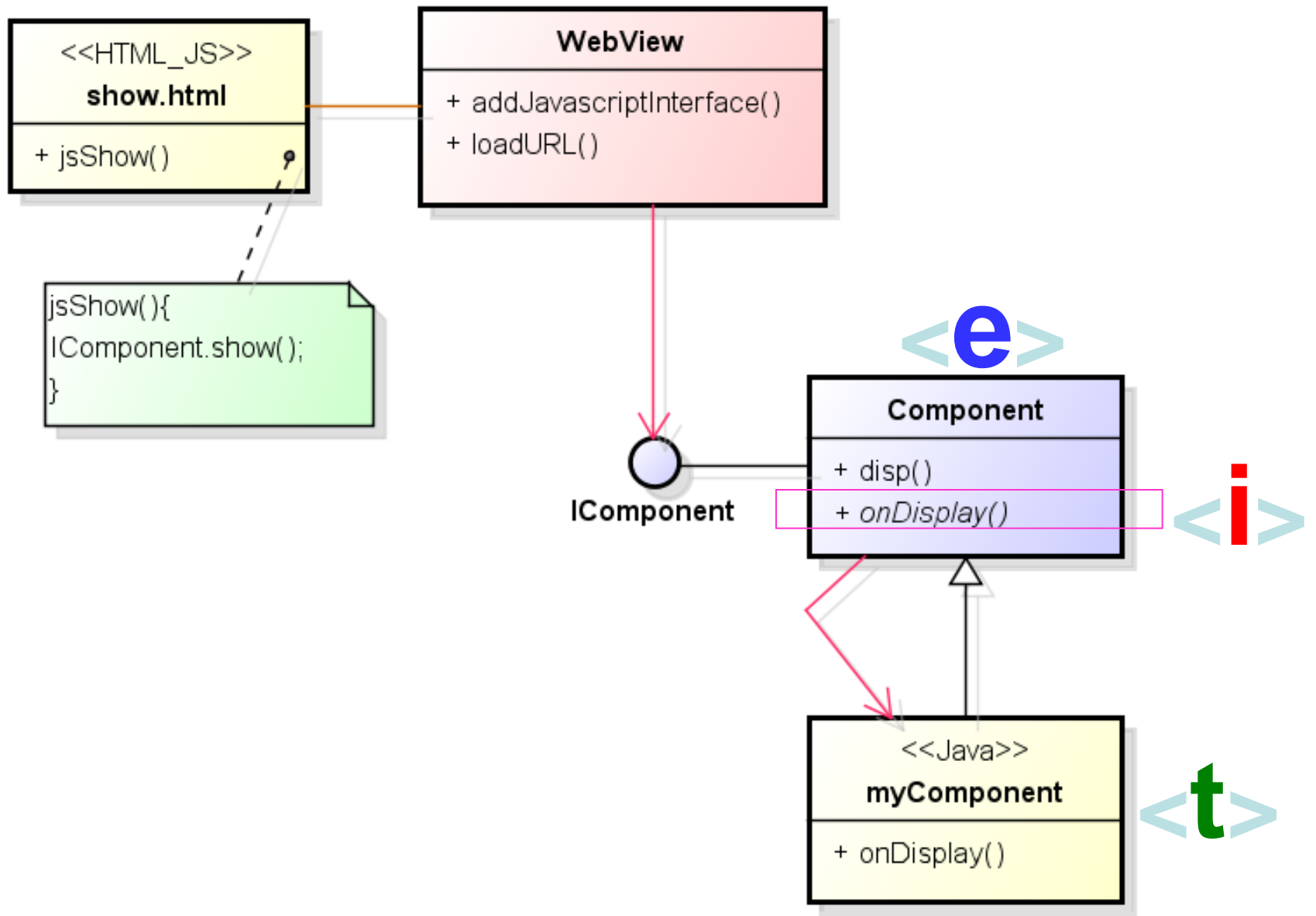
设计通用性基类

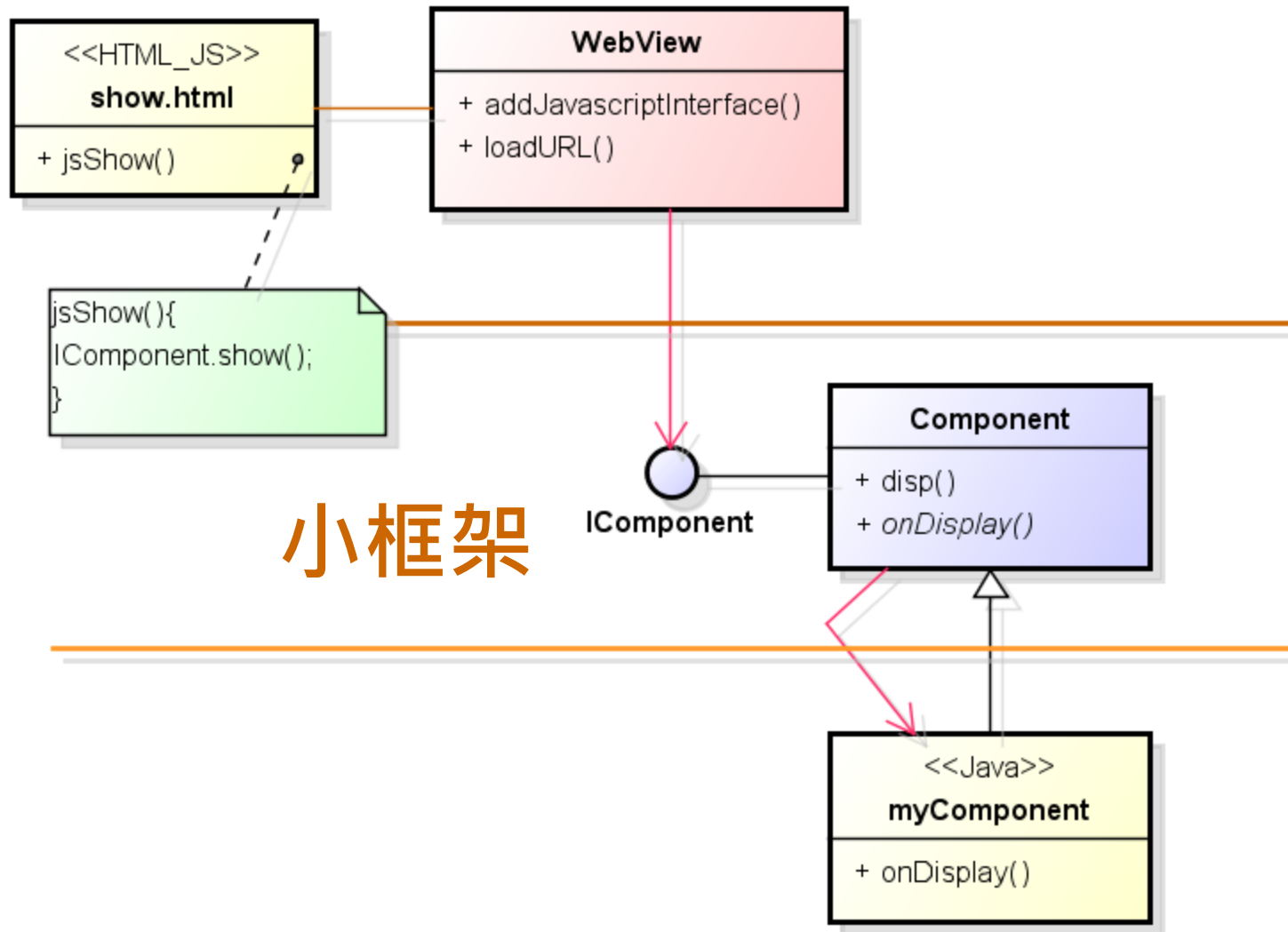
- 以上的IComponent接口已经转变成为通用性接口了。可以设计一个通用性基类来实现此通用性接口。你知道为什么需要这个通用性基类吗?那么,又应该由谁来定义这项基类呢?













~ Continued ~