OLAT: Inserting multiple custom links into ForumController and WikiMainController

Dirk Albrecht, Sebastian Volke

Universität Leipzig, Fakultät für Informatik

March – July 2009

Abstract

The OLAT extension framework has some very disturbing limitations. OLAT extensions can insert new actions into existing modules, provided an extension point exists within that specific module. But even then the extension is limited to just one action per module, which is quite restrictive. Furthermore only a title, a description and a controller to handle the events in case the action is triggered can be specified using the `ActionExtension` provided by the OLAT framework.

The goal of this paper is provide a way to overcome those restrictions and give new extensions the power to set up almost any thinkable detail of an extension point.

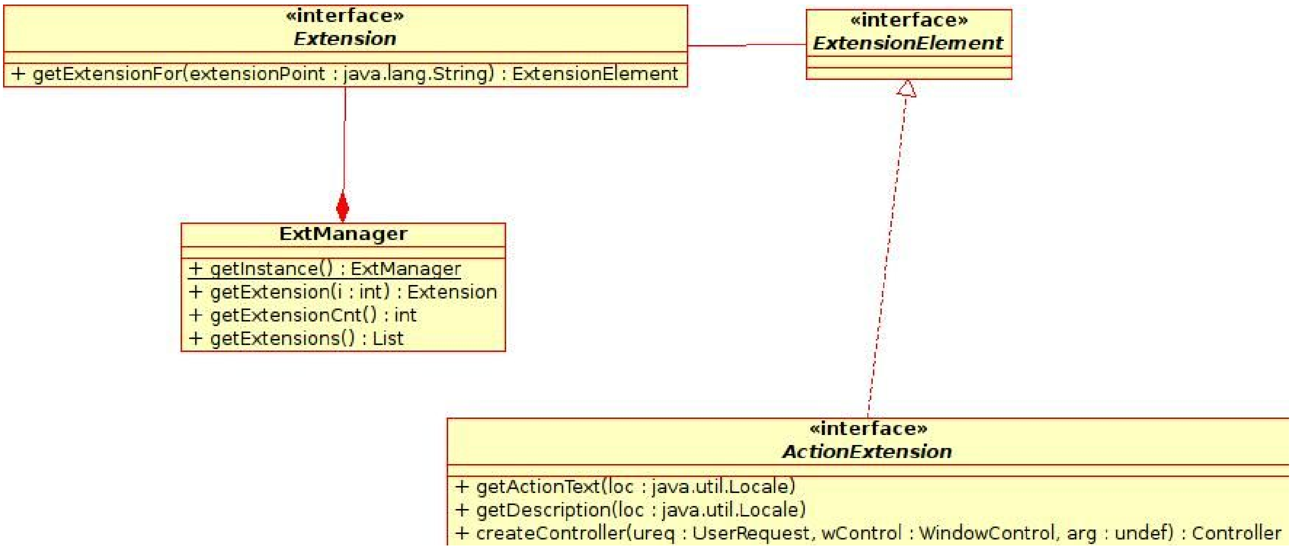
Creation extensions and extension points in OLAT

Olat extension framework

The OLAT extension framework consists of 2 interfaces (`Extension` and `ExtensionElement`) and the `ExtManager` class. As shown in the UML diagram the `ExtManager` manages all extensions which are implementations of the `Extension` interface. In turn every extension contains one ore more implementations of the `ExtensionElement` interface. The extension works as a container for the actual items to be insert into the respective module.

`ExtensionElement` is an empty interface. Anything can be an `ExtensionElement`, but that doesn't make to much sense in most cases. OLAT provides one sub-interface of `ExtensionElement`: `ActionExtension`, which is also part of the diagram.

In OLAT every module, that is: every class, can contain an extension point. This means, the class has to ask the `ExtManager` for a list of all extensions. Then it has to query every extension for `ExtensionElements` by passing a string to the `getExtensionFor(String id)` function of it. In most cases it sensible to pass the name of the querying class. If `getExtensionFor(String id)` returns an `ExtensionElement` (i.e. not null), this


«interface»
Extension
+ getExtensionFor(extensionPoint : java.lang.String) : ExtensionElement
«interface»
ExtensionElement
ExtManager
+ getInstance() : ExtManager
+ getExtension(i : int) : Extension
+ getExtensionCnt() : int
+ getExtensions() : List
«interface»
ActionExtension
+ getActionText(loc : java.util.Locale)
+ getDescription(loc : java.util.Locale)
+ createController(ureq : UserRequest, wControl : WindowControl, arg : undef) : Controller

element can be used to extend the functionality of the class. A great restriction at this point is, that for each id string only one ExtensionElement can be returned.

The preferred ExtensionElements in OLAT are implementations of ActionExtension. This interface allows to extend modules with additional actions. It provides functions to pass a title, a description and a controller. The controller is intended to be called, when the action is triggered. Note that title and description have to be passed as pre-translated strings. Doing the translation within the extensible module doesn't make sense for a real extension.

Extension point for forum and wiki

In our case we tried to create an extension point in ForumController and WikiMainController, i.e. to allow extensions to place custom links on top of the thread list in ForumController (next to the export-button) and within the toolbox of the WikiMainController.

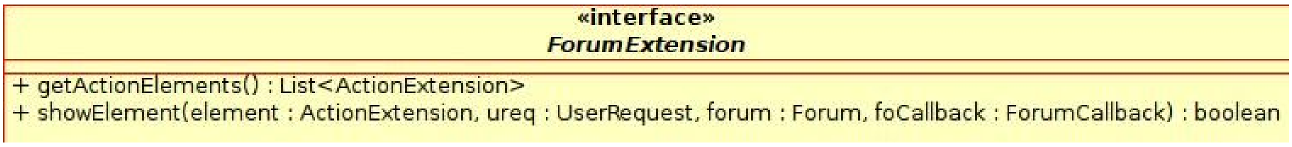
To achieve this, the respective controllers skim through the list of extensions from ExtManager and get the ExtensionElement from each of them (if it isn't null). In case the ExtensionElement is also an ActionExtension, a custom link is created (text from getActionText(), tooltip from getDescription()) and on click createController(...) is called.

Very limiting is: the programmer of the extension cannot configure, where the link is placed and he cannot determine, whether a user can see the link or not. This needs to be fixed.

When the user clicks the link, the ForumController (or respectively the WikiMainController) receives an event, creates the controller, that in turn could launch a wizard or some kind of dialog box etc.

Creating a more powerful extension point for ForumController and WikiMainController

For our purposes we needed to control the creation of the links much more fine-grained. We needed to determine whether a user should see the link or not. So we chose to create a new subinterface to ExtensionElement with a new level of abstraction:



```
«interface»
ForumExtension

+ getActionElements() : List<ActionExtension>
+ showElement(element : ActionExtension, ureq : UserRequest, forum : Forum, foCallback : ForumCallback) : boolean
```

This ExtensionElement is itself a container for ExtensionElements, precisely ActionExtensions. The only classes in OLAT that have a state, are the controllers. The extension doesn't have a state und thus can't track the rights or views of a specific user. But it is possible to have a backdoor. ForumExtensions returns a list of ActionsExtensions to the ForumController, that could possibly be shown to the user. Then the ForumController passes some stateful parameters to the ForumExtension, that in turn returns whether to show the link (true) or not (false).

Concerning the WikiMainController it could even be determined, where the link should be placed: in the toolbox, within the history, etc.