Seamless UI Integration of Extensions into OpenOffice.org

Carsten Driesner
OpenOffice.org Framework Project Lead
Sun Microsystems, Inc.

Overview

- What's the purpose of this presentation?
- Addons.xcu – A short introduction
- What features are available?
  - Introduction
  - Examples
  - How to use or implement it
  - Planned improvements.
- Future outlook

What's the purpose of this presentation?

- Some months ago I looked over the OpenOffice.org extension repository.
- What I found out?
  - Overall the quality of extensions is good.
  - User interface integration could be better.
- What could help extension developers?
  - A cookbook regarding user interface integration.

Addons.xcu – A short introduction

- Central OpenOffice.org configuration file (XML based) to define certain user interface aspects for an extension.
  - Define menus and toolbars
  - Define and associate custom images to commands
  - Merging for menus and toolbars
- Normally every extension with a user interface contains an Addons.xcu file.
- It will be enhanced in the future.

What features are available?

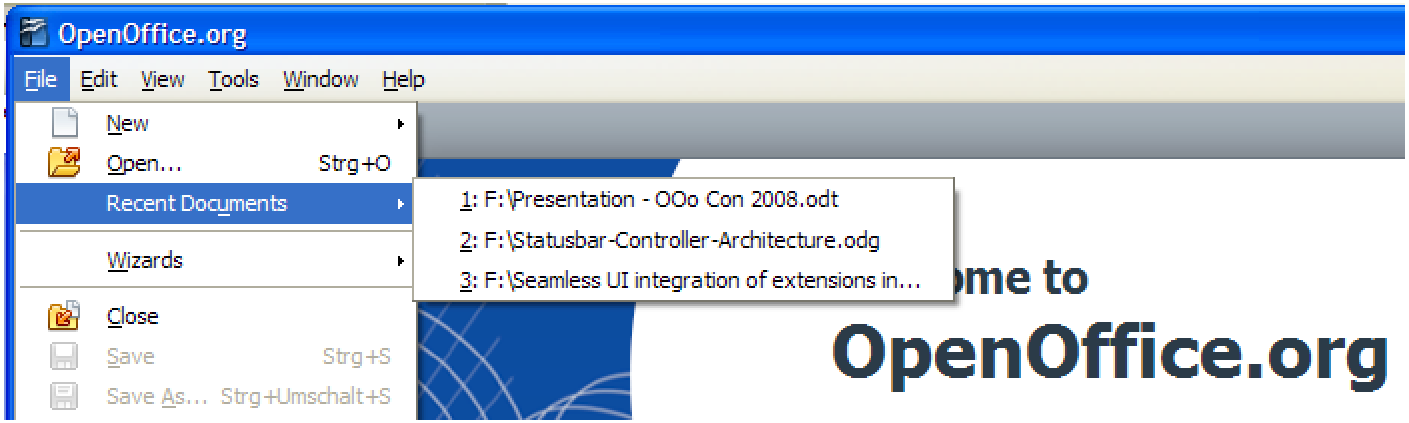
- Popup Menu Controller
- Complex Toolbar Controller
- Toolbar and Menu Merging
- Accelerators
- Context Menu Interception

Popup Menu Controller

- What's a popup menu controller?
  - A popup menu controller manages the content of a popup menu which is part of the OpenOffice.org menu bar.
- When should one use a popup menu controller?
  - Whenever the content is dynamic or only known at runtime.
  - If you want to exchange an existing popup menu controller.

Popup Menu Controller

- Examples
  - The recent file list (File – Recent Documents)
  - Create a new document (File – New)
  - Available toolbar list (View - Toolbars)


OpenOffice.org
File
Edit
View
Tools
Window
Help
New
Open... Strg+O
Recent Documents
Wizards
Close
Save Strg+S
Save As... Strg+Umschalt+S
1: F:\Presentation - OOo Con 2008.odt
2: F:\Statusbar-Controller-Architecture.odg
3: F:\Seamless UI integration of extensions in...
ome to
OpenOffice.org

Popup Menu Controller

- How to implement a popup menu controller?
  - Two different parts are needed.
    - 1. Add your popup menu controller to the Controller.xcu configuration file.
    - 2. Implement the UNO com.sun.star.frame.PopupMenuController service.

Popup Menu Controller

Sample Controller.xcu

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE oor:component-data SYSTEM "../../../../../component-update.dtd">
<oor:component-data oor:name="Controller" oor:package="org.openoffice.Office.UI"
xmlns:oor="http://openoffice.org/2001/registry" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<node oor:name="Registered">
  <node oor:name="PopupMenu">
    <node oor:name="org.openoffice.Office.MyNewPopupMenuController" oor:op="replace">
      <prop oor:name="Command">
        <value>org.openoffice.Office.MyExtension:MyCommand</value>
      </prop>
      <prop oor:name="Module">
        <value/>
      </prop>
      <prop oor:name="Controller">
        <value>org.openoffice.office.myExtension.ControllerImplementationName</value>
      </prop>
    </node>
  </node>
</node>
```

- A very simple format for controller registration.

Popup Menu Controller

UNO Service: com.sun.star.frame.PopupMenuController

- There are two central interfaces.

```
module com { module sun { module star { module frame {
interface XPopupMenuController : com::sun::star::uno::XInterface
{
    void setPopupMenu( [in] com::sun::star::awt::XPopupMenu PopupMenu );

    void updatePopupMenu();
};
}; }; }; };
```

```
module com { module sun { module star { module awt {
interface XMenuListener : com::sun::star::lang::XEventListener
{
    [oneway] void highlight( [in] com::sun::star::awt::MenuEvent rEvent);

    [oneway] void select( [in] com::sun::star::awt::MenuEvent rEvent);

    [oneway] void activate( [in] com::sun::star::awt::MenuEvent rEvent);

    [oneway] void deactivate( [in] com::sun::star::awt::MenuEvent rEvent);
};
}; }; }; };
```

- Comprehensive tutorial is available here:
  - > http://wiki.services.openoffice.org/wiki/Framework/Tutorial/Popup_Menu_Controller

Popup Menu Controller

- Planned improvements for OpenOffice.org 3.1
  - Enhanced UNO AWT* menu API (Mainly done by community member Ariel Constenla Haile)
    - Support for images (get/set/remove/rotate/mirror)
    - Get/set tool tip and help text
    - Get/set keyboard shortcut information (accelerators)
    - New style services for ease of use
  - New UNO service for simple access to custom images (Mainly done by Shi Zhoubo from RedFlag 2000)
    - Retrieve your custom images via command URL

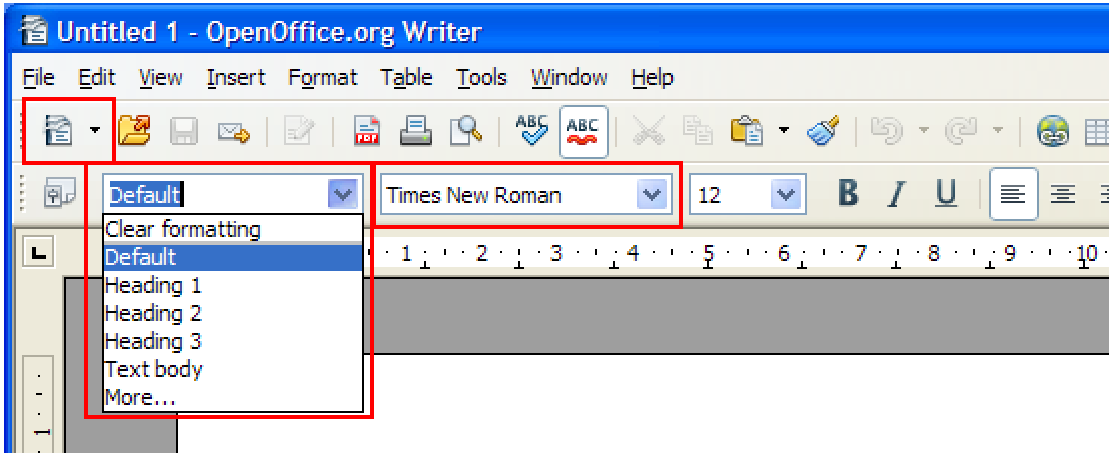
* Abstract Window Toolkit

Complex Toolbar Controller

- What's a complex toolbar controller?
  - An implementation which manages all aspects of a single toolbar item.
- When should one use a complex toolbar controller?
  - If you want to provide a set of functions / values at one place.
  - The user should be able to enter information.
  - To sum up:
    - Whenever a simple button doesn't provide enough flexibility.

Complex Toolbar Controller

- Examples
  - > Apply Style Combo Box (Formatting Bar)
  - > Font Name Combo Box (Formatting Bar)
  - > New Document Button (Standard Bar)



Complex Toolbar Controller

- How to implement a complex toolbar controller?
  - > Two different parts are needed. Extensions providing a user interface integration already have both of them.
    - – Add a controller type property to your Addons.xcu
    - – Extend your protocol handler implementation
    - – If you want to receive notifications from the toolbar control implement: com.sun.star.frame.XControlNotificationListener

Complex Toolbar Controller

Sample Addons.xcu

```xml
<?xml version='1.0' encoding='UTF-8'?>
<oor:component-data xmlns:oor="http://openoffice.org/2001/registry"
xmlns:xs="http://www.w3.org/2001/XMLSchema" oor:name="Addons"
oor:package="org.openoffice.Office">
  <node oor:name="AddonUI">
    <node oor:name="OfficeToolBar">
      <node oor:name="org.openoffice.Office.example.Search" oor:op="replace">
        <prop oor:name="URL">
          <value>org.openoffice.Office.example.Search:SearchText</value>
        </prop>
        <prop oor:name="Title">
          <value xml:lang="en-US">Search Text</value>
        </prop>
        <prop oor:name="Context" oor:type="xs:string">
          <value>com.sun.star.text.TextDocument</value>
        </prop>
        <prop oor:name="ControlType" oor:type="xs:string">
          <value>Combobox</value>
        </prop>
        <prop oor:name="Width" oor:type="xs:long">
          <value>200</value>
        </prop>
      </node>
    </node>
  </node>
</oor:component-data>
```

- Just two additional properties to define a combo box with a certain width.

Complex Toolbar Controller

Sample code (C++) to set a list

```
// The command "SetList" currently supports only one parameter
Sequence< NamedValue > aArgs( 1 );

// create a sequence of strings for the dropdown menu
Sequence< rtl::OUString > aList( 3 );
aContextMenu[0] = rtl::OUString::createFromAscii( "Entry 1" );
aContextMenu[1] = rtl::OUString::createFromAscii( "Entry 2" );
aContextMenu[2] = rtl::OUString::createFromAscii( "Entry 3" );

// argument is called "List"
aArgs[0].Name = rtl::OUString::createFromAscii( "List" );
aArgs[0].Value <<= aList;

// initialize control command
com::sun::star::frame::ControlCommand aCtrlCmd;
aCtrlCmd.Command = rtl::OUString::createFromAscii( "SetList" );

aCtrlCmd.Arguments = rArgs;
com::sun::star::frame::FeatureStateEvent aEvent;
aEvent.FeatureURL = aURL;
aEvent.Source     = (::com::sun::star::frame::XDispatch*)this;
aEvent.IsEnabled  = bEnabled;
aEvent.Requery    = sal_False;
aEvent.State    <<= aCtrlCmd;

// send command via statusChanged to the control
xControl->statusChanged( aEvent );
```

- More information:
  - > http://wiki.services.openoffice.org/wiki/Framework/Article/Generic_UNO_Interfaces_for_complex_toolbar_controls

Complex Toolbar Controller

- Planed improvements
  - > Currently we have no plans.
  - > The design is very flexible therefore enhancements can be done very easily.
  - > This is a very good playground for people who want to help us.
- Feedback is required from developers.
  - > What do you miss?

Toolbar and Menu Merging

- What's toolbar and menu merging?
  - A feature for extensions to manipulate toolbars and menu bars without writing any code.
- When should one use toolbar and menu merging?
  - If you want to add a your commands into existing user interface elements.
  - If you want to exchange or remove items from existing user interface elements.

Toolbar and Menu Merging

- Examples (use case)
  - Your content management system should be available for OpenOffice.org. Loading and storing functions should be seamlessly integrated.
- How to use toolbar and menu merging?
  - Add your merging instructions into the Addons.xcu file.
    - There are different sections for Toolbar and Menu Bar merging.

Toolbar and Menu Merging

Sample Addons.xcu to insert a toolbar item to the standardbar

```
<?xml version='1.0' encoding='UTF-8'?>
<oor:component-data xmlns:oor="http://openoffice.org/2001/registry"
xmlns:xs="http://www.w3.org/2001/XMLSchema" oor:name="Addons"
oor:package="org.openoffice.Office">
  <node oor:name="AddonUI">
    <node oor:name="OfficeToolbarMerging">
      <node oor:name="org.openoffice.test.testcomponent" oor:op="replace">
        <node oor:name="T1" oor:op="replace">
          <prop oor:name="MergeToolBar">
            <value>standardbar</value>
          </prop>
          <prop oor:name="MergePoint">
            <value>.uno:Open</value>
          </prop>
          <prop oor:name="MergeCommand">
            <value>AddAfter</value>
          </prop>
          <prop oor:name="MergeContext">
            <value/>
          </prop>
          <node oor:name="ToolBarItems">
            <node oor:name="B1" oor:op="replace">
              <prop oor:name="URL">
                <value>.uno:CloseDoc</value>
              </prop>
              <prop oor:name="Title">
                <value>Close Document</value>
              </prop>
            </node>
</node> </node> </node> </node> </node> </oor:component-data>
```

- More information about merging:
  - http://wiki.services.openoffice.org/wiki/Framework/Article/Addon_Menu_Toolbar_Merging

Toolbar and Menu Merging

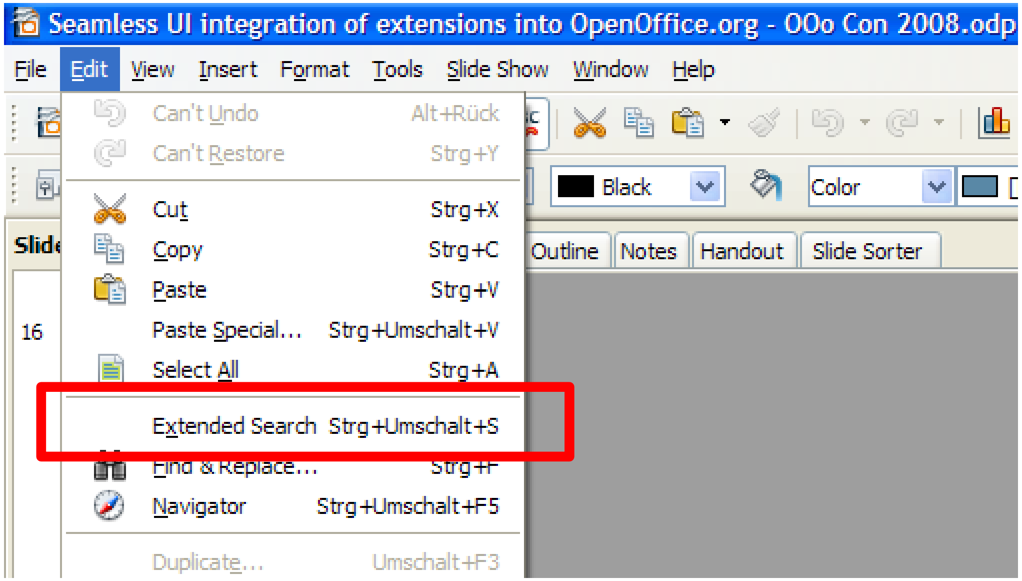
- Planned improvements
  - No plans for improvements.
  - The design is very flexible therefore enhancements can be done easily.
- Feedback is required from developers.
  - What do you miss?
  - Did you encounter performance issues?

Accelerators

- What are accelerators?
  - An accelerator key is a key or set of keys that performs a predefined function.
- When should one use accelerators?
  - Whenever you want a seamless integration of your extension into OpenOffice.org. There are several users who prefer to use accelerators to execute commands.

Accelerators

- Example (use case)
  - > Your extended search function should be accessible via an accelerator.



Accelerators

- How to define accelerators?
  - There are two steps needed to register accelerators.
    - Add a job to your extension to start your accelerator registration code.
    - Use the user interface configuration API to register your accelerators.

Accelerators

Sample code (C++) to add an accelerator (CTRL+SHIFT+S) to the global accelerator configuration.

```
...

// Create the global accelerator configuration service
Reference< com::sun::star::ui::XacceleratorConfiguration > xGlobalAccelCfg(
  m_xServiceManager->createInstance(
    ::rtl::OUString("com.sun.star.ui.GlobalAcceleratorConfiguration")),
  UNO_QUERY_THROW );

// set key event to define the accelerator
com::sun:star::awt::KeyEvent aKeyEvent;
aKeyEvent.KeyCode   = com::sun::star::awt::Key::S;
aKeyEvent.Modifiers = com::sun::star::awt::KeyModifier::SHIFT|
                      com::sun::star::awt::KeyModifier::MOD1;

// associate the accelerator with a command
xGlobalAccelCfg->setKeyEvent(aKeyEvent, rtl::OUString::createFromAscii(“.uno:Open”));

// store the changes in the configuration
xGlobalAccelCfg->store();

...
```

- More information can be found here:
  - http://wiki.services.openoffice.org/wiki/Framework/Tutorial/Accelerators

Accelerators

- Planned improvements for OOo 3.1
  - Accelerators are stored in the configuration (Cooperation with Wu Yan from RedFlag 2000).
  - Extensions can add, remove or replace accelerators using a configuration file.

Sample Accelerators.xcu

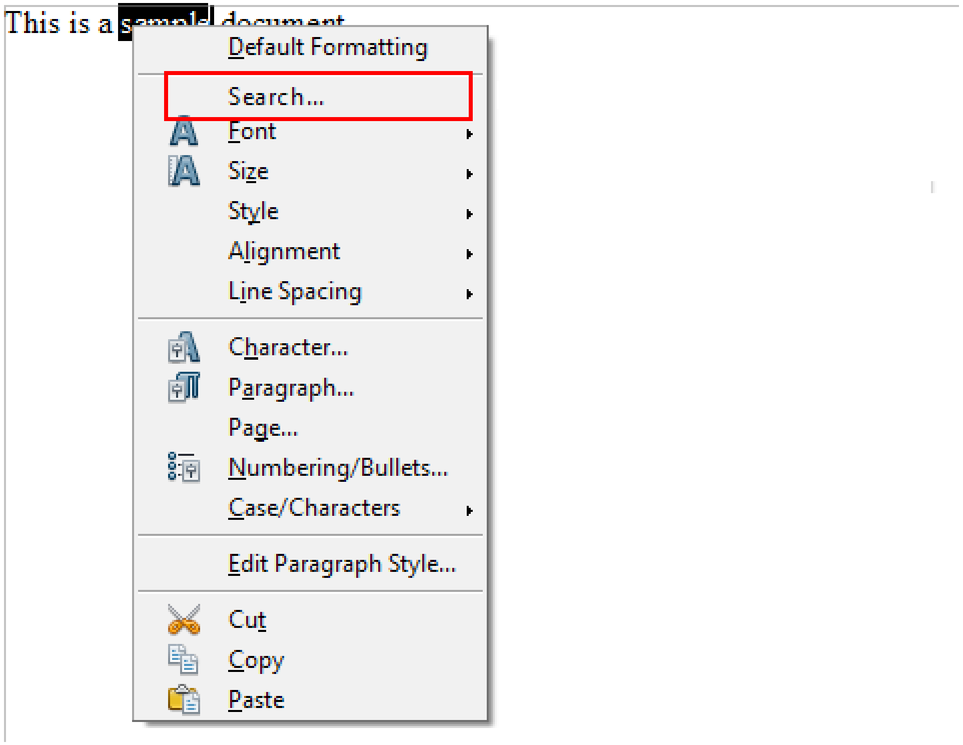
```
<?xml version="1.0" encoding="UTF-8"?>
<oor:component-data xmlns:oor="http://openoffice.org/2001/registry"
xmlns:xs="http://www.w3.org/2001/XMLSchema" oor:name="Accelerators"
oor:package="org.openoffice.Office">
  <node oor:name="PrimaryKeys">
    <node oor:name="Global">
      <node oor:name="V_MOD1" oor:op="remove"/>
      <node oor:name="V_SHIFT_MOD1" oor:op="replace">
        <prop oor:name="Command">
          <value xml:lang="en-US">.uno:Paste</value>
        </prop>
      </node>
    </node>
  </node>
  <node oor:name="SecondaryKeys">
    <node oor:name="Global">
      <node oor:name="V_MOD1" oor:op="replace">
        <prop oor:name="Command">
          <value xml:lang="en-US">.uno:Paste</value>
        </prop>
      </node>
    </node>
  </node>
</oor:component-data>
```

Context Menu Interception

- What's context menu interception?
  - A feature to manipulate context menus. You can add, remove and exchange menu items. Additionally it's possible to completely prevent that context menus are shown.
- When should one use context menu interception?
  - If your extension wants to add context dependent functions to a menu.
  - If you want to suppress a certain or all context menus.

Context Menu Interception

- Examples (use case)
  - You want to add your search function to the Writer. Adding this function to a context menu looks natural.



This is a sample document

Context Menu Interception

- How to use context menu interception?
  - There are several steps needed to have a working context menu interceptor.
    - Add a job to your extension to receive events when a new frame is created (OnViewCreated).
    - Implement a frame action listener to receive frame action events. That's necessary to register your interceptor on certain events (e.g. component attached)
    - Implement a context menu interceptor that manipulates context menus.

Context Menu Interception

Very simple sample code to modify a context menu

```
ui::ContextMenuInterceptorAction SAL_CALL MyContextMenuInterceptor::notifyContextMenuExecute(
    const ui::ContextMenuExecuteEvent& rEvent )
throw (uno::RuntimeException)
{

    // The action trigger container is also a factory for sub containers and entries
    uno::Reference< lang::XMultiServiceFactory > xFactory( rEvent.ActionTriggerContainer,
                                                           uno::UNO_QUERY);

    // Create a simple menu entry using "com.sun.star.ui.ActionTrigger"
    uno::Reference< beans::XPropertySet > xMenuEntry( xFactory->createInstance(
        ::rtl::OUString::createFromAscii("com.sun.star.ui.ActionTrigger")), uno::UNO_QUERY);

    // Initialize label and command URL for the entry
    ::rtl::OUString aLabel      = ::rtl::OUString::createFromAscuu("Search...");
    ::rtl::OUString aCommandURL = ::rtl::OUString::createFromAscii(
                                          "org.openoffice.Office.Example:Search");

    // Initialize label and command URL for the entry
    xMenuEntry->setPropertyValue(::rtl::OUString::createFromAscii("Text"), uno::Any(aLabel)));
    xMenuEntry->setPropertyValue(::rtl::OUString::createFromAscii("CommandURL"),
                                                       uno::Any(aCommandURL)));

    // Insert entry to the third position
    rEvent.ActionTriggerContainer->insertByIndex(2, uno::Any(xMenuEntry));

    // Provide information to the framework how to proceed
    return ui::ContextMenuInterceptorAction_EXECUTE_MODIFIED;
}
```

- More information about context menu interception:
  - > http://wiki.services.openoffice.org/wiki/Framework/Tutorial/Context_Menu_Interception

Context Menu Interception

- Planned improvements (no target yet)
  - Every context menu should have a unique identifier.
    - E.g. “private:resource/contextmenu/spellcheck”
  - Convert resource based context menus to XML files.
  - Provide context menu merging via Addons.xcu file.

Future outlook

- Docking window support for extensions (Cooperation with RedFlag 2000)
  - Integrate modeless UNO dialogs into docking windows
  - Layout manager support for docking windows.
- Context menus defined by XML files.
  - Merging via configuration files.
- Extended UNO AWT.
  - More controls
    - Grid control
  - Access to more features/functions.
    - Retrieve system properties
    - Notifications when settings has been changed (colors, fonts).

Q & A

Meet the Sun Experts at the Sun Booth.