OpenOffice.org and XUL – Embedding Gecko in OOo

Stephan Schäfer

Project Lead GSL

Sun microsystems

About Me

- StarOffice developer since 2000
- Project Lead GSL, main topic VCL
  - VCL architecture and Win32 support
  - System windows, Accessibility, RTL, NWF, Toolbars
  - Writing prototypes...
- Before Sun
  - GIS Software
  - 3D Computer Graphics

Outline

Goal: Evaluate Alternative UI Framework

- Overview of OOo's UI
- Motivation
- Previous Work
- XUL
- Embedding Gecko
- UNO/XPCOM bridge
- Demo
- Evaluation and Consequences

Overview of OOo's UI Layer

- Single system window as application frame
  - > inner windows are just clip regions
  - > all components are hand drawn, controls use NWF
  - > rich control set, platform independent
  - > supports UI mirroring
- Accessibility
  - > platform independent through Java access bridge
- Resource system
  - > localization framework

Motivation: UI

- Resource system
  - > file format, needs compiler
  - > no GUI editor
  - > customization requires full build environment
- Dialog contents
  - > fixed position and sizes
  - > string placeholder must allow for largest translation
  - > no layout management
- UI support for API programmers
  - > AWT-like toolkit on top of VCL, no resource system
  - > Java UI: theming issues, modal dialogs difficult

Sample Resource File

```
HelpId = SID_ATTR_ZOOM ;
Size = MAP_APPFONT ( 160 , 142 ) ;
Text [ de ] = "Maßstab" ;
Moveable = TRUE ;
RadioButton BTN_WHOLE_PAGE
{
    Pos = MAP_APPFONT ( 12 , 14 ) ;
    Size = MAP_APPFONT ( 80 , 10 ) ;
    Text [ de ] = "~Ganze Seite" ;
    Text [ en-US ] = "~Entire Page" ;
    Text [ x-comment ] = " ";
};
RadioButton BTN_PAGE_WIDTH
{
    Pos = MAP_APPFONT ( 12 , 27 ) ;
    Size = MAP_APPFONT ( 80 , 10 ) ;
    Text [ de ] = "Seiten~breite" ;
    Text [ en-US ] = "~Page Width" ;
    Text [ x-comment ] = " ";
};
```

Motivation: Applications

- Tight connection between core and UI
  - class inheritance throughout modules
  - C++ API (VCL), pretty hard to replace
- Example: Dialogs
  - no common API, argument handling inconsistent
  - calls into core via pointer, bypassing UNO API
- UNO API for UI development
  - different toolkits, platforms, small devices, etc.
  - better code through required refactoring

Motivation: Maintainability

- Huge existing code base
  - > still growing
  - > things get not removed
- Changes are expensive
  - > improvements only if we do it
  - > regressions as well...
  - > benefit from speed-ups in external projects
- But then: new dependencies
  - > quality, support, integration, build process
  - > align road maps

Our Goal Revisited

- Separate applications and UI
  - > define dialog API
  - > implement dialogs with any toolkit
  - > use a powerful toolkit
- Still not really customizable
  - > instantiation is hard-coded
  - > limited to cosmetic changes, context is fixed
- Long-term goal (*Vision*): active UI
  - > provide rich API for arbitrary UI actions
  - > dialogs could actively modify any document parameters
  - > embed application modules in external UI framework

Previous Work

- Need for change is obvious
  - > OOoCon 2003: New Toolkit/Canvas for OOo
  - > OOoCon 2004: Simplifying the OOo UI Layout Process
- Qt/GTK/wxWidgets/...
  - > too close to VCL
  - > small benefit since native widget rendering
  - > integration costs too high just for UI layout
- Write layouter, change resource format
  - > still an option
- But XUL is more...

What is XUL ?

- XML dialect: XML User Interface Language
  - > XML: UI elements, layout etc.
  - > CSS: appearance, skins
  - > JavaScript: programmatic control
- Used by the Mozilla Application Suite
  - > Mozilla, Firefox, Thunderbird,...
- Gecko rendering engine
  - > renders UI and web contents
  - > has an embedding API
  - > available standalone as *XULRunner*

Comparison sheet

Feature	VCL	XUL
Basic control set	Yes	Yes
Extended Control Set	Yes	Requires XTF
Localization support	Yes	Yes
CJK/CTL/BiDi UI	Yes	Yes
Accessibility	Yes	Platform dependent
System integration (NWF)	Yes	Yes
Layout manager	No	Yes
Rapid prototyping	No	Yes
Style sheets	No	Yes
Resource Editor	No	Probably
Resource format	Binary	XML

Embedding Gecko

- Dialog based approach
  - rewrite embedding sample using VCL
    - `nsIWebBrowserChrome`, `nsIEmbeddingSiteWindow`, `nsIWebProgressListener`
    - e.g., `<mozsource>/embedding/tests/winEmbed`
  - window management by VCL, rendering by Gecko
- Features
  - renders arbitrary XUL/HTML/etc. in VCL control
  - supports chrome URLs
  - allows for DOM traversal
  - can intercept key events (e.g., F1 for online-help)

The Prototype

Replace An Existing VCL Dialog By Using XUL

- XPCOM/UNO bridge
  - > written by Daniel Boelzle
  - > designed as XPCOM component
  - > bridges between JavaScript (XPCOM) and OOo (UNO)
- The Zoom dialog
  - > simple enough, lives in SVX
  - > use VCL's Gecko control, XPCOM/UNO bridge
  - > provide chrome package
  - > UNO service that provides access to dialog parameters

The Prototype

Demo...

Evaluation

- Proof of concept successful
- But...
  - > Missing UNO API to access ItemSets
  - > Accessibility API mismatch (Java vs. native APIs)
  - > Complex controls (even SpinFields) missing -> XTF
  - > Deployment unclear (XULRunner)
  - > Mozilla components in OOo use different bootstrapping
  - > Localization framework must support XUL files
  - > Online-Help and Testtool depend on HelpIDs
  - > Testtool library expects VCL controls

Consequences

- XUL Dialogs probably not in OOo 3.0 ☹
- Evaluation continues
- Short-term goal: Migrate Help Viewer
  - > currently Writer/Web displays HTML help
  - > embed Gecko instead
  - > can be reverted easily
  - > addresses several important goals: build environment, run-time issues, deployment, accessibility
- Support for SDK users ?
  - > would require UNO wrapper for new XULDialog class
  - > quality unclear if technology not used inside OOo

Credits And Links

- Daniel Boelzle
  - > XPCOM/UNO bridge
- Philipp Lohmann
  - > GTK support
- Kay Ramme
  - > Initiator and XUL promoter
- Web Resources:

  http://www.mozilla.org/projects/embedding/

  http://www.xulplanet.com

  Nigel McFarlane: Rapid Application Development with Mozilla (http://phptr.com/Perens)

  O'Reilly's Creating Applications With Mozilla (http://books.mozdev.org/)

OpenOffice.org and XUL

Stephan Schäfer

stephan.schaefer@sun.com

ssa@openoffice.org