TkGecko: Another Attempt for an HTML Renderer for Tk

Georgios Petasis

Software and Knowledge Engineering Laboratory,
Institute of Informatics and Telecommunications,
National Centre for Scientific Research "Demokritos",
Athens, Greece
petasis@iit.demokritos.gr

Overview

- Tk and HTML
  - Tkhtml & Hv3
  - Embedding popular browsers
- Gecko
  - TkGecko: embedding Gecko
- Examples
  - Rendering a URL
  - Retrieving information from the DOM tree
- Conclusions

Tk and HTML

- Displaying HTML in Tk has always been an issue
- This shortcoming has been the motivation for a large number of attempts:
  - From simple rendering of HTML subsets on the text or canvas widget
    - ✓ i.e. for implementing help systems)
  - To full-featured Web browsers
    - ✓ like Hv3 or BrowseX
- The relevant Wiki page lists more than 20 projects
  - Does not cover all approaches trying to embed existing browsers in Tk (COM, X11, etc)

Tkhtml

- Tkhtml is one of the most popular extensions
  - An implementation of an HTML rendering component for the Tk toolkit in C
  - Actively maintained
  - Supports many HTML 4 features
    - ✓CCS
    - ✓JavaScript (through the Simple ECMAScript Engine)
- Despite the impressive supported list of features, Tkhtml is missing features like:
  - Complete JavaScript support
  - Flash
  - Java, ...

Embedding popular browsers

- Several approaches that try to embed a full-featured Web browser have been presented
- Internet Explorer is a popular target (Windows)
  - Through COM, either with Tcom or Optcl
- Under Unix, a similar effort has been made by the Entice project
  - Which embeds Firefox through the TkXext extension for X11
- TkGecko is a similar approach: tries to embed a popular and cross-platform browser
  - The rendering engine of Firefox was chosen, known as Gecko

TkGecko

- Not the first project that tries to embed Gecko
  - An earlier attempt has been sponsored by a company(Eolas Technologies)
  - Presented at the 7th Tcl conference (2000)
  - Closed-source project
- Newer TkGecko is open source
  - Under the BSD license
  - Sources hosted at SourceForge

Mozilla's Gecko

- Gecko is a cross-platform, standards-compliant and feature-complete rendering engine
- Quite popular:
  - Firefox, Thunderbird, Camino, Flock, SeaMonkey, k-Meleon, Netscape 9, etc.
- Quite complex
  - Its complexity is evident not only by the size of its source code, but also by the effort required in order to embed it in a C++ application
- Embedding is performed through XPCOM
  - A protocol similar to COM

Embedding Gecko

- Documentation for embedding Gecko is available
- Embedding Gecko is not easy:
  - Stability of the API: more fluid than stable
    - ✓ Multiple embedding APIs available
    - ✓ A new one has appeared since TkGecko development started (less than a year ago!)
  - Complexity of the API: functionality scattered among many interfaces
  - Thread-safety: unknown internal threading model
  - Dependence upon toolkits: i.e. GTK+ under Linux
    - ✓ The toolkit must be initialised along Tk, and Tk windows to be mapped in windows of the toolkit

TkGecko

- TkGecko implements a large percent of the required interfaces
  - Supporting a wide range of features
- The source code is a mixture of static code, and code dynamically generated through SWIG
  - SWIG is mostly used to interface DOM classes
- Building TkGecko is not difficult, due to CMake
  - Provided that a Development XULRunner SDK is available
- Under Linux, GTK+ is initialised in a separate thread, under which Gecko is embedded
  - Messages are exchanged among this thread and Tk ones

Using TkGecko: an example (1)

CODE

Initialise XPCOM/XUL

Create a TkGecko widget

Register Virtual Events Callbacks

```
package require Tk
package require tkgecko
set paths {
  {C:\Program Files (x86)\Mozilla Firefox\xpcom.dll}
  {C:\Program Files\Mozilla Firefox\xpcom.dll}
  /usr/lib64/xulrunner-1.9.1/libxpcom.so
  /usr/lib64/xulrunner-sdk-1.9.1/sdk/lib/libxpcom.so
  /usr/lib/xulrunner-1.9.1/libxpcom.so
  /usr/lib/xulrunner-sdk-1.9.1/sdk/lib/libxpcom.so
}
set xpcom [tkgecko::initializeXPCOM {*}$paths]
puts "XPCOM library: $xpcom"
```

Using TkGecko: an example (2)

```
set xuldir [file nativename [file dirname $xpcom]]
set appdir {} ;# Same as xuldir...
set profiledir [file native [file normalize ~/.tkgecko]]

puts "XUL directory:     $xuldir"
puts "APP directory:     $appdir"
puts "Profile directory: $profiledir"

puts "tkgecko::initializeXUL:\
  [tkgecko::initializeXUL $xuldir $appdir $profiledir]"

puts ================================================================
puts "              Initialisation completed!"
puts ================================================================
```

TkGecko: creating a widget

```
set URI https://developer.mozilla.org/en-US/
grid [ttk::button .back    -text { < }    -command onBack] \
     [ttk::button .forward -text { > }    -command onForward] \
     [ttk::button .reload  -text {Reload} -command onReload] \
     [ttk::entry  .uri     -textvariable URI] \
     [ttk::button .load    -text {Load}   -command onLoad] \
  -padx 2 -pady 2 -sticky snew
grid [tkgecko::browser .browser -width 800 \
            -height 600 -highlightthickness 1] \
  -columnspan 5 -sticky snew -padx 2 -pady 2
grid [ttk::label .status -textvariable STATUS] - - -\
     [ttk::progressbar .progress] \
  -sticky snew -padx 2 -pady 2
grid columnconfigure . 3 -weight 1
grid rowconfigure    . 1 -weight 1
```

TkGecko: adding bindings

```
## Bindings:
bind .browser <<OnStatusScriptChange>> {set ::STATUS [lindex %d 0]}
bind .browser <<OnStatusLinkChange>>   {set ::STATUS [lindex %d 0]}
bind .browser <<OnStatusChange>>       {set ::STATUS [lindex %d 0]}
bind .browser <<OnSetTitle>>           {wm  title  . [lindex %d 0]}
bind .browser <<OnProgressChange>>     {onProgress {*}%d}
## Other virtual events...
# bind .browser <<OnLocationChange>>     {}
# bind .browser <<OnSetDimensions>>      {}
# bind .browser <<OnStop>>               {}
# bind .browser <<OnStateChange>>        {}
# bind .browser <<OnFocusNextElement>>   {}
# bind .browser <<OnFocusPrevElement>>   {}
# bind .browser <<OnSetFocus>>           {}
# bind .browser <<OnRemoveFocus>>        {}
# bind .browser <<OnVisibilityChange>>   {}
# bind .browser <<OnShowTooltip>>        {}
# bind .browser <<OnHideTooltip>>        {}
bind .browser <<OnDocumentLoadInit>>     {onLoadInit   {*}%d}
bind .browser <<OnDocumentLoadFinish>>   {onLoadFinish {*}%d}
```

Callbacks (1)

```
proc onLoadInit {args} {
  puts "<<onLoadInit>>: $args"
  .progress state !disabled
  .progress configure -maximum 100 -value 0
};# onLoadInit

proc onLoadFinish {args} {
  puts "<<onLoadFinish>>: $args"
  .progress state disabled
  update idle
  after 1000 {set ::STATUS {}}
  testDOM
};# onLoadFinish

proc onLoad {} {
  .browser navigate $::URI
  onNewPage
};# onLoad
```

```
proc onBack {} {
  .browser back
};# onBack

proc onForward {} {
  .browser forward
};# onForward

proc onReload {} {
  .browser reload
  onNewPage
};# onReload
```

Callbacks (2)

```
proc onNewPage {} {
  if {[.browser can_go_back]} {
    .back state !disabled} else {.back state disabled}
  if {[.browser can_go_forward]} {
    .forward state !disabled} else {.forward state disabled}
};# onNewPage

proc onProgress {uri curUriProgress   maxUriProgress
                     curTotalProgress maxTotalProgress} {
  # puts "$curTotalProgress $maxTotalProgress"
  set curTotalProgress [expr {abs($curTotalProgress)}]
  set maxTotalProgress [expr {abs($maxTotalProgress)}]
  if {$maxTotalProgress >= $curTotalProgress} {
    .progress configure -maximum $maxTotalProgress \
                        -value $curTotalProgress
  }
};# onProgress
```

Result


Mozilla Developer Network
Reload
https://developer.mozilla.org/en-US/
Load
MOZILLA DEVELOPER NETWORK
A comprehensive, usable, and accurate resource for everyone developing for the Open Web.
Log in Become an MDN member
Search
SEARCH
powered by Google
MDN
WEB MOBILE ADD-ONS APPLICATIONS
DOCS COMMUNITY
Welcome to the Mozilla Developer Network, a.k.a.:
DEVMO
LEARN THE HISTORY BEHIND PROJECT DEVMO »
New to MDN?
Want to help make our network even better?
Here's how to get started »
Already a member?
Customize your profile & fine-tune your participation within MDN.
Log in »
HTML5
Mozilla continues to lead the charge in the adoption of HTML5 and other evolving open Web technologies. Find out more about the HTML5 spec and the latest enhancements in HTML5, CSS3 and other developer features for Firefox 4.
MORE »
MDN ON TWITTER
Correspondence on Twitter from around the Mozilla Developer Network.
@mozamo from ADD-ONS says:
RT @fligtar: AMO developers working hard on our big release this Thursday: http://www.flickr.com/photos/fligtar/4946014203/
4 weeks ago
@mozamo from ADD-ONS says:
New categories on AMO: Games & Entertainment, Shopping, and (mobile) Photos & Media http://mzl.la/c1125d
4 weeks, 1 day ago
@mozhacks from WEB says:
MDC has evolved... read more about the new MDN website: http://mzl.la/newmdnsite
1 month ago
@MozMobile from MOBILE says:
Fennec Alpha for Android and Nokia N900 is here! Enjoy Firefox Sync, Add-ons, Awesome Screen and more! http://bit.ly/aR4qMG
1 month ago
@planetmozilla from APPLICATIONS says:
The Mozilla Blog: Fennec Alpha Released for Android and Nokia N900: The Alpha release of the next major version of... http://bit.ly/b5LmNN
1 month ago
HIGHLIGHTS FROM AROUND THE NETWORK
DOCS
NEWS & UPDATES
TECHNOLOGY GLOSSARY FOR WEB DEVELOPERS
HTML
CSS
JavaScript
HTML5
DOM
SVG
Canvas
AJAX
Media Queries
WebGL
Web Storage
Audio
Video
Web Workers
Files
Drag and Drop
CSS Transforms
CSS Gradients
Security
IndexedDB
MOBILE
ADD-ONS
APPLICATIONS
https://developer.mozilla.org/en/HTML/HTML5

TkGecko widget subcommands

- *pathname* **focus_activate**
- *pathname* **focus_deactivate**
- *pathname* **navigate URI ?flags?**
- *pathname* **load**
- *pathname* **parse ?-base base_uri? ?-mime mime_type? ?--? Data**
- *pathname* **stop ?flags?**
- *pathname* **save ?-data_dir data_dir? ?-mime mime_type? ?-flags flags? ?-pflags persist_flags? ?-col wrap_col? ?--? uri**

TkGecko and the DOM tree

- Extensive support for accessing the DOM tree is provided
  - nsIDOMHTMLCollection
  - nsIDOMNodeList
  - nsIDOMNamedNodeMap
  - nsIDOMNode
  - nsIDOMElement
  - nsIDOMHTMLElement
  - nsIDOMAttr
  - nsIDOMDocument
  - nsIDOMHTMLDocument
  - nsIWebBrowserPersist
  - nsIDocumentEncoder
  - nsIWebBrowserSetup
  - nsIWebNavigation

Example: retrieving HTML

```
set dom      [$browser document]
set body     [$dom GetBody]
set content [$dom SerializeToString $body]
$body -delete
$dom  -delete
```

Example: retrieving formatted text

```
set dom      [$browser document]
set body     [$dom GetBody]
set encoder [$dom GetEncoder text/plain 0]
if {$encoder ne "NULL"} {
  $encoder SetNode $body
  set content [$encoder EncodeToString]
  $encoder SetNode NULL
} else {
  set content "NULL encoder!"
}
$body    -delete
$encoder -delete
$dom     -delete
```

Conclusions

- TkGecko embeds Gecko under Tk
  - Supported platforms: Windows and Linux
- Basic functionality already available
  - Displaying HTML
  - Manipulating DOM tree
- Thread safety is an issue
  - Not tested at all
- Some stability issues do exist
  - Random lockups after a number of pages

Thank you!