Wt, The Witty Web Toolkit

FOSDEM – Lightning talk

Koen Deforche – Wim Dumon – Pieter Libin
Wt Basics

• Library to develop web applications
  – Rich Internet Applications
  – Web 2.0

• Available for C++ (now), and Java (soon).

• Object Oriented Widget tree model
  – API is inspired by existing GUI libraries

• Provides abstraction of web technologies
  – CGI/HTML/AJAX/JavaScript/Forms/DOM
Hello Britney!
Hello Britney!

![Image of Britney Spears with a web browser window showing a song lyrics]

- Baby
- Hit me!
- One more time, Baby

http://witty.sf.net/
Is Britney Witty or a Cuty?

```cpp
HitmeWidget::HitmeWidget(QWidget *parent) :
    QWidget(parent)
{
    nameEdit_ = new QLineEdit("", this);
    hitButton_ = new QPushButton("Hit me!", this);
    oneMoreTime_ = new QLabel("", this);
    connect(hitButton_, SIGNAL(clicked()), this, SLOT(oneMoreTime()));

    QHBoxLayout *layout = new QHBoxLayout(this);
    layout->addWidget(nameEdit_);
    layout->addWidget(hitButton_);
    layout->addWidget(oneMoreTime_);
}

void HitmeWidget::oneMoreTime()
{
    oneMoreTime_->setText("One more time, " + nameEdit_->text());
}

HitmeWidget::HitmeWidget(WContainerWidget *parent)
    : WContainerWidget(parent)
{
    nameEdit_ = new QLineEdit("", this);
    hitButton_ = new QPushButton("Hit me!", this);
    hitButton_->setMargin((QLength(5), Left | Right);
    oneMoreTime_ = new WText("", this);
    hitButton_->clicked().connect(SLOT(this, HitmeWidget::oneMoreTime()));
}

void HitmeWidget::oneMoreTime()
{
    oneMoreTime_->setText("One more time, " + nameEdit_->text());
}
```
Browser and Desktop

```cpp
HitmeWidget::HitmeWidget(WContainerWidget *parent)
    : WContainerWidget(parent)
{
    nameEdit_ = new WLineEdit("", this);
    hitButton_ = new WPushButton("Hit me!", this);
    hitButton_->setMargin(Widgets::Margin(5), Left | Right);
    oneMoreTime_ = new WLineEdit("", this);
    hitButton_->clicked.connect(SLOT(this, HitmeWidget::oneMoreTime));
}

void HitmeWidget::oneMoreTime()
{
    oneMoreTime_->setText("One more time, " + nameEdit_->text());
}
```
Browser and Desktop

Wim Dumon
Standard Wt Widgets
## WLabel interface

### Public Member Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WLabel (WContainerWidget *parent=0)</code></td>
<td>Construct a <strong>WLabel</strong> with empty text and optional parent.</td>
</tr>
<tr>
<td><code>WLabel (const WString &amp;text, WContainerWidget *parent=0)</code></td>
<td>Construct a <strong>WLabel</strong> with a given text.</td>
</tr>
<tr>
<td><code>WLabel (WImage *image, WContainerWidget *parent=0)</code></td>
<td>Construct a <strong>WLabel</strong> with an image.</td>
</tr>
<tr>
<td><code>WFormWidget * buddy () const</code></td>
<td>Return the buddy of this label.</td>
</tr>
<tr>
<td><code>void setBuddy (WFormWidget *buddy)</code></td>
<td>Set the buddy of this label.</td>
</tr>
<tr>
<td><code>void setText (const WString &amp;text)</code></td>
<td>Set the label text.</td>
</tr>
<tr>
<td><code>const WString &amp; text () const</code></td>
<td>Get the label text.</td>
</tr>
<tr>
<td><code>void setImage (WImage *image)</code></td>
<td>Set the image.</td>
</tr>
<tr>
<td><code>WImage * image () const</code></td>
<td>Get the image.</td>
</tr>
</tbody>
</table>
Wt: Event Handling

Web browser

Events & Widget state changes

Wt application

- event handling: signal/slots
- business logic
- modify widget tree

Render Widget changes
Wt: Session Architecture

- Wt manages multiple sessions

- event handling: signal/slots
- business logic
- modify widget tree
Wt: Session Management

Two modes (runtime option):

● Dedicated processes
  – One process for each session
  – High session cost
  – Kernel-level isolation between sessions
    • Suitable for application like e-banking
  – Easy to debug

● Shared processes
  – Fixed number of processes handle all sessions
  – Multi-threaded request handling
  – Low session cost
  – Suitable for 'open' applications like google maps.
Wt: Deployment

- Linux & Unix:

  - Web server: apache, lighttpd
  - FastCGI module: mod_fastcgi, mod_fcgid
  - Wt session manager
  - Wt session processes

- Linux, Unix & Win32:

  - Web server: apache, lighttpd
  - Wt httpd (soon)
  - Wt session processes
Wt: Abstraction of web technologies

- Specify what the page should look like, not how it should be updated

- Specify event handling in C++

- Library tries to use AJAX and JavaScript, if not available uses plain HTML page refreshes

- Wt may optimize away server roundtrip
GMail Composer Example

We could have, but did not send the following email:

To: "Koen Deforce" <koen.deforce@gmail.com>
To: "Koen alias1" <koen.alias1@yahoo.com>
Cc: "Bartje" <jafar@hotmail.com>
Subject: "That's cool! Want to start your own google?"
Attachment: "uiloog.jpg" (image/jpeg), was in spool file: /tmp/wtx62RG5
Message body:
Witty message
Wt: sample applications

Rega DB Software
HIV Data Management and Analysis Environment
Clinical Edition

Login form
Login
Username: kdforc0
Password: ********
Login Help
Wt: sample applications

Rega DB Software
HIV Data Management and Analysis Environment
Clinical Edition

Login form
Something went wrong when validating your input, please verify your input and try again!

Login
Username: kdforc0
Password: ****

Login  Help
# Wt: sample applications

## Rega DB Software

**HIV Data Management and Analysis Environment**

**Clinical Edition**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Patient ID</th>
<th>Name</th>
<th>Surname</th>
</tr>
</thead>
<tbody>
<tr>
<td>PT</td>
<td>9903</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>9889</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>9881</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>9861</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>9859</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>9856</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>9855</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>9760</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>9728</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PT</td>
<td>9674</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

FOSDEM 2007  http://witty.sf.net/  Wim Dumon
Wt: development status

- Wt 2.0.0 (C++): stable
- JWt (Java)
  - Runs in Java Application Servers
  - First release planned in March.
Wt: developers

• Koen Deforche
  – Initial developer, maintains C++ Wt.

• Wim Dumon
  – Documentation
  – Windows port

• Abdiel Janulguide
  – First template based signal/slot implementation

• Pieter Libin
  – Port to Java, maintains Java JQt.
Wt on the web

Availability, documentation, etc...

http://witty.sf.net
Thank you!