Designing and developing portable large-scale JavaScript web applications within the Experiment Dashboard framework

CHEP New York 2012

J. Andreeva, I. Dzhunov, E. Karavakis, L. Kokoszkiewicz, M. Nowotka, P. Saiz, D. Tuckett
Contents

• Traditional Web UIs
  – Overview & Example
  – Pros & Cons
• JavaScript Web UIs
  – Overview & Example
  – Pros & Cons
  – Cons fixed
• False Start
• Our Approach
  – Overview
  – Technology cocktail
  – Configurable UIs
• Xbrowse
  – Overview
  – Data flow
  – Adaptors
• Hbrowse
• SSB
• Conclusions
Traditional Web UIs

- Server-side view generation

- Multiple page interface
  - Hypertext-style
  - Full page loaded for each user interaction

- Example: ATLAS DDM Dashboard 1.0
  http://dashb-atlas-data.cern.ch/dashboard/request.py/site
Traditional Web UIs: Example

**Links**
- Static plots

**Overview**
- Activity Period
- Activity in Last Hour
- Activity in Last 4 Hours
- Activity in Last 24 Hours
- Activity in Last 7 Days
- Activity in Last 30 Days
- Activity in ...

**Selected Activities**
- Production
- T0 Export
- Functional Test
- User Subscriptions
- Staging
- Data Consolidation
- Data Brokering
- Group Subscriptions

**Selected Clouds**
- CA Cloud
- CERN Cloud
- DE Cloud
- ES Cloud
- FR Cloud
- IT Cloud
- NL Cloud
- TW Cloud
- UK Cloud
- US Cloud

**Activity Summary (2011-04-05 03:10 to 2011-04-05 07:10 UTC)**
Click on the cloud name to view list of sites

<table>
<thead>
<tr>
<th>Cloud</th>
<th>Efficiency</th>
<th>Throughput</th>
<th>Successes</th>
<th>Registrations</th>
<th>Errors</th>
<th>Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>100%</td>
<td>106 MB/s</td>
<td>7482</td>
<td>115</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CERN</td>
<td>100%</td>
<td>92 MB/s</td>
<td>13418</td>
<td>120</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>DE</td>
<td>96%</td>
<td>450 MB/s</td>
<td>18359</td>
<td>781</td>
<td>18360</td>
<td>0</td>
</tr>
<tr>
<td>ES</td>
<td>32%</td>
<td>60 MB/s</td>
<td>1550</td>
<td>87</td>
<td>1950</td>
<td>4233</td>
</tr>
</tbody>
</table>
Traditional Web UIs: Pros & Cons

• Benefits
  – Cross-browser compatibility
  – Modularity for team-working (page per feature)
  – Bookmarkable URLs
  – Search-engine friendly
  – Accessibility friendly
  – Expertise

• Issues
  – Low interactivity
    • User cannot easily customize data visualization
  – Slow loading
    • Full page loaded for each user interaction
    • Server-side call for new view of same data

• Issues hinder novel analysis of data
• Issues cannot be fixed within the paradigm
JavaScript Web UIs

- Client-side view generation
- Single page interface
  - GUI-style
  - Data loaded on-demand
- Example: ATLAS DDM Dashboard 2.0
  http://dashb-atlas-data.cern.ch/ddm2/
JavaScript Web UIs: Example

Accordian
Sliders
Dynamic plots
Movable Panels
JavaScript Web UIs: Pros & Cons

• Benefits
  – High interactivity
    • User can easily customize data visualization
    • Leverage 3rd-party plugins
  – Fast loading
    • Data loaded on-demand
    • Client-side call for new view of same data

• Benefits facilitate novel analysis of data

• Issues
  – Browser incompatibilities
  – Slow client-side rendering
  – Lack of modularity for team-working (single-page)
  – Non-bookmarkable URLs
  – Not search-engine friendly
  – Not accessibility friendly
  – Lack of expertise

• Issues can be fixed within the paradigm
JavaScript Web UIs: Cons Fixed

- **Fixed Issues**
  - Browser incompatibilities
    - Standards compliance: [Acid3](#)
    - JavaScript libraries: [jQuery](#)
  - Slow client-side rendering
    - Faster JavaScript engines
    - Faster processors
  - Lack of modularity for team-working
    - Client-side MVC frameworks
    - View object per feature
  - Non-bookmarkable URLs
    - URL hash management

- **Issues still in progress**
  - Not search-engine friendly
  - Not accessibility friendly
  - Lack of expertise
False Start

- Our early attempts at JavaScript Web UIs
  - No client-side MVC framework
  - State / data stored in form inputs / DOM
  - No common JavaScript library choice
  - Limited use of 3rd-party plugins
  - Non-modular, highly-coupled “spaghetti” code

- Difficult to extend or maintain
Our Approach

- **Client-side MVC**
  - **Model**
    - state – user selected filters
    - data – application data from server
  - **Views**
    - update / reflect state
    - request data from web API
    - update / reflect data
  - **Controller**
    - initialize model and views
    - synchronize state with URL

- **Web API**
  - AJAX requests
  - JSON responses
  - Data source decoupling

- **Client-side MVC + Web API** → Configurable UIs
ES

Technology Cocktail

• jQuery Core & UI
  – DOM manipulation
  – UI widgets
  – Popular! →

• Plugins
  – URL hash: BBQ
  – MVC events: Backbone
  – Templating: Handlebars
  – Plotting: Highcharts
  – Tables: DataTables
  – Utilities: Underscore
  – …
Configurable UIs

- Skeleton UIs adaptable to many use cases
- Design features
  - Client-side MVC
    - Configurable model
    - Pluggable views
  - Pre-defined views
  - Server interaction via web API
    - UI treated as external application
    - Data source decoupling
- Building new applications for new use cases
  - Define model state default values
  - Combine / adapt pre-defined views
  - Few dozen lines of code
- 2 configurable UIs
  - Hbrowse for hierarchical data
  - Xbrowse for matrix data (in more detail next)
Xbrowse: Overview

- Xbrowse configurable UI
- Client-side MVC

- Pre-defined view objects
  - sidebar, controls, tabs, matrix, plots, toolbars, ...
  - particularly suited to matrix data

- Brief documentation: [http://tinyurl.com/xbrowse](http://tinyurl.com/xbrowse)
Xbrowse: Pre-defined Views

- **Titlebar**: Display of the dashboard's title.
- **Toolbar**: Contains options for filtering and adjusting the view.
- **Matrix**: Main data matrix displayed in the center.
- **Sidebar**: Sidebar with options to select sources and destinations.
- **Plots**: Graphs showing data trends.
- **Samples**: Detailed samples of data error.
Xbrowse: Data Flow

Client-side MVC + Web API

User -> View: input
View -> Model: update state
Model -> URL: update state
URL -> Web API
Web API -> Model: change
Model -> AJAX
AJAX -> JSON
JSON -> update data
update data -> change
change -> Model
Model -> View
View -> User: change
User -> View
View -> Model: change
Model -> Web API
Web API -> URL
URL -> Web API

www.cern.ch/it
Xbrowse: Model Adaptor

```javascript
/* Model Adaptor */
xmvc.adaptor = $.extend(true, xmvc.adaptor, {
  model: {
    post_init: function(model) {
      // define state defaults
      model.state.opts.defaults = {
        date:{interval:240, from:'', to:''},
        vo:['atlas', 'cms', 'lhcb'],
        ...
      };
      // define state types for automatic coercion
      model.state.opts.types = {
        date:{interval:'number', from:'date', to:'date'},
        ...
      };
      // define state change side-effects
      model.state.on('change:date', function(e) {
        this.set({plot:{bin:null}}); //reset plot.bin
      });
    }
  }
});
```
Xbrowser: View Adaptor

/* View Adaptor */
xmvc.adaptor = $.extend(true, xmvc.adaptor, {
  view:
    create: function() { // define view plugins
      return [
        new ViewDates(), view_matrix, ...
      ];
    }
};

var view_matrix = new ViewMatrix({
  ctx:'#t_tab_matrix', // DOM ID
  ajax:
    url:'/dashboard/request.py/transfer-matrix', // AJAX URL
    params:function(state, data, opts) { // AJAX parameters
      return {
        from_date:state.date.from,
        to_date:state.date.to,
        vo:state.vo,
        ...
      },
      template: ... // template (optional)
    }
});
(new xmvc.Controller()).init(); // initialize controller
Xbrowse: Matrix View

ATLAS DDM DASHBOARD 2.0 M3.2
TRANSFER DASHBOARD (2012-02-10 21:00 to 2012-02-11 01:00 UTC LOCKED)

Matrix View

Total SOURCES

CA+ 99 % 254 MB/s 99 % 256 MB/s 81 % 605 MB/s 81 % 179 MB/s 87 % 807 MB/s 81 % 211 MB/s 34 % 52 MB/s 73 % 208 MB/s 83 % 624 MB/s
CERN+ 99 % 472 MB/s 100 % 302 MB/s 99 % 142 MB/s 100 % 121 MB/s 96 % 168 MB/s 100 % 568 MB/s 93 % 25 MB/s 100 % 60 MB/s 96 % 22 MB/s
DE+ 80 % 277 MB/s 100 % 72 MB/s 77 % 27 MB/s 98 % 100 MB/s 98 % 413 MB/s 100 % 92 MB/s 70 % 92 MB/s 100 % 11 MB/s 100 % 44 MB/s
ES+ 53 % 122 MB/s 0 % 0 MB/s 0 % 0 MB/s 0 % 0 MB/s 3 % 107 MB/s 0 % 0 MB/s 12 % 157 MB/s 100 % 57 MB/s 91 % 10 MB/s
FR+ 59 % 386 MB/s 59 % 386 MB/s 59 % 386 MB/s 59 % 386 MB/s 59 % 386 MB/s 59 % 386 MB/s 59 % 386 MB/s 59 % 386 MB/s 59 % 386 MB/s 59 % 386 MB/s
IT+ 77 % 209 MB/s 11 % 14 MB/s 11 % 14 MB/s 11 % 14 MB/s 11 % 14 MB/s 11 % 14 MB/s 11 % 14 MB/s 11 % 14 MB/s 11 % 14 MB/s 11 % 14 MB/s
ND+ 80 % 7 MB/s 100 % 35 MB/s 100 % 64 MB/s 100 % 72 MB/s 100 % 2 MB/s 100 % 4 MB/s 100 % 6 MB/s 100 % 81 MB/s 100 % 2 MB/s
NL+ 90 % 190 MB/s 20 % 10 MB/s 100 % 10 MB/s 100 % 10 MB/s 100 % 0 MB/s 100 % 0 MB/s 100 % 0 MB/s 100 % 0 MB/s 100 % 0 MB/s
TW+ 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s
UK+ 79 % 241 MB/s 99 % 29 MB/s 100 % 138 MB/s 100 % 98 MB/s 100 % 95 MB/s 100 % 100 MB/s 100 % 68 MB/s 100 % 87 MB/s 100 % 89 MB/s 100 % 96 MB/s
US+ 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s 10 % 10 MB/s

TOTAL DESTINATIONS

ERROR SAMPLES: CA --> FR


Total /439

436
2
1
Hbrowse Overview

- Hbrowse configurable UI
  - JavaScript client framework for hierarchical data visualization
  - Highly configurable environment
  - Bookmarking support
  - Highcharts and Google charts support
  - Server-side independent
  - Infinite number of data levels

- Full documentation: [http://www.hbrowse.net/](http://www.hbrowse.net/)

- See poster: [152] Hbrowse framework
Example Applications

- Hbrowse for hierarchical data
  - ATLAS Task Analysis
    https://dashb-atlas-prodsys-prototype.cern.ch/templates/task-analysis/#timerange=lastMonth&demo=on
  - CMS Interactive View
  - ATLAS Dataset distribution
    http://dashb-atlas-task.cern.ch/templates/pandadatasetdist/
  - ...

- Xbrowse for matrix data
  - ATLAS DDM Dashboard
    http://dashb-atlas-data.cern.ch/ddm2/
  - WLCG Transfers Dashboard
    http://dashb-wlcg-transfers.cern.ch/ui/
    (See poster: [289] Providing WLCG Global Transfer Monitoring)
  - ...

(See poster: [289] Providing WLCG Global Transfer Monitoring)
Site Status Board Overview

- **Site Status Board**
  - Monitoring of site and service status
  - Deployed for four main LHC experiments
  - Heavily used by CMS and ATLAS
    - [http://dashb-ssb.cern.ch](http://dashb-ssb.cern.ch)
    - [http://dashb-atlas-ssb.cern.ch](http://dashb-atlas-ssb.cern.ch)

- **Developed using common approach**
  - Client-side MVC
  - Web API
  - jQuery + plugins

- **Uses Backbone as MVC framework**
  - [http://documentcloud.github.com/backbone/](http://documentcloud.github.com/backbone/)

- **See poster:**
  - [514] Recipe for a successful application
Conclusions

- It is possible to build sustainable large-scale JavaScript applications with limited resources
- Our approach works for us …
  - Client-side MVC + Web API
  - jQuery + plugins: BBQ, Highcharts, DataTables
  - Configurable UIs
- … but it could be improved
  - Standardize on common MVC framework
  - Fix remaining issues:
    - search engine support
    - accessibility support
  - Testing
  - Documentation
  - Mobile support
  - Optimization
  - …
Thank you for your time

http://dashboard.cern.ch/

dashboard-support@cern.ch