

OpenMRS 2.0 User Interface Redesign

Darius Jazayeri
djazayeri@pih.org
darius@openmrs.org

Goals

- Build a **cool** and **effective** user interface
- **Easy** to program
- But also **flexible**
 - **Role-based**
 - Can be **reconfigured** by the implementer
 - Modules can **override** behavior
 - Configurable by the **end-user** (maybe in 3.0)

UI Framework Needs (for devs)

- Dynamic role-based handling of page requests
- Easy to build pages out of a common library of useful components
 - Reuse these components in modules
 - Reuse them for completely new UIs
- Good patterns that are easy to copy

Dynamic Page Handling

- **Dynamic** page request handler that determines what page to display to user
- Based on **Role**, User preferences, etc.
- The **administrator** maps what page different roles see for homepage, patient page, etc.
- Core will provide **good default pages** for standard roles, but administrators can **create their own pages** (a la Google homepage or JIRA dashboards)
- Eventually let end-users customize pages

Dynamic Page Handling

<https://xyz.org/patient.page?patientId=123>

If your role is
'Clinician'

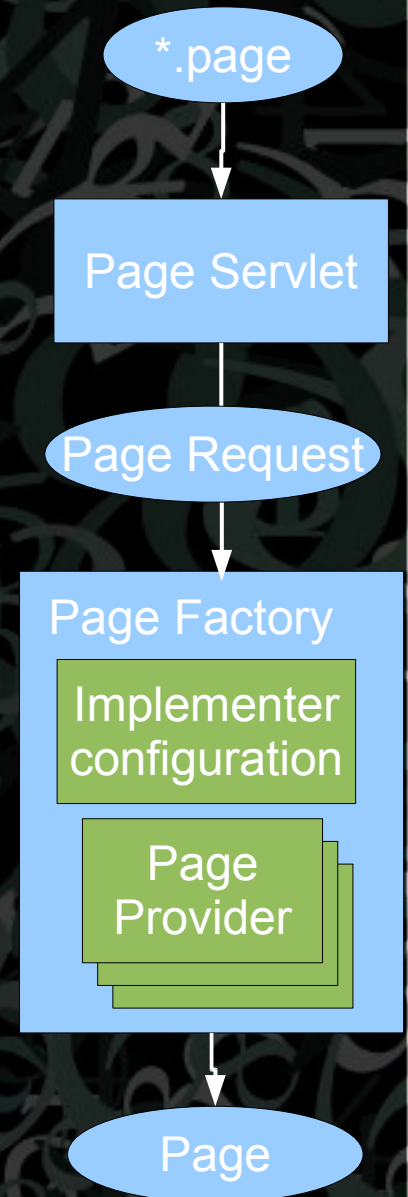
clinician-patient-page

| | | | |
|------------|--------------|---------------|----------|
| Bob Smith | Alice Jones | Charles Brown | |
| Bob SMITH | ID: ABC-1234 | | |
| Overview | Date | Type | Location |
| | 01/01/2010 | ... | ... |
| Encounters | 01/01/2010 | ... | ... |
| Allergies | 01/01/2010 | ... | ... |

If your role is
'Data Clerk'

pihrwanda-clerk-patient-page

| | |
|-------------|-----------------|
| Bob SMITH | ID: ABC-1234 |
| Overview | Missing weight! |
| Form Entry | |
| Data Errors | |



Page Fragments

- Pages are composed of Page Fragments (which may be recursively composed of other Page Fragments)

The screenshot shows a web application interface with several annotated components:

- Site Header:** A red-bordered bar at the top containing four fragments: "Logo" (Your Logo), "Login info" (Hi Darius | Log Out), "Notification bar" (2 messages, 5 likes, power icon), and "Shortcuts" (Find patient).
- Data Clerk Patient Header:** A blue-bordered bar containing patient tabs (1234 Jones Darius, 4567 Smith Bob), a profile card for "Jones, Darius" (28 years | TB Program), and status boxes ("Male in PMCT Program", "No Height Recorded").
- Data Clerk Patient Page:** A green-bordered bar at the bottom containing a navigation menu with tabs: "Basic Information", "Encounters", "Orders", "Programs", "Forms", and "Overview".

Annotations with arrows point to these sections from the left side of the slide.

Page Fragments

- Recursively composed
- Page Fragments from **multiple sources**
 - OpenMRS Core
 - Modules
 - User-created (in DB, like HTML Forms)

| | |
|---------|---------------------------------|
| | |
| Tab One | Vertical tabs (OpenMRS core) |
| Tab Two | |
| More... | |
| | |

| (Data Entry Statistics module) | | |
|--------------------------------|------------------|---------------|
| When | Patients Created | Forms Entered |
| Today | 3 | 15 |
| Last 7 days | 16 | 72 |

Page Fragments

- Recursively composed
- Multiple Sources
- Encapsulate **reusable** functionality
 - Infrastructure
 - vertical tab panel
 - box with title
 - Functional widgets
 - encounter list
 - find a patient

An Example Page Fragment

- Encounter List

Default usage on a patient page

Also usable on the home page

| Encounters | | | |
|-------------|-------------|------------|----------------|
| Date | Type | Location | Provider |
| 01/Sep/2010 | TB Followup | Cange PTJW | Jean Toussaint |
| 02/Aug/2010 | TB Followup | Cange PTJW | Jean Toussaint |
| 01/Jul/2010 | TB Intake | Cange PTJW | Jean Toussaint |
| ... | ... | ... | ... |

| Today's Encounters | | | | |
|--------------------|-------------|-------------|------------|----------------|
| Patient | Date | Type | Location | Provider |
| Alice Marie | today 07:15 | TB Followup | Cange PTJW | Jean Toussaint |
| Bob Smith | today 07:32 | TB Followup | Cange PTJW | Jean Toussaint |
| Claire Fleur | today 07:41 | TB Intake | Cange PTJW | Jean Toussaint |
| ... | ... | ... | ... | ... |

```
<fragment name="encounter-list"/>
```

```
<fragment view="encounter-list"  
  controller="todays-encounters"  
  title="Today's Encounters"  
  showPatientColumn="true"/>
```

PS- you should be able to
compose page fragments
without really programming

Good Patterns, Easy to Copy

- If it's easy to do things right, people will do them right.
- Provide consistent formatting of domain objects
- Provide reusable CSS
- Provide reusable Javascript functions
- Provide good examples of how to use these
- Provide a style guide

Dynamic Page Mapping

```
// Framework allows you to plug in PageRequestMappers  
public class PageFactory {
```

```
    ...  
    @Autowired  
    List<PageRequestMapper> requestMappers;  
    ...  
    private void mapInternalPageName(PageRequest request) { ... }  
    ...  
}
```

```
// One simple implementation out of the box
```

```
public class HibernateRoleBasedPageRequestMapper  
    implements RoleBasedPageRequestMapper {  
    ...  
    public String mapRequest(PageRequest request) {  
        String pageName = request.getPageName();  
        Role role = request.getSession().getRole();  
        if (role == null) return pageName;  
  
        RoleBasedPageRequestMapping mapping = getMappingForPageAndRole(pageName,  
            role);  
        return mapping == null ? null : mapping.getMapping();  
    }  
    ...  
}
```

Groovy Templates as views

```
<%  
  ui.decorateWith("standardPage")  
%>  
  
<div class="column col60">  
  
  <h4>${ ui.includeFragment("welcomeMessage") }</h4>  
  
  ${ ui.includeFragment("findOrCreatePatient") }  
  
</div>  
  
<div class="column col40">  
  
  ${ ui.includeFragment("dataEntryStats",  
    [ decorator: "widget",  
      decoratorConfig: [title: ui.message("dataEntryStats.mine.title")] ] ) }  
  
</div>
```

Fragment Controllers

```
public class FindPatientFragmentController {  
  
    public void controller(FragmentConfiguration config, FragmentModel model) {  
        String query = null;  
        try {  
            query = config.getAttribute("query").toString();  
        } catch (Exception ex) {}  
  
        if (query == null) {  
            model.addAttribute("patients", new ArrayList<Patient>());  
            model.addAttribute("query", "");  
        } else {  
            List<Patient> patients = Context.getPatientService().getPatients(query);  
            model.addAttribute("patients", patients);  
            model.addAttribute("query", query);  
        }  
    }  
  
    public ObjectResult search(FragmentActionRequest request) throws Exception {  
        String query = request.getRequiredParameter("query", String.class, "query");  
        List<Patient> patients = Context.getPatientService().getPatients(query);  
        return new ObjectResult(ObjectResult.build(request.getUiUtils(),  
            patients, "patientId", "patientIdentifier", "personName",  
            "age", "gender", "birthdate"));  
    }  
}
```

Timeline

- In 2 weeks: Developers start dabbling by writing new pages (e.g. "Data Clerk patient dashboard")
- In 6 weeks: Technology demo: show a skeletal version of the new UI to implementers
- ??: OpenMRS 2.0 will be a reimplementations of 'all' current functionality and workflow in the new UI framework, plus a couple new features (e.g. Role-based homepages)