Customizing ERModernLook Applications

David Holt
CSCW Systems Corporation

Overview

- Context
- D2W is awesome! - David LeBer WOWODC 2011
- Developer workflow when using D2W
- Brand new demo has examples of everything I'll talk about
- Strategies and tips for development with ERModernLook

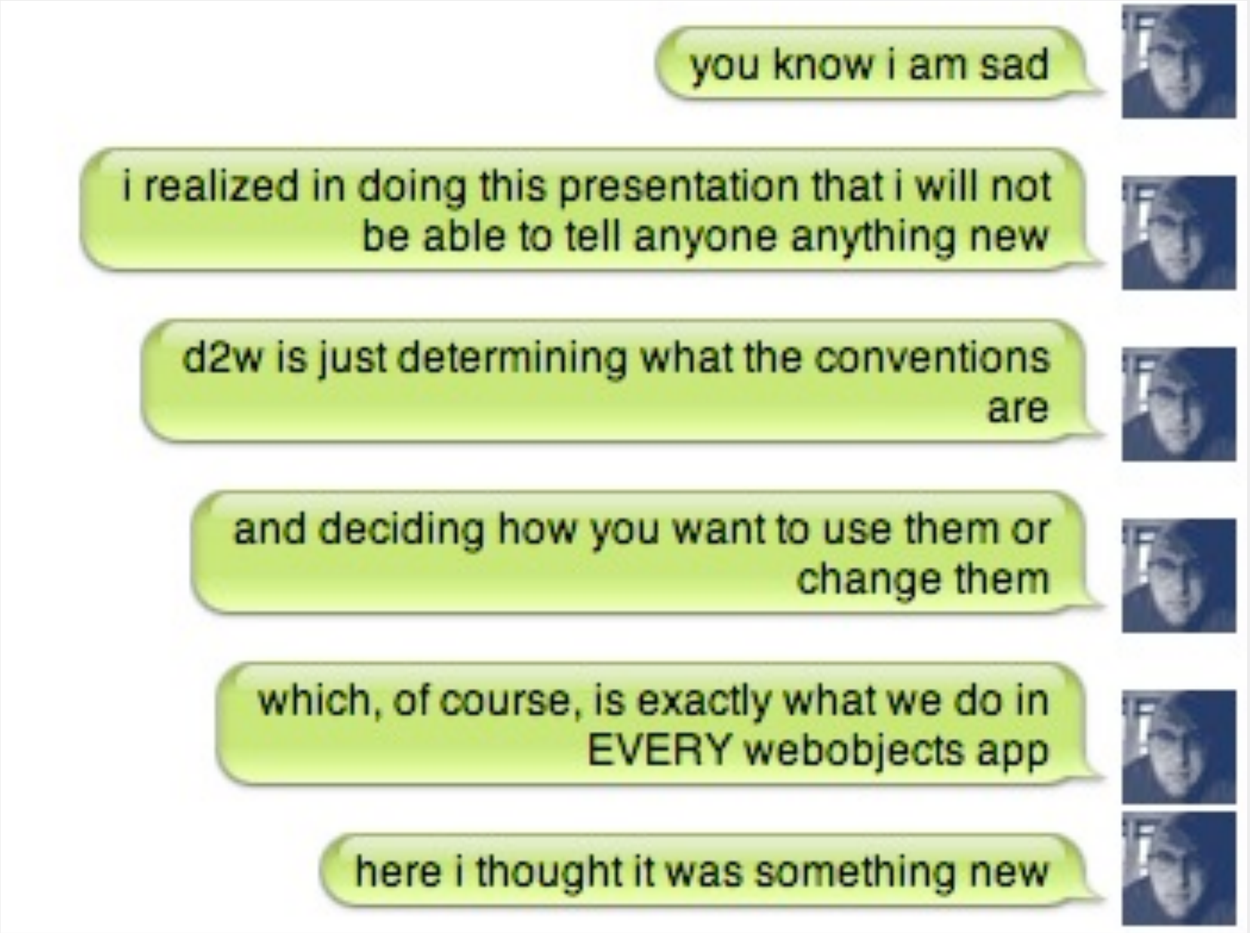
Context

- Attended WWDC in 2000 and 2002 when WO still had heavy presence and saw the WOnder release presentation
- Developer with WO since 2003
- Lone developer with additional support when necessary
- Started with WOnder - my preference is to use what exists
- Our applications are tools that attempt to direct user behaviour as much as possible - form based, no advertising, simple and utilitarian, minimize training

Our experience with ERModernLook

- Made the commitment to start using D2W about a year ago
- Core functionality of main app took about a month to get going
- Simple data capture applications take about two weeks start to finish
- Fast prototyping and iterations for the clients
- We have found ERModernLook is ready for client facing applications “out of the box”

Overview


you know i am sad
i realized in doing this presentation that i will not be able to tell anyone anything new
d2w is just determining what the conventions are
and deciding how you want to use them or change them
which, of course, is exactly what we do in EVERY webobjects app
here i thought it was something new

“Just” Determining what the Conventions Are

- Your job is to understand what default behaviour ERModernLook has and what frameworks are used to provide that behaviour
- There is default behaviour where you may not expect (CSS)
- I am NOT going to talk about creating a new look, because then you are creating your own conventions/default behaviours for your applications to follow

Hopefully Conventions are "Best Practices"

- proper subclassing of pages and components
- separate business logic from your view and controller logic
- correct editing context handling (EC per page, peer EC for edit pages so changes can be discarded when the user Cancels)
- all string labels in the UI are localized
- data entered in forms is validated / errors displayed to the user
- PageWrapper/WOComponentContent + ERXNavigationMenu

Over Time, Conventions Change

1. The days only Chuck remembers (you will likely see this in “legacy” code)

```
Project eo = new Project();
ec.insertObject(eo);
// manually set up your mandatory attributes and relationships
```

2. The days that everyone remembers (you will likely see this in a lot of code)

```
EOUtilities.createAndInsertInstance(ec, "Project");
// manually set up your mandatory attributes and relationships
```

3. The current convention (you will likely see more adoption going forward)

```
Person user = _session._user;
Project.createProject(ec, "Write Slides", user); // constructor sets mandatory attributes and relationships
```

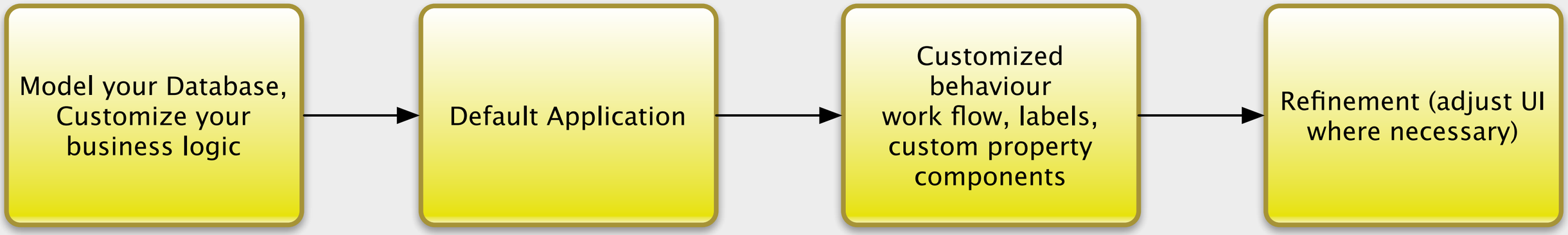
A brief segue

- What is D2W? See wiki for list of resources
- I found it helpful to hear it referred to as an expert system
- What does it know?
- How to build web application front ends to data stores
- It has been DESIGNED to take care of most of your front end development tasks

ERModernLook is not just D2W

- ERXNavigation
- Localization
- Validation
- Object Oriented CSS
- Ajax

Steps for Developer



Steps for the Developer (Model)

- Create or supply a Model
- how you name your attributes and relationships affects their default labels
- user info dictionaries
- default behaviour with respect to your model is determined by rules

Steps for the Developer (Login)

- Decide whether you need login handling
- If you do, you can handle where your user lands after login
- If you don't, then your Main component can offer whatever functionality you choose
- ERXD2WDirectAction (maybe you want to give direct access to your app via email, web link, etc.)

Steps for the Developer (Page)

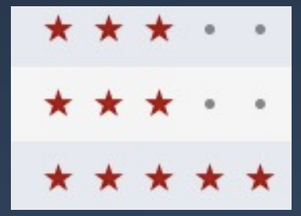
- Choose an entity to work with
- Decide on a task. Maybe a Sub-task (wizard, tab, group, csv)
- Default page or a named PageConfiguration - default naming conventions
- Decide on template for the page (i.e. List, GroupingList)
- Labels (you can use variables too)
- These behaviours are set up with rules

Steps for Developer (Page cont.)

- Is default or modified behaviour of the page template adequate?
- Make new template for this page?
- displayPropertyKeys - what attributes and relationships will you choose to display?
- Are there permission-based considerations for things you're displaying on the page?
- Modify Rules, Templates, Business Logic

Steps for Developer (Attributes)

- Are the default components adequate? Do any customizable characteristics need to be modified? (i.e. height, width)
- How do you plan to label them? (I strongly recommend Localizable.strings even if you are using only one language)
- Any special characteristics apply? Custom property level component?



Steps for Developer (Relationships)

- Which component applies?
- ERD2WEditToManyRelationship is worth the effort to learn
- Likely need to modify default characteristics depending on your number of options (checkboxes, browser, radio)
- What do you plan to use to label the items in the relationship? (userPresentableDescription, custom business logic, a specific attribute on the related entity?)

ERModernLook (navigation)

- Set up your navigation tab for the chosen entity/page
- Nice convention is something like Tab-Entity, SubTab-Task1, SubTab-Task2

- Work out labels for the Localizable.strings file
- Set up your rules for NavigationState. Navigation State is controlled by the page you load.

ERModernLook (navigation cont.)

- ERXNavigation allows you to use up to three levels of tabs
- You can do all sorts of fancy things (see Ramsey's notes from HelloD2W NavigationMenu.plist)
- You can vary the configuration of tabs and sub-tabs based on stored values (tabs for no login, tabs after login)
- I use ERXThreadStorage in my session (i.e. based on a whether a user is logged in, whether a user has permission to see a given tab configuration)

Navigation configuration example (session class)

```
 public String navigationRootChoice() {
  Person user = (Person) user();
  if(user != null ) {
    if(user.isAdmin()==true) {
      return "adminuser";
    }
    return "home";
  }
  return "none";
}
```

Navigation configuration example (NavigationMenu.plist)

```
   name = Root;
directActionClass = DirectAction;
directActionName = default;
children = "session.navigationRootChoice";
childrenChoices = {
   home = (
      Instructions, Blog,
   );
   adminuser = (
   Instructions, Admin,
   );
   none = (
      PublicBlog,
   );
};
```

ERModernLook (pageflow)

- Page creation logic for each tab goes in the MainNavigationController.
- After creating a few pages, workflow issues begin to appear
- You may not need a tab for every page (i.e. some tasks can have Navigation State left as the Tab for the Entity)

ERModernLook (pageflow cont.)

- You can use the “actions” key to set up your right and left actions in Lists
- You can use ERDControllerButton in rows for additional, custom actions
- For page level actions (such as download excel version) you can use ERDActionBar which appears in the BottomActions section

How to manage the complexity?

- I have created a set of custom EOModelDoc templates that give you generated worksheets to record some of the decisions you are making at the entity/task level
- More importantly, the worksheets can help you communicate to your client which property keys will be displayed on which page, the labels that will be used, the label for the page)
- Of course you can also just build and iterate which has a very low overhead once you get comfortable with what you are doing

Models

All Entities
SimpleBlog

SimpleBlog

Entities

BlogCategory
BlogCategoryEntry
BlogEntry
Person
Role
XPersonRole

```
SimpleBlog
pageConfigurationName:
displayNameForPageConfiguration:
navigationState:
Task:
Entity: BlogEntry


        propertyKey     displayNameForProperty  componentName


                Attributes      ---     ---
                    body    -       -
                    lastModifed     -       -
                    timestampCreation       -       -
                    title   -       -


                Relationships   ---     ---
                    categories      -       -
                    person  -       -


                displayPropertyKeys / tabSectionContents
```

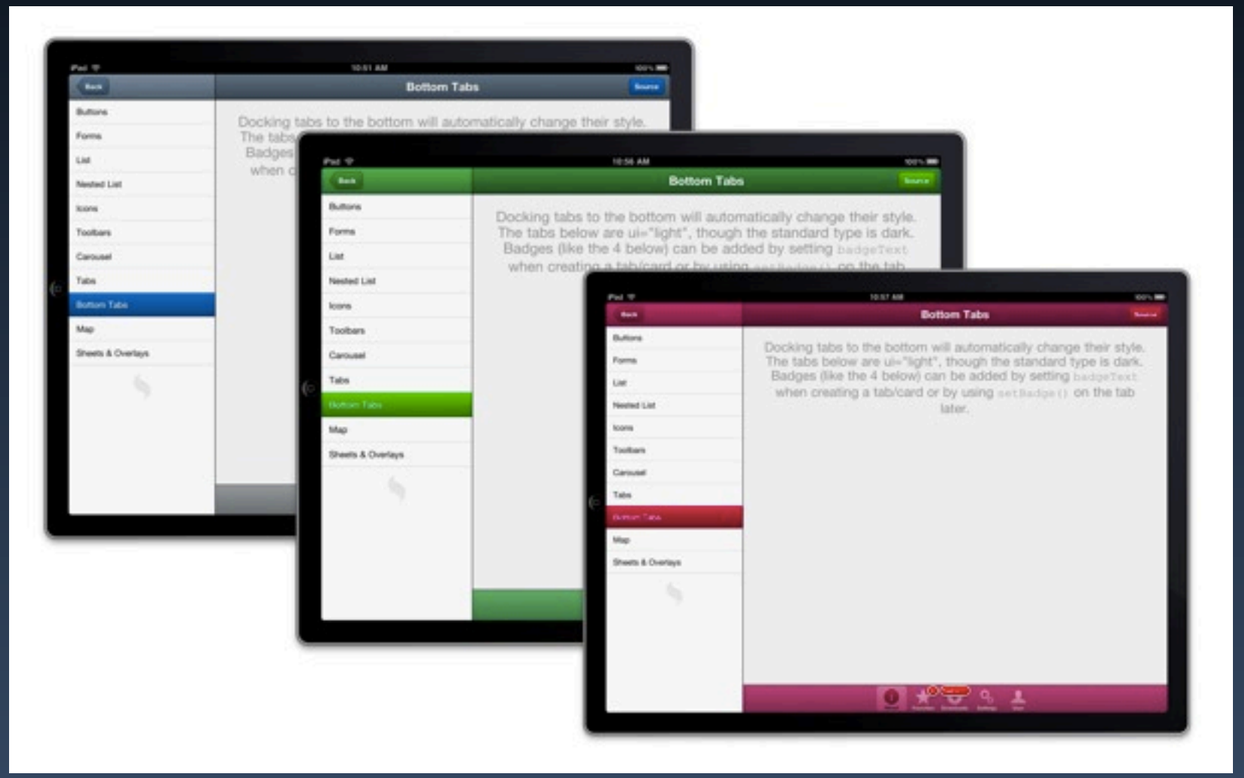
- ☐ pageConfigurationName: createBlogEntry
- ☐ displayNameForPageConfiguration: "Create Blog Entry"
- ☐ navigationState: Blog.CreateBlog
- ☐ Task: create
- ⊟ Entity: BlogEntry
  - ⊟ propertyKey
    - ⊟ **Attributes**
      - ☑ body
      - ☐ lastModifed
      - ☑ timestampCreation
      - ☑ title
    - ☑ **Relationships**
      - ☑ categories
      - ☑ person
    - ☐ **displayPropertyKeys / tabSectionContents**
      - ☐ (
        "person.name",
        "timestampCreation",
        "title",
        "body",
        "categories"
)

displayNameForProperty	componentName
—	—
- Body	- ERD2WEditLargeString
-	-
- Timestamp Creation	- D2WDisplayDate
- Title	- ERD2WEditString
—	—
- categories.categoryDescription	- ERD2WEditToManyRelationship / checkbox
- person.name	- ERD2WMultiItemCustomComponent / DisplayStrings

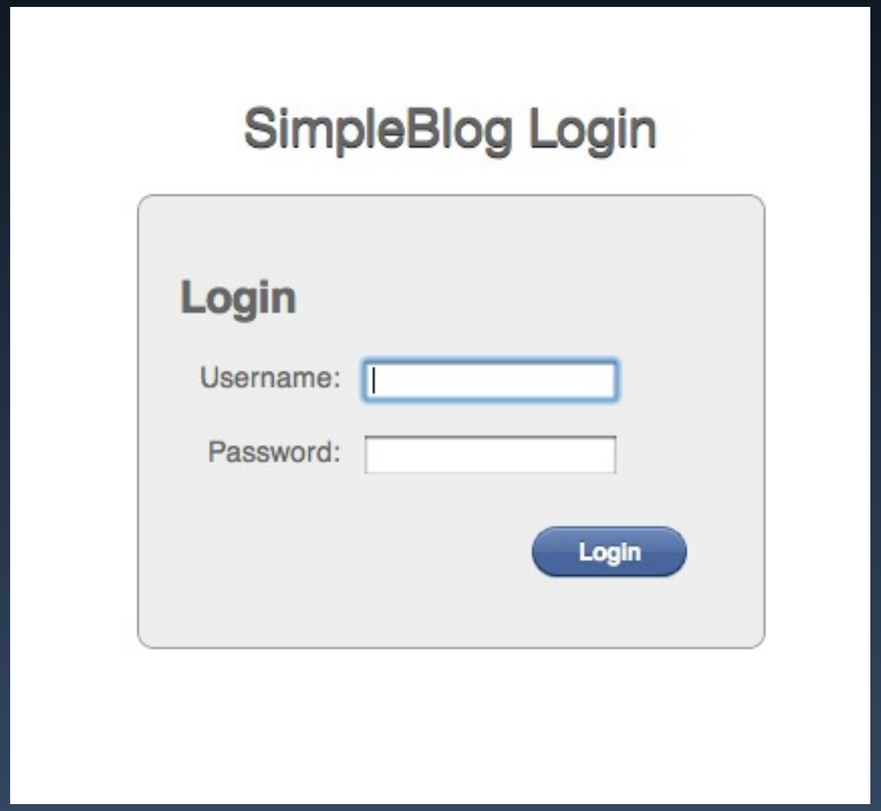
CSS

- ERModernLook makes extensive use of CSS
- You can use the default skin and easily add a custom stylesheet to your application so you don't have to start from scratch
- You can also copy the skin framework entirely and then change the stylesheets as much as you like
- I have begun the work of generating the skin framework with Sass and Compass. This allows the use of variables in stylesheets and the opportunity to quickly create entirely new skins rapidly

Sencha Touch as an Example



Demo of SimpleBlog



Suggestions and Tips

- Take it slow, practice practice practice- Zen and the Art of Archery
- Limit yourself to existing templates/components at the start (don't forget ERD2W, D2W includes many components)
- Learn simple rules first (displayPropertyKeys, show/hide buttons)
- Learn how to configure embedded page configurations such as ERMODEditRelationshipPage (this is among my most used components)

More Suggestions and Tips

- One of my favourite threads all time was a question about how to create a button for D2W that would create an email and send it to users. The first response was for people to dive into how to create a custom component, but then they realized that the action could be triggered in the business logic.
- When it comes to creating new components, think it through carefully. You'll be surprised how much can be moved into business logic using methods such as...

ERXGenericRecord

- willInsert
- didInsert
- willUpdate
- didUpdate
- willDelete
- didDelete

Additional things that could have been covered

- ERSelenium (functional testing is crucial since rule changes are relatively unpredictable until run time). See WOWODC presentation from 2008 on testing.
- Migrations
- ERAttachment
- BugTracker - especially the Factory class
- ERCoreBusinessLogic

Resources

- http://wiki.objectstyle.org/confluence/display/WO/Direct+To+Web+%28D2W+and+ERD2W%29
- https://github.com/daveeed/SimpleBlog

Q&A