The Next Generation of Eclipse: e4

Mike Milinkovich
Executive Director
Eclipse Foundation

Changing Environment

- New Technologies: RIA Applications and Cloud Computing
  - AJAX, Flash, Silverlight
  - Amazon E2 and S3, Google Docs, etc
- Dynamic languages becoming more mainstream
  - JavaScript, Ruby, Python, PHP
- UI programming is changing
  - XAML, CSS

Goals of e4

- Make it easier to write and deploy applications across computing environments (RIA, Desktop, Server, Cloud, Embedded)
  - Support Eclipse for the web

- Make it easier to write plugins
  - Support dynamic languages for plugin development

- Allow better control over the look of Eclipse based products
  - Remove the IDEness of Eclipse based products and applications

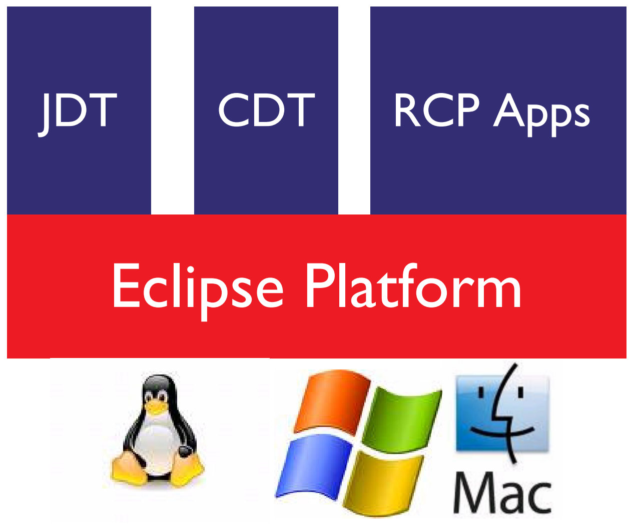
- More diversity of contributors to the core platform

Commitment to 3.x Compatibility

- We are committed to protecting your investment in the current 3.x SDK and RAP
- Ongoing development in 3.x for >5 years
  - Targeted enhancements, bug fixes, new platforms
  - e4 items that are backwards compatible
- Co-existence as long as needed
  - Think Apache 1.x and Apache 2

Eclipse Today

- Eclipse Platform delivers extensible frameworks for building applications
- Desktop oriented applications
- Java centric



What can we learn from the web?

Separate style information from implementation

- CSS based skinning

Include a scripting language (JavaScript is an “obvious” candidate)

Model the Workbench / make it available as a “DOM”

What's not great about the web?

Confusing array of choices leads to business risk

- Ajax: Dojo/GWT//Rico/Qooxdoo/…
- RIA: Silverlight vs. Flash/Flex/AIR

No component model or programming model beyond JavaScript

- It's OK for today's generation of Web applications, but for development scalability and broad appeal, the status quo won't work

Eclipse Tomorrow

Mac

Desktop

Logging

Help

Search

Preferences

Shared Resources

Persisting Data

Eclipse Application Services

python™

php

Dynamic Languages

RIA

AJAX

amazon webservices™

Google App Engine

Cloud Services

Key Architectural Goals

- **Eclipse Platform as Services**
  - Mixed use web applications
- Modeled and Declarative UI
- SWT for the Web
  - Mixed use web applications

Eclipse Platform as Services

- Eclipse SDK provides a number of frameworks
  - e.g. Selection, Drag&Drop, Progress, Help, Registry, Preferences, etc.
  - These are what you write your plugins against
- Migrate the existing SDK frameworks to be 1st class services
  - Clearer component boundaries to allow reuse in new contexts
  - Want well defined and documented set of RESTful interfaces for each service
  - Existing Eclipse plug-in model will remain

Services allow for…

Write Plug-ins in non-Java languages

- Wider audience, skillset
- Some languages more appropriate for some problems
- Initially JavaScript

JavaScript in the Workbench possibilities

- Write/recording/edit macros
- Runtime modification of behaviour
- Workflow orchestration
  - e.g. selection side effect behaviour

Plug-ins can now be distributed

- Running in different memory spaces

Key Architectural Goals

- Eclipse Platform as Services
- **Modeled and Declarative UI**
- SWT for the Web

Modeled UI

Presently

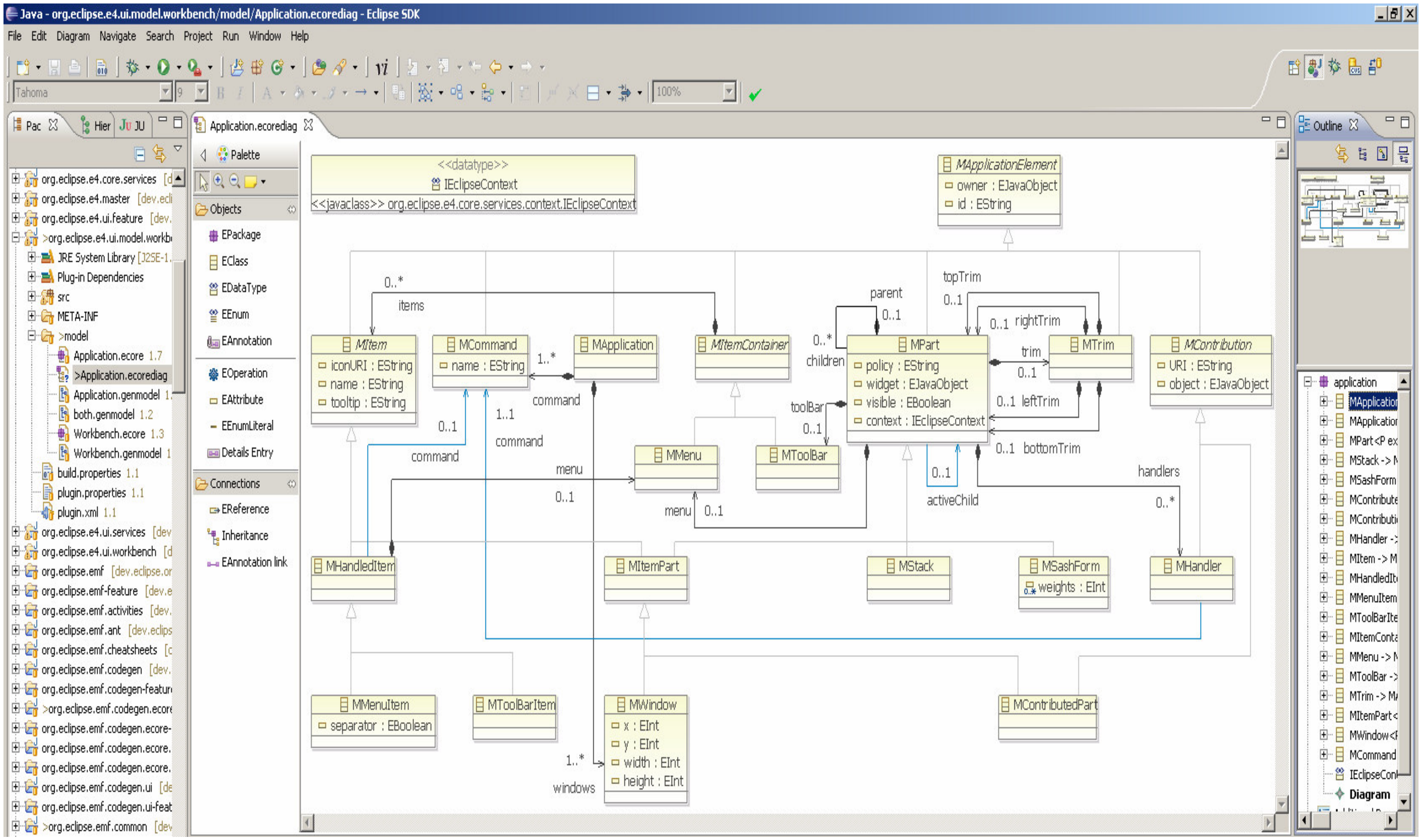
- Eclipse UI components tightly couple look and behavior
- The hierarchy of containment is fixed
  - e.g. workbench window->perspective->view/editor
- Results in IDEness creep into RCP applications

Modeled UI

- Create a DOM-like model of the workbench using EMF
- Model can be changed using external tools

- Create domain specific UI models to customize programming model
  - Workbench is no longer the starting point
  - Ex. UI Model for Investment Banking

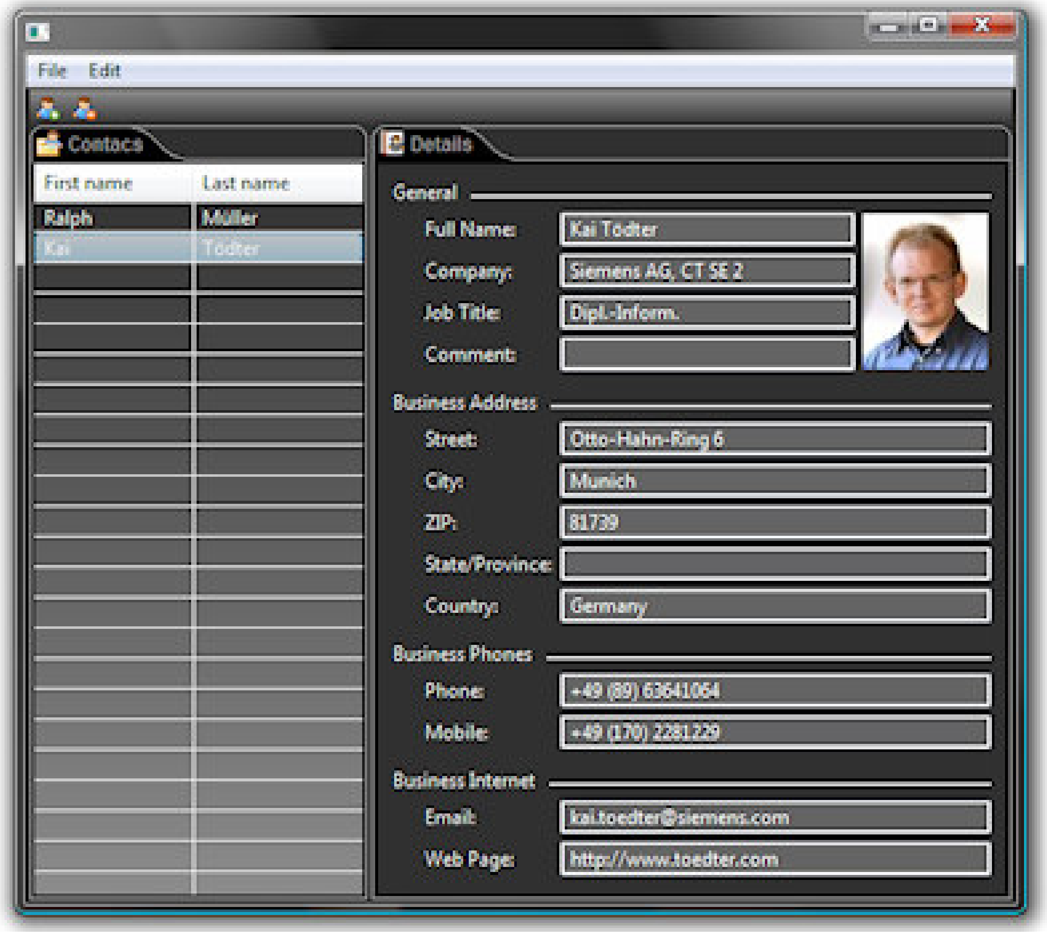
Workbench Model



Advantages to Modeled UI

- Forces us to clean up our architecture
- Simplifies part assembly, allows new structures
  - E,g, nesting of Java editors in the Compare editor
- More flexible RCP applications
  - Wider audience for RCP applications

Lose the "Eclipseness" of RCP Apps



First name	Last name
Ralph	Müller
Kai	Tödter

General

Full Name:	Kai Tödter
Company:	Siemens AG, CT SE 2
Job Title:	Dipl.-Inform.
Comment:	

Business Address

Street:	Otto-Hahn-Ring 6
City:	Munich
ZIP:	81739
State/Province:	
Country:	Germany

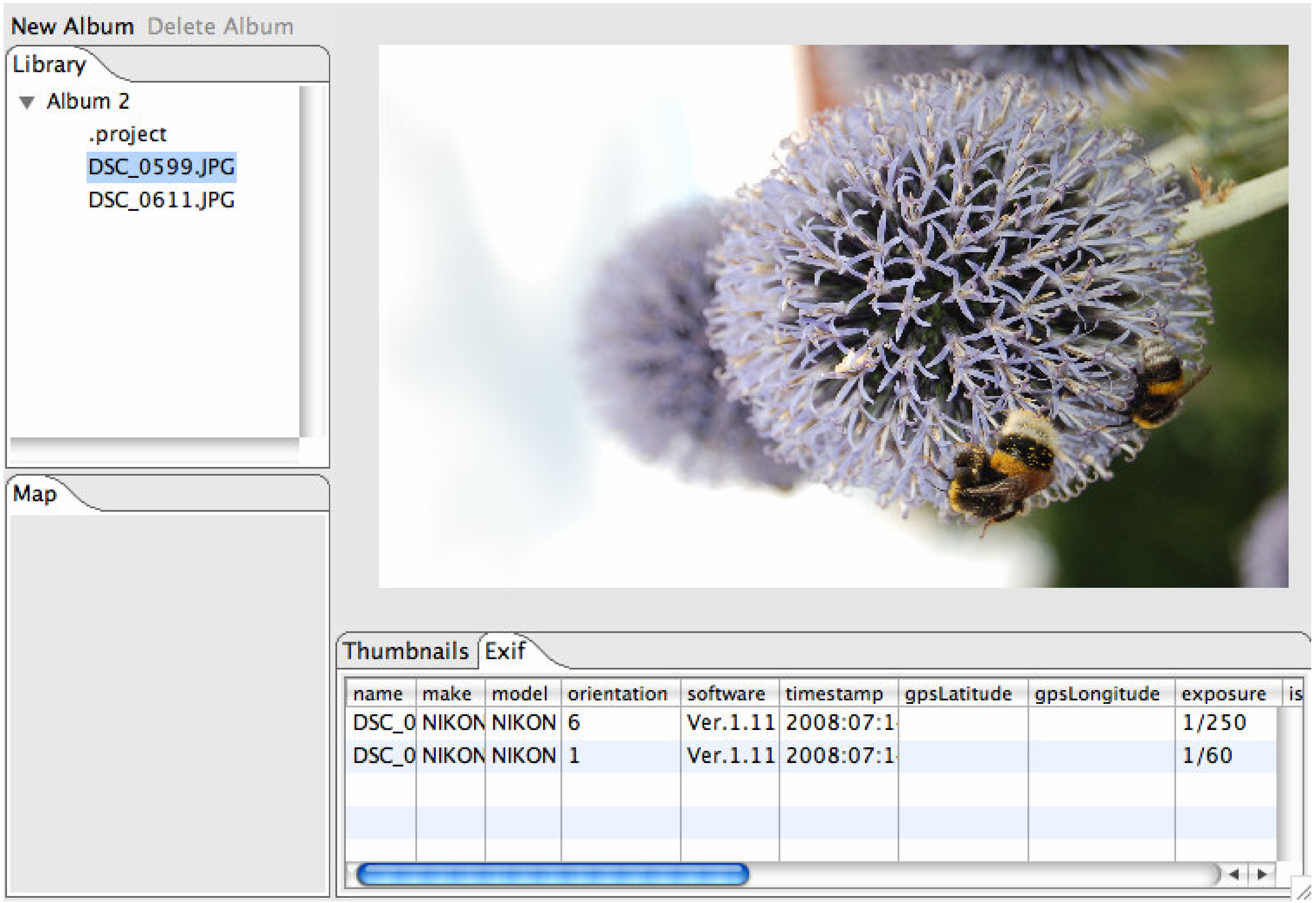
Business Phones

Phone:	+49 (89) 63641064
Mobile:	+49 (170) 2281229

Business Internet

Email:	kai.toedter@siemens.com
Web Page:	http://www.toedter.com

Flexible User Interfaces



name	make	model	orientation	software	timestamp	gpsLatitude	gpsLongitude	exposure	is
DSC_0	NIKON	NIKON	6	Ver.1.11	2008:07:1			1/250	
DSC_0	NIKON	NIKON	1	Ver.1.11	2008:07:1			1/60	

Declarative UI

Declarative construction

- E.g. XSWT, XUL, …

Declarative styling via CSS

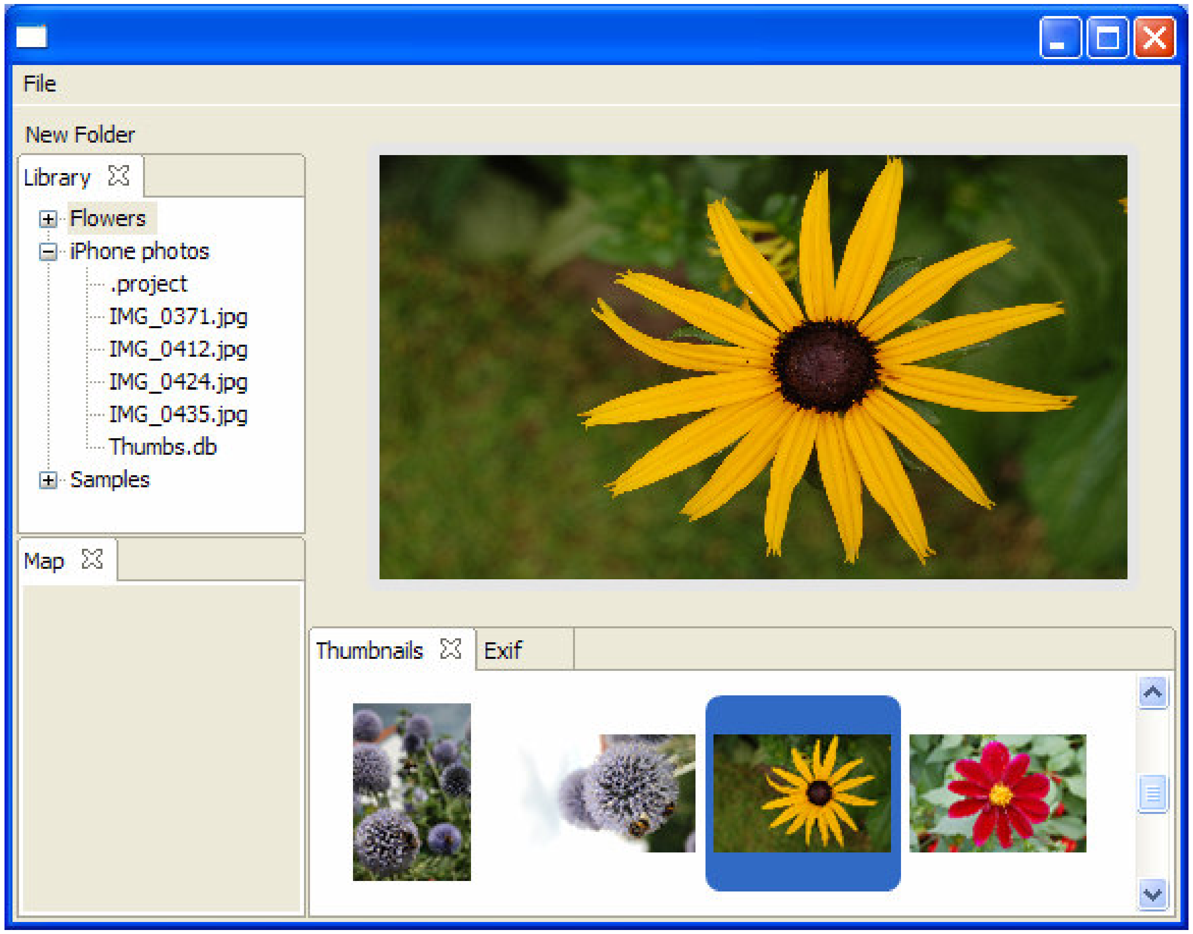
- Radically restyle Eclipse
- Simplify task of styling

Use a separate, pluggable styling engine

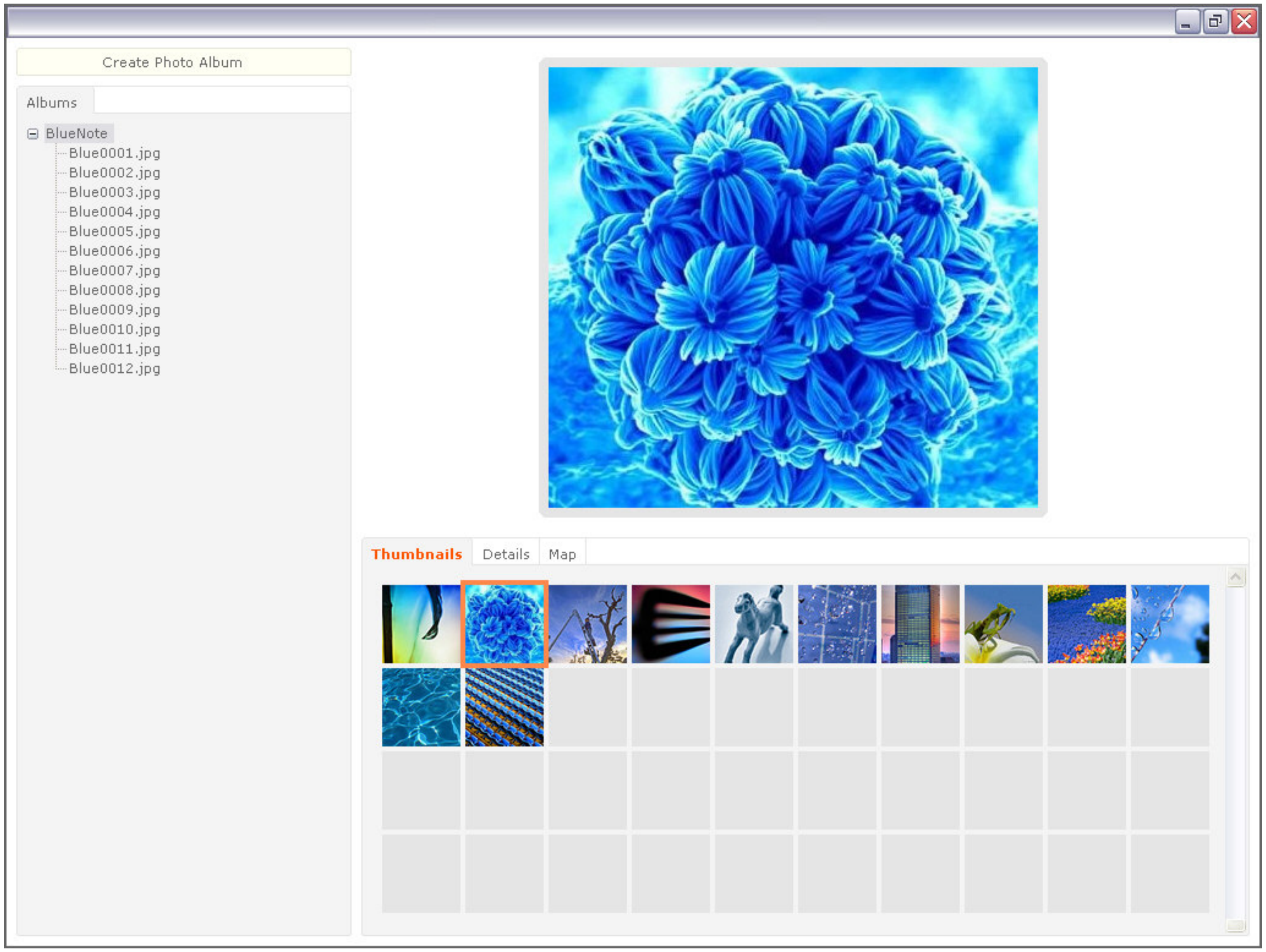
- CSS would allow sharing of styling information between desktop and related web pages

```
tab {font-family: Verdana; height: 23px; }
tab.active {start-color: #afc0eb; end-color: #7a96df; }
tab.inactive {start-color: #ffffff; end-color: #ece9d8; }
```

The Makeover



The Designers Vision…



Key Architectural Goals

- Eclipse Platform as Services
- Modeled UI
- **SWT for the Web**

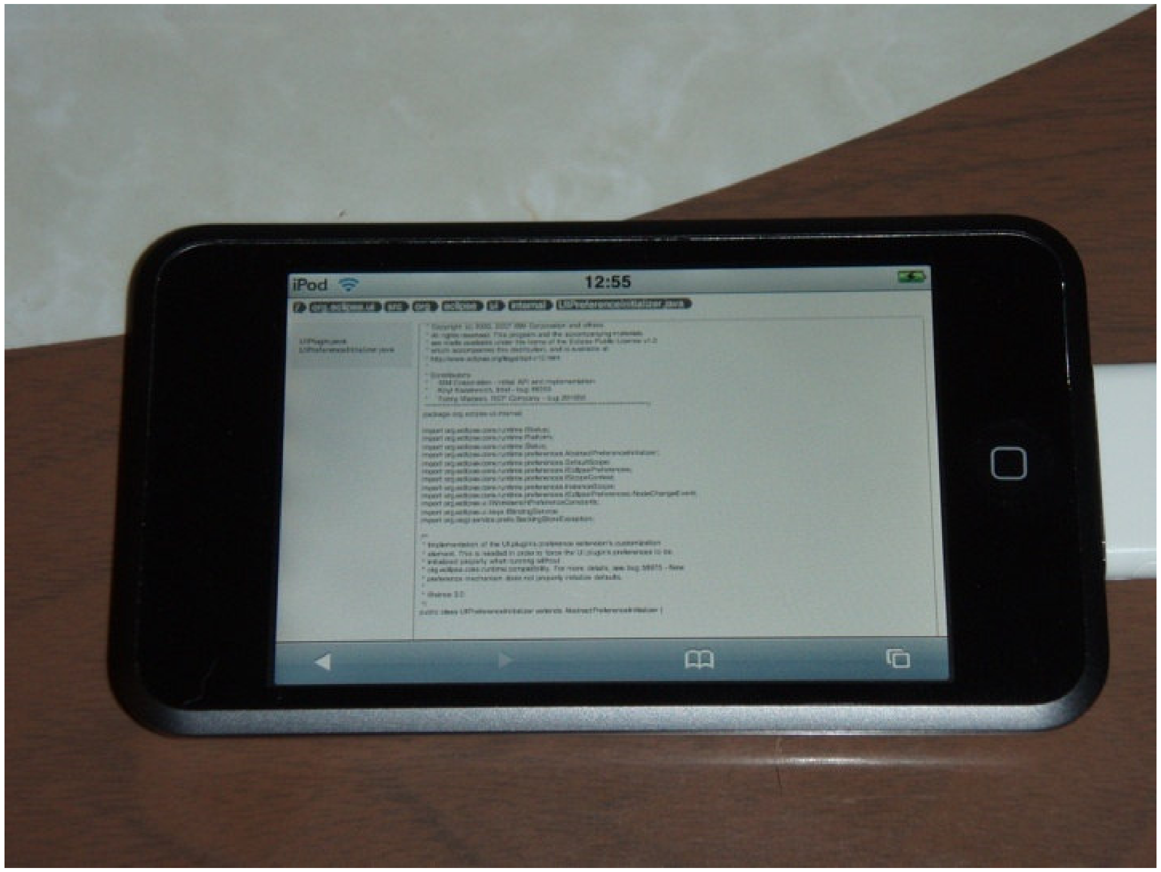
SWT on the Web

- SWT Browser Edition
  - Selective migration of high value widgets
  - E.g. StyledText cross compiled to run in web browser
- Write plugins in Javascript, run in either location
- Access common services which can be remoted
  - Dependency injection of services from any language
- Multi-user enable the workbench

Demo – SWT Browser Edition

Flickr x-compliled Javascript with Dojo
Flickr x-compiled to ActionScript in Flash
Flickr in Java in WPF native

Eclipse on an iPhone?



Timeline

- Individual work areas will move at own pace
  - Graduate as they become ready
- Overall “e4” platform builds with regular milestones
- Need to sync up with changes in 3.x code
- Checkpoint / re-assess after 1 year
- Deliver in 2 years

We Want Participation!

- Building a new, innovative platform
- New code, less complexity - your opportunity
- The SDK and RAP committers are totally psyched, but we must support the 3.x stream too
- Join the conversation!

  https://dev.eclipse.org/mailman/listinfo/e4-dev

Questions?

THANKS!