

EtoileUI

Quick Overview

Why a “new” UI toolkit?

- Everything can be changed at runtime
- Simple, compact and highly polymorphic API
- Write less code and develop faster
- Feeling of manipulating real objects

Key Features

- High level and uniform UI object tree
- UI objects are model objects
- Pluggables layouts and styles
- Squeak-like environment
- Pick & drop
- CoreObject integration

Developer Tool Goals

- Simplifies the development of new controls
- Permanent prototyping (no need to prototype, the application is the prototype)
- Live development
- Visual programming
- Generates UI from model objects

Document Goals

- Converges object manager and document
- Converges UI builder and document builder
 - UI and documents are both described in term of UI object tree
 - Composite document (static or dynamic)

MVC & UI as Model

View	Model
UI Object	Model Object
UI Object (meta view)	UI Object

■ Usual MVC

■ Reflective MVC

Pick & Drop

- Orthogonal copy/paste and drag/drop
- Drag/drop is a subcase of pick/drop
- Supported on any objects without writing code
- Pick collection to put several objects together on a pickboard

Separation of Concerns

- No monolithic view
- UI aspects stored in a repository
 - Styles
 - Layouts
 - Event Handlers
 - Views

UI Objects as AST

- If you consider that layout items are
 - abstract representations
 - organized in tree structure
 - transformable to concrete representations
- Layout item tree can be seen as an UI AST and the rendering backends as equivalent to code generation or translation backends

UI Plasticity Goals

(long term)

- Multiple UIs output by applying tree transformations on a shared high level UI description (an existing UI object tree)
- UI customization at runtime
- Web deployment

File Manager with CoreObject

```
id objectManagerView = [[ETContainer alloc] initWithFrame: myFrame];
```

```
[objectManagerView setSource: [objectManagerView layoutItem]];
```

```
[[objectManagerView layoutItem] setRepresentedObject: [CODirectory  
objectWithURL: [NSURL fileURLWithPath: @"/"]];
```

```
[[objectManagerView layoutItem] setShouldMutateRepresentedObject:YES];
```

```
[objectManagerView setLayout: [ETOutlineLayout layout]];
```

```
[[ETLayoutItemGroup windowGroup] addItem: [objectManagerView  
layoutItem]];
```