Project Manager Notes

Christopher Armstrong (carmstrong@fastmail.com.au)

January 30, 2011

Contents

0.1 Abstract

Project Manager is an Étoilé service for managing the user's workspace. This is a draft set of notes about the concepts underpinning Project Manager and the way a user might interact with it. Furthermore, this document proposes an architectural model for how Project Manager might integrate with other Étoilé components (such as EtoileUI or CoreObject) in order to realise these use cases.

This document is not an official guide to Étoilé concepts, imagined usage scenarios or architecture. Instead, its a memory dump on the part of its author to try and grapple with the ideas set forth by various persons working on the project and try and imagine a concrete implementation of them.

Chapter 1

Overview

1.1 Terminology

Active project The open project that is currently selected in the user interface and which is displaying its open views. There is one active project at a time.

Document A document is a top-level object in the Étoilé workspace. It belongs to a project, and may be represented by multiple views in the workspace. Documents can be created by the Project Manager or through some action in another application.

Inactive project A project that is open and running but is not currently selected.

Open project A project that has all its views opened and running.

Project A container for holding a set of views. The project remembers the position and stacking order of views on the screen.

Project Overlay A window that is temporarily displayed over the top of the workspace to allow the user the select a view to open.

Project Switcher An onscreen toolbar for switching between projects and closing active projects.

Service A service is a GNUstep application that displays the contents of documents in views on the screen. A service has one instance in the user's workspace, but may be displaying multiple documents and views.

View A window on the screen. These are called views in Étoilé because they are more than just windows: a view in Etoile remembers where on the screen it was placed, the internal layout and the data being edited.

1.2 Introduction

Project Manager is the Étoilé service for managing your workspace.

Chapter 2

Interactions

2.1 Use Cases

2.1.1 Create new project

The user creates a new project by selecting an option from within the Project Manager menus or the project switcher. The user gives the project a title, and then, Project Manager will switch to the project.

2.1.2 Open a project

The user clicks the projects icon and the projects overlay appears, displaying all the inactive projects. The user double clicks on one of the projects to open it. Project Manager opens the project and switches to it.

2.1.3 Switch to an active project

The user switches to an open project by clicking on its icon in project switcher. Before switching, Project Manager will hide all the views of the currently active project. Project Manager will then restore all the active views of the selected project and make the project's icon active on the project switcher.

2.1.4 Close a project

The user holds their mouse over the representation of the project in the project switcher until the close button appears. They then click the close button to make the project close. All of its open views will be closed and the project representation will be removed from the project switcher. If this was the active project, Project Manager will switch to the last active project.

2.1.5 Create a new view

A new view can materialise in a few ways:

- The user goes to the Document menu and selects a new document type to create.
- The user drags or copies some content out of an existing view into the project so such that it is linked or copied into a new view.
- The user goes to the template chooser and selects a template to instantiate as a new document.

A view isn't necessarily the same as a document. For example, a user may create a new document using the template chooser, but the service creating the document may spawn two views to show the document. Each view will have a unique view ID, but when the service that runs the view is asked to restore it, the data stored under that view identifier is will contain a reference to the document data that it contains. Both views will contain a reference to the same document, but it is up to the service to load the document only once.

2.1.6 Search

2.1.7 Open a view within the project

The user double-clicks on an empty portion of the desktop to open the Project Overlay. Inside the project overlay is the views that are part of the current project but are not currently open. The user double-clicks on a view and it drops down to the desktop and open.

At this point, the user can open another view by double-clicking on it, but they instead close the Project Overlay by just clicking outside of its overlay window.

2.1.8 Switch between open views

The Project Manager user interface could provide a few ways to select and switch between open views:

- a shortcut key combination to make each view in the stack active each time it is pressed
- an overlay that displays the active views in the current project which the user can select between

2.1.9 Close a view

The user clicks the close button on the active view. The view disappears from the screen, but is still available in the project switcher to be opened again.

2.1.10 Minimise or restore a view

The user clicks the minimise button on the active view

2.2 User Interface

Chapter 3

Architecture

3.1 Concepts

3.1.1 Views

The concept of a view in Étoilé, and especially in Project Manager, is the ability for the environment to remember the placement and internal layout of a window. The responsibility for remembering this is split between different parts of the environment.

A view is essentially a window from the perspective of Project Manager. From a visual perspective, Project Manager is responsible for remembering everything from the border outwards. This means that it will remember the position of the window on the screen, where it appeared in the z-order, the width and height, and whether it was running, minimised or closed. Everything internal to the window such as the document being edited and the layout of user interface items is the responsibility of the service displaying the window. This will be provided by the EtoileUI framework.

A view is not actually created by Project Manager even if it initiates the process of instantiating a new view. Instead, an Étoilé service will create a view in response to some kind of user command. This means that Project Manager is not responsible for loading the view, but only tracking when it is created or restored, moved around on the screen and when it is closed. So that Project Manager can track the window corresponding to that view, the view needs to uniquely identify itself. The process of restoring views and tracking them on the screen is described in section 4.1 (p.10) on X11 integration.

The runtime architecture of Project Manager is split into two parts. The first part is a window manager, which offers tight integration with Étoilé and GNUstep to integrate both closely in an X11 environment. The second part is a private service that displays the user interface used for managing projects and views and facilitates some auxiliary environment tasks.

3.1.2 Services

Each service is a GNUstep application. We use the name service to distinguish them from ordinary applications because users don't interact with them in the traditional sense of an application. Instead of starting up an application as a direct user action, the user does something in the environment like creating a new document or opening a view that might cause a service to begin running.

Because a service is a GNUstep application, and in turn, a GNUstep application is a programme, the lifetime of a service is just that of an ordinary application i.e. it is started as a new process, it can be tracked using its process identifier, we can communicate with it using inter-process communication primitives like sockets and shared memory, and it will die when it is closed. Almost every operating system support these fundamental ideas in essentially the same way that GNUstep is portable and can maintain the same Application Programming Interface on each OS without much difficulty.

Chapter 4

Operating System Integration

4.1 X11 Integration

4.1.1 Events

The events listed below are used by the application portion of Project Manager to determine when views are created, restored, minimised and closed.

MapWindow

This event is used to track when a view is restored or newly created. The window can be identified from its window properties.

UnmapWindow

This event is used to track when a view is minimised or closed. We can distinguish between a minimise and a close by inspecting the WM_STATE property that is set by the window manager portion.

4.1.2 Window Properties

__ETOILE_VIEW_ID

A string containing the view identifier as created by Project Manager and given to the service when asking it to restore the view.

This property is placed on a view's top-level window by the service before it is mapped. It is not updated by the service while the window is still mapped.

__ETOILE_PROJECT_ID

A string containing the identifier of the project. It is placed on a view's top-level window by a service to indicate the project that a view belongs to. This property may be updated by a service to indicate that the view belongs to a new project, which means the window manager should monitor this property and only display the view window in the currently set project.

The Project Manager will place this property on the root window of the workspace to indicate the current project. When it changes this property, the window manager should re-map the windows on the display so that only the views of the newly selected project are shown.

4.1.3 Messages