

Study of User Interface Distribution in Multi-Device and Multi-User Environments with Dynamically Migrating Engines

ADAPTION DES INTERFACES À L'ENVIRONNEMENT

Date : 11/11/2014

Student : Lifeng TAO

1. Introduction

In the last decade a wide variety of interactive devices have penetrated the mass market, and people spend more and more time using them. This has made it possible to create many environments where people spend a long time interacting with various devices sequentially or in parallel. In order to better exploit such technological offer often people would like to better use multiple devices while interacting with their applications, for example to dynamically move components of their interactive applications across different devices with various interaction resources.

So the authors of this article proposed a framework which can be exploited both in Web and Java applications in order to obtain more easily application user interfaces (UIs) that can be dynamically distributed and/or migrated in multi-device and multi-user environments. The framework also allows dynamically creating multiple simultaneous sessions for applications used by groups of devices where the UI is distributed. Furthermore, it does not require a fixed server to manage the distribution. The elements of the UI can be distributed by specifying specific device(s), group(s) of devices, specific user(s), and groups of users according to roles.

2. Context of use

This framework is currently available for Android and Web-based applications. The authors plan to provide a version for iOS as well. As for the specific equipment, it can be a mobile phone, a Tablet PC, a PC, a WebTV etc.

3. Time of use

The technique is used not only at design time but also at runtime. The developer can use it to design applications which can dynamically distribute UIs into other devices very easily, so it facilitates the developer's life. The user of the application can focus on the information in which he is interested, because on his screen, only the appropriate component will be displayed, so it facilitates the user's life too.

4. Solution

4.1. Underlying model

The technique presented in this paper use a CIM/PIM model, it can be used to develop different kinds of applications on different platforms (computer, phone, tablet, WebTV) using different languages (Java, web languages).

The model is composed of a library and runtime support. The library is used by the developers to introduce UI distribution in their applications. The runtime support can run on a dedicated server or in one of the devices participating in the distribution. The Engine side is the runtime support and is responsible for managing the requests of distribution changes, processing them, and calculating the new distribution state. For each distributable element of the UI a JSON document is created and memorized in a document-oriented database (CouchDB) located in the engine. The engine is dynamical and can be moved. We just need to call the *MoveEngine Command*, the current engine will serialize the Distribution State and sends the

database to the requester device. Then an *EngineMoved Notification* command will be sent by the current engine, and all others devices have to subscribe to the new engine.

Some basic notion for the model:

Two concepts (Type and Role) to address the devices involved in the distribution:

Type: The type concept is associated with a set of device capabilities.

Role: The role concept is related to the type of tasks carried out by the user using a device in the distributed application and is independent of the device type.

The Distribution API

ASSIGN Command: used to change the devices that can display or allow manipulation by the user of a certain element in the UI.

Feedback Command: Used to inform devices of a change in the value of an element

ASSIGN Notification: engine’s ASSIGN Command response

Feedback Notification: engine’s Feedback command response

4.2. An example of using scenario

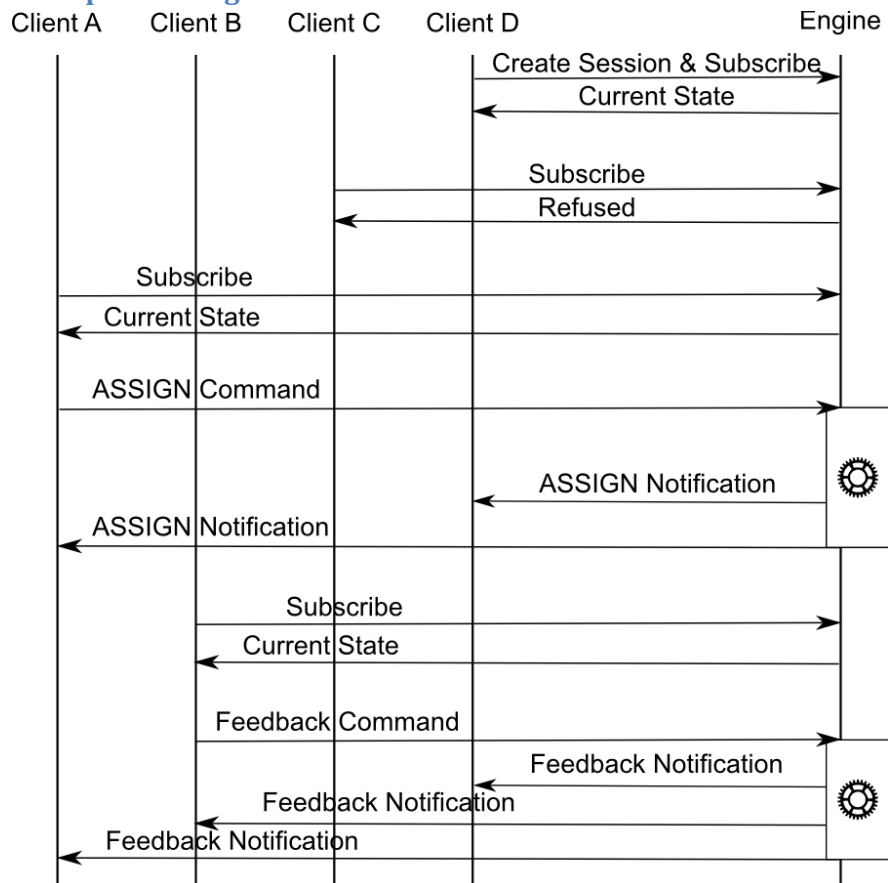


Figure 1: Sequence Diagram describing examples of subscription requests (accepted and refused) and commands sent by multiple devices.

In this example, Client D creates a session and subscribes itself. After that client C tries to subscribe but the request is refused by the engine because it does not provide the necessary credentials. Client A instead subscribes successfully and then sends an ASSIGN command. The engine calculates the new state and sends the corresponding notification to the involved devices. This is repeated by client B, which after a successful subscription sends a Feedback Command.

4.3. A specific example (City guide)

It is an example of a multi-user, multidevice application for outdoor environments. There are two kinds of roles in this example: the guide and the tourists. The user interface elements are in the Enabled state for the guide while they can be Disabled or even Invisible for the tourists. The guide can interactively change the states for the elements shown in the tourist's user interfaces.

Look at the image1 "Guide UI" and image2 "Tourist UI", the guide can select the different images to be shown to tourists. The big image (element with id FeaturedImage) is visible to tourists and enabled to guide. The right block is enabled to guide and invisible to tourist. The guide can change the FeaturedImage, thus, a Feedback Command is sent to the Engine containing the URI of the image to be shown. And on the tablet for the tourist, the selected image will appear.

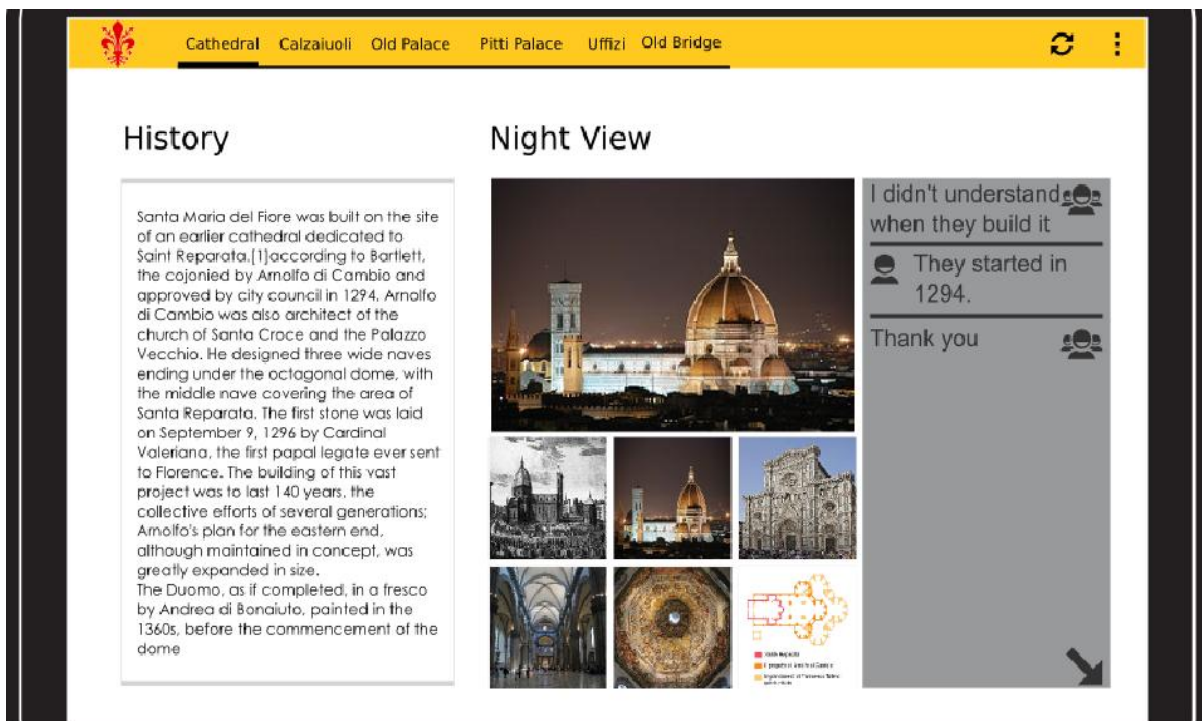


Figure 2: Guide UI

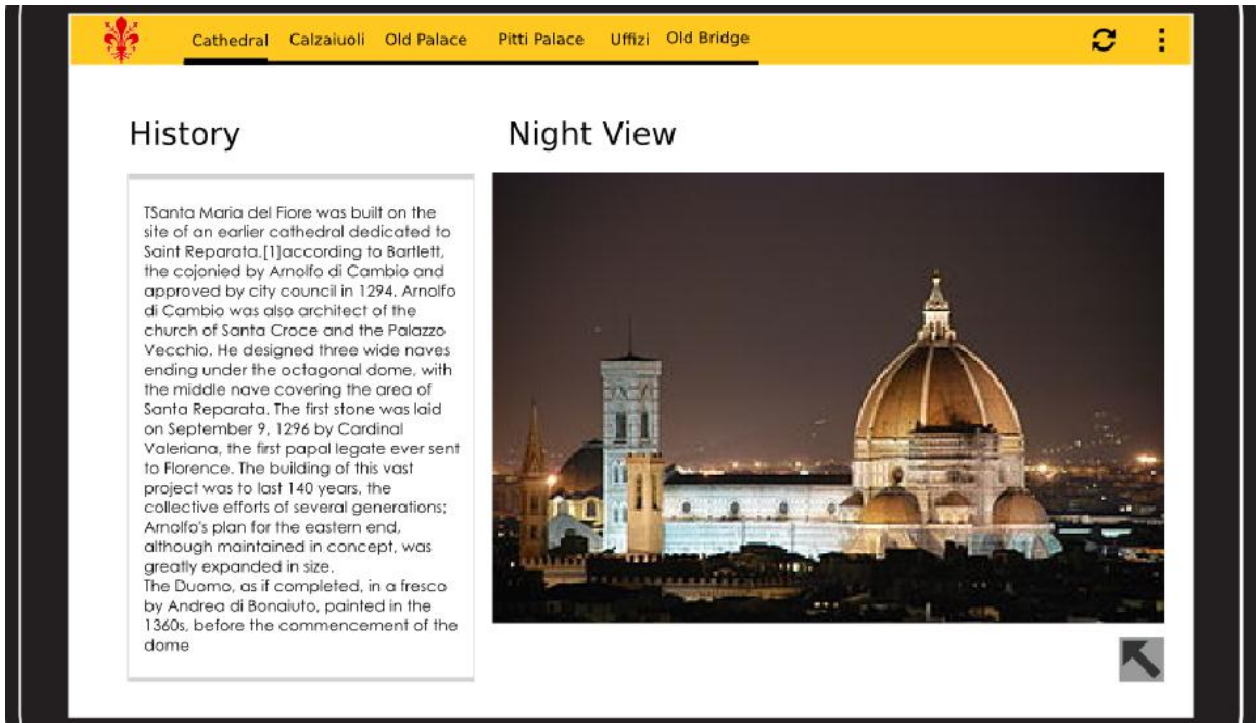


Figure 3: Tourist UI

5. My opinion

The big advantage is that different roles have different views on different devices. It can be very helpful for different user to focus on the specific information. Using different devices can make the applications adapted to the environment. And according to the authors, the impact of the framework in the application code is limited; the framework has a good performance.

The disadvantage is that the platform and development language supported by this framework is very limited. And for now, there isn't any security ensurance.