Flash in OSGi

Confidential, Draft

7 Pages

Abstract

The OSGi framework offers an environment where services from different nature and vendors van coexist in a robust and standard way. To make applic ation with user interfaces OSGi does not offer any support. Macromedia Flash is a tool chain very popular among designers for its ease of use and the high quality of its user interfaces. This RFP tries to describe the benefit of integrating both technologi es.

0 Document Information

0.1 Table of Contents

0.2 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be inter preted as described in [1].

```
Source code is shown in this typeface .
```

0.3 Revision History

The last named individual in this history is currently responsible for this document.

All Page Within This Box

Revision	Date	Comments
Initial	03 17 2006	rob van den b erg, Siemens VDO *rob.vandenberg@siemens.com*
0.2	04 26 2006	Added Use Case Scope broadened to SVG -T

1 Introduction

Over the years there has been a n ongoing discussion within the OSGi community how to put user interfaces on top of bundles. However, no single sol ution seems to cover all uses of OSGi:

- Swing: too big for embedded devices

- AWT: too low level

- SW: requires native code

Moreover, none of these solutions are very appealing for designers: they prefer tooling that is more catered for their need, is more vi sual appealing. Macromedia Flash offers a solution for building of user interfaces that is popular among designers and can be used both in embedded systems as well as in web -based applications. However Flash does not offer the services infrastructure offer ed by the OSGi framework. This RFP tries to describe the advantages of using Flash within an OSGi environment and what is needed for an efficient collaboration.

All Page Within This Box

OSGi™ Alliance

2 Application Domain

2.1 Flash[1] and SVG-T

Macromedia Flash is a popular development environment for o ffering attractive user interfaces. Traditionally it was mainly used for games and other animations, but gradually it offered more interactivity so that rich user interfaces could be built in Flash. As its development environment resembles tools like Adobe   Illustrator, it is very appealing for graphic designers. One of the big advantages of Flash is that it uses a vector   description of screens. This makes that its user interfaces can be used both in a Web environment as well as on embedded devices: many mobile phones nowadays have support for flash.

Many of the advantages of Flash are now also available as a W3C standard called Scalable Vector Graphics (SVG). SVG is available in multiple profiles and can also be supported on mobile devices. The Macromedi   a Flash Lite 2.0 player can actually support both SVG -T and Flash. There is a lot of development in tools but they are currently not as good and widespread as the Flash tools.

2.2 Interactivity

In Flash user interfaces are designed using graphical tools. The i nteractivity is achieved by means of a scripting language (ActionScript or JavaScript) that binds screen elements to functionality. These functions are typically implemented by means of, native, callback functions.

2.3 Terminology + Abbreviations

ActionScrip t	Prototype language used for adding actions to Flash User Interfaces
FLASH	Popular Authoring enviroment developed by MacroMedia
JavaScript	Prototype language used for adding actions to SVG -T User Interfaces
SVG	Scalable Vector Graphics
SVG-T	SVG Tiny
UI	User Interface

---

[1] No te that the term Flash in this case is used for both the development environment as well as for the player used for playing the user interfaces

All Page Within This Box

3 Problem Description

3.1 UI & OSGi

As Flash is very popular for developing user interfaces, and the fact that they can be used in an embedded environment, it would be a great benefit if we could easily combine this ease of development with the OS Gi service framework.

By offering services to integrate Flash/SVG applications with OSGi services it would make it easy to deploy bundles that require a user interface. Specifically for the following situations where offering FLASH/SVG based OSGi services would prove to be a great benefit.

3.1.1 Developing User Interfaces

A typical OSGi-based application consists of a number of cooperating OSGi bundles. A Flash/SVG -OSGi integration would be of great help: it would make it possible to easily develop a user interfa ce for this application

3.1.2 Providing a bundle with its own user interfaces

A vendor wants to develop a bundle with its own distinctive user interface. In this case it would be nice if the Flash/SVG UI can be a part of a bundle.

3.1.3 Upgrading a user interface

Finally by embedding Flash/SVG UIs into OSGi bundles, we can benefit from the upgradeability feature of the OSGi framework for user interfaces.

3.2 Flash-OSGi

Key to this integration that the coupling between the two worlds should be minimal: the OSGi bundle shou ld not rely on their user interface, nor should the Flash/SVG UI make any assumptions on the services.

The following section will detail a use case mentioned in the previous sections.

4 Use Case

Acme Inc. develops a Bluetooth based blood pressure meter. The device can be used with OSGi devices (mobile phones, PDAs, PCs, etc) . They develop 2 bundles. One bundle contains the device driver, the other bundle contains a high resolution version of the UI intended for PCs and a low resolution version intended for PD As and mobile phones.

All Page Within This Box

The developers design their code. Important aspects of the design are the events they must receive from the UI and the events th ey will send to the UI. These artifacts are placed in a design docu ment that is given to the graphic designers. They outsource the GUI development to a graphic design company. The GUIs are tested on a PC until the management team approves them.

The bundles are now placed on a management system for deployment. When they d eploy their bundles, the management system downloads the bundles. For small devices, the high res GUI is removed from the bundle during download.
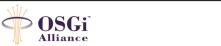
5 Requirements

- The GUI must be able to bind to actions in the bundle's code
- The GUI must be able to bind to acti ons in other bundle's
- The bundle must be able to influence the GUI (select scenes, provide texts, etc).
- The GUI must look identical on devices with the same graphic capabilities.
- It must be possible to use popular editing tools for the GUI (Like MacroMedia Flash)
- It must be possible to replace a GUI dynamically without system restart
- Must support localization
- Both SVG-T and Flash must be supported
- Future format extensions and new formats should be supported
- The runtime must be able to choose the between mu ltiple GUIs depending on the environment. That is, it must for example be possible to insert a big screen version as well as a small screen version in the same bundle.
- The GUI must be able to express requirements on the environment so that management syste ms can decide what bundles are suitable for a specific device
- The overhead introduced by using Flas h/SVG-T should be minimal .

6 Document Support

6.1 References

[1]. Bradner, S., Key words for use in RFCs to Indicate Requirement Levels, RFC2119, March 1997.

[2]. Software Re quirements & Specifications. Michael Jackson. ISBN 0 -201-87712-0

[3]. http://en.wikipedia.org/wiki/Macromedia_Flash

All Page Within This Box

[4].   http://www.w3.org/Graphics/SVG/Group/

6.2 Author's Address

Name	rob van den berg
Company	Siemens VDO Automotive
Address	Luchthavenweg 48
Voice	+31 40 8444859
e-mail	rob.vandenberg@siemens.com

Name	Peter Kriens
Company	aQute
Address	9C, Avenue St. Drézéry
Voice	+33467542167
e-mail	Peter.Kriens@aQute.se

6.3 End of Document