

SVG Tiny (SVGT) et XHTML basic : langages pour des composants JSF mobiles

Tayeb Lemlouma

Novembre 2004, Projet WAM, INRIA Rhône Alpes
Tayeb.Lemlouma@inrialpes.fr

1. Introduction

Dans une architecture d'adaptation et de négociation [14] des composants JSF [15] pour les terminaux mobiles, il est important d'identifier un format qui sera utilisé pour la génération des présentations mobiles des différents composants. Le format choisi doit être supporté par les terminaux mobiles cibles. Qui dit supporté, dit que ces différents terminaux vont être capables de traiter et d'afficher correctement les différentes présentations reçues. Vu le caractère automatique et dynamique de la génération des présentations, il est nécessaire que le langage cible choisi soit flexible et adaptable, afin de couvrir une large gamme de clients mobiles en prenant en compte la diversité de leurs caractéristiques matérielles et logicielles. Ce document étudie le langage cible qui pourra être choisi pour l'adaptation des présentations de composants JSF.

2. Le standard SVG

SVG (*Scalable Vector Graphics*) [1] est un format standardisé proposé par le consortium W3C pour les images et les animations (Web, mais pas seulement) en deux dimensions. SVG est basé sur le langage XML, ce qui lui confère une souplesse et une clarté d'utilisation très forte. Le format gère des ressources textuelles, des dessins, des images, du son et des animations. Le langage SVG comprend plusieurs avantages, basé sur XML, il permet d'utiliser tous outils XML existants dont les parseurs, les mécanismes de transformation comme avec les langages de transformation (tels que XSLT) et l'adaptation pour différents contextes.

Grâce à son principe de dessin vectoriel, SVG assure une adaptation d'affichage des différentes ressources médias à des tailles différentes de dispositifs d'affichage. Les formes vectorielles peuvent être réduites ou zoomées pour une zone ou une région particulière. Il est également possible d'associer des styles différents, adaptés au contexte cible, au contenu SVG d'origine. Comme SVG est du XML, il est possible d'accéder à toutes les entités élémentaires d'une présentation donnée. Cet avantage permet de manipuler facilement les objets de la présentation, ce qui donne une souplesse dans l'adaptation et la transformation des différentes ressources multimédias, afin de les rendre adaptables à un contexte bien particulier. La taille d'une présentation SVG est relativement raisonnable par rapport aux autres formats non vectoriels, en outre, SVG supporte des mécanismes de compression et de binarisation, ce qui le rend efficace pour les environnements ayant des limitations de bande passante. Du côté génération de présentations, il est facile de créer du contenu SVG, avec de complexes objets multimédias.

Pour répondre à une forte demande de la part de l'industrie et de la communauté des utilisateurs du langage SVG, le groupe de travail du W3C a travaillé pour la définition d'une forme de SVG qui peut être utilisée pour des présentations graphiques sur des types dispositifs d'affichage. Afin de satisfaire cette forte demande, le groupe de travail SVG s'est investi dans un effort ciblé sur la création de spécifications de profils qui adressent les terminaux mobiles. Comme il n'est pas suffisant de définir un

unique profil pour toute la diversité des clients mobiles, vu leurs différentes caractéristiques : vitesse de traitement, capacité mémoire, support de couleurs, etc.. le groupe de travail a défini deux profils SVG : SVG Tiny (SVGT) et SVG Basic (SVGB), voir Figure 1. Le premier profil (SVGT) [2] convient aux terminaux mobiles à caractéristiques limitées de bas niveau ; tandis que le deuxième profil (SVGB) vise des terminaux mobiles de niveau de support plus élevé. En raison des caractéristiques et des limitations des ressources des terminaux mobiles tels que la capacité de traitement et d'affichage, les profils mobiles de SVG présentent de nombreuses contraintes sur le contenu, les types d'attribut, les propriétés et le comportement envisagé des agents utilisateurs.



Figure 1 : profils du langage SVG

Les profils SVG Tiny et SVG Basic utilisent certains modules SVG 1.1 [3]. Pour chaque module (voir Figures 2 et 3), un profil donné peut contenir la version complète, une version basique ou aucune version de ce module.

SVG Tiny	
Structure	<i>Core Attribute Module Basic Structure Module</i>
Painting	<i>Basic Paint Attribute Module Basic Graphics Attribute Module</i>
Hyperlinking	<i>Hyperlinking XLink Attribute Module</i>
Conditional Processing	<i>Conditional Processing Module</i>
Shapes	<i>Shape Module</i>
Images	<i>Image Module</i>
Text	<i>Basic Text Module</i>
Fonts	<i>Basic Font Module</i>
Animation	<i>Animation Module</i>
Extensibility	<i>Extensibility Module</i>

Figure 2 : profil du langage SVG Tiny

SVG Basic	
Structure	<i>Core Attribute Module Structure Module Viewport Attribute Module</i>
Style	<i>Style Module</i>
Painting	<i>Paint Attribute Module Opacity Attribute Module Graphics Attribute Module</i>
Hyperlinking	<i>Hyperlinking XLink Attribute Module External Resources Attribute Module</i>
Conditional Processing	<i>Conditional Processing Module</i>
Shapes	<i>Shape Module</i>
Images	<i>Image Module</i>
Text	<i>Text Module</i>
Color Profile	<i>Color Profile Module</i>
Gradients	<i>Gradient Module</i>
Patterns	<i>Pattern Module</i>
Clipping	<i>Basic Clip Module</i>
Masking	<i>Mask Module</i>
Fonts	<i>Font Module</i>
Interactivity	<i>Document Events Attribute Module Graphical Element Events Attribute Module Animation Events Attribute Module</i>
Scripting	<i>Scripting Module</i>
Views	<i>View Module</i>
Filters	<i>Basic Filter Module</i>
Animation	<i>Animation Module</i>
Extensibility	<i>Extensibility Module</i>
Document Model	<i>Basic Document Model Module</i>

Figure 3 : profil du langage SVG Basic

3. SVG dans les plateformes mobiles

La grande majorité des terminaux mobiles supporte déjà le format SVG, que ce soit d'une manière native ou après une configuration explicite. En effet, certains terminaux sont dotés d'une plateforme de supports du standard SVG et certains autres supportent l'installation d'un navigateur approprié. Tous les terminaux mobiles qui fonctionnent sous Symbian OS, ce qui est le cas de la majorité des terminaux mobiles, supportent le format SVG. Les téléphones mobiles suivants supportent au moins le profil SVG Tiny : Nokia 3650/3600, Nokia 3660/3620, Nokia 6600, Nokia 7650, Nokia 9210 Communicators, Nokia 9290 Communicator, Nokia N-Gage, Sony Ericsson P800, Sony Ericsson P900, Motorola A920, Motorola A925, FOMA F900i, FOMA F2102V, FOMA F2051, Siemens SX1, Nokia 6620, Nokia 7700, Nokia 9500, Motorola A1000, Panasonic X700, Sendo X, BenQ P30.

En 2002, Bitboys [4] développe un accélérateur matériel qui permet, entre autres, de visualiser plus rapidement les fichiers au format SVG, ce qui est très avantageux pour les appareils mobiles. A partir d'octobre de l'année dernière, le noyau du processeur graphique *The Bitboys Acceleon G10* a été intégré au dispositif d'affichage par NEC Electronics.

Les assistants personnels peuvent facilement supporter le format SVG en installant un navigateur SVG. De nombreux navigateurs sont disponibles pour les terminaux de type Windows Smartphones.

Exemples : BitFlash [5], CSIRO [6], eSVG [7]. Pour les terminaux mobiles qui supportent Java, des navigateurs, tels que TinyLine SVG Viewer [8], SVG peuvent être utilisés.

La première organisation qui a adopté le format SVG Mobile, est le 3GPP (3rd Generation Partnership Project) [9], qui rassemble tous les leaders de l'industrie des terminaux mobiles. 3GPP adopte le format SVG Mobile, comme un format vectoriel essentiel dans le format MMS (Multimedia Messaging Service) et PSS (Packet-Switched Streaming Service) dans leur version Release 5. Dans ces deux spécifications, SVG Tiny 1.1 est exigé et SVG Basic 1.1 est optionnel.

Le Java Community Process (JCP) a créé un groupe expert JSR (Java Specifications Request) [10], présidé par Nokia et Sun pour travailler sur la création d'APIs standardisés, de SVG Tiny pour J2ME. SVG Tiny deviendra donc, à court terme, un composant qui fait partie de la plateforme J2ME. Dans un appel à proposition, daté de mars dernier, MPEG recommande un travail sur un format binaire, compatible avec SVG Tiny, pour la représentation des scènes dans les environnements mobiles.

4. Quelques ressources logicielles

De nombreux navigateurs supportent le format SVG. Pour les terminaux mobiles qui fonctionnent sous le système d'exploitation Symbian OS, le navigateur ZOOMON SVG Player [11] supporte SVG Tiny. Le téléphone mobile Siemens CX65 est le premier appareil mobile commercialisé avec un navigateur ZOOMON SVG Player intégré. Le Sony Ericsson K700 est intègre aussi le même navigateur. Pour les systèmes Symbian OS, une version du navigateur Opera intègre le navigateur ZOOMON SVG Player depuis avril 2003. CSIRO fournit un navigateur PocketSVG, qui supporte SVG Tiny, pour les assistants personnels Pocket PC et Pocket PC 2002. Sharp fournit un navigateur SharpMotionART player pour Palm, Pocket PC et Zaurus. BitFlash Mobile SVG Player est un navigateur qui supporte SVG Tiny et SVG Basic pour les systèmes Windows 2000/XP, Pocket PC, Pocket PC 2002 et Symbian OS 6/7. En 2003, Sharp et BitFlash ont été rassemblés dans le développement et l'intégration d'un nouveau navigateur, MotionART player, dans Sharp's V601SH.

Intesis fournit le navigateur eSVG pour les systèmes embarqués, les assistants personnels et les plateformes qui tournent sous Symbian OS. Le toolkit TinyLine inclue un navigateur mobile SVG qui supporte SVG Tiny pour les appareils J2ME. Le navigateur NetFront fourni par ACCESS, inclue un module d'extension multimédia qui permet d'installer un navigateur SMIL et SVG. Les plateformes supportées sont, entre autres, Palm OS, Symbian OS, Pocket PC, Windows CE et Linux. Plazmic offre des navigateurs interactifs pour les appareils i-mode et MIDP.

5. Les formulaires

Plusieurs applications Web nécessitent l'utilisation des formulaires qui permettent à l'utilisateur de saisir des informations qui seront transmises au serveur. Ces applications incluent généralement une partie de traitement du coté du serveur ; le serveur traite les informations reçues et envoi un résultat qui dépend de ces informations. Un exemple de ces applications est la gestion des d'authentifications vers un serveur donné, où le client est invité à saisir son login et son mot de passe. Le langage SVG Tiny ne supporte pas, d'une manière native, les formulaires. Pour supporter les formulaires, dans le monde des terminaux mobiles, l'utilisateur final doit installer des modules supplémentaires sur son terminal. L'objectif de l'adaptation étant de fournir du contenu dont la présentation est adaptée au contexte cible sans faire des suppositions fortes sur les modules installés sur le terminal, le langage SVG Tiny doit être enrichi par un autre langage cible qui assure un support de formulaires et qui soit supporté par la communauté des terminaux mobiles.

6. XHTML basic

XHTML, a été défini par le W3C en reformulant le langage HTML 4 en XML. XHTML Basic [12] comprend un ensemble minimal de modules XHTML qui inclut le support des formulaires, des images, des tableaux, etc. Cet ensemble est conçu comme une base commune qui peut être étendue et utilisé par des clients Web qui ne supportent pas toutes les fonctionnalités de XHTML ; tels que les téléphones portables, les assistants personnels, les *paggers*, etc. XHTML sert donc de langage commun à un ensemble hétérogène de plateformes et de user agents.

La spécification XHTML mobile profile (XHTML MP), définie par le WAP Forum (actuellement OMA : Open Mobile Alliance [13]), est basée sur XHTML Basic. Les implémentations des navigateurs mobiles des industriels de la téléphonie sont hétérogènes et s'approche plutôt de la spécification XHTML Basic du W3C que de la spécification XHTML MP. Par exemple, les téléphones mobiles japonais DoCoMo supportent les fonctionnalités de XHTML Basic plutôt que le profile entier XHTML MP. Les récents téléphones mobiles Nokia peuvent traiter des documents XHTML MP mais sans le support du namespace WAP. Le navigateur Opera 6, conçu pour les plates formes Symbian supporte, avec quelques exceptions, XHTML Basic avec le *mime type* `text/html` (et non pas `application/vnd.wap.xhtml+xml` tel qu'il est recommandé par OMA). Afin de résoudre le problème d'hétérogénéité du support de XHTML, la meilleure approche consiste à utiliser XHTML Basic qui représente un dénominateur commun pour les terminaux qui supporte XHTML.

Le langage XHTML basic complète SVG pour remédier au problème du manque de support des formulaires dans SVG Tiny. En effet, XHTML basic assure la prise en compte de l'édition de page Web qui comporte des zones d'édition tel que *textarea*, *input*.

Exemple d'utilisation des formulaires dans XHTML Basic:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
    "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type"
    content="text/html; charset=iso-8859-1" />
  <title>Exemple de document XHTML Basic</title>
</head>

<body>
<form action="http://www.foo.com/foo" method="post">
  <table>
    <tr>
      <td colspan="2">
        <textarea cols="25" rows="5" name="text_area">
          Ligne 1
          Ligne 2
        </textarea>
      </td>
    </tr>
    <tr>
      <td>
        <label for="fname">Prénom:</label>
        <input type="text" name="firstname" id="fname" />
        <label for="lname">Nom:</label>
        <input type="text" name="lastname" id="lname" />
      </td>
      <td>
        <select name="component-select">
          <option selected="selected">Option_1</option>
          <option>Option_2</option>
          <option>Option_3</option>
          <option>Option_4</option>
          <option>Option_5</option>
        </select>
        <input type="submit" value="Envoyer" />
      </td>
    </tr>
  </table>
</form>
</body>
</html>
```



Ligne 1 Ligne 2	
Prénom: <input type="text"/>	Option_1 ▼
Nom: <input type="text"/>	Envoyer

Afin de profiter du large support de SVGT par l'ensemble des téléphones mobiles ainsi que ses possibilités d'adaptation évoluée absentes dans XHTML Basic, le langage SVGT peut être utilisé pour transmettre du contenu selon les fonctionnalités de l'application Web et le profil du terminal final. La détermination du type de présentation, à envoyer au terminal, est la responsabilité du module de négociation dans l'architecture proposée.

Références

1. Scalable Vector Graphics (SVG), <http://www.w3.org/Graphics/SVG/>.
2. Mobile SVG Profiles: SVG Tiny and SVG Basic, <http://www.w3.org/TR/2003/REC-SVGMobile-20030114/>.
3. Scalable Vector Graphics (SVG) 1.1 Specification, W3C Recommendation *14 January 2003*. <http://www.w3.org/TR/SVG11/>.
4. Bitboys, <http://www.bitboys.fi/>.
5. BitFlash, <http://www.bitflash.com/index.asp>.
6. CSIRO, <http://sis.cmis.csiro.au/svg/>.
7. eSVG, <http://esvg.ultimodule.com/>.
8. TinyLine SVG Viewer, <http://www.tinyline.com/>.
9. 3GPP, <http://www.3gpp.org/>.
10. JSR 266, <http://jcp.org/en/jsr/detail?id=226>.
11. ZOOMON SVG Player, <http://www.zoomon.com/player.htm>.
12. XHTML Basic, <http://www.w3.org/TR/xhtml-basic/>
13. OMA (Open Mobile Alliance), <http://www.openmobilealliance.org/>
14. Tayeb Lemlouma, Architecture de Négociation et d'Adaptation de Services Multimédia dans des Environnements Hétérogènes, Thèse de doctorat, 9 juin 2004, INPG, Grenoble, France.
15. JavaServer Faces, Sun microsystems, <http://java.sun.com/j2ee/javaserverfaces/index.jsp>