

Comparaison de méthodes de transport de séquences de graphiques animés

Cyril Concolato

Jean Le Feuvre

Jean-Claude Moissinac

¹ Ecole Nationale Supérieure des Télécommunications
Paris, France

{concolato,lefeuvre,moissinac}@enst.fr

Résumé

Les formats de représentation de données multimédia tendent à être utilisés à la fois pour l'Internet, pour la télévision et pour les mobiles. Plusieurs formats sont proposés pour représenter des scènes multimédia animées et interactives. Cet article étudie les possibilités offertes par SVG 1.2 pour représenter de telles scènes et des techniques à mettre en œuvre pour une exploitation intelligente des ressources réseaux et mémoires. Ces considérations sont particulièrement importantes dans le cas de l'utilisation sur téléphone mobile.

Mots clefs

SVG, streaming, format ISO, hinting., multimedia, mobile

1 Introduction

En janvier 2003, le World Wide Web Consortium (W3C) a publié la spécification SVG – Scalable Vector Graphics, version 1.1 [1]. Cette spécification définit un langage XML pour décrire des graphismes 2D vectoriels animés, qui peuvent être combinés avec des images bitmap. Le groupe SVG travaille actuellement sur la version 1.2 du langage [3]. Les évolutions introduites font, notamment, de SVG un langage de description de scènes multimédia riches, pouvant être composées de graphique 2D, de vidéo, d'images et d'audio. Comme d'autres langages de ce genre, SVG permet de décrire comment un ensemble d'éléments graphiques et de médias sont agencés spatialement et temporellement. XHTML, BIFS, SMIL et Flash sont d'autres langages de ce type. L'acheminement efficace de contenus complexes décrits dans ces langages est un défi important. Dans le cas de SVG, ce défi est amplifié par la nature a priori monolithique de tout document XML.

Le contenu audio-vidéo consiste d'habitude en séquences de données à haut débit nécessitant la diffusion synchronisée de l'audio et de la vidéo. Un procédé simple de diffusion est le téléchargement puis la lecture du fichier téléchargé. Cela peut impliquer, selon la taille du fichier, un retard important pour l'utilisateur qui doit attendre le temps du téléchargement avant de voir le

contenu. Pour réduire ce retard, deux catégories de méthodes existent : le téléchargement progressif et le streaming.

Le téléchargement progressif est très populaire sur Internet pour voir des annonces ou des extraits de films, utilisant par exemple le lecteur QuickTime [4]. Dans ce scénario, un fichier est téléchargé, mais ce fichier est formé de telle façon qu'il peut commencer à être joué en cours de téléchargement.

Le streaming est aussi très répandu, particulièrement pour la distribution d'évènements en direct. Dans ce cas, une information de configuration est envoyée avant le contenu proprement dit, utilisant des protocoles tels que SDP [5], puis des flux de médias sont transportés, à l'aide de protocoles comme RTP [6] ou MPEG 2 TS [7]. Le broadcasting peut être vu comme un cas particulier de streaming, avec un grand nombre d'utilisateurs, mais sans canal de retour.

Dans cet article, nous étudions comment du contenu multimédia basé sur des descriptions de scènes SVG peut être acheminé. À cette fin, nous décrivons dans la Section 2, quelques possibilités utiles du langage SVG et proposons une classification, relative à la distribution, de scènes SVG. Dans la Section 3, nous présentons de nouvelles façons de livrer des scènes SVG basées sur cette classification et illustrons ces principes sur la diffusion de longues séquences de dessin animé. Dans la Section 4, nous proposons quelques mesures de l'efficacité de ces méthodes de distribution. Finalement, dans la Section 5, nous concluons cet article et présentons les travaux futurs.

2 Classes de contenus SVG

La distribution de contenus multimédia exige la distribution de la description de scène (partie graphique et textuelle de la présentation ainsi que la description de l'agencement spatial et temporel des médias) et celle des médias qui la composent. Si la scène contient du graphique animé (ou du texte évolutif, tel des sous-titres), plusieurs états successifs de la scène doivent être fournis, mais un état initial de la scène doit être livré avant les

médias. Cela est nécessaire au lecteur multimédia pour paramétrer ses canaux de communication pour les flux de médias et d'initialiser les structures de décodage (zone d'affichage, mémoires tampons des décodeurs).

Pour comprendre comment acheminer une description de scène multimédia SVG, nous introduisons ici quelques notions. Nous décrivons la particularité de scènes SVG en ce qui concerne la distribution et proposons une classification qui sera alors utilisée pour proposer des mécanismes de distribution appropriés.

Une scène SVG peut être animée en utilisant des éléments d'animation SMIL, comme 'set' ou 'animate', ou en utilisant des langages de script, comme ECMAScript [8] et des interfaces pour la manipulation de documents XML, comme DOM [9].

De telles scènes SVG permettent de décrire de longues séquences de graphiques animés, éventuellement composés avec de l'audio et de la vidéo. Il paraît naturel d'appliquer à de telles scènes les techniques utilisées pour acheminer de la vidéo. Mais comme une scène SVG est un document XML, elle est normalement utilisée comme un tout infractionnable. Les techniques de distribution de la vidéo ne peuvent donc pas s'appliquer simplement.

Tandis que des médias audio et vidéo sont par nature faite de fragments facilement identifiables, les documents XML ne le sont généralement pas. Dans la plupart des cas, les documents XML doivent être entièrement analysés syntaxiquement avant d'être exploités.

Pour contourner cette difficulté, SVG définit la notion d'affichage progressif qui permet à un lecteur SVG de charger progressivement le document et d'afficher au fur et à mesure ce qui a été chargé. Cette technique est semblable à ce que des navigateurs Web font quand ils lisent des pages HTML : l'affichage est progressif et simultané au chargement. Dans SVG, l'affichage progressif est le comportement par défaut.

La combinaison de l'affichage progressif et des éléments SMIL pour l'animation nous permet de proposer une classification des contenus SVG, relativement aux possibilités de distribution continues et progressives de séquences. Nous considérons deux classes de scènes : les scènes qui peuvent être fragmentées temporellement et les autres.

Une scène SVG est fragmentable temporellement si le document peut être divisé en fragments XML respectant les contraintes suivantes. Chaque fragment est formé d'instructions de composition/rendu datées (elles possèdent une estampille temporelle, une indication de l'instant où elles doivent être utilisées). Ces instructions peuvent concerner : la définition de nouveaux éléments

graphiques, qui doivent être injectés dans l'arbre de scène à partir d'un certain instant; la définition d'animations (modifications de paramètres graphiques, de position...), qui doivent commencer à un instant précis; et l'effacement d'éléments. Ce dernier type de commande exploite l'élément 'discard' introduit dans SVG 1.2. Finalement, toutes les instructions appartenant au même fragment doivent avoir la même information de temps. Par analogie avec un vocabulaire utilisé en vidéo numérique, nous définissons un fragment XML respectant ces contraintes comme une unité d'accès (Access Unit – AU) et l'information de timing associée comme une estampille temporelle MPEG-2, un CTS dans MPEG-4 ou un NPT dans RTP. Selon cette définition, nous pouvons former un flux SVG comme la séquence ordonnée temporellement des AU. La figure 1 illustre la fragmentation d'un document SVG simple dans un flux SVG, fait de quatre AU.

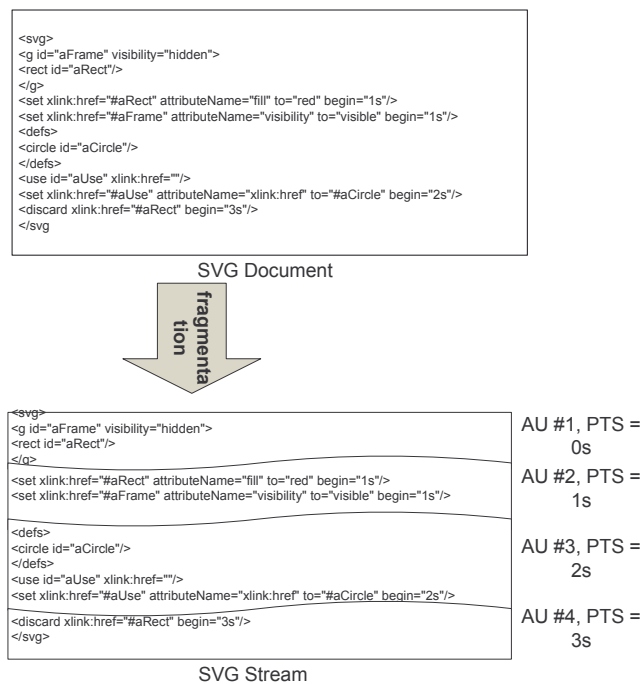


Figure 1: fragmentation en Access Unit d'une séquence SVG

Nous considérons deux classes de contenus SVG : les contenus fragmentables temporellement, à partir desquels un flux peut être créé; et les autres contenus, qui doivent être traités comme des documents monolithiques, comme des fichiers. Il faut noter que les deux classes de contenu peuvent décrire des scènes animées.

En prenant en compte cette division des contenus SVG, la durée de la scène et l'utilisation éventuelle de médias externes (audio, vidéo), la Table 1 propose le mécanisme

de distribution le plus approprié pour chaque classe et sous-classe de contenus SVG.

Tableau 1: classes de contenus SVG et mode de distribution

Classes de contenus SVG	Mécanisme de Distribution Appropriés
Scène longue ou courte, non fragmentable, n'utilisant pas de médias continus (audio, vidéo)	La scène peut seulement être téléchargée, parce qu'elle ne peut pas être fragmentée. Une longue scène devrait être restituée en utilisant l'affichage progressif
Scène longue, non fragmentable, utilisant des médias continus externes (audio, vidéo)	La scène et les médias devraient être acheminés séparément pour permettre la lecture des médias avant que la scène ne soit entièrement récupérée. La scène devrait être livrée comme un fichier simple. Les médias peuvent utiliser n'importe quelle méthode de distribution, utilisant le téléchargement progressif, le streaming ou le broadcasting
Scène courte, non fragmentable, utilisant des médias continus externes (audio, vidéo)	Le téléchargement progressif (voir la Section 3.1.1) et le streaming (voir la Section 3.1.2 de Section) peuvent être utilisés.
Scène longue ou courte, fragmentable temporellement, utilisant éventuellement des médias continus externes (audio, vidéo).	N'importe quelle méthode de distribution peut être utilisée (incluant celles de la section 3.1) mais si la scène est longue, elle devrait être streamée (nouvelle technique proposée dans la Section 3.2.).

3 Acheminement de contenus SVG

Cette section présente les détails techniques de quelques méthodes intéressantes pour la distribution de scènes multimédia SVG. Elle ne détaille pas les méthodes existantes (ex : chargement de fichier), mais se concentre sur de nouvelles techniques permettant d'acheminer une scène multimédia, accompagnée de ses éventuels médias continus, avec un meilleur contrôle de l'utilisation des ressources de lecteur, du déroulement temporel et de l'exploitation du réseau d'acheminement.

3.1 Scènes SVG non fragmentables temporellement

Cette section décrit deux méthodes pour acheminer des scènes SVG non fragmentables. La contrainte pour leur distribution consiste en ce que, puisque l'état initial ne peut pas être extrait du reste de la scène, la scène entière doit être livrée d'avance pour que le lecteur assure son paramétrage (ressources réseau, ressources mémoire).

3.1.1 Fichiers stockés et chargement progressif

La solution proposée est ici d'utiliser les fonctionnalités offertes par l'ISO Base Media File Format [10] pour stocker la scène et les médias dans un fichier autorisant le téléchargement progressif simultané à la lecture de ce fichier.

La spécification de l'ISO Base Media File Format est la base d'un ensemble de formats de fichier multimédia (MP4, 3GP, MJ2). Ces fichiers se prêtent bien au stockage de données multimédia. De tels fichiers sont structurés en utilisant des éléments de bloc appelés des Box (boîte). Dans un fichier conforme, l'information nécessaire pour comprendre le fichier (c'est-à-dire le type du fichier; le nombre de flux qu'il contient; le type, la taille, la durée et l'information de configuration pour chaque flux) sont contenus dans les boîtes de niveau supérieur (File Type Box et Movie Box). Les données d'Access Unit sont stockées dans la Media Data Box. On peut construire des fichiers ISO pour le téléchargement progressif en stockant la Movie Box et la File Type Box au début du fichier et en entrelaçant les Access Unit dans la Media Data Box, placée à la fin du fichier.

L'amendement 1 du format de fichier ISO spécifie comment inclure des données complémentaires dans des fichiers ISO en utilisant les boîtes Meta Box et XML Box. Le plus évident est alors de stocker le contenu de SVG dans une XML Box. Pour permettre le téléchargement progressif de tels fichiers, une Meta Box doit aussi être placée au début, avant toute Media Data Box. La figure 2 décrit la structure d'un tel fichier.



Figure 2 – Structure d'un fichier ISO pour le téléchargement progressif d'un contenu SVG non fragmentable accompagné de médias continus (A:audio, V:video)

3.1.2 Streaming

Nous proposons ici une méthode d'acheminement de flux SVG qui s'appuie sur les protocoles internet RTP, RTSP et SDP. Cette solution est adaptée de la spécification du streaming de MPEG 4 sur IP [11]. Puisque nous considérons le cas où le contenu SVG n'est pas fragmentable temporellement, cela n'a aucun sens de chercher à le streamer sur un canal RTP. De plus, puisque la scène SVG est nécessaire pour le paramétrage initial du lecteur, nous proposons d'envoyer la scène dans la description SDP avec l'information de session, en utilisant "a=" et l'utilisation d'un codage base64 pour acheminer le SVG, tandis que les médias continus seront streamés de façon classique.

3.2. Scènes SVG fragmentables temporellement

L'ISO Base Media File Format a été initialement conçu pour le stockage de données de médias continus comme des flux audio ou vidéo. Cependant, avec la définition donnée dans la Section 2 d'un flux SVG, il est facile de stocker le contenu SVG dans un fichier ISO. Nous proposons en effet de stocker les fragments XML décrivant la scène SVG tel que montré dans la Figure 4, de la même manière que des fragments audio ou vidéo (AU), c'est-à-dire dans une piste du fichier ISO. En utilisant cette technique, il est possible de profiter des fonctionnalités du format de fichier ISO ; par exemple, il est possible d'intercaler les fragments de scène SVG avec les AUs des médias continus. Cette méthode de stockage, permet le téléchargement progressif du fichier résultant sans nécessiter la longue attente que pouvait constituer le chargement initial de la scène SVG (cf 3.1).



Figure 3 – Structure d'un fichier ISO pour le téléchargement progressif d'un contenu SVG fragmentable accompagné de médias continus (A:audio, V:video)

Le hinting est un processus complémentaire pour préparer le contenu SVG en vue de sa transmission sur la base de mécanismes introduits par l'ISO. Cela consiste à découper le contenu en morceaux bien identifiés, associés à un instant précis, prêts à constituer des paquets à SVG à acheminer sur le réseau à un instant connu. En adaptant la technique générique de hinting habituel de l'audio et de la vidéo, au contenu multimédia entier, y compris le SVG proprement dit, il devient possible de streamer l'ensemble à l'aide du protocole RTP, avec un serveur standard capable de servir des fichiers ISO. Nous proposons d'étendre SDP pour permettre la description de l'utilisation de flux SVG en attribuant à l'item 'm' du SDP le MIME TYPE image/svg+xml.

4 Résultats

Cette section présente les résultats des tests de distribution qui ont été réalisés en utilisant les méthodes exposées dans la Section 3. Il décrit d'abord l'environnement logiciel qui a été utilisé, puis présente alors les scènes SVG sur lesquelles les méthodes de distribution proposées ont été testées; et finalement, il donne quelques éléments d'évaluation.

4.1 Environnement logiciel

Nous avons mis en oeuvre les méthodes de distribution proposées à l'aide du logiciel Open Source GPAC [12]. Nous avons créé des fichiers ISO avec une Meta Box contenant le SVG ou des fichiers ISO avec des pistes de

SVG grâce à l'outil de GPAC appelé MP4Box. Nous avons préparé les pistes de hinting de SVG pour le streaming avec le même outil. Le serveur de streaming utilisé est Darwin [13]. Finalement, le lecteur SVG est le lecteur Osmo4 du projet de GPAC, que nous avons complété pour inclure : analyse de XML basé sur SAX; affichage de SVG conforme à SVG Tiny 1.1 et à quelques fonctionnalités de SVG Tiny 1.2, comme l'élément 'discard' ou le rendu progressif.

4.2 Etude de cas

Nous avons avec succès testé les méthodes de distribution proposées pour le contenu de SVG, sur de longues séquences de dessin animé, définies avec du graphique vectoriel SVG, comme décrit dans [14].

4.3 Mesures

Nous avons évalué les différentes méthodes de distribution en tenant compte des critères suivants : attente avant le début de l'affichage et utilisation de mémoire par le lecteur. Nous avons mesuré le retard entre la demande du contenu et le début de l'affichage. Les résultats figurent dans la Table 2. Dans cet exemple, le contenu dure 14 secondes, avec un débit moyen de 500 Kbps et un niveau maximal à 1 Mbps. La bande passante HTTP utilisée est 800 kbps.

Scénario de Distribution	Attente (ms)
Plein Fichier Chargeant alors Rendu	730
Chargement de Fichier Progressif et Rendu	80
Chargement Progressif HTTP et Rendu	120
Coulant(En continu) et Rendu Progressif	380

Table 2 - Attente de temps selon le scénario de distribution

Comme nous pouvons voir, la deuxième technique permet de commencer à jouer la séquence environ 600 ms plus tôt que la première technique. Pour les autres techniques, le temps d'attente est semblable à la deuxième, mais la différence porte sur le temps de paramétrage initial lié au réseau et au remplissage du tampon de stockage temporaire (environ 100 ms).

Nous avons mesuré la consommation de mémoire pendant la réception et la lecture du contenu. La figure 4 montre l'utilisation de mémoire en fonction du temps pour les quatre techniques précédentes. Les axes de temps ont été alignés pour que l'instant de début d'affichage, nommé t_0 , coïncide.

L'avantage du streaming apparaît clairement sur cette figure, en particulier pour des terminaux à capacité réduite comme des mobiles. Nous avons ainsi proposé un procédé efficace d'acheminement de longues séquences de graphiques animés vers des mobiles.

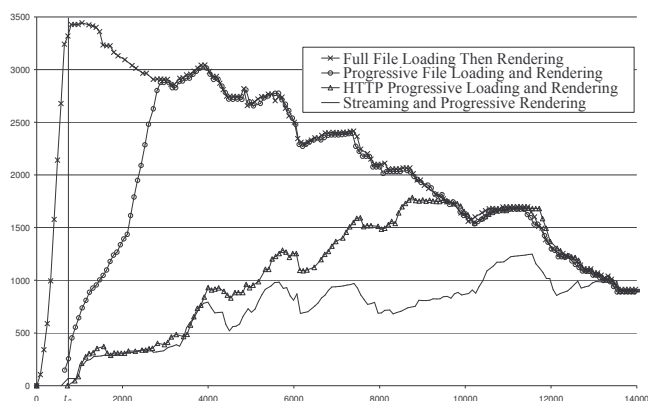


Figure 4 Occupation mémoire dans le lecteur en fonction de la méthode d'acheminement d'une séquence d'animation décrite avec SVG

5 Conclusion

Dans cet article, nous avons défini la notion d'Access Unit sur du SVG et avons défini des classes de contenu SVG en vue de leur distribution. Nous avons présenté comment distribuer des scènes SVG en utilisant des mécanismes de synchronisation habituels. Nous avons montré qu'en utilisant ces mécanismes et en les adaptant aux contenus SVG, il est possible de réduire le temps d'attente requis pour visualiser le SVG. Nous avons vu aussi que notre méthode permet l'utilisation de moins de mémoire sur le lecteur, notamment grâce à l'utilisation conjointe de l'élément 'discard' introduit dans SVG 1.2 et à un mécanisme de régulation côté serveur. Dans le futur, nous voudrions examiner les mécanismes de compression permettant de réduire l'utilisation de bande passante pour le streaming de SVG. Nous voudrions aussi évaluer les mécanismes d'animation SVG comparés à ceux utilisés dans MPEG 4 LAsER [15].

6 Remerciements

Le travail décrit dans cet article a été partiellement financé par la Commission européenne au cours du projet IST DANAE [16].

Références

- [1] Scalable Vector Graphics (SVG) 1.1 Specification, W3C Recommendation 14 January 2003, <http://www.w3.org/TR/SVG>
- [2] A. Quint, "Scalable vector graphics", Multimedia, IEEE Volume 10, Issue 3, pp. 99-102, July-Sept. 2003.
- [3] Scalable Vector Graphics (SVG) Tiny 1.2 Specification, W3C Working Draft, Dec. 2005, <http://www.w3.org/TR/SVGMobile12>
- [4] QuickTime Player, <http://www.apple.com/quicktime>
- [5] SDP: Session Description Protocol, <http://www.ietf.org/rfc/rfc2327.txt>
- [6] RTP: A Transport Protocol for Real-Time Applications, <http://www.ietf.org/rfc/rfc3550.txt>

- [7] MPEG-2 Systems White Paper, P. Schirling, <http://www.chiariglione.org/mpeg/technologies/mp02-ts/index.htm>
- [8] ECMAScript Language Specification, 3rd edition (December 1999), Standard ECMA-262, <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [9] Document Object Model (DOM) Level 3 Core Specification, Version 1.0, W3C Recommendation 07 April 2004, <http://www.w3.org/TR/DOM-Level-3-Core>
- [10] MPEG-4 File Formats white paper, D. Singer, M. Zubair Visharam, <http://www.chiariglione.org/mpeg/technologies/mp04-ff/index.htm>
- [11] RTP Payload Format for Transport of MPEG-4 Elementary Streams, <http://www.rfc-editor.org/rfc/rfc3640.txt>
- [12] GPAC Open Source Project, multimedia framework based on the MPEG-4 Systems standard, <http://gpac.sourceforge.net>
- [13] Darwin Streaming Server, <http://developer.apple.com/darwin/projects/streaming>
- [14] C. Concolato, J.-C. Moissinac, J.-C. Dufourd, "Representing 2D Cartoons using SVG", Proceedings of SMIL Europe 2003, Paris, Feb. 2003
- [15] MPEG-4 LAsER, www.mpeg-laser.org
- [16] IST DANAE Project, Dynamic and Distributed Adaptation of Scalable Multimedia Content in a Context-Aware Environment, <http://danae.rd.francetelecom.com>