

Composition d'objets multimédia à base d'opérateurs

Stéphane Lo Presti

Didier Bert

Andrzej Duda

LSR-IMAG - Grenoble, France

BP 72, F-38402 Saint-Martin d'Hères Cedex

Tel +33 476.82.72.23 - Fax +33 476.82.72.87

E-mail: {Stephane.Lo-Presti, Didier.Bert, Andrzej.Duda}@imag.fr

Résumé

Les présentations multimédia sont définies par une composition spatio-temporelle complexe d'objets monomédia. Des standards et des propositions existent pour spécifier ces compositions. Les langages proposés sont d'un bas niveau d'abstraction et ont une sémantique opérationnelle. Nous proposons un langage de haut niveau qui permet d'assembler des objets à partir d'opérateurs dont la sémantique est définie formellement dans le cadre d'un nouveau modèle temporel. Il est ensuite possible de manipuler et présenter ces scénarios.

Mots Clef

Présentation multimédia, composition, opérateurs, modèle temporel.

1 Introduction

Depuis des années déjà, les médias audio-visuels constituent un vecteur majeur de l'information. Chaque média fait appel à ses propres techniques, tant vis-à-vis du contenu que du contenant, c'est-à-dire le format de stockage des données.

Les objets monomédia encapsulent les données, le plus souvent audio ou visuelles. On assemble ensuite ces objets afin de les organiser selon diverses dimensions, spatiale, temporelle ou basée sur le contenu. C'est de cette manière que l'on crée une présentation, ou objet, multimédia qui pourra être présenté à un utilisateur.

Un objet audiovisuel est l'exemple le plus commun de présentation multimédia. Une vidéo est la mise en parallèle, en les synchronisant finement, d'une animation et d'une bande son lui correspondant. Pour des applications plus spécifiques, on peut même faire intervenir de nouveaux médias, comme le toucher ou le goût. Il est aussi indispensable de rendre ces objets interactifs, par le biais de périphériques de plus en plus divers et variés (joystick, équipement de réalité virtuelle, écrans tactiles, interface vocale, etc.). Ces objets prennent donc une forme de plus en plus complexe qu'il faut pouvoir maîtriser.

2 Création de présentations multimédia

Afin de pouvoir créer des présentations multimédia, de nombreux travaux ont proposé des approches de composition des objets monomédia.

L'organisme de standardisation ISO a créé le standard HyTime (Hypermedia/Time-based Structuring Language) [9]. Les scénarios écrits dans ce langage placent sur des axes spatiaux et temporels les éléments à composer, selon une sémantique complexe qui rend l'écriture des scénarios difficile.

Plus tard, le standard MHEG (Multimedia and Hypermedia information coding Expert Group) [7] a permis d'élever le niveau d'abstraction en utilisant un modèle à objet afin d'encapsuler les données et leur composition. La composition temporelle repose, d'une part, sur l'usage d'attributs BEGIN et END pour spécifier la mise en séquence ou en parallèle des objets, et d'autre part sur les attributs START et STOP pour préciser d'autres égalités entre points temporels. La syntaxe du langage ne permet pas d'exprimer de manière concise les scénarios car beaucoup d'attributs sont attachés aux objets.

Le standard MPEG (Moving Picture Coding Expert Group) de l'ISO a pour objectif l'encodage efficace du son et d'animation numériques et de leurs combinaisons. La partie MPEG-4 Version 2 [4] de ce standard définit le codage d'objets dans les scènes et aussi les moyens de composer ces objets grâce au langage BIFS (Binary Format for Scene Description). La partie MPEG-7 [6] spécifie la description du contenu multimédia grâce au langage DDL (Description Definition Language).

Flash [10] de la société Macromedia permet de créer des animations vectorielles au sein desquelles l'on mêle des sons et des boutons. Orienté vers le web, ce format binaire de fichiers a permis d'animer et de rendre interactives les pages web et est associé à l'outil Flash. La composition temporelle est rudimentaire, car elle consiste à organiser soi-même l'arrangement des divers

objets dans le temps.

IBM a défini le format de fichier HotMedia [5] afin d'intégrer des données hétérogènes, y compris des programmes. On peut attacher des actions à ces données. Une application d'édition lui est aussi associée. Là encore c'est un modèle temporel linéaire qui est utilisé.

Plus récemment dans le cadre du World Wide Web, le W3C a recommandé le langage SMIL 2.0 (Synchronized Multimedia Integration Language) [1]. SMIL est un langage qui permet d'indiquer la composition d'éléments multimédia dans l'espace et le temps grâce à des balises. Il est aussi possible de lier les débuts et fins de certains éléments du scénario grâce à des attributs de balises et de créer des conteneurs temporels en imbriquant les balises. Cette hétérogénéité de la spécification temporelle rend la sémantique du scénario sophistiquée. Elle dépend en effet du mélange des balises et des attributs qui lient les instants d'éléments et qui définissent leurs répétitions.

3 Langage de composition de présentations multimédia

C'est dans ce contexte que nous proposons un langage de haut niveau d'abstraction permettant de composer des objets multimédia.

Ce langage permet d'assembler des objets multimédia, dits "primitifs", par le biais d'opérateurs, pour l'instant uniquement temporels. Dans l'exemple de programme qui suit, deux objets représentant un texte et une vidéo sont créés et sont présentés en séquence:

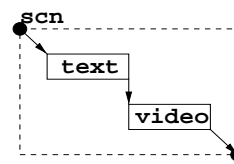
```
Text text = new Text("example.txt").
Video video = new Video("example.mpg").
Scenario scn = text ; video.
```

L'opérateur `;` indique comment présenter dans le temps les objets `text` et `video` et construit un nouvel objet composé `scn`.

Dix opérateurs sont prédéfinis [2]. On y retrouve la mise en séquence (notée `;`), la mise en boucle (`loop`) et diverses façons de composer en parallèle des intervalles. Parmi ces façons, on trouve les opérateurs `master`, `min` et `max` (qui arrêtent l'objet composé respectivement lorsque le premier argument, le premier objet à s'arrêter et le dernier objet à s'arrêter se termine) ou `alt` (qui permet de faire un choix conditionné sur un ensemble d'intervalles selon la durée) ou encore `par`, `∨` et `∧` (le premier présentant en parallèle des objets définis au sein de "branches" et les deux autres permettant respectivement d'arrêter une branche depuis une autre branche ou d'être arrêté par la fin d'une branche).

Chaque opérateur a une sémantique bien définie, exprimée à l'aide d'un schéma et de façon formelle grâce à un ensemble de règles causales [2]. Ces règles causales lient les points temporels, c'est-à-dire leurs débuts et

fins, des arguments de l'opérateur conformément à la sémantique de l'opérateur.



Ainsi dans l'exemple précédent, la fin de l'objet `text`, notée $end(text)$, est liée au début de l'objet `video`, noté $start(video)$, par la règle causale:

$$end(text) \mapsto start(video)$$

ce que l'on représente par le schéma ci-dessus.

Les points temporels sont groupés en deux catégories: les points observés (notés `beg` et `end`) et les points générés (notés `start` et `stop`). Ces points temporels sont respectivement en partie gauche, comme "cause", ou droite, comme "conséquence", d'une règle causale. Les objets primitifs peuvent avoir des durées indéterminées puisque les règles causales lient des points temporels, qui s'abstraient de l'instant d'occurrence du point temporel.

Les règles causales peuvent être conditionnées afin d'exprimer une sémantique d'opérateur dépendant de comparaisons entre points temporels. Ceci permet d'exprimer un comportement dynamique correspondant à plusieurs cas d'exécution possibles. Les conditions sont composées de comparaisons des points temporels selon une relation d'ordre particulière, qui prend en compte les règles de causalité.

Les opérateurs `;`, `loop`, `max`, `alt` et `par` permettent de créer une grande variété de scénarios tout en gardant une sémantique simple. Les autres opérateurs introduisent des situations plus complexes.

La présentation, des objets composés consiste à organiser dans le temps le rendu des différents objets primitifs selon la sémantique du scénario. Cette présentation peut à son tour être utilisée comme argument d'un opérateur dans une autre présentation. L'auteur peut ainsi définir incrémentalement son scénario et réutiliser son travail.

4 Sémantique des scénarios

La sémantique de tout scénario est l'ensemble des règles causales définies par les opérateurs utilisés. Elle est obtenue à l'issue d'un processus de normalisation.

Ce processus détermine les règles causales des objets composés à partir de celles de leurs arguments. Il manipule des structures de niveau, noté $\psi.S$ où ψ est une sous-expression du scénario et S est l'ensemble de ses règles de causalité. La normalisation est définie par des règles d'inférence qui déterminent l'ensemble des règles causales d'un niveau ψ à partir des ensembles des niveaux inférieurs. Ces niveaux inférieurs correspondent aux sous-expressions de ψ .

Par exemple la règle (lift) rajoute, sous certaines conditions (le prédicat `normal` n'est pas ici précisé), les règles causales d'un niveau ψ' dans un niveau qui

lui est supérieur ψ , ce qui est représenté par un ajout d'une règle causale à l'ensemble ψ :

Condition	Substitution
$\psi \leq \psi' \wedge \text{normal}(\psi')$ $\wedge p \neq \text{beg} \wedge q \neq \text{end}$ $\wedge (p \mapsto q \text{ if } c) \in \psi'$	$\psi.\{ p \mapsto q \text{ if } c \}$

Les règles (p-beg) et (p-stop) remplacent respectivement les démarrage et fin d'un niveau (noté ψ') dans un niveau parent (noté ψ) par leurs valeurs données par l'ensemble de règles causales du niveau ψ' :

(p-beg)	
$\psi \leq \psi' \wedge$ $(\text{beg} \mapsto q) \in \psi'$	$\psi.\{ p \mapsto \text{start}(\psi') \text{ if } c \}$ <hr/> $\psi.\{ p \mapsto q \text{ if } c \}$
(p-stop)	
$\psi \leq \psi' \wedge$ $(p \mapsto \text{stop} \text{ if } c') \in \psi'$	$\psi.\{ \text{end}(\psi') \mapsto q \text{ if } c \}$ <hr/> $\psi.\{ p \mapsto q \text{ if } c \wedge c' \}$

Une autre partie de ces règles d'inférences sont dues à l'influence des opérateurs **master**, **min**, \searrow et \swarrow , qui ont pour particularité d'arrêter un ou plusieurs de leur arguments. Ces règles, dites de coupure, sont obtenues par instanciation sur les opérateurs d'une méta-règle, qui exprime la sémantique de l'arrêt d'un opérateur pendant son exécution. A l'issue du processus de normalisation, un scénario à base d'opérateurs est traduit en un ensemble de règles causales s'appliquant aux objets monomédia. Les conditions de ces règles causales expriment le ou les cas de figure dans lequel la règle causale s'exécute.

L'interprétation des règles causales, comprenant la définition de la relation d'ordre sur les points temporels, et la définition des points temporels forment un modèle qui sous-tend les scénarios. Le modèle permet de s'assurer la cohérence temporelle des scénarios définis dans ce langage.

Sans exprimer la sémantique des scénarios, il est possible de travailler sur leur structure afin d'en tirer directement des informations intéressantes. La structure arborescente des scénarios du langage permet des calculs recursifs de propriétés et de fonctions [3].

On peut par exemple adapter les scénarios aux spécificités d'une machine de présentation particulière. Ceci est possible car certains opérateurs peuvent être redéfinis en fonction des autres, comme **min** et **max**:

$$\begin{aligned} \min[A \parallel B] &= \text{par}[a : A \searrow \{b\} \parallel b : B \searrow \{a\}] \\ \max[A \parallel B] &= \text{par}[a : A \parallel b : B] \end{aligned}$$

Notons que ces opérateurs sont fournis afin de faciliter l'écriture des scénarios. Divers scénarios peuvent être écrits différemment mais avoir la même sémantique. On peut donc définir une égalité sur les termes qui permet de remplacer certaines parties d'un scénario par un autre arrangement d'opérateurs. Ce choix d'un opérateur plutôt qu'un autre se fera en fonction des spécificités de la machine de présentation.

D'autre part, les durées des objets primitifs, lorsqu'elles sont connues, peuvent permettre de simplifier l'ensemble de règles causales. Ceci correspond à une optimisation qui est utile avant d'entamer le rendu de la présentation multimédia. De plus il est possible de répondre à des questions comme "Y a-t-il des objets dont la présentation n'est pas arrêtée?" ou bien "Dans quel cas, tel objet ne sera pas présenté?" en définissant des fonctions adéquates [3].

Une traduction vers des ensembles de relations d'ordre sur les points temporels permet d'exploiter d'autres techniques associées (formatage temporel [11], vérification de relations, etc.) en affaiblissant l'information des scénarios. En particulier, la causalité et l'organisation conditionnelle des règles causales sont alors perdues.

5 Compilation des scénarios

Par ce langage à base d'opérateur, l'auteur de présentations multimédia a donc le moyen de spécifier un scénario en n'indiquant que les éléments utiles: objets monomédia et opérateurs de synchronisation. Les détails, tels que le rendu des divers objets monomédia ou la mise en œuvre de la synchronisation, sont gommés à ce niveau.

Ces aspects sont gérés au niveau de la plate-forme d'exécution des scénarios. Cela correspond à une machine de présentation qui peut être construite pour des plate-formes aussi diverses qu'un ordinateur personnel, un terminal télévisuel numérique ou bien un téléphone mobile de troisième génération.

L'implémentation de cette machine de présentation fait intervenir des transformations propres à chaque plate-forme (dégradation de contenu, réservation de ressources, etc.) afin de rendre de la manière la plus adéquate le scénario.

Toute machine de présentation doit respecter le modèle temporel sous-tendant le langage afin d'assurer la cohérence temporelle des scénarios joués. Le modèle implique ainsi certaines contraintes, comme tenir à jour une liste des objets couramment présentés ou bien stoppés.

Nous avons spécifié à partir de ces contraintes une machine générique de présentation qui est structurée en trois parties (rendu, exécution et objets) [2]. La

partie objet peut reposer sur une base de donnée et la partie rendu peut correspondre à un moteur de rendu, comme on en trouve en infographie ou dans les jeux vidéos.

La partie exécution pourrait gérer les règles causales définies dans le scénario par les opérateurs. Cette gestion est peu efficace, car elle nécessite de manipuler des formules logiques coûteuses en temps de résolution. Les opérateurs sont donc compilés en des règles d'exécution. Une règle d'exécution consiste en une ou plusieurs instructions lancées en même temps. Une instruction peut être une causalité liant un point temporel à un autre, mais sans condition booléenne contrairement aux règles causales. Ce peut être aussi une instruction de branchement avec laquelle le comportement dynamique des opérateurs est traduit. Chaque opérateur est réécrit comme une séquence de règles d'exécution et les opérateurs utilisés dans le scénario correspondent à des contextes, c'est-à-dire des groupes de règles d'exécution dont la machine de présentation déclenchera le démarrage et l'arrêt.

Nous comptons traduire cette machine de présentation et ces règles d'exécution vers une machine de présentation préexistante au sein de nos équipes de recherche [8]. Cette dernière machine reposant sur un langage de bas niveau, la traduction fait intervenir des paramètres externes, précisant par exemple la synchronisation fine entre les divers objets.

6 Conclusion

Nous avons proposé un langage pour spécifier des présentations multimédia basé sur des opérateurs temporels de haut niveau d'abstraction. Ces opérateurs composent les objets monomédia selon une sémantique précise et permettent à l'auteur de créer sa présentation de façon incrémentale. Les opérateurs sont au nombre de dix. La sémantique des opérateurs est définie à partir de règles causales qui s'appliquent sur les points temporels des arguments de l'opérateur. Celle des scénarios fait intervenir un processus de normalisation. Les points temporels et les règles causales forment le cadre d'un modèle temporel qui permet de manipuler les scénarios, procéder à des calculs sur eux et prouver des propriétés.

Le travail en cours consiste en l'écriture d'un compilateur pour le langage à base d'opérateur en se basant sur le travail de spécification d'une machine de présentation. Nous voulons aussi augmenter notre langage d'une spécification des relations spatiales qui soit cohérente avec l'approche par opérateurs.

Références

[1] Jeff Ayars, Dick Bulterman, Aaron Cohen, and al. Synchronized Multimedia Integration Language (SMIL 2.0). Technical report, W3C: World Wide

Web Consortium, August 2001. REC-smil20-20010807.

- [2] Didier Bert and Stéphane Lo Presti. Algebraic Operators and Causal Relations for Modeling Multimedia Presentations. In *Presentation at the Symposium on Modal and Temporal Logic-based Planning for Open Networked Multimedia Systems (Boston)*, November 1999. To appear in IOS Press.
- [3] Didier Bert and Stéphane Lo Presti. Algebraic Specification of Operator-based Multimedia Scenarios. In Springer-Verlag, editor, *Recent Trends in Algebraic Development Techniques, WADT'99. Selected Papers. LNCS 1827*, pages 382–399, 2000.
- [4] Rob Koenen. Overview of the MPEG-4 Standard. Technical Report ISO/IEC JTC1/SC29/WG 11 N4030, ISO/IEC, March 2001.
- [5] Keeranoor G. Kumar, James S. Lipscomb, Arun Ramchandra, and al. The HotMedia Architecture: Progressive and Interactive Rich Media for the Internet. Technical Report RC21519(97069)6JUL1999, IBM Internet Media Group, July 1999.
- [6] José M. Martínez. Overview of the MPEG-7 Standard. Technical Report ISO/IEC JTC1/SC29/WG 11 N4031, ISO/IEC, March 2001.
- [7] T. Meyer-Boudnik and W. Effelsberg. Mheg explained. *IEEE Multimedia*, 2(1):26–38, Spring 1995.
- [8] F. Rousseau and A. Duda. An Execution Architecture for Synchronized Multimedia Presentations. In *Proceedings of the Third European Conference on Multimedia Applications, Services and Techniques (ECMAST'98)*, pages 42–55, Berlin, Germany, 1998.
- [9] International Standard. Information technology hypermedia/time-based structuring language (HyTime), April 1992.
- [10] Macromedia team. Macromedia Flash file format (SWF) SDK (Software Development Kit). Technical report, Macromedia, 2001.
- [11] M. Vilain and H. Kautz. Constraint Propagation Algorithms for Temporal Reasoning. In *Fifth National Conference on Artificial Intelligence*, pages 377–382, Philadelphia, 1986.