

PROGRAMMATION ÉVÉNEMENTIELLE DE PARTITIONS MUSICALES INTERACTIVES.

D. Fober, S. Letz, Y. Orlarey

{fober,letz,orlarey}@grame.fr

Grame - Centre national de création musicale

RÉSUMÉ

INSORE est un environnement pour la conception de partitions musicales interactives qui intègre un système original d'interaction basé sur des événements et sur un langage de script permettant d'associer des messages arbitraires à ces événements. Initialement conçu pour être piloté via OSC, la version textuelle des messages OSC s'est rapidement constituée en format de stockage, puis étendue en un langage de script permettant une plus grande souplesse dans la description des partitions et des interactions avec ces partitions. Cet article présente ce langage de script et illustre notamment ses capacités à décrire des interactions sous forme événementielle, tout en restant dans l'espace temporel.

1. INTRODUCTION

INSORE est un environnement pour la conception de partitions musicales qui propose une approche de la notation musicale étendue à des objets graphiques arbitraires [4], incluant la représentation de l'interprétation musicale. Cette approche nouvelle de la notation permet également la représentation des relations temporelles entre les objets de la partition [3].

La conception d'INSORE fait réponse à une carence des outils informatiques actuels pour la notation de la musique, qui n'ont pas évolué en proportion des nouvelles formes de création musicale (voir par exemple [8] [5]). En particulier, il y a un fossé significatif entre les musiques interactives et la manière statique de les noter. Les technologies d'aujourd'hui permettent le calcul et l'interaction avec la musique en temps réel, mais la dimension symbolique de la notation est généralement exclue du processus d'interaction excepté dans des travaux très récents [6].

Conçu pour être piloté par des messages OSC [9], INSCORE se prête naturellement à une utilisation interactive. Cette approche de la programmation de partition par messages est également déclinée en un langage de script, basé sur une extension des messages OSC, et fournissant des primitives d'interaction reposant sur des notions d'événements. Ces événements sont similaires à ceux typiquement disponibles pour la gestion des interfaces utilisateurs (par exemple en Javascript via le DOM [7]), avec une extension dans le domaine temporel.

Cet article présente tout d'abord deux exemples de partitions interactives, mises en oeuvre dans des créations récentes. Il présente ensuite le système de messages et les événements d'interaction, qui permettent à la fois de décrire la partition et d'interagir avec elle. Des exemples d'usages viennent enfin illustrer les capacités d'expression du système.

2. PARTITIONS INTERACTIVES

Les musiques interactives sont aujourd'hui l'objet d'intérêts artistiques et scientifiques convergents. L'interaction soulève des problèmes à la fois pour la composition, la description et pour l'exécution des oeuvres. Ces problèmes sont abordés dans les aspects temporels de la partition interactive [1] ou du contrôle [2], et sont liés au calcul de l'oeuvre.

Pour la notation de l'oeuvre interactive, deux créations récentes ont mis en oeuvre INSCORE pour créer des partitions dynamiques avec des approches originales, qui témoignent également des besoins de la création contemporaine. Il s'agit de *Calder's Violin* et de *Alien Lands*.

2.1. Calder's Violin

Calder's violin, composé par Richard Hoadley, a été créé à Cambridge en Octobre 2011. La pièce est définie comme "*musique automatique pour violon et ordinateur*" et présente dynamiquement au musicien, de la notation musicale symbolique générée de manière algorithmique (figure 1). Cette partition est alors jouée par le musicien en parallèle des sons générés par l'ordinateur ou par d'autres musiciens. Les technologies utilisées reposent sur SuperCollider pour l'environnement de programmation audio et sur INSCORE pour la notation. Pour plus de détails se référer à [6].

2.2. Alien Lands

Alien Lands est un ensemble de pièces pour percussions et quatuor à corde, composées par Sandeep Bhagwati. Dans leur version interactive, ces pièces ont été données à Montréal en Février 2011. L'utilisation d'INSORE relève de quatre catégories :

- partition traditionnelle avec tourne de page automatique,



Figure 1. Calder's Violin : exemple de partition.

- partition avec choix automatiques réalisés par l'ordinateur : ordre des mesures, sélection de lignes,
- partition complexe automatique, comprenant des éléments générés de manière algorithmique (figure 2),
- partition complexe interactive, où les éléments algorithmiques sont générés à la demande des musiciens.

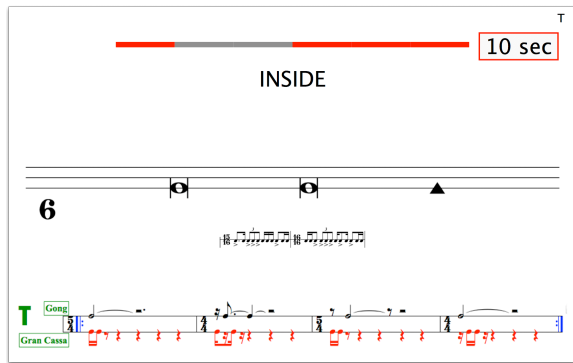


Figure 2. Alien Lands : une partition complexe automatique.

3. DESIGN DE PARTITION PAR MESSAGES

Le principe de base pour la description d'une partition, consiste à envoyer des messages OSC au système pour créer les différents composants de la partition et pour contrôler leurs attributs, aussi bien dans l'espace graphique que temporel.

3.1. Format général des messages

Le format général des messages INSCORE est illustré figure 3. Il s'agit d'une spécialisation des messages OSC qui peut être vue comme *orientée objet*, où l'adresse désigne l'objet cible du message, *method* désigne une méthode de l'objet cible et *params*, les paramètres de la méthode. Un message INSCORE peut donc être vu comme l'appel d'une méthode d'un objet de la partition.

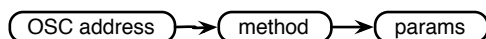


Figure 3. Format général des messages INScore.

Le système inclut des messages pour gérer les attributs graphiques des composants de la partition (position, couleur, échelle, rotations, effets...), pour gérer leurs attributs temporels (date, durée), pour exprimer les relations entre espaces graphiques et temporels, pour synchroniser des objets, pour représenter des signaux et pour gérer des événements d'interaction.

Exemple 1 Change la position *x* de l'objet *obj*. L'adresse décrit la hiérarchie des objets : *obj* est contenu dans une scène nommée *scene* qui est incluse dans l'application d'adresse ITL.

```
/ITL/scene/obj x -0.5
```

3.2. Le langage de script

Bien que prévu pour être émis sous forme de paquets sur un réseau, les messages OSC peuvent s'exprimer sous forme textuelle. C'est cette forme textuelle qui constitue le format de sauvegarde des partitions. Elle a été rapidement étendue pour en faire un langage de script.

3.2.1. Adresses étendues

Les adresses OSC sont étendues pour permettre l'émission de messages aussi bien vers INSCORE qu'à destination d'une machine et/ou application externes (figure 4). Cela permet d'initialiser à la fois la partition et les ressources externes qui peuvent y être associées.



Figure 4. Extension du schéma d'adressage.

Exemple 2 Initialisation d'une partition et d'une application externe à l'écoute du port 12000, sur une machine nommée *host.adomain.net*. Dans les scripts, le point virgule (;) est utilisé comme terminaison de message.

```
/ITL/scene/score set gmnf 'myscore.gmn';
host.adomain.net:12000/run 1;
```

3.2.2. Variables

Des variables ont été introduites pour permettre le partage de paramètres entre messages. Une variable associe un identificateur et une liste de paramètres ou une liste de messages (figure 5). Les variables peuvent être utilisées dans les paramètres des messages sous la forme *\$identificateur*.

Exemple 3 Déclaration de variables et utilisation comme paramètre de couleur. Le caractère '!' marque le début d'un commentaire.

```
color = 200 200 200;
colorwithalpha = $color 100;
msgsvr= (! une liste de messages
  localhost:7001/world "Hello world",
  localhost:7001/world "how are you?" );
/ITL/scene/obj color $colorwithalpha;
```

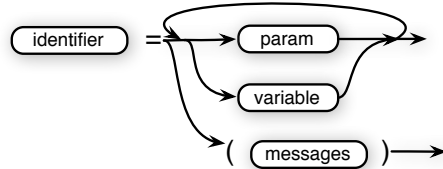


Figure 5. Variables.

3.2.3. Langages

Les scripts supportent également l'inclusion de langages de programmation comme javascript (par défaut) ou lua. Les sections correspondantes sont indiquées de manière similaire à html (figure 6). Le code est évalué au moment de la lecture du script et le résultat attendu de l'évaluation est un ensemble de messages INSCORE qui vont remplacer le script correspondant.

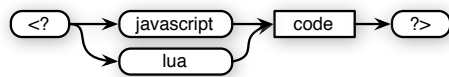


Figure 6. Langages.

4. INTERACTION ÉVÉNEMENTIELLE

Le processus d'interaction événementielle repose sur l'association de messages à des événements du système. Ces messages sont émis lorsque l'événement auquel ils sont associés se produit. Le format général des messages pour créer de telles associations est décrit en figure 7.

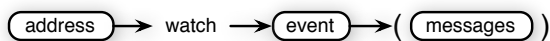


Figure 7. Format général d'un message d'interaction.

4.1. Typologie des événements

Les événements définis par le système sont d'une part des événements typiques d'interface utilisateur (tels que clics de souris par exemple) ainsi que des événements définis dans le domaine temporel (table 1)

Dans le domaine temporel, un événement est déclenché quand un objet entre ou quitte une zone temporelle (timeEnter, timeLeave) définie par deux dates, ou quand la durée d'un objet entre ou quitte une zone temporelle (durEnter, durLeave) définie par deux durées.

4.2. Variables contextuelles

Lorsqu'un événement se produit, les messages associés sont évalués, notamment parce qu'ils peuvent faire référence à des *variables contextuelles*. Ces variables concernent

Domaine graphique	Domaine temporel
mouseDown	timeEnter
mouseUp	timeLeave
mouseenter	durEnter
mouseleave	durLeave
mousemove	

Table 1. Principaux événements du système.

principalement les événements du domaine graphique et permettent d'accéder à la position de la souris au moment de l'événement dans différents référentiels (\$x \$y \$sx \$sy) ou encore à la date correspondant à cette position (\$date).

Exemple 4 Un objet qui suit les clics de la souris. La virgule (,) est utilisée comme séparateur de message dans les listes de messages.

```

/ITL/scene/obj watch mouseDown (
  /ITL/scene/obj x '$sx',
  /ITL/scene/obj y '$sy' );
  
```

5. CAS D'USAGE

5.1. Tourne de page

Une application simple des événements temporels consiste à décrire des tournes de page automatique. Un objet surveille les zones temporelles correspondants aux différentes pages et rappelle ces pages lorsque qu'il entre dans ces zones. A noter que cet objet peut aussi bien être un curseur qui se déplace sur la partition.

```

/ITL/scene/obj watch timeEnter 0 12
  (/ITL/scene/score page 1);
/ITL/scene/obj watch timeEnter 12 24
  (/ITL/scene/score page 2);
etc.
  
```

5.2. Séquences d'interactions

Les messages d'interaction décrits en figure 7 acceptent des messages arbitraires du système en tant que messages associés à un événement. Il est donc possible d'associer un message d'interaction à un événement et ainsi, de décrire des séquences d'interactions.

Exemple 5 Description d'une séquence d'interactions basées sur le clic de souris : le premier clic change la couleur de l'objet, le second modifie l'échelle, le troisième fait une rotation, le quatrième modifie également l'échelle. La profondeur de ces associations n'est pas limitée.

```

/ITL/scene/obj watch mouseDown (
  /ITL/scene/obj color 100 100 255,
  /ITL/scene/obj watch mouseDown (
    /ITL/scene/obj scale 1.4,
    /ITL/scene/obj watch mouseDown (
      /ITL/scene/obj angle 45. ,
      /ITL/scene/obj watch mouseDown (
        /ITL/scene/obj scale 0.8 ))));

```

5.3. Séquences dans le domaine temporel

La séquence d'interactions décrites ci-dessus (section 5.2) peut se décliner dans le domaine temporel en associant les changements d'état à des événements temporels et en déplaçant l'objet dans le temps. Avec cette approche, il est à la fois possible de changer l'ordre des événements, mais également de contrôler le déroulement de ces événements dans le temps.

Ce type de description permet de combiner dans une même expression, des approches événementielles, des accès non-séquentiel et du contrôle temporel.

Exemple 6 Description d'une séquence d'interactions par association à des événements temporels. Les événements temporels sont déclenchés par l'entrée de l'objet dans des zones temporelles consécutives, et dont la durée est la ronde.

```

/ITL/scene/obj watch timeEnter 1 2
  (/ITL/scene/obj color 100 100 255);
/ITL/scene/obj watch timeEnter 2 3
  (/ITL/scene/obj scale 1.4);
/ITL/scene/obj watch timeEnter 3 4
  (/ITL/scene/obj angle 45.);
/ITL/scene/obj watch timeEnter 4 5
  (/ITL/scene/obj scale 0.8);

```

5.4. Structuration de l'espace temps

Dans le cadre d'expériences pédagogiques réalisées aux Ateliers des Feuillantines, la reconstruction du canon par ton de l'Offrande Musicale de J.S. Bach a été réalisée sous forme d'un objet à la fois *partition et instrument*, et INSCORE a été utilisé pour calculer dynamiquement la représentation symbolique de la musique jouée¹. L'objet réalisé représente la structure de l'œuvre en trois dimensions (figure 8). Chaque barreau, sur lequel un capteur peut être posé, est muni d'une étiquette indiquant le chemin à suivre, et correspond à une transposition du canon.

Du point de vue de la représentation symbolique, les différentes transpositions peuplent l'espace temporel à des dates différentes, de telle sorte que sur l'instrument, l'association d'une position à une date permet de rappeler la transposition correspondante via les événements temporels d'INSCORE.

1. <http://tinyurl.com/ah3jftu>



Figure 8. L'objet *partition / instrument* de l'Offrande musicale.

6. CONCLUSION

L'association de messages à des événements se révèle être un mécanisme simple, puissant et homogène pour la description de partitions dynamiques. Il est désormais possible de décrire l'ensemble des comportements d'une partition dans le langage de script d'INSCORE. L'utilisation d'applications externes pour piloter ces comportements peut dès lors se limiter au déplacement d'objets dans le temps en associant à ces objets, des comportements liés à des espaces temporels.

Remerciements

Cette recherche a été menée dans le cadre du projet INEDIT qui est soutenu par l'Agence Nationale pour la Recherche [ANR-12-CORD-009-03].

7. REFERENCES

- [1] A. Allombert, "Aspects temporels d'un système de partitions musicales interactives pour la composition et l'exécution," Ph.D. dissertation, Université Bordeaux 1, Oct. 2009. [Online]. Available : <http://tel.archives-ouvertes.fr/tel-00516350>
- [2] A. Cont, "Antescofo : Anticipatory synchronization and control of interactive parameters in computer music," in *Proceedings of International Computer Music Conference*, ICMA, Ed., 2008.
- [3] D. Fober, C. Daudin, S. Letz, and Y. Orlarey, "Time synchronization in graphic domain - a new paradigm for augmented music scores," in *Proceedings of the International Computer Music Conference*, ICMA, Ed., 2010, pp. 458–461.
- [4] D. Fober, C. Daudin, Y. Orlarey, and S. Letz, "Interlude - a framework for augmented music scores," in *Proceedings of the Sound and Music Computing conference - SMC'10*, 2010, pp. 233–240.

- [5] J. Freeman, “Bringing instrumental musicians into interactive music systems through notation,” *Leonardo Music Journal*, vol. 21, no. 15-16, 2011.
- [6] R. Hoadley, “Calder’s violin : Real-time notation and performance through musically expressive algorithms,” in *Proceedings of International Computer Music Conference*, ICMA, Ed., 2012, pp. 188–193.
- [7] B. Höhrmann, P. Le Hégaré, and T. Pixley, “Document object model (dom) level 3 events specification,” World Wide Web Consortium, Working Draft WD-DOM-Level-3-Events-20071221, December 2007.
- [8] T. Magnusson, “Algorithms as scores : Coding live music,” *Leonardo Music Journal*, vol. 21, pp. 19–23, 2011.
- [9] M. Wright, *Open Sound Control 1.0 Specification*, 2002. [Online]. Available : http://opensoundcontrol.org/spec-1_0