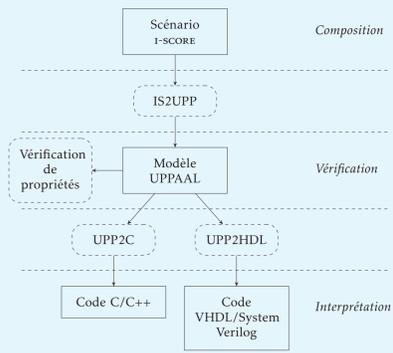


Introduction



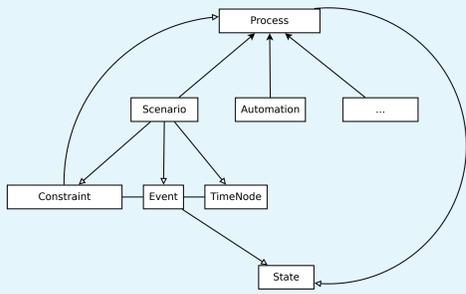
- ▶ i-score permet de décrire le temps de manière fine avec des outils issus de la programmation, pour des personnes n'ayant pas de pratique de développement logiciel.
 - ▶ Laboratoires : **LaBRI, GMEA.**
 - ▶ Entreprises : **Blue Yeti, RSF.**
 - ▶ Artistes impliqués dans conception (conception dirigée par experts).
- Le projet est issu de la fusion des séquenceurs **Virage** et **Acousmoscribe**.

Scénario interactif

Éléments du modèle :

- ▶ **Processus** : exécution via scheduler. Possède une durée propre.
- ▶ **Contrainte** : représente une durée (connue ou non). Contient des processus.
- ▶ **Évènement** : porte des conditions et relie des contraintes.
- ▶ **TimeNode** : synchronise des évènements.
- ▶ **État** : porte des données.
- ▶ **Scénario** : un processus qui agence contrainte, évènement, timenode. Permet d'obtenir une hiérarchie infinie.

Difficulté : gestion à l'édition des cas qui ne rentrent pas dans le modèle pour faciliter l'écriture.



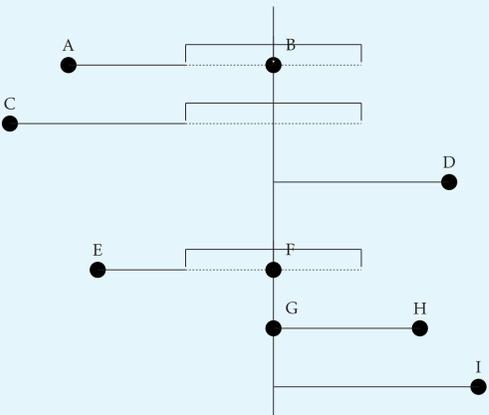
Exemple de notation

Une représentation qui contient les éléments d'un scénario.

- ▶ **Contraintes** : traits horizontaux.
- ▶ **TimeNode** : trait vertical.
- ▶ **Évènement** : point.

Notion de temps souple : un évènement peut arriver dans un intervalle donné par le compositeur.

- ▶ Par rapport aux séquenceurs traditionnels, on ne peut plus avoir de ligne de temps simple, de graduation...
- ▶ Problématique analogue au débogage : comment n'écouter qu'une partie du spectacle (au milieu par exemple)? Nécessaire lors de la création.

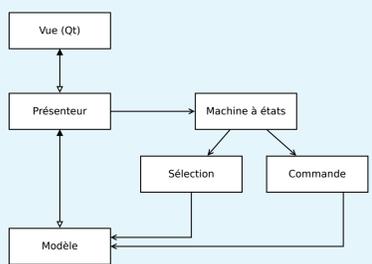


Interopérabilité

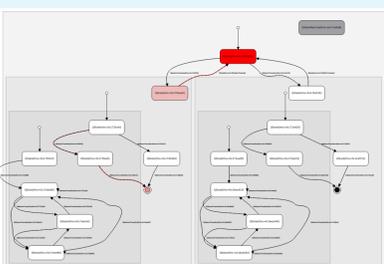
- ▶ Notion de paramètre adaptée aux créations artistiques. Organisation en arbres.
- ▶ Basé sur la spécification OSC.
- ▶ Implémentations pour OSC, Midi, Minuit (découverte), Qt...

Le protocole OSC est simple et a pu être implémenté en une semaine dans une application spécialisée pour contrôler des Metabot.

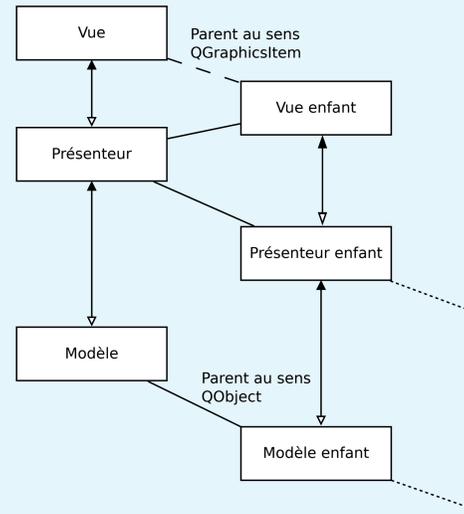
UI d'édition



- ▶ Édition répartie. Problématique : undo-redo réparti.
- ▶ Pattern pour édition à la volée fluide : machines à états interagissant avec le pattern Command (cf. figure). Utilisation de *GammaRay* (KDAB) pour débogage.
- ▶ Recherches en cours pour résolution des problèmes de contraintes qui se posent.
- ▶ En retour, fonctionnement cross-platform (mais UI adaptée requise sur mobile pour que ce soit utile).

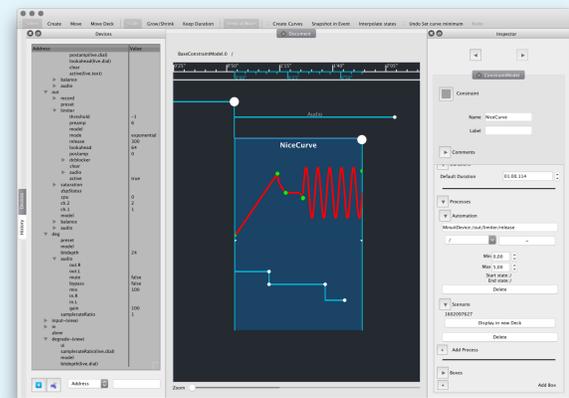


Modèles et technologies



- ▶ Relation entre modèles : Passive View, Hierarchical MVP (hiérarchie infinie).
- ▶ Le modèle s'expose lui-même dans l'arbre de paramètres pour être contrôlé de l'extérieur.
- ▶ Complexité du projet : doit servir de base pour une famille de logiciels ayant des buts différents par la suite, mais pouvant communiquer facilement et être affichés dans une seule fenêtre. Nécessité d'une logique commune pour menus, panneaux, sélection, undo-redo, etc...
- ▶ Utilisation des capacités de plug-ins de la bibliothèque Qt.
- ▶ Analogue à un petit système d'exploitation avec système de fenêtrage.

Capture d'écran



Formalisme et exécution sur FPGAs

- ▶ I-score est construit sur un modèle formel appelé *Automates Temporisés*.
- ▶ Pour la première fois, il est possible de vérifier automatiquement des propriétés des scénarios. Par exemple, il est possible d'assurer que les relations temporelles sont toujours respectées ou que le scénario se termine toujours.

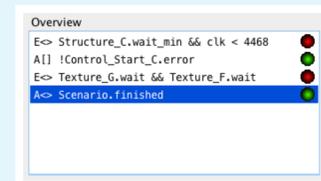


FIGURE 1: Vérification automatique avec UPPAAL.

- ▶ Les partitions interactives réalisent des tâches en temps réel et des calculs intensifs avec une grande ensemble de données qui, parfois, ne peuvent être traités par des ordinateurs ordinaires.
- ▶ L'utilisation de *Field Programmable Gate Arrays* (FPGAs) est une alternative raisonnable pour atteindre le niveau de performance nécessaire pour les applications multimédias.
- ▶ Les FPGAs offrent certains avantages, par exemple, de parallélisme physique, de reconfiguration, une vitesse élevée, ainsi que la fiabilité et la réutilisation de modules.
- ▶ Intuitivement, les automates temporisés peuvent être considérés comme plusieurs machines à états finies étendues avec des horloges logiques et s'exécutant en parallèle. Ainsi, il est possible de les mettre en œuvre facilement sur un FPGA en utilisant un langage de description matériel (HDL) comme *SystemVerilog*.

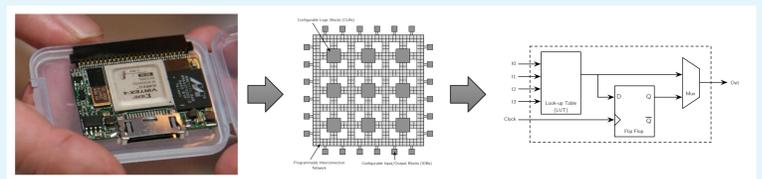


FIGURE 2: Une vue détaillée d'un FPGA.

Conclusion et perspectives

- ▶ Sources d'inspiration : réseaux de Petri, diagrammes de Gantt.
- ▶ Utilisé par artistes en résidence.
- ▶ Pas assez de tests (en développement depuis octobre).
- ▶ Application aux robots : compiler en code pour firmware metabot.
- ▶ Vérification automatique de scénarios afin d'éviter des erreurs lors de l'exécution.
- ▶ Exécution de scénarios avec une faible latence et en temps réel en utilisant des FPGAs.