

# Transcription rythmique dans *OpenMusic*

Arbres de rythme et règles de réécriture

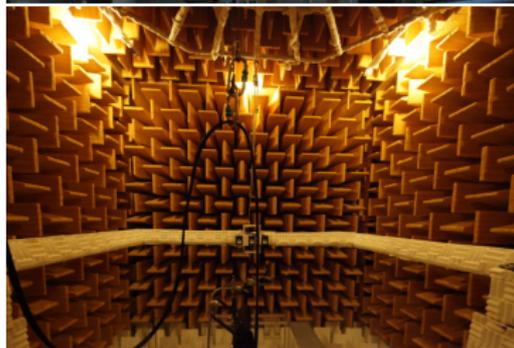
Pierre DONAT-BOUILLUD

ENS Cachan Bretagne

Ircam  
Equipe Repmus

Du 13 Mai au 19 Juillet 2013





L'IRCAM :

- un centre de recherche
- une salle de concert
- un institut pour compositeurs

L'équipe REPMUS :  
Représentation musicale

## 1 Contexte du problème

- OpenMusic
- La transcription rythmique

## 2 Transcription par réécriture

- Les arbres symboliques de rythme
- L'algorithme de quantification

## 3 Résultats

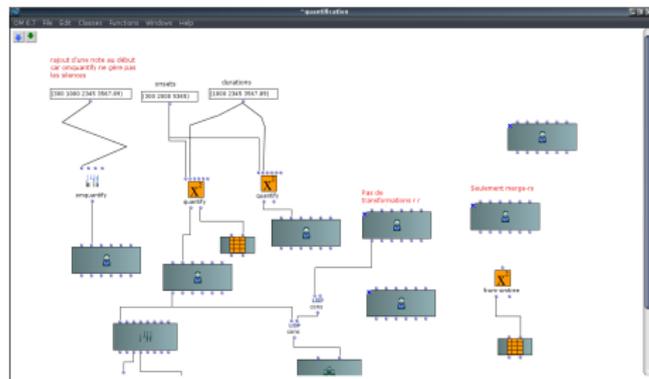
- La bibliothèque de fonctions `OMRewrite`
- Comparaison de `omquantify` et d'`OMRewrite`

# OpenMusic

Un environnement de composition assistée par ordinateur

## Un *Common Lisp* graphique

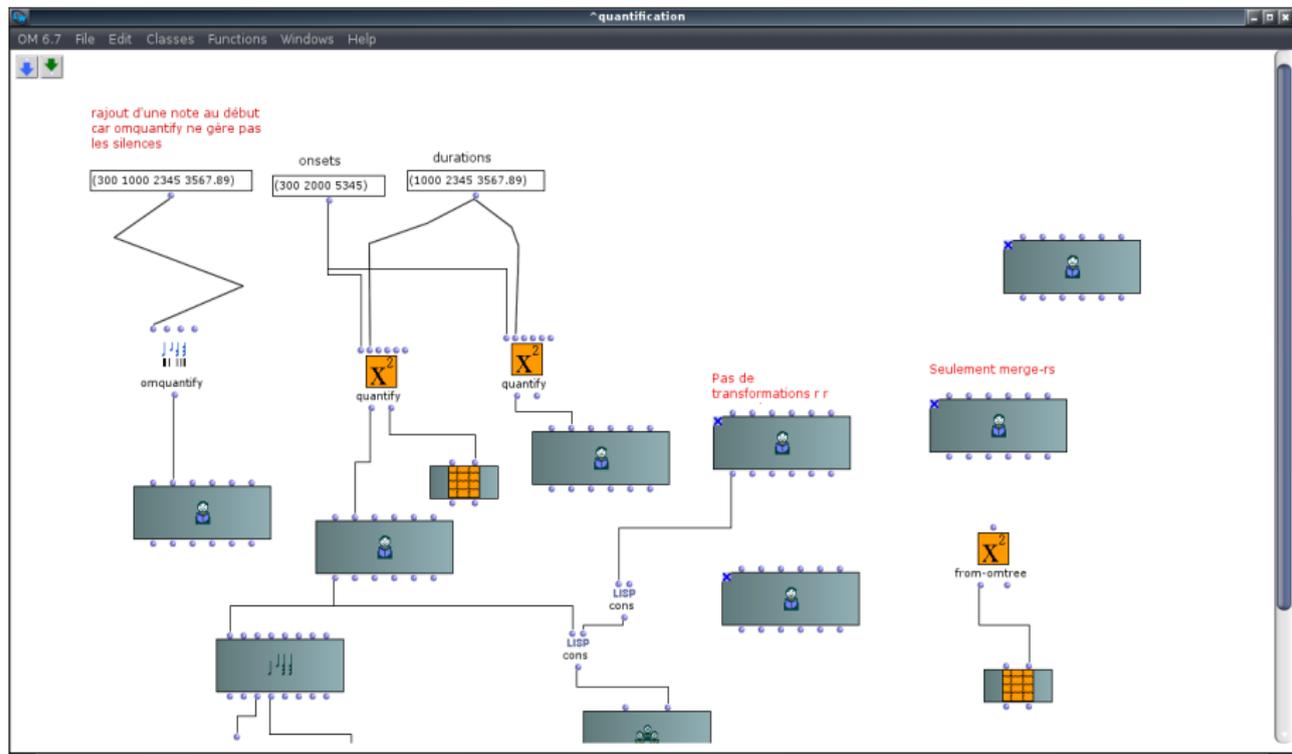
- boites fonctionnelles interactives connectées entre elles
- un programme est mis dans un patch
- possibilité de créer de nouvelles boites graphiquement, ou bien en *Common Lisp*



Un patch dans  
*OpenMusic*.

# OpenMusic

## Un environnement de composition assistée par ordinateur



# OpenMusic

Un environnement de composition assistée par ordinateur

## Un *Common Lisp* graphique

- boites fonctionnelles interactives connectées entre elles
- un programme est mis dans un patch
- possibilité de créer de nouvelles boites graphiquement, ou bien en *Common Lisp*

## Avec des bibliothèques de fonctions consacrées à la musique et au son

- Représentation, analyse, traitement symbolique de la musique 🎵
- Traitement du signal sonore

# Qu'est-ce que le rythme ?

## Le rythme

C'est la perception d'une structure dans un ensemble d'événements sonores.

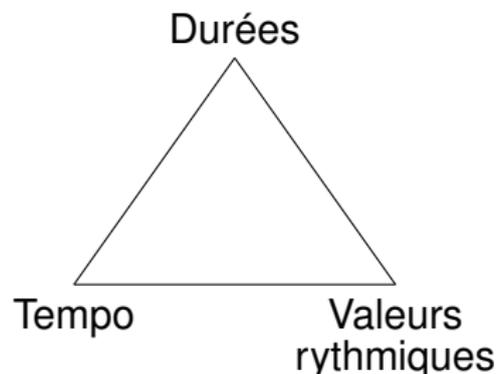
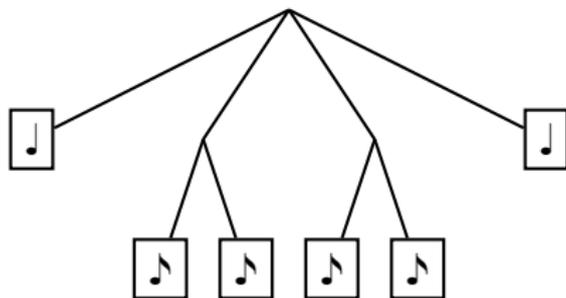


FIGURE : Relation « triadique » du rythme

# Le rythme

Une structure arborescente







# Les approches existantes pour la quantification rythmique

## Quantification naïve sur grille

- générer une grille régulière
- aligner les notes sur la grille

## Limites

Echoue face à des rythmes complexes ou si la subdivision choisie ne correspond à la subdivision du morceau.

Ex : binaire et ternaire

# Les approches existantes pour la quantification rythmique

## Quantification par heuristique

- générer un ensemble de grilles pas forcément régulières
- noter les grilles pour les durées à quantifier
- choisir la grille qui a la meilleure note

## Limites

Peu flexible.

Compliqué à paramétrer.

Les rythmes obtenus peuvent être compliqués.

## 1 Contexte du problème

- OpenMusic
- La transcription rythmique

## 2 Transcription par réécriture

- Les arbres symboliques de rythme
- L'algorithme de quantification

## 3 Résultats

- La bibliothèque de fonctions `OMRewrite`
- Comparaison de `omquantify` et d'`OMRewrite`

# Les arbres symboliques de rythme

Une notation plus adaptée pour la réécriture

Chaque nœud représente la même durée que ses frères ;

## Un petit nombre de symboles

n : note

r : silence

s : liaison avec la note précédente

= : le premier symbole différent de = dure "1 + le nombre de = précédents adjacents" .

# Les arbres symboliques de rythme

## Un exemple

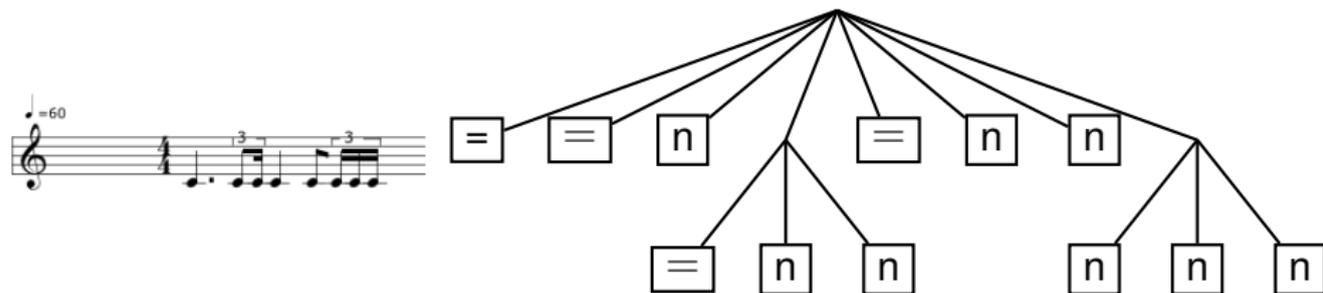
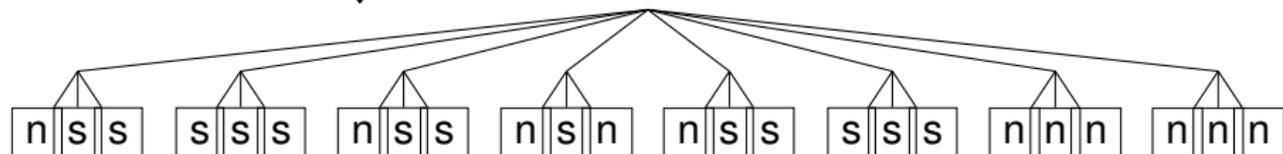


FIGURE : Arbre symbolique de rythme ( = = n ( = n n ) = n n ( n n n ) )

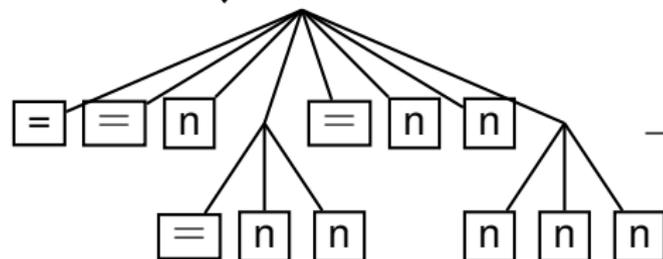
# L'algorithme de quantification



↓ Génération d'un arbre complet



↓ Application de règles de réécriture



# Exemples simples de règles de réécriture

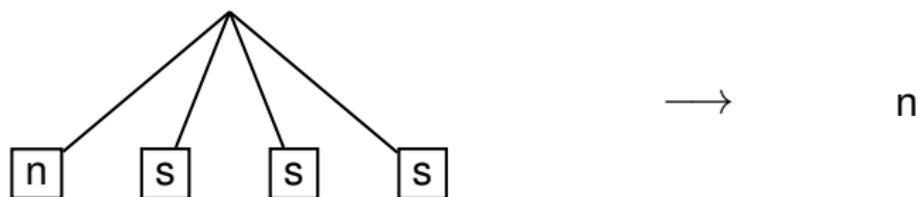


FIGURE : Règle de réécriture de fusion de notes liées

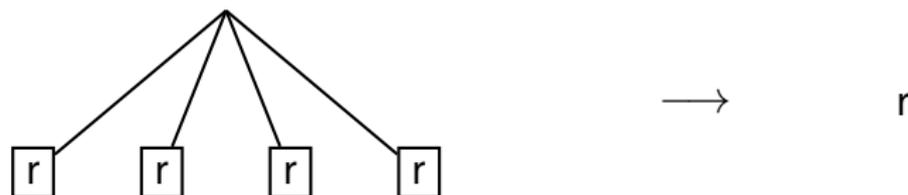


FIGURE : Règle de réécriture de fusion de silences.

## 1 Contexte du problème

- OpenMusic
- La transcription rythmique

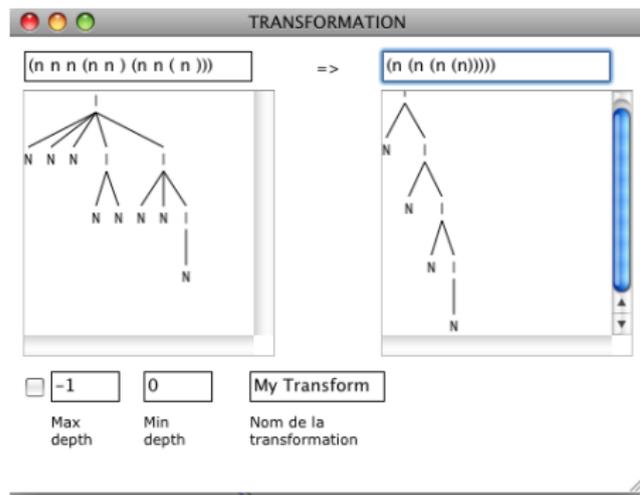
## 2 Transcription par réécriture

- Les arbres symboliques de rythme
- L'algorithme de quantification

## 3 Résultats

- La bibliothèque de fonctions `OMRewrite`
- Comparaison de `omquantify` et d'`OMRewrite`

# La bibliothèque de fonctions *OmRewrite*



Outils graphiques :

- quantifier
- définir ses propres règles de réécriture

Un outil créatif pour compositeurs.

# Comparaison de `omquantify` et d'*OMRewrite*

Exemple où *OMRewrite* manque de précision en durée

	Attaques (ms)	Durées (ms)
En entrée	(300 2000 5345)	(1600 2345 3567.89)
<i>OMRewrite</i>	(500 2000 5500)	(1250 2250 3250)
<code>omquantify</code>	(286 2000 5333)	(1589 2333 3542)

**TABLE :** Comparaison entre `omquantify` avec paramètres 60, (4, 4), 8 et *OMRewrite* avec paramètres 4, 3, 60 et *silence-strategy*

<i>OMRewrite</i>	0.22	0.04	0.09
<code>omquantify</code>	0.007	0.005	0.007

**TABLE :** Erreur relative sur les durées

## Comparaison de `omquantify` et d'*OMRewrite*

- Moindre précision d'*OMRewrite* en durée et en attaque
- Pas de propagation d'erreurs
- Plus facilement paramétrable

Est-ce que précision en attaques et en durée est un bon critère pour une transcription rythmique ?

Le bon critère : la simplicité de la notation.

→ critère subjectif a priori

# Bilan et perspectives

- notation et principe de l'algorithme de quantification présentés lors d'un *groupe de travail* sur le rythme à l'IRCAM, qui est un séminaire public régulier où chercheurs et compositeurs discutent de sujets sur le rythme
- projet d'article principalement sur les règles de réécriture utilisées
- d'autres approches à l'IRCAM sont explorées pour la transcription rythmique : apprentissage d'automates d'arbre

FIN

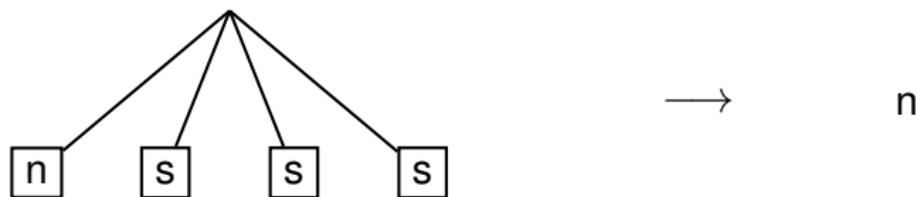
# Règles de réécriture

On ne présente ici qu'un exemple de la règle ; il en existe en réalité plusieurs versions différentes :

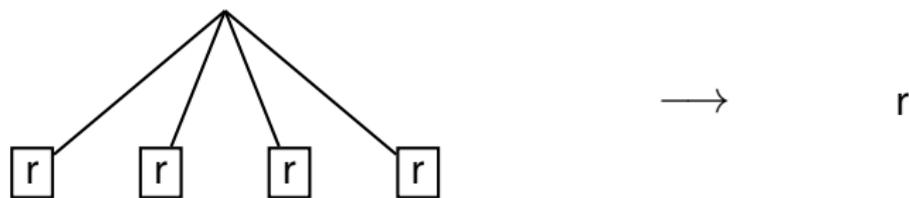
- une pour chaque nombre de fils ( $\leq 32$ )
- une pour chaque possibilité de début d'occurrence d'un motif (pour certaines règles comme `gen=`).

# Règle de réécriture de fusion de notes liées

merge-ns

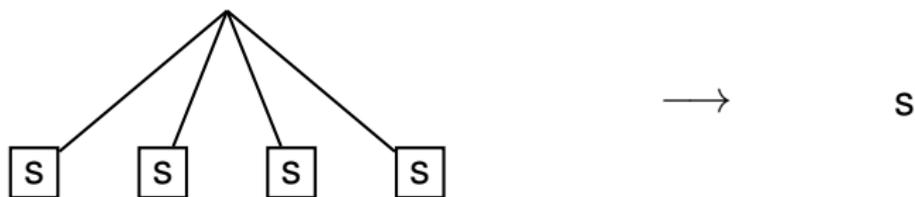


# Règle de réécriture de fusion de silence `merge-rs`



# Règle de réécriture de fusion de liaisons

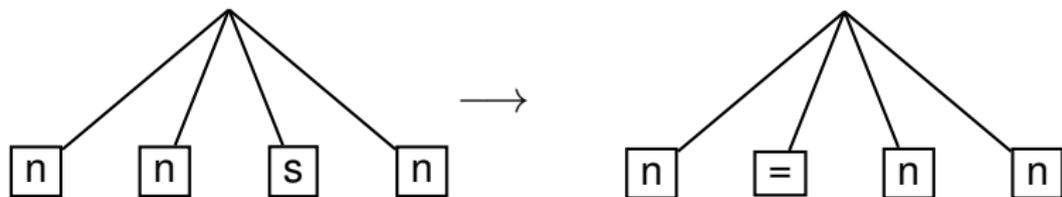
merge-rs



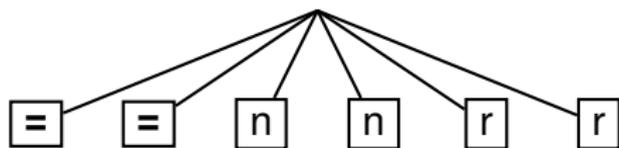
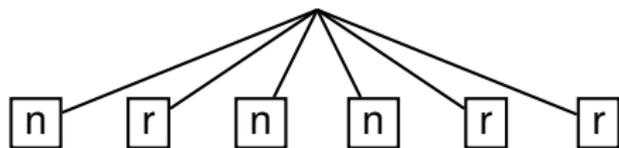
# Règle de remontée de subdivision seule `move-up`



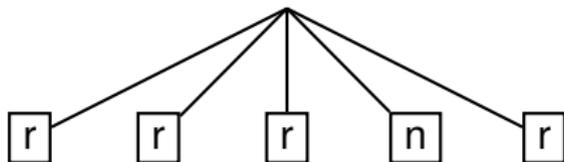
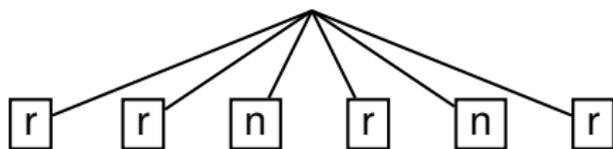
# Règle de génération des symboles = gen- =



# Règle de réécriture de dilatation dilate



# Règle de réécriture de dilatation puis suppression des nouveaux silences apparus `metric-reduce`



# Règle de réécriture d'élagage cut-sides

