Retour sur le développement d'un outil de composition musicale à l'aide de GWT

Steven Varoumas

Objectif

➡ Créer un outil d'édition musicale accessible sur internet (pour l'éducation, projets ultérieurs,…).

Quelle technologie ?

- **Javascript + HTML5**
  - \+ Moderne, multi-plateforme, grande communauté, ne nécessite pas d'installation.
  - \- Javascript : n'est pas type-safe, syntaxe parfois étrange (ex: tests d'égalité)
  - \- HTML5 : Tous les navigateurs n'ont pas forcément le même comportement.
- **Flash + ActionScript**
  - \- N'est pas supporté par les appareils mobiles : iOS (depuis toujours), Android (depuis 4.1 Jelly Beans en 2011) : à l'abandon ?
  - \- Plugin flash à installer par l'utilisateur.
- **Applets Java**
  - \+ Java : Large API, langage "populaire" : souvent enseigné et utilisé par les entreprises.
  - \- Lourd, vieillissant …
  - \- La plupart des navigateurs web les bloquent désormais par défaut.

Google Web Toolkit (GWT)

- Outil permettant de convertir une app client/serveur écrite en **Java vers les technologies web**.
  - Partie client : convertie en JavaScript/HTML/CSS
  - Partie serveur : services fournis par servlets (protocole GWT RPC)

\+ Communauté importante (Google)

\+ Librairie de widgets graphiques (boutons, onglets,…) intégrée

\+ Support possible du code JavaScript

\+ Une extension Chrome permet d'exécuter et de déboguer en direct.

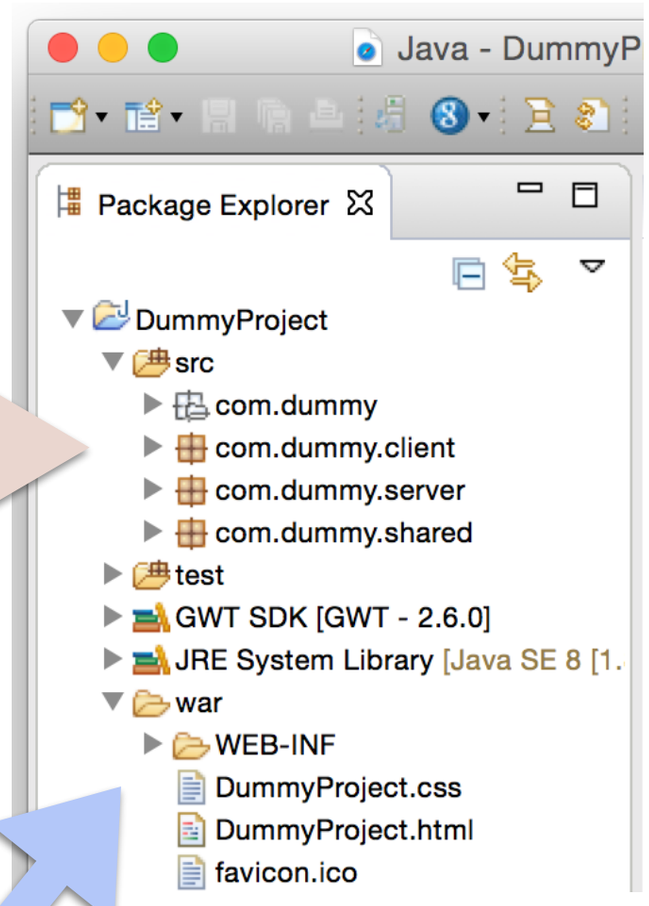
\+ Gère les différences de comportements entre les navigateurs.

➡ www.gwtproject.org/

Structure d'un projet GWT et restrictions

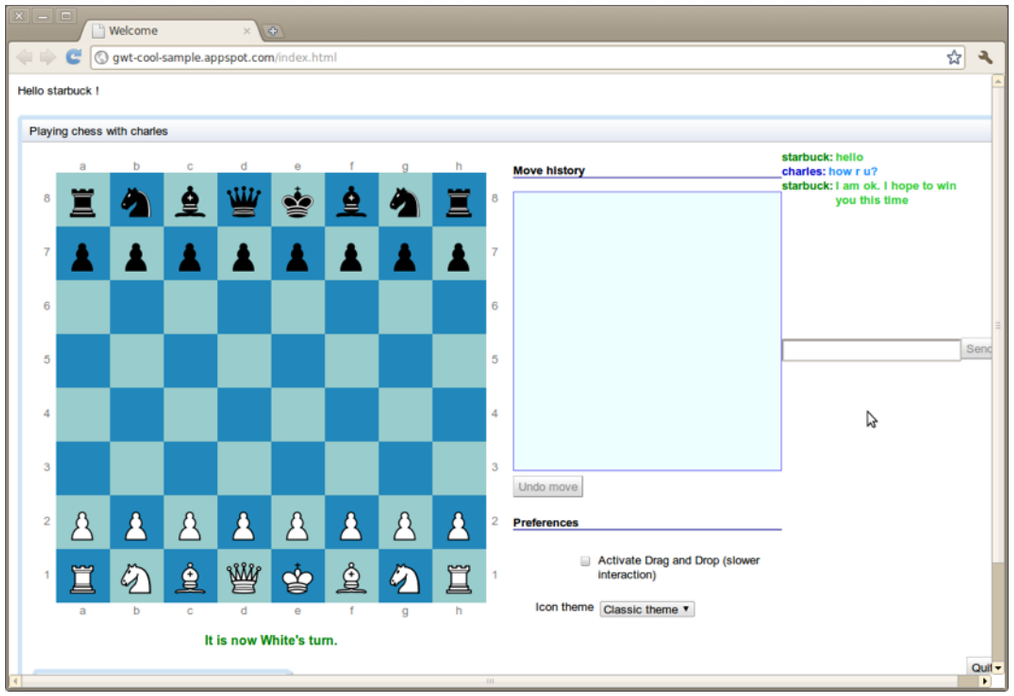
- Plugin Eclipse

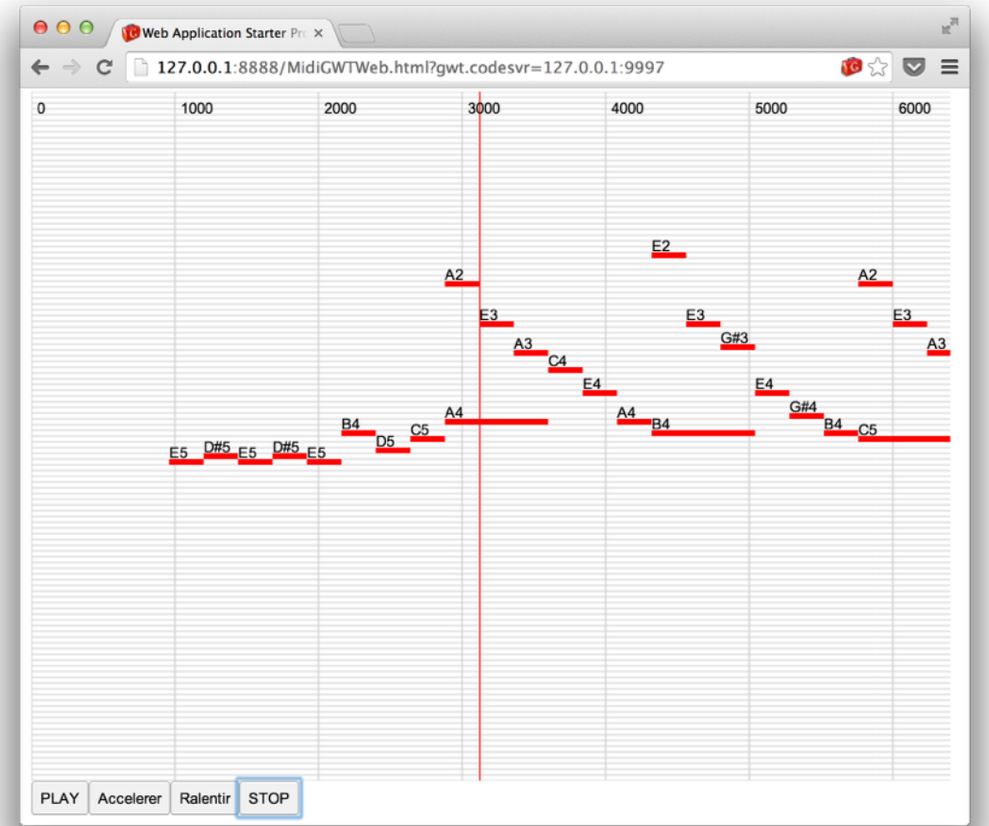
- Le code d'un projet GWT se divise en trois parties distinctes :
  - un package ***`client`*** pour le code à compiler en JavaScript : compatible `java.lang` et (en partie) `java.util` (+ *`com.google.gwt`*)
  - un package ***`server`*** compatible avec toute l'API Java
  - un package ***`shared`***

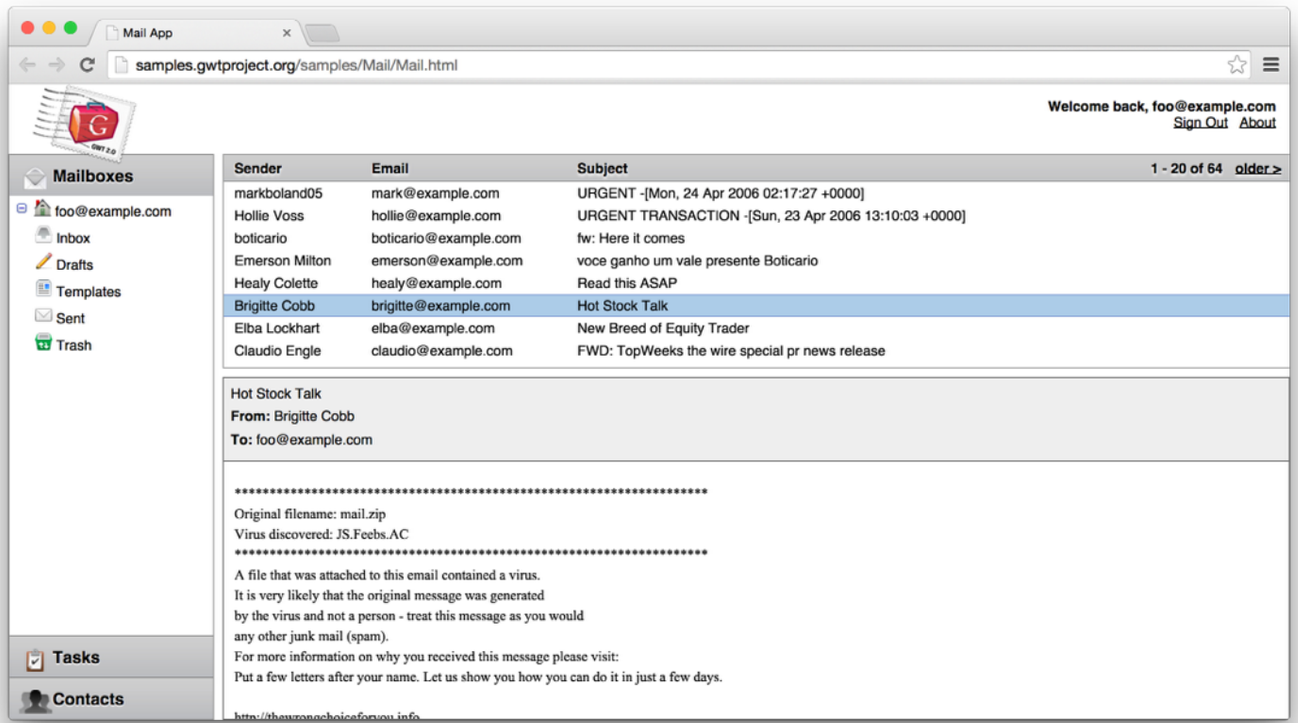


+ un dossier ***`war/`*** contenant le .html, .css, et les autres fichiers nécessaires au chargement de la page (images, polices, librairies, etc).

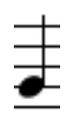
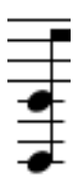
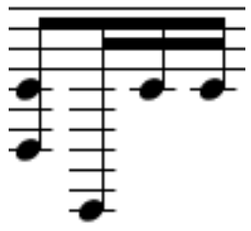
Google Web Toolkit (GWT) : exemples

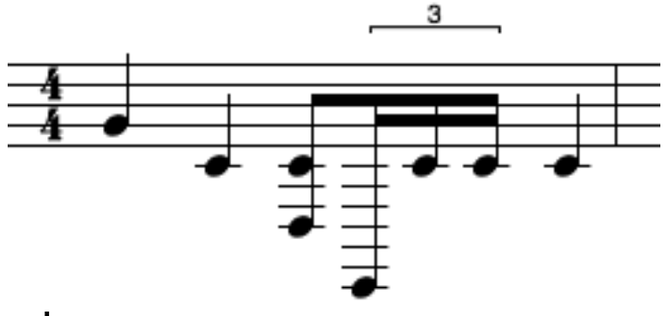






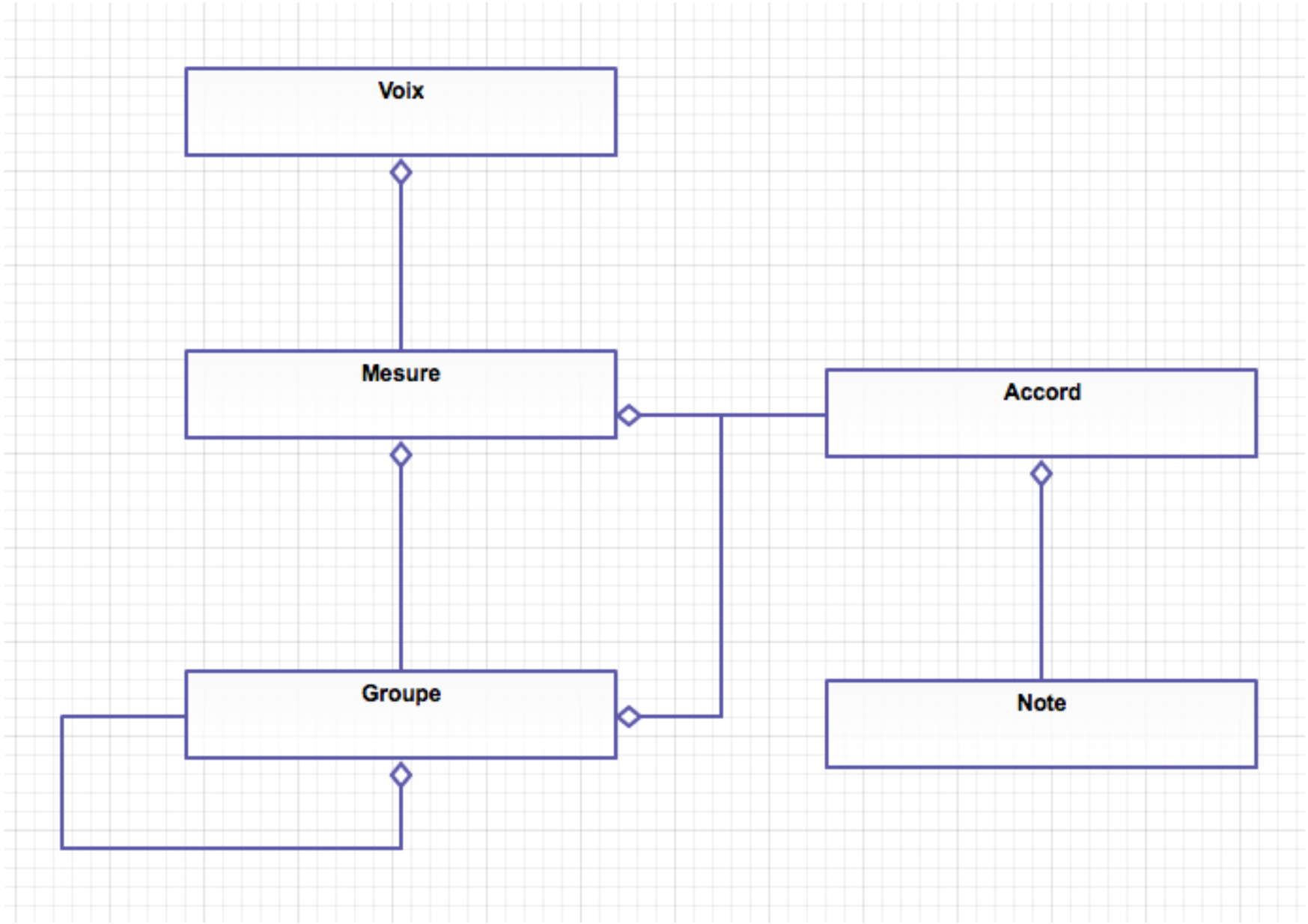
Modélisation : Les différents éléments

- **La note** : élément de base (hauteur, vélocité, durée,…)

- **L'accord** : composé de notes

- **Le groupe** : composé d'accords / sous-groupes

- **La mesure** : composée d'accords / groupes



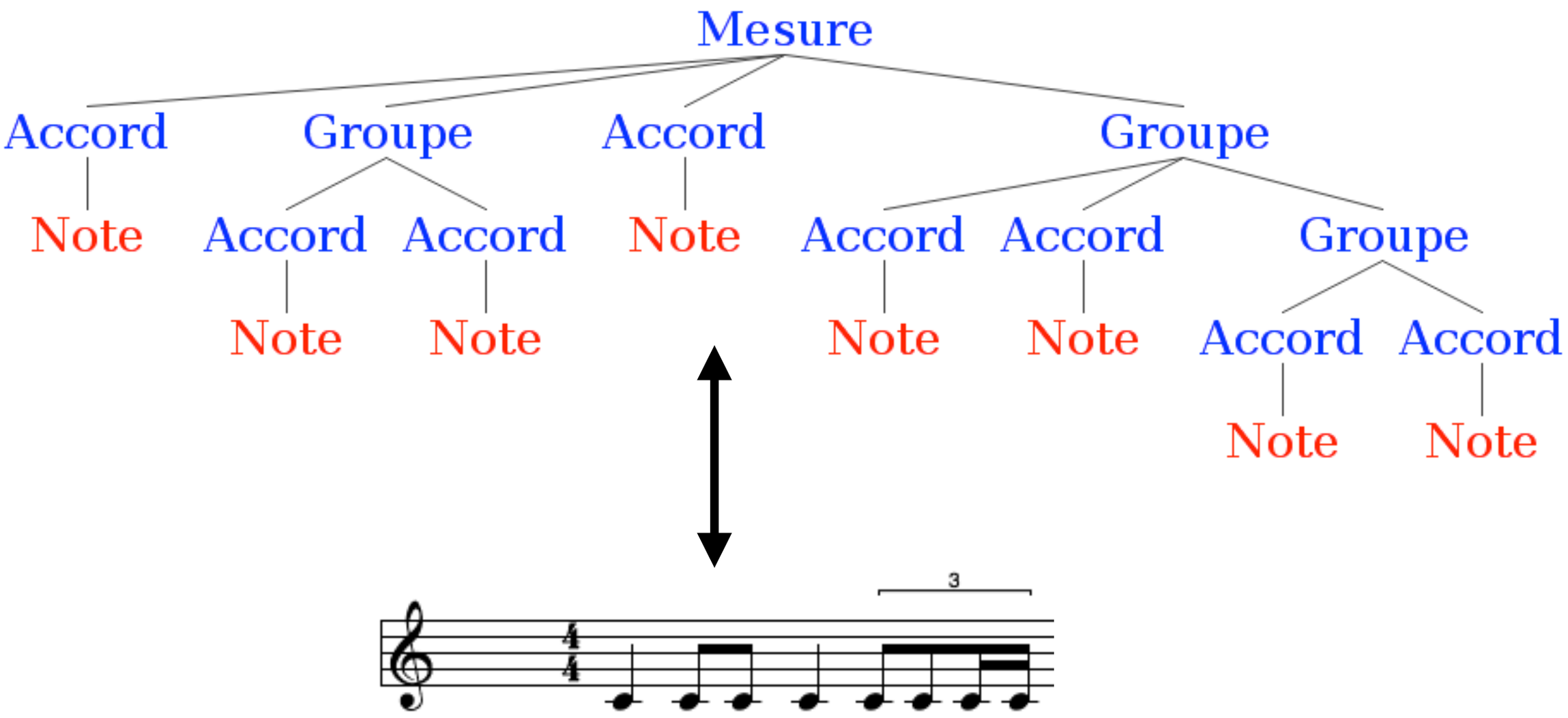
- **La voix** : composée de mesures

Modélisation des éléments: diagramme de classes



Structure de données

- Les mesures sont des arbres :

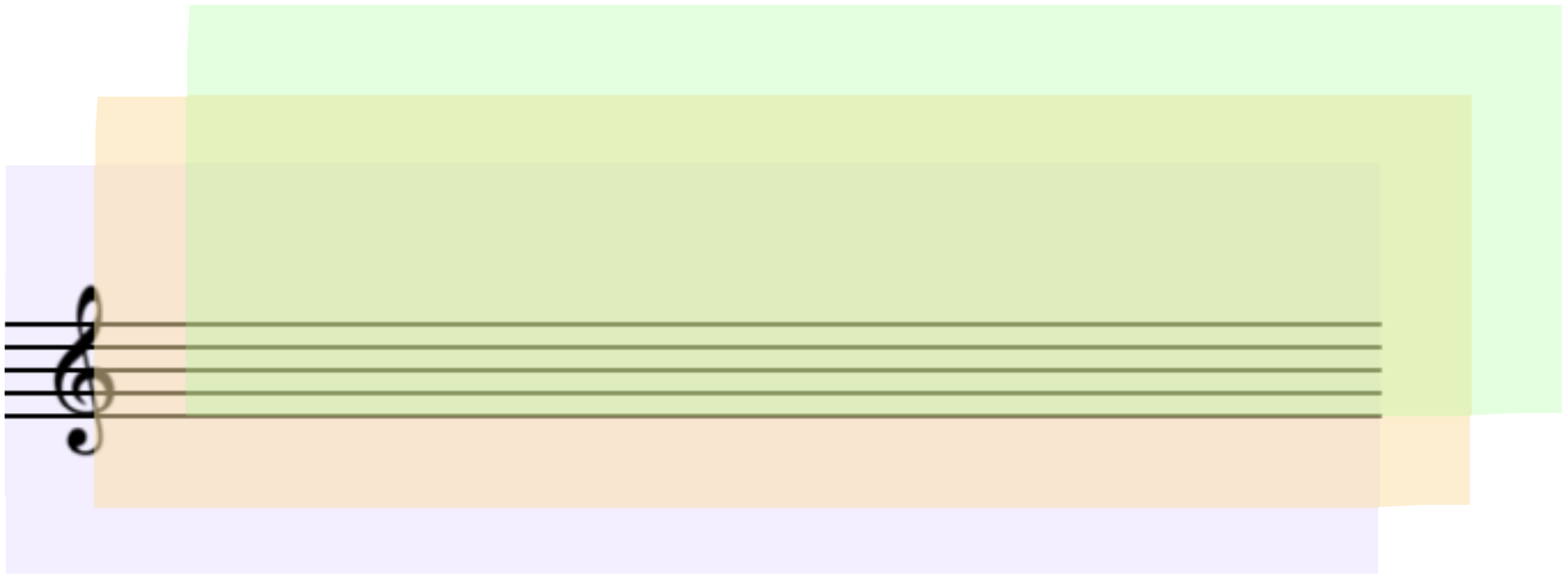


Dessiner les objets - La portée

- **Canvas** HTML5 : un élément permettant le dessin 2D (formes géométriques, images, texte, …) à l'intérieur d'une zone d'une page web (~JPanel en Swing).

- GWT fournit une classe **`Canvas`** pour les manipuler.

- Dans l'éditeur, on utilise plusieurs canvas ("calques") superposés:

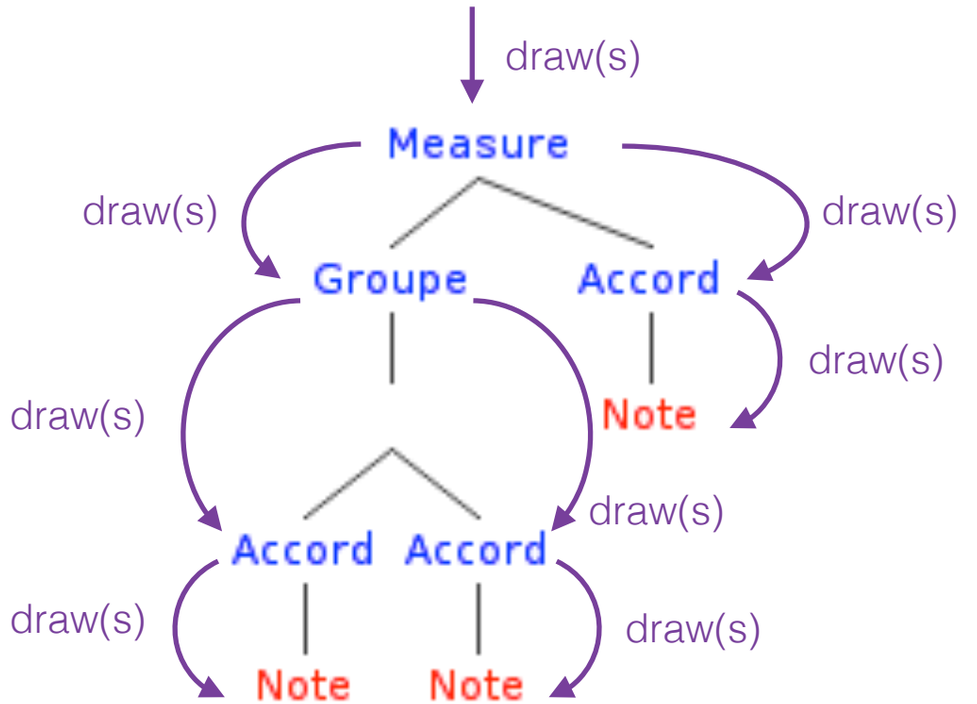


Dessiner les objets - Les notes

- On dessine sur la portée en parcourant les arbres représentant les mesures.

- Chaque élément se dessine lui-même sur la portée et demande à ses fils de se dessiner à leur tour :

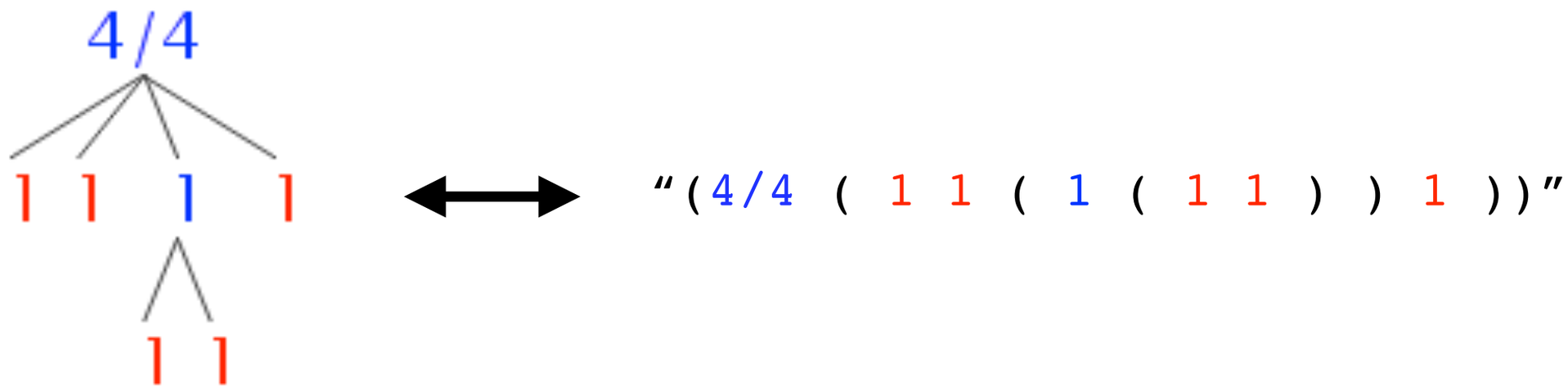
```
//in Measure :
public void draw(Staff s){
    drawSignature(s);
    for(IElement e : this.elements){
        e.draw(s);
        //e est un Chord ou un Group
    }
}
```



Une fonction midi2pixel() permet de calculer la hauteur (en pixels) d'une note sur la portée par rapport à sa valeur midi.

Construire les objets - Arbres Rythmiques

- On construit tous les objets à partir d'arbres rythmiques (qui représentent les "divisions" des mesures).
- Un arbre rythmique peut se représenter par une chaîne de caractères :



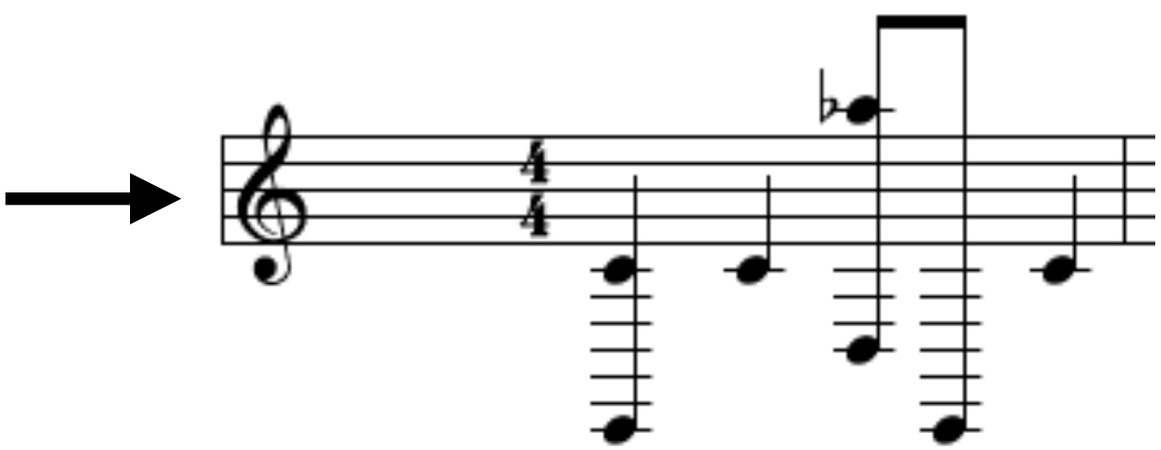
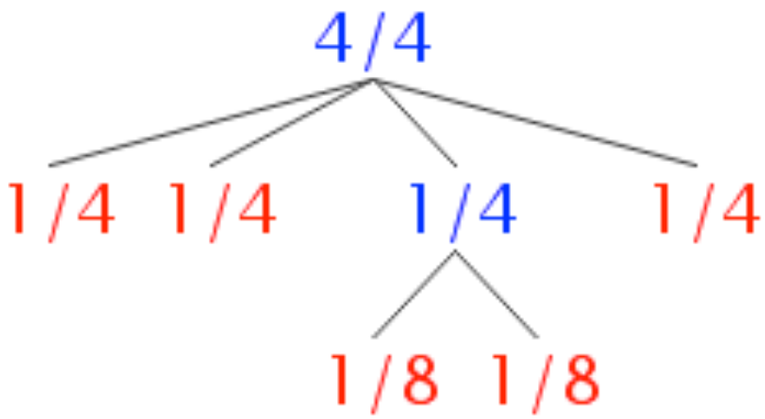
Construire les objets - Mesure

- La mesure est créée à partir d'un arbre + d'une liste d'accords (valeurs MIDI):

(4/4 (1 1 (1 (1 1)) 1))

+

((6000 4000) (6000) (5000 8000) (4000) (6000))

un accord



La durée de chaque noeud est calculée en fonction de sa profondeur dans l'arbre.

Arbres rythmiques : Parsing avec JavaCC

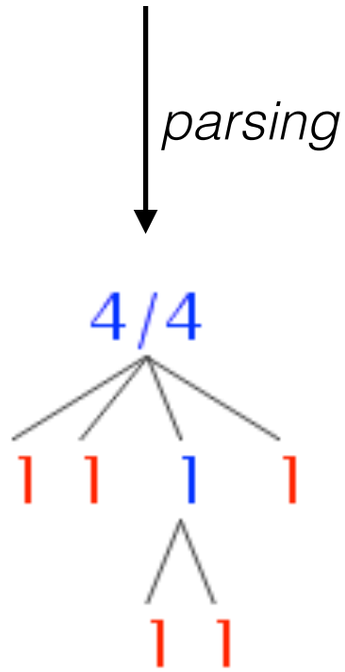
- Afin de “transformer” la représentation textuelle de l’arbre rythmique en objet RhythmicTree, il faut un parser.

- **JavaCC (Java Compiler Compiler)** est un générateur automatique de parsers (en Java) à partir d’une grammaire.

- Notre grammaire :

NUMBER := [0-9]
RATIO := NUMBER/NUMBER | NUMBER
EXP := (NUMBER | TREE)+
TREE := (RATIO EXP)

“(4/4 (1 1 (1 (1 1)) 1))”



Jouer les notes : exemple d'interfaçage avec JSNI

- Pour jouer les compositions, on utilise un plugin qui fonctionne avec une librairie JavaScript : *Jazz*.

- Il faut pouvoir appeler directement du code JavaScript dans le projet Java : **JSNI** (JavaScript Native Interface)

fichier Java :

```
public final native void playNoteMidi(int chan, int code, int velocity)
/*-{
    $doc.playNote(chan,code,velocity);
}-*/;
```

JSNI

fichier .js:

```
document.playNote = function playNote(chan,code,vel){
    Jazz.MidiOut(chan,code,vel);
}
```

Démo

http://stevenvar.comli.com

Liens

- GWT :
  https://www.gwtproject.org

- JSNI :
  https://developers.google.com/eclipse/docs/gwt_jsni?hl=fr
- JavaCC :
  https://javacc.java.net
- Jazz Plugin : :
  http://jazz-soft.net
- Pour tester l'éditeur :
  http://stevenvar.comli.com