

MuBu for Max/MSP

IMTR IRCAM – Centre Pompidou

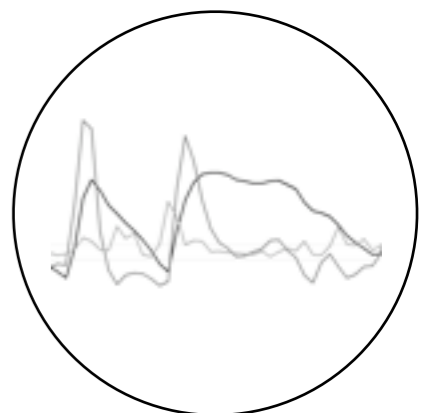
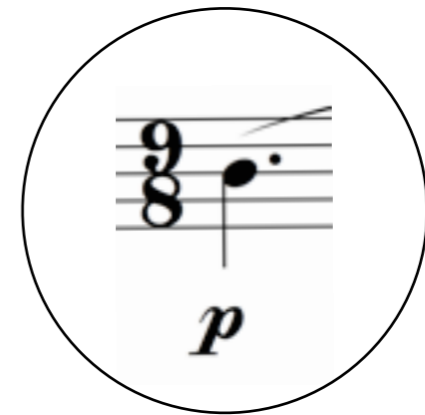
Norbert Schnell
Riccardo Borghesi

20/10/2010

Motivation

Create a solid and open framework for the experimentation with recorded data streams of multiple representations in *Max/MSP*

- audio samples
- audio descriptors
- gesture and motion capture data
- spectral audio representations
- symbolic representations
- segmentation and annotations
- ... what else?



Converging Applications

- **Analysis/re-synthesis** (*sound, music and speech*)
- **Corpus based granular synthesis** (*textures*)
- **Following and recognition** (*gestures/sounds and scores*)
- **Computer aided improvisation**
- ...

Context

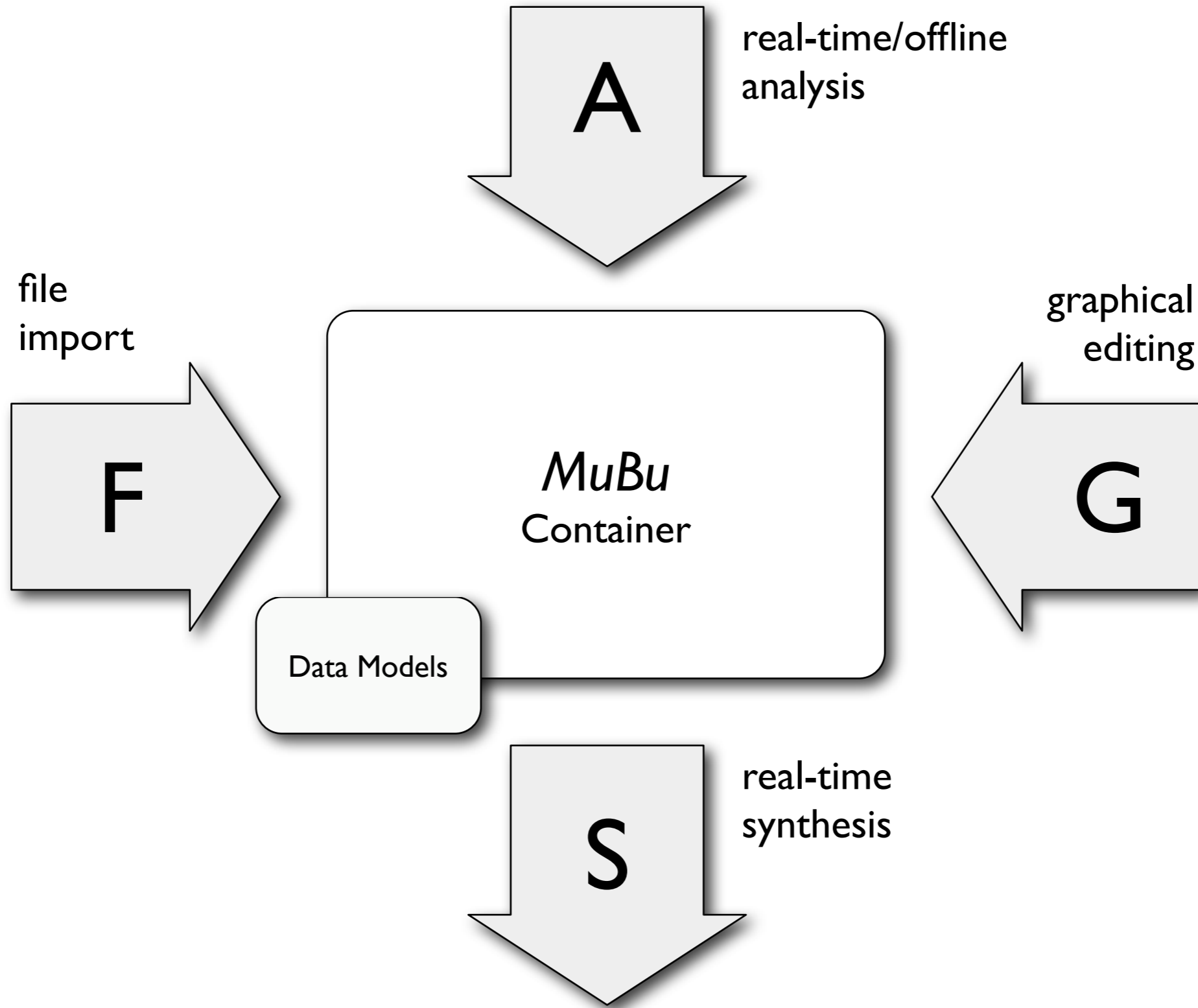
- *Créations IRCAM & Forum*
- *Recherche musicale*
- Prototyping of interactive real-time applications

Relaxed Real-Time Analysis/Synthesis

- Real-time *analysis, recognition and classification*
 - *structured* sound and gesture data
 - “*emerging symbolics*”
- Content based audio *processing and rendering*
 - synthesis based on offline or real-time analysis
 - “*post-sampling*” and “*re-performance*”

MuBu Container Functionalities

- Store *heterogeneous synchronous* data streams
- File import/export
- Visualization and editing
- Optimized congruent access to shared data
- Handling of dynamic configurations (data and references)

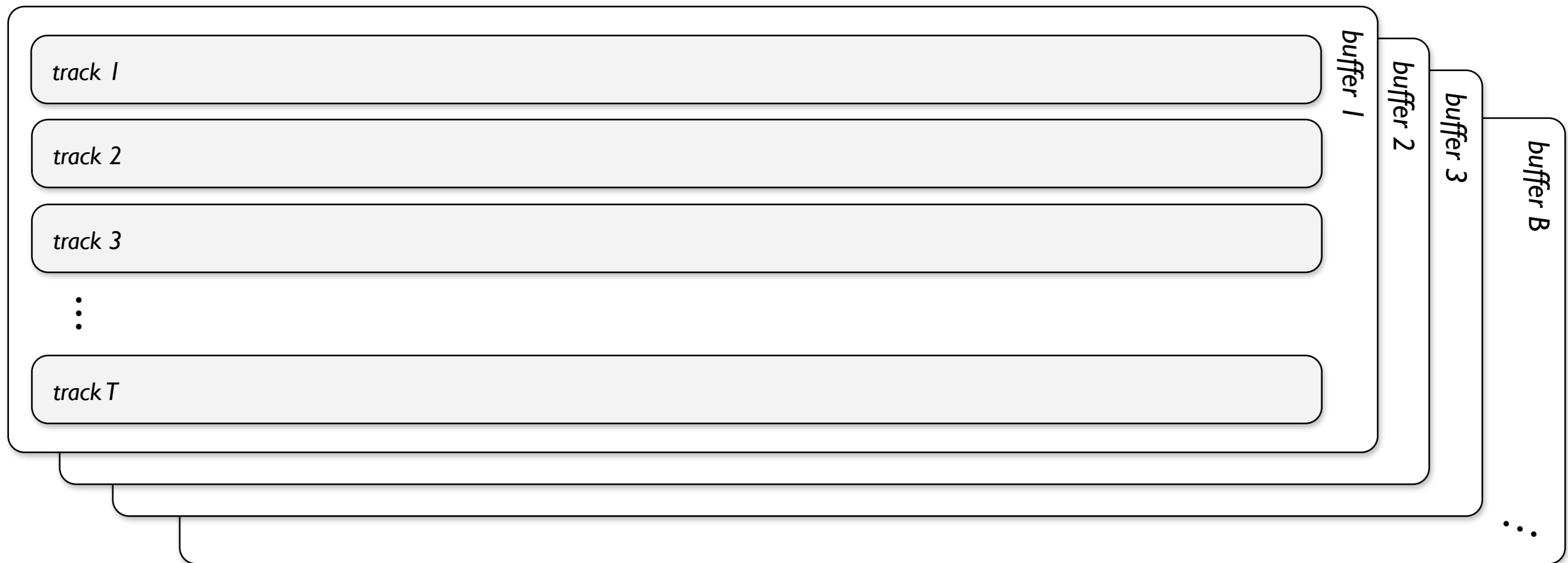


MuBu Container Design

- *Simple* and *easy* to maintain
- *Minimizing* dependencies on other libraries
- *Thread-safe* and *lock-free* congruent data access

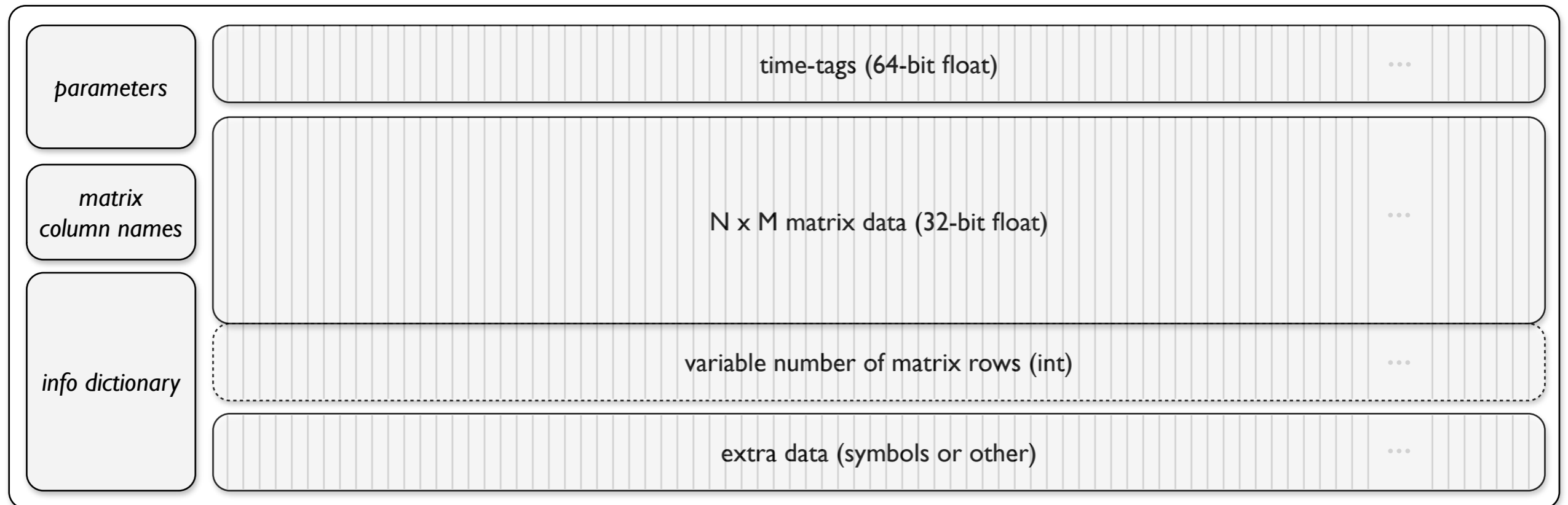
Container Data Structure

- Array of *buffers* being arrays of *tracks*



Track Data Structure

- Array of 2-dim matrix data (optional *time-tags* and *matrix rows*)
- Array of (non-matrix) “extra” data (currently *labels* only)
- Track meta-data



Modularity and Thread-Safeness

- Distinction of *configuration* and *real-time* access
 - *thread-locked* access to container (using *pthread r/w locks*)
 - *lock-free* access to track data (using *compare-and-swap*)
- Track implemented as *immutable* data structure
 - fixed capacity
 - re-configuration by copying data into a new track
- Container manages dynamic track configuration
 - locked *callbacks* to modules that refer to a container
 - simple *garbage collector* (by reference count)

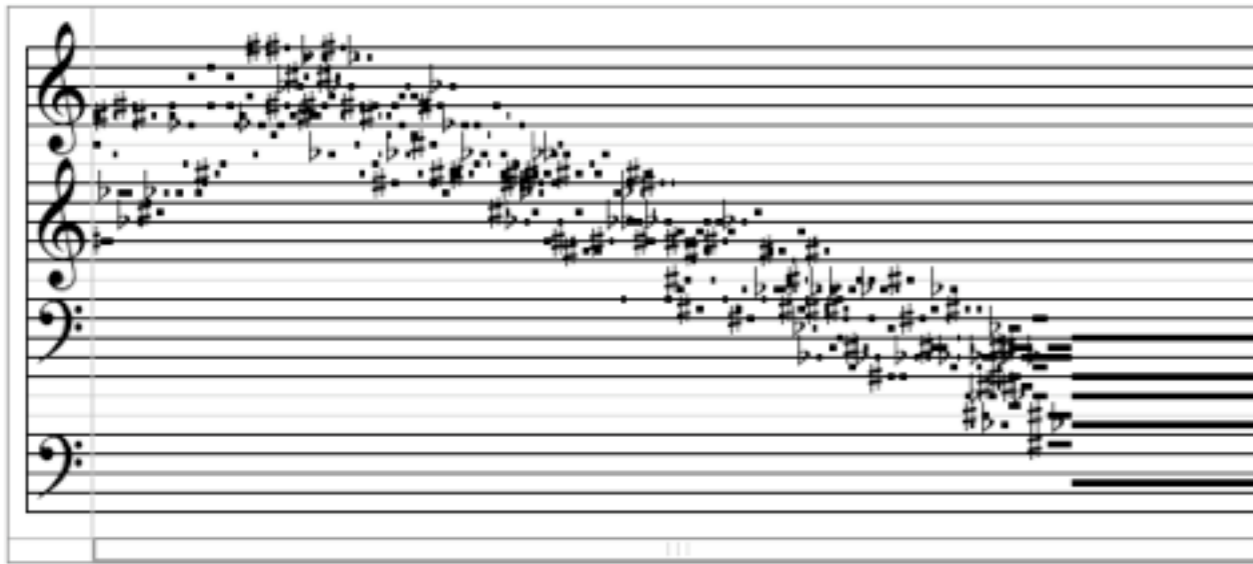
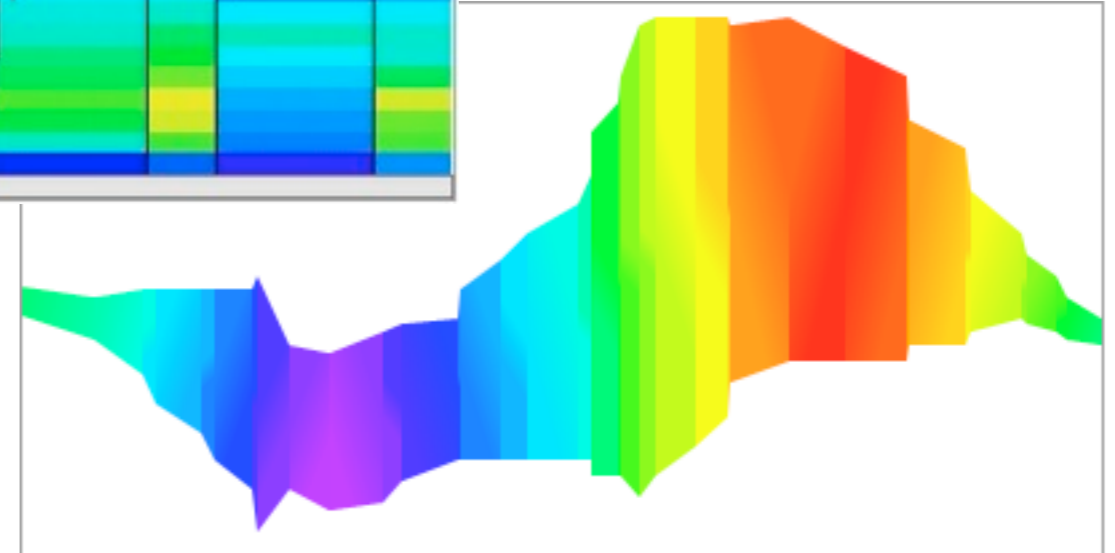
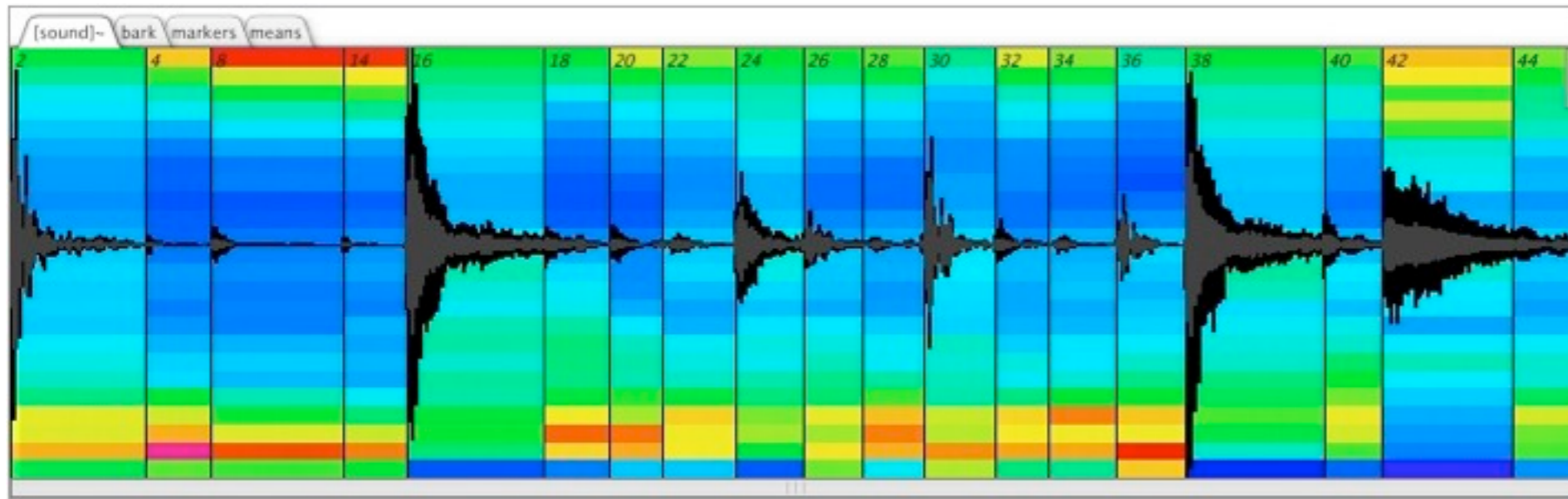
File Format Support

- *SDIF* import and export*
- Plain text file import and export
- *MIDI standard file* import
- *MusicXML* import
- Save/load data with Max patcher (for small amounts of data)
- * ... what else?

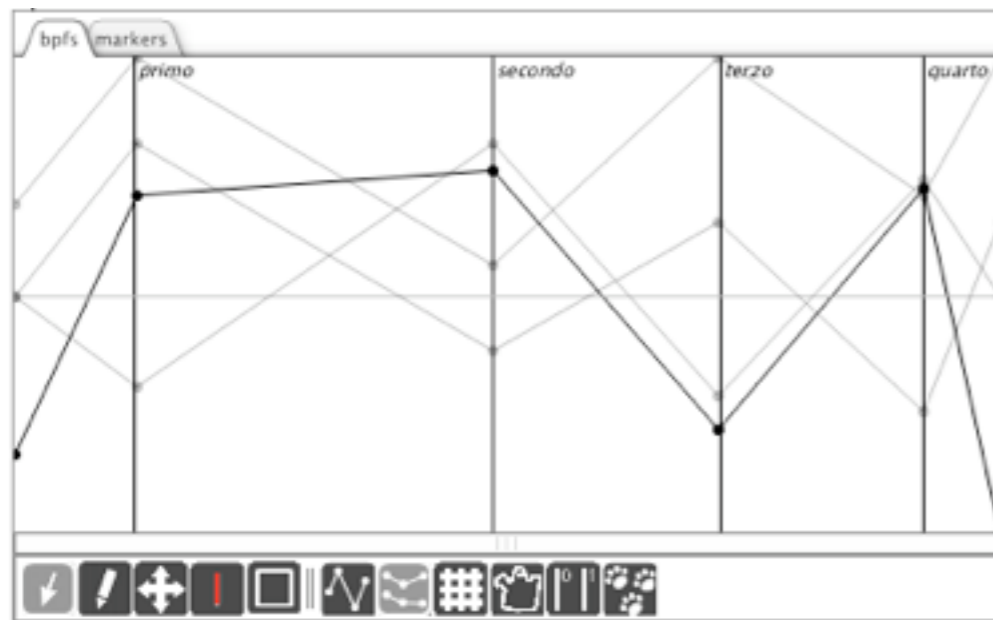
* *features to be implemented*

Visualization and Editing

- Based on the *Juce* framework
- **Abstract data interface** (also implemented for *FTM* data structures)
- **Set of editor/visualization components**
 - **waveform** (single or multi-channel)
 - **break-point function** (single or also multi-channel)
 - **sonogram** (of sampled or time-tagged data)
 - **markers** (with *duration* and *offset*)
 - **textual tables/matrices**
 - **piano roll** (or simplified staves)
 - **traces** (waveform or bpf with *color* and *thickness*)
- **Control components** (*scroll bar, rulers, toolbar, tabs, buffer chooser, etc.*)

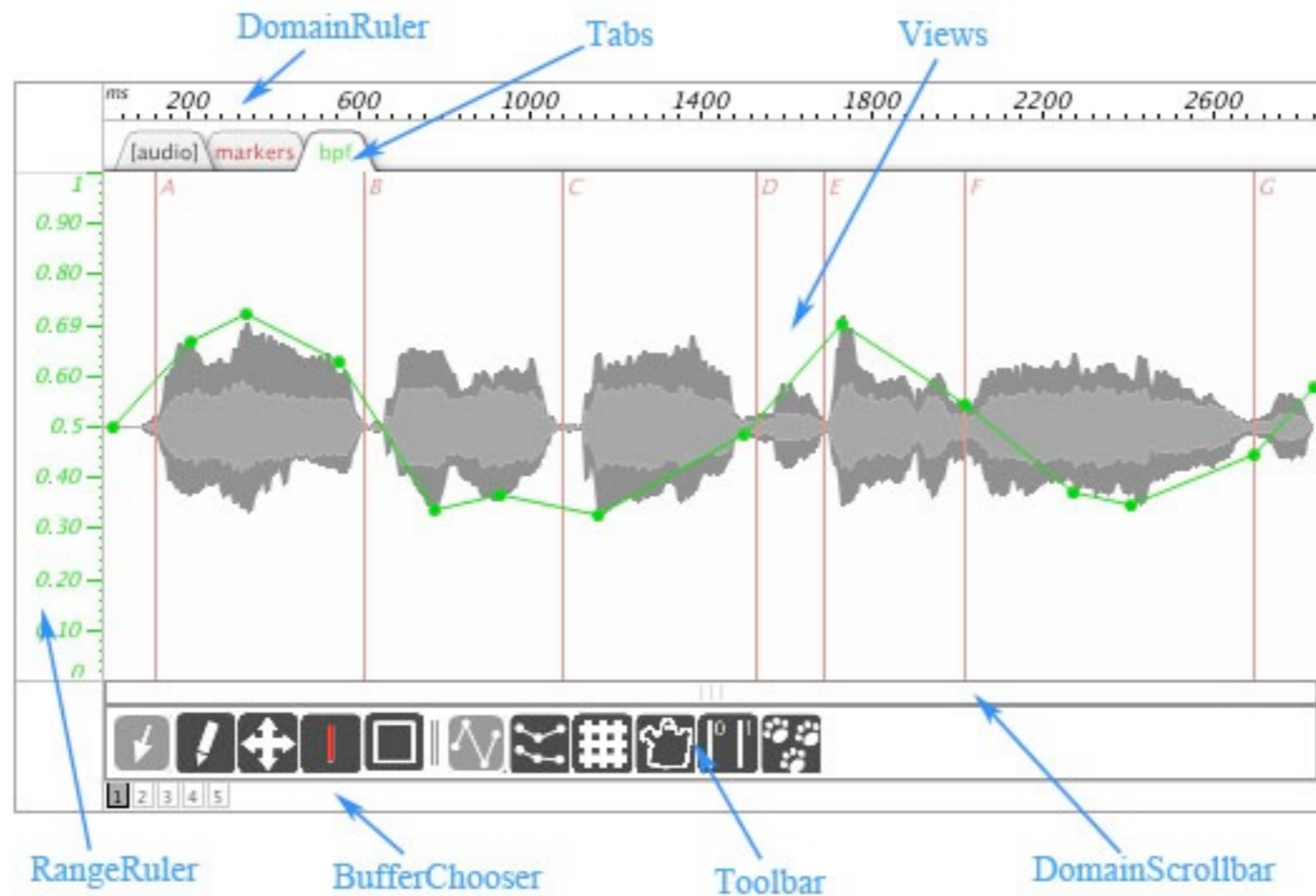


	1 <i>first</i>	2 <i>zweite</i>	3 <i>terzo</i>	4 <i>quatrieme</i>	5 <i>last</i>
0	1	0.2	colognole	0	0
1	0	bpf [#1]	0	0	0
2	0	0	fmat [#1]	0	0
3	123	0	maggiorana	0	cotoletta
4	0	0	0	0	0

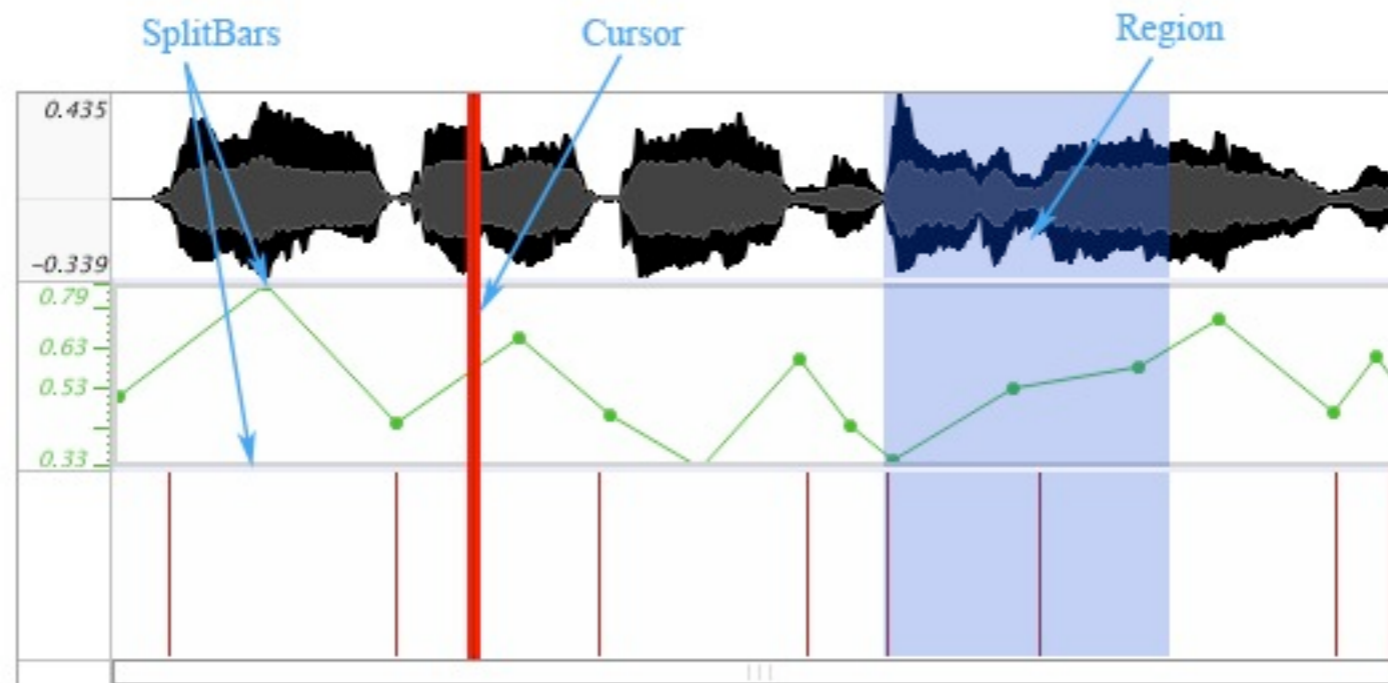


0	1	0.2	colognole	0	0
1	0	bpf [#1]	0	0	0
2	0	0	fmat [#1]	0	0
3	123	0	maggiorana	0	cotoletta
4	0	0	0	0	0

Editor Control Components (1/2)



Editor Control Components (2/2)



Generic Data Access

- Generic data container
 - mubu* ... basic multi-buffer container
 - imubu* ... container with graphical user interface
- Reference to a specific track
 - mubu.track* ... optimized access to track data
- * Generic *record* and *play* externals
- * JavaScript interface

Analysis

- *FTM & Co* prototype patch
 - loudness
 - power spectrum and spectral moments
 - MEL bands & MFCC coefs
 - YIN
 - loudness based segmentation
- * PSOLA markers
- * Partial and harmonics
- * *SuperVP*

Synthesis

- *ZsaZsa* overlap-add engine
 - concatenative synthesis
 - granular synthesis
 - * PSOLA
 - * *FFT-I* based additive synthesis
 - * *SuperVP*
- “*spectral/hybrid concatenative synthesis*”

Data Models

- KD-tree based k-nearest neighbors
- * N-Grams?
- * Factor oracle?
- * ... what else?

Current Release

- Beta version of first set of externals
 - *mubu* ... container
 - *imubu* ... container with graphical interface
 - *mubu.track* ... optimized access to track data
 - *mubu.knn* ... k-nearest neighbors
 - *mubu.concat~* ... concatenative synthesis
 - *mubu.granular~* ... granular synthesis

Future Developments

- Finalize SDIF export
- Integrate analysis and segmentation (real-time/offline)
- Integrate spectral analysis/synthesis (real-time/offline)
 - PSOLA (*ZsaZsa* synthesis)
 - additive (FFT-I synthesis)
 - phase vocoder (*SuperVP* and *simple*)
- Implement further data models

Links

- *MuBu* on the IMTR web pages
→ <http://imtr.ircam.fr/imtr/MuBu>
- *MuBu* beta-test mailing list (subscription required)
→ <http://listes.ircam.fr/www/info/mubu-beta>