FEEVER: 18-Month Review Meeting

ARMINES/MINES ParisTech, GRAME, INRIA, UJM/CIEREC

October 13, 2015

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

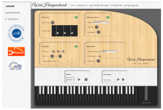
ANR

FEEVER: FAUST Environment Everyware

What are we aiming for?

- Notation: standard language for audio signal processing
- Efficiency: fast and reliable implementations
- Deployment: target all platforms (desktop, Internet, mobile...)
- Accessibility: seamless and universal compilation technology
- Education: tools and content for audio teaching

▸ The FAUST world (OWL, Awabot, Smartphones, moForte Guitar, Web Hapsichord, moForte Geo Shred)



FEEVER: FAUST Everyware

How do we get there?

FEEVER package structure:

- Notation: T1 Models, T2 Compiler
- Efficiency: T1 Models, T2 Compiler
- Deployment: T3 Ubiquity.
- Accessibility: T3 Ubiquity, T4 Education
- Education: T4 Education

An interdisciplinary team
... a wide-ranging project
...... a fast-moving world (standards, devices, interactions)

Outline

- Packages (results, perspectives):
  - Models
  - Compiler
  - Ubiquity
  - Education
- Highlights:
  - Liquid Clocks
  - FAUST Everyware Development
  - jsCoq
  - Clavecin+ and Web Hapsichord
- Conclusion

FEEVER Packages

T1 Models

Deliverables:

- Extensive survey of literature on synchronous programming
- Lots of work on Coq formalism and proofs (stability, DFT, unitarity, linearity)
- (new) Theory of Liquid Clocks

Assessment:

- 60%
- Multi-rate vector model (towards “spectral” FAUST)
- Code optimisation (theory)

FEEVER Packages

T2 Compiler

Deliverables:

- Multi-rate code generator (prototype)
- (new) “control” signal processor

Assessment:

- 20%
- Type and rate inference (formalism, algorithm)
- Multi-rate code generator (production)
- Code optimisation (implementation)
- Real-time constraints

FEEVER Packages

T3 Ubiquity (1/2)

Deliverables:

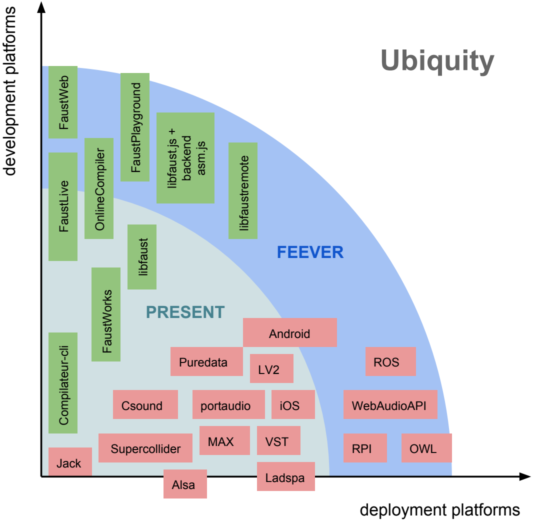
- Lots of cool new technologies
- Compilation:
  - FaustWeb (*Compiler-as-a-Service*)
  - Embeddings (Max/MSP, CSound)
- FaustLive (browser-based Faust development, smartphones)
- (new) javascript backend and library for WebAudio API
- (new) Architecture files (ROS, OWL)

Assessment:

- 90%
- Multi-rate architecture files

FEEVER Packages

T3 Ubiquity (2/2)



FEEVER Packages

T4 Education

Deliverables:

- RIM, Licence- and Master-level material (oral, projects)
- (new) FaustPlayground web environment for kids
- (new) Coq-to-javascript compiler
- and lots of events
  (Clavecin exhibit, *Turenas* and J. Smith's course at Stanford...)

Assessment:

- 30%
- Updated documentation (multi-rate)
- Tutorial on analysis-synthesis techniques
- Licence- and Master-level material (text)

Highlights

Liquid Clocks (T1)

- Generalization of FAUST type and rate system with value-dependent logico-numerical properties

$$\begin{aligned} s &::= \langle \nu : b \mid p \rangle & \qquad \textit{stream type} \\ p &::= q \mid p \wedge p \mid p \Rightarrow p & \qquad \textit{properties} \end{aligned}$$

- Quantitative properties of time-critical stream functions

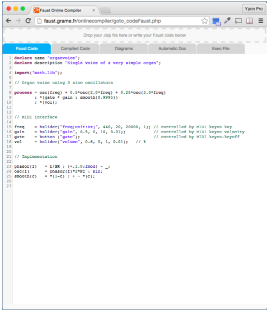
$$\begin{aligned} E \vdash \mathrm{mod} \ : \ & \langle x : \mathrm{int} \rangle \rightarrow \langle y : \mathrm{int} \mid y > 0 \rangle \rightarrow \\ & \langle \nu : \mathrm{int} \mid (\hat{x} \Leftrightarrow \hat{y} \Leftrightarrow \hat{\nu}) \wedge \hat{\nu} \Rightarrow (\nu = x \bmod y) \rangle \end{aligned}$$

- Subject reduction theorem
- *Perspectives*: SAT/SMT, safety properties (progression, determinism), FAUST property checking (bounds, on/off)

Highlights

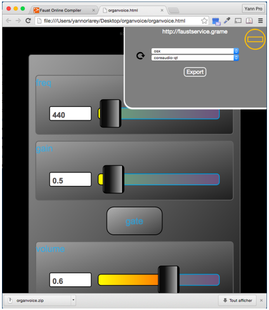
FAUST Everyware Development (T2, T3, T4)

- A unique sound technology on every platform
- Development via compilation anywhere: FaustLive, Max/MSP, FaustPlayground, online compiler, Csound, Antescofo...
- Compilation chain



Faust Online Compiler
Yann Pro
faust.grame.fr/onlinecompiler/goto_codeFaust.php
Drop your .dsp file here or write your Faust code below
Faust Code
Compiled Code
Diagrams
Automatic Doc
Exec File

```
declare name "organvoice";
declare description "Single voice of a very simple organ";

import("math.lib");

// Organ voice using 3 sine oscillators

process = osc(freq) + 0.5*osc(2.0*freq) + 0.25*osc(3.0*freq)
        : *(gate * gain : smooth(0.9995))
        : *(vol);


// MIDI interface

freq    = hslider("freq[unit:Hz]", 440, 20, 20000, 1); // controlled by MIDI keyon key
gain    = hslider("gain", 0.5, 0, 10, 0.01);           // controlled by MIDI keyon velocity
gate    = button ("gate");                             // controlled by MIDI keyon-keyoff
vol     = hslider("volume", 0.6, 0, 1, 0.01);   // %


// Implementation

phasor(f)   = f/SR : (+,1.0:fmod) ~ _;
osc(f)      = phasor(f)*2*PI : sin;
smooth(c)   = *(1-c) : + ~ *(c);
```



freq 440

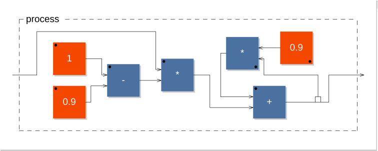
gain 0.5

volume 0.6

- ▸ FAUST Everyware development
- *Perspectives*: Multi-rate integration

Highlights

jsCoq (T1, T3, T4)



- Rich and interactive platform for DSP programming, teaching and proof checking
- “Literate programming” vision (*à la* Max/MSP)
- Coq-to-Ocaml-to-javascript compilation technology
- ▸ Bounded feedback loop
- *Perspectives*: integration with online compiler, web sound prototyping, augmented courseware (Stanford)

Highlights

Clavecin+ and Web Hapsichord (T3, T4)

- Clavecin+ dedicated synthesizer box
- FAUST-based physical modeling
- Musée d'art et d'industrie de Saint-Etienne, Musée de la Cité de la musique (Paris)
- Just inaugurated (September, 2015)
- *Perspectives*: more advanced physical modeling, ▸ Web Hapsichord

Conclusion

Self-assessment:

- Project on track (roadmap, finance)
- Exciting new prospects for science and technology advances
- *Key challenge*: multi-rate model and compiler
- Rich academic output (see report)
- Excellent interactions and cooperation between partners
- Formalized governance
- Good relationships with Imaginove and innovative companies

FEEVER: 18-Month Review Meeting

ARMINES/MINES ParisTech, GRAME, INRIA, UJM/CIEREC

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR

ANR

`www.feever.fr`