Interfacing Pd with Faust

Albert Gräf

Dept. of Music Informatics

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Interfacing Pd with Faust

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

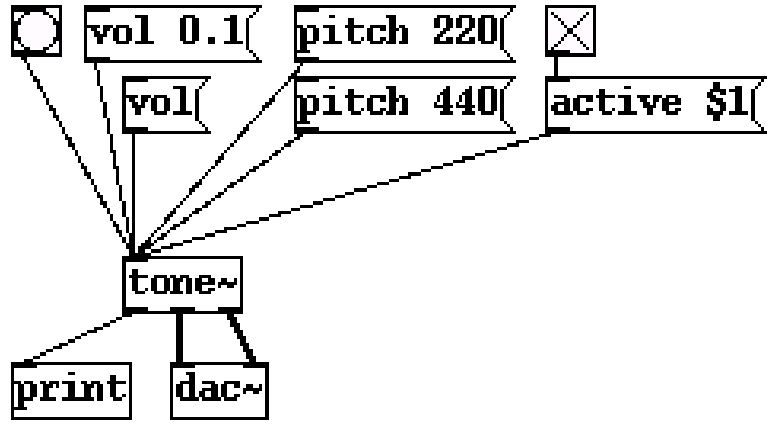
```
// tone.dsp

vol      = nentry("vol", 0.3, 0, 10, 0.01);// %
pan      = nentry("pan", 0.5, 0, 1, 0.01); // %
freq     = nentry("pitch", 440, 20, 20000, 0.01);

// simple sine tone generator

process  = osci(freq)*vol : panner(pan);
```

- **Faust programmers:** use Pd as a graphical test environment
- **Pd users:** extend Pd with audio externals programmed in Faust (Karplus-Strong etc.)



Interfacing Pd with Faust

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

```
// tone.dsp

vol      = nentry("vol", 0.3, 0, 10, 0.01);// %
pan      = nentry("pan", 0.5, 0, 1, 0.01); // %
freq     = nentry("pitch", 440, 20, 20000, 0.01);

// simple sine tone generator

process  = osci(freq)*vol : panner(pan);
```

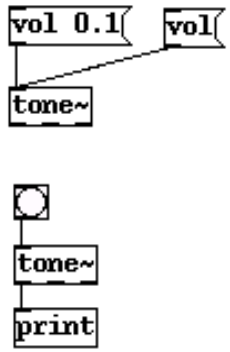
Why Faust?

- **Faust is convenient:** high-level functional programming language
- **Faust is powerful:** can do lots of things which are awkward or impossible in Pd
- **Faust is fast:** sophisticated automatic optimizations, generates C++ code
- **Faust is portable:** works with different platforms and environments, just recompile

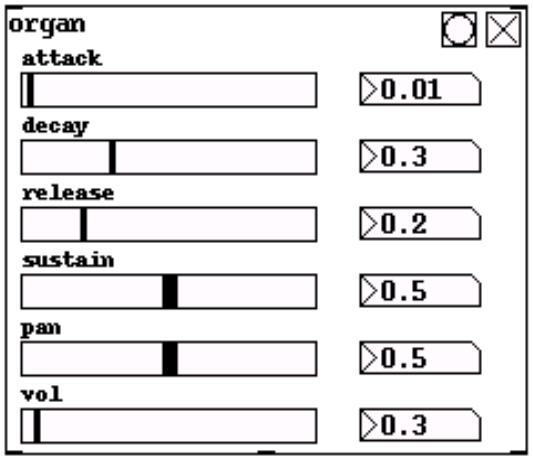
JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Main Features of Pd/Faust

- **automatic mapping** of Faust controls (`button`, `checkbox`, `nentry`, `hslider`, `vslider`; also passive controls: `hbargraph`, `vbargraph`)
- **inspection** (`bang` reports all controls)
- **control pathnames** following Faust group structure (`hgroup`, `vgroup`, `tgroup`)
- default **`active`** control (mute, bypass)
- **faust2pd:** automatic gop ("graph on parent") patches

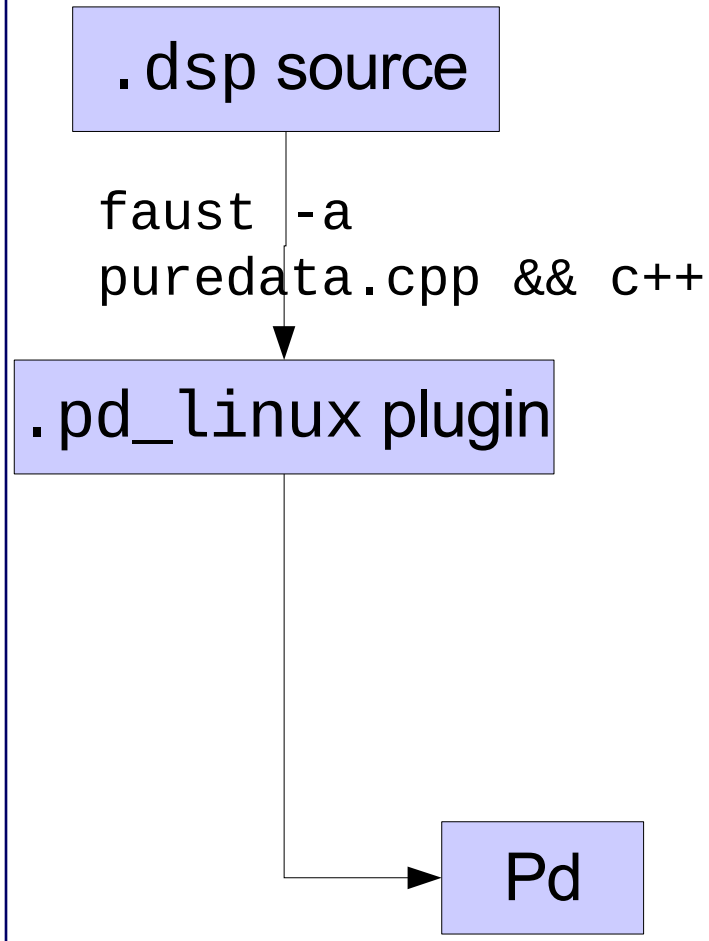


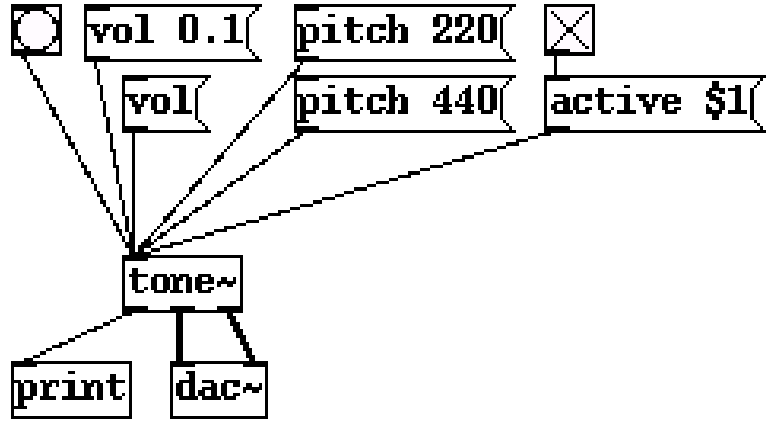

```
nentry /faust/pan 0.5 0.5 0 1 0.01
nentry /faust/pitch 440 440 20 20000 0.01
nentry /faust/vol 0.3 0.3 0 10 0.01
```



JOHANNES GUTENBERG UNIVERSITÄT MAINZ

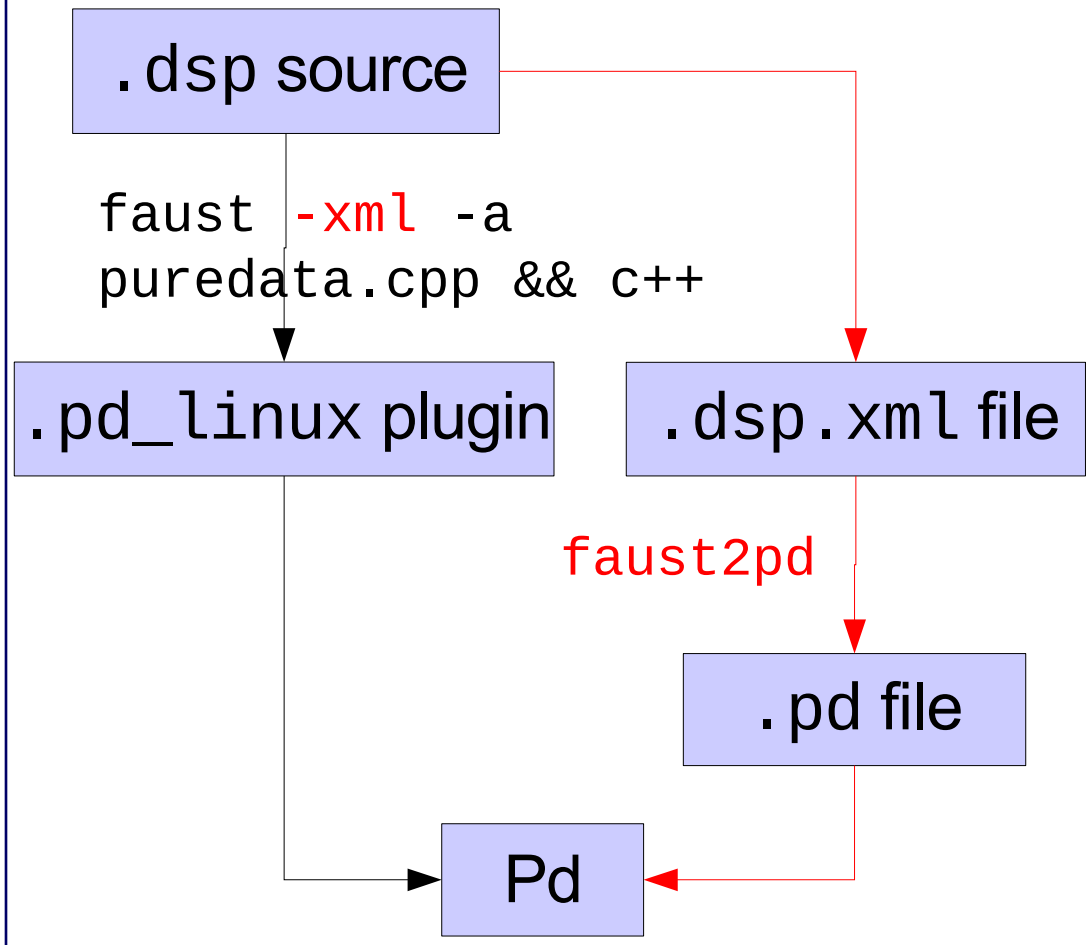
Plain Interface

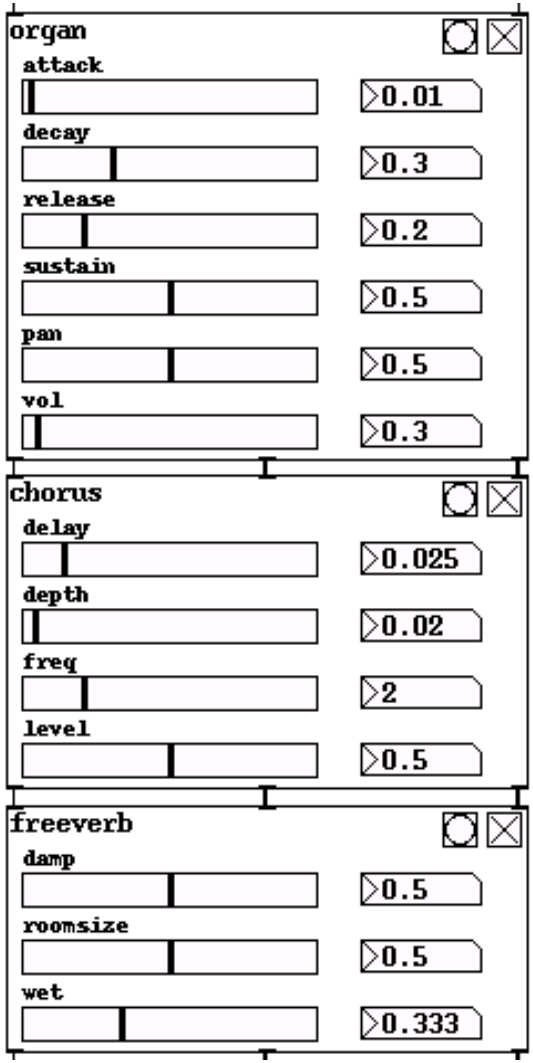




JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Deluxe Interface





JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Pd+Faust **+Q**

```
pattern   = repeat [60,60,choose [63,67]];
repeat X  = {X|repeat X};
choose Xs = Xs!rand 0 (#Xs-1);
```

- **Faust** only does audio processing
- **Q** is another functional programming language tailored for symbolic processing
- **Pd/Q external** allows Pd control objects to be implemented in Q
- **Pd+Faust+Q** = visual patching + functional programming of sophisticated audio and control objects

JOHANNES GUTENBERG UNIVERSITÄT MAINZ

Where To Get

- **Faust** (includes Pd interface): http://faust.grame.fr/ (also Web-based Faust compiler)
- **Q website** (many examples, Pd/Q interface): http://q-lang.sf.net/examples.html#Pd

Don't miss the **Faust Hands On** workshop tomorrow, Sat, 16.00!