SOUND GATES

Projectplan

Outline

- Introduction
- Generative music
- Technologies
- Soundgates
- Workplan

SOUND GATES

Introduction

Music

- **Traditional:**
  Musician performs and people perceive music

- **Trend***:*
  Interact with music (even without knowledge)
  - Cheering and shouting at a concert
  - Guitar Hero, Rockband, DJ Hero, Singstar, …

2 Level of sound generation

- **Goal:**
  Generate music in Hardware on a FPGA

- **Level 1:**
  Musician builds a synthesizer

- **Level 2:**
  User interacts with system at runtime to modify the output
  - Motion Sensors
  - 3D depth camera (i.e. Kinect)

SOUND GATES

Generative music

Approaches to generative music

- Creative / Procedural
- Interactive / Behavioural

Approaches to generative music

- Creative / Procedural
- Interactive / Behavioural

Creative / Procedural

- Generate music from precomposed options
- Musician writes song with different parts
  - parts are exchangeable and randomly played

- ie. “Mozart’s Musikalisches Würfespiel“
  - next played section was randomly chosen by rolling a dice

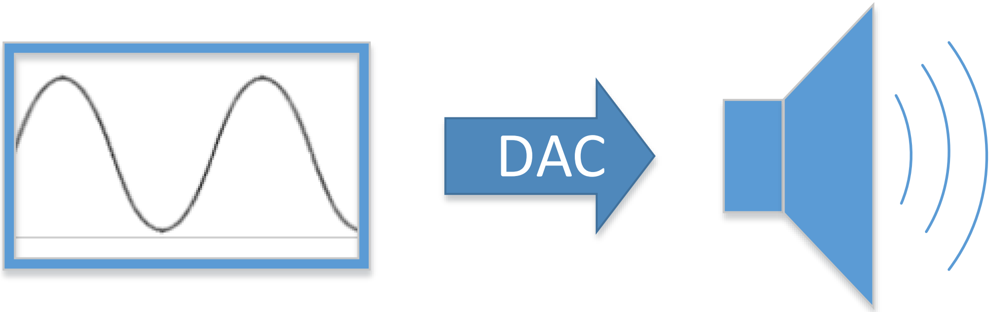
Approaches to generative music

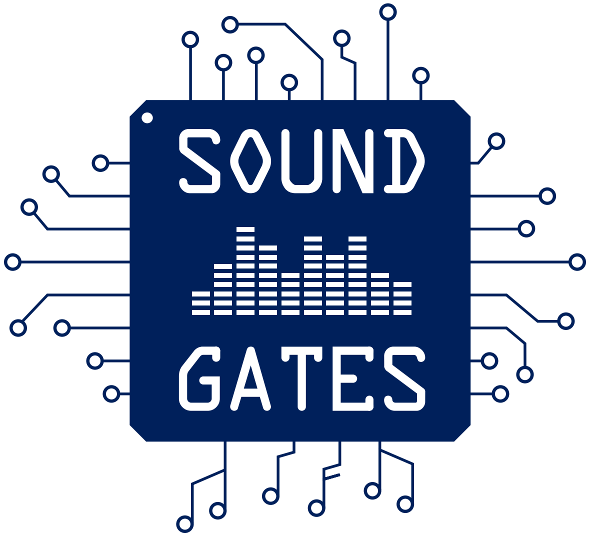
- Creative / Procedural
- Interactive / Behavioural

Interactive / Behavioural

- Results from processes without discernable musical inputs
  - uses:
    synthesized music, recorded or filtered samples
- Music generation fully controlled by user input and interaction
  - combined and filtered synthesized waveforms
  - input modified with sensors

Generate Sound on a digital System - Simple synthesizer





Soundgates

Soundgates

- Editor
- Simulator
- COSMIC

Soundgates

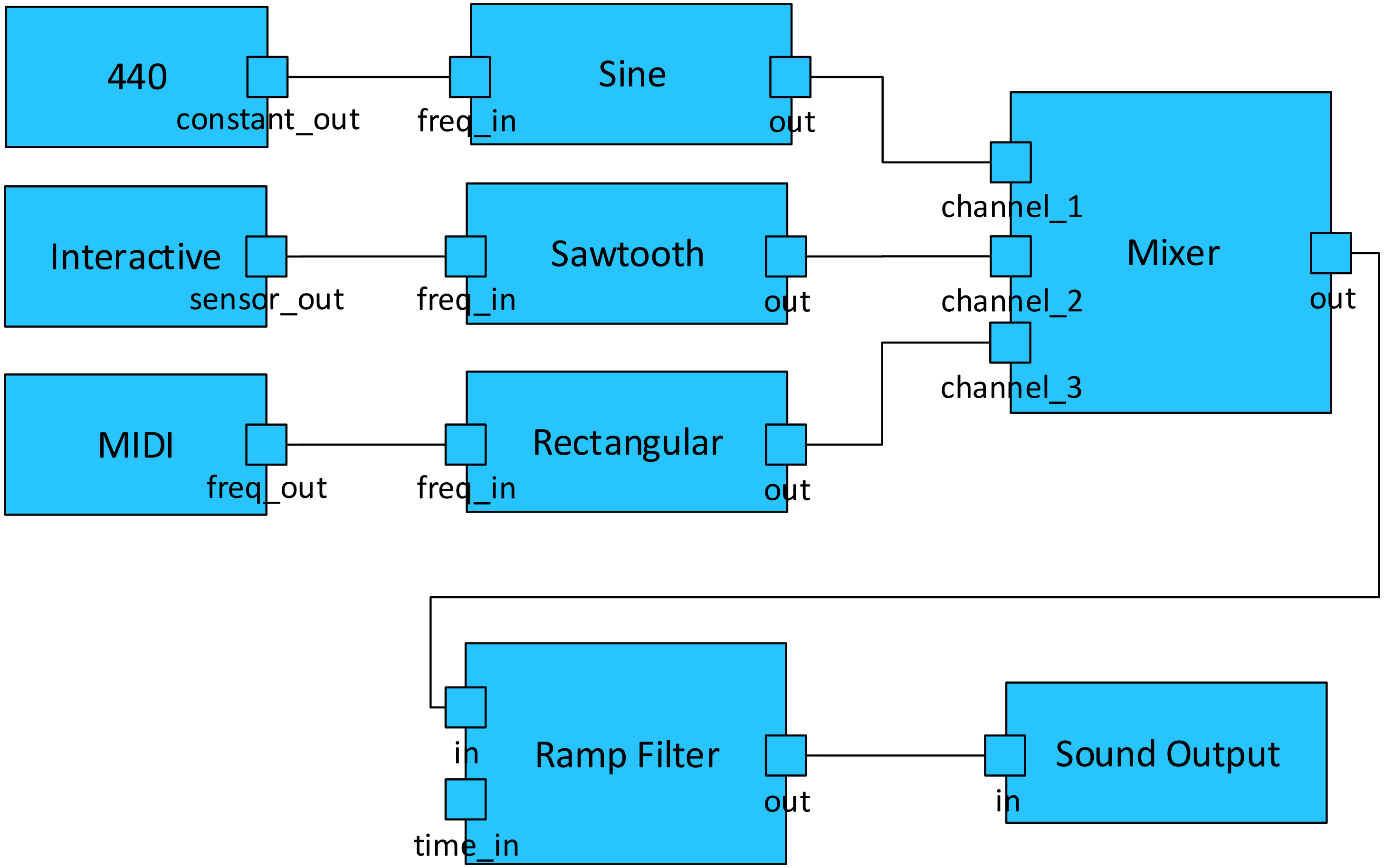
- Editor
- Simulator
- COSMIC

Editor

- Musician builds/loads a patch
  - consists of sound-components and connections

- Sound-components
  - wave generators (sine, sawtooth, rectangular)
  - arithmetic functions (i.e. addition, multiplication)
  - filters (i.e. low pass)
  - mixers
  - composite sound components

Example patch



Editor functions

- Define interface of component
  - possible to modify at runtime with sensors
- Validates patch
  - i.e. every port has an input
- Export patch to VHDL code

Soundgates

- Editor
- Simulator
- COSMIC

Simulator

- **Problem:**
  Testing the output is not possible until VHDL code is synthesized

- **Solution:**
- Test the developed system on PC
  - Every component will be implemented in SW & HW

Soundgates

- Editor
- Simulator
- COSMIC

COSMIC

- **Co**mputer **S**cientists **M**aking Mus**ic**
- The generated Bitstream is put on FPGA
- Performer maps sensors to interfaces
  - starts session by pushing a button
- Creates input values with sensors
  - music will be generated / modified

SOUND GATES

Technologies

GMF

- “Model Driven Software Development” approach for graphical editors
- Eclipse framework
- Specify Metamodel and generate software
- **Used for:**
  Create graphical editor to build patches and generate VHDL code

ReconOS

- Operating System on FPGA with soft core CPU
- Linux
- Integration of soft- and hardware threads
  - communication abstracted by method calls

- **Used for:**

  Sensor input processed in software (IPC)
  - modifies parameters of HW

Open Sound Control (OSC)

- Message based communication protocol
- Independent of transport protocol
- OSC message
  - i.e. /synthesizer1/oscillator/sine1/freq “int32” 440

- **Used for:**
  Sensors will send OSC messages to FPGA system to modify parameters

SOUND GATES

Workplan

Agile inspired development process

- 5 milestones
  - each consists of a set of tasks
  - approximately five to six weeks per milestone
- “Github” for versioning and sharing of code
- “Redmine” to represent milestones and tracking of tasks and bugs
- Functional system at the end of every milestone

Milestones

1. Prototyping infrastructure / environment
   - fundamental infrastructure is prototyped
   - no direct communication between them

2. Prototype of a digital synthesizer
   - basic digital synthesizer can be modeled with the editor
   - transform patch to HDL description

Milestones

3. Polishing editing environment
   - emulate the sound of an analog synthesizer (OF EVERY ANALOG SYN?)
   - create Android application to stream senor data to the COSMIC system
   - additional audio processing components
4. System integration and benchmarking
   - evaluate system limits

Milestones

5. Documentation, Testing, Presentation
   - polishing phase