MARS - a music mixing language

Overview

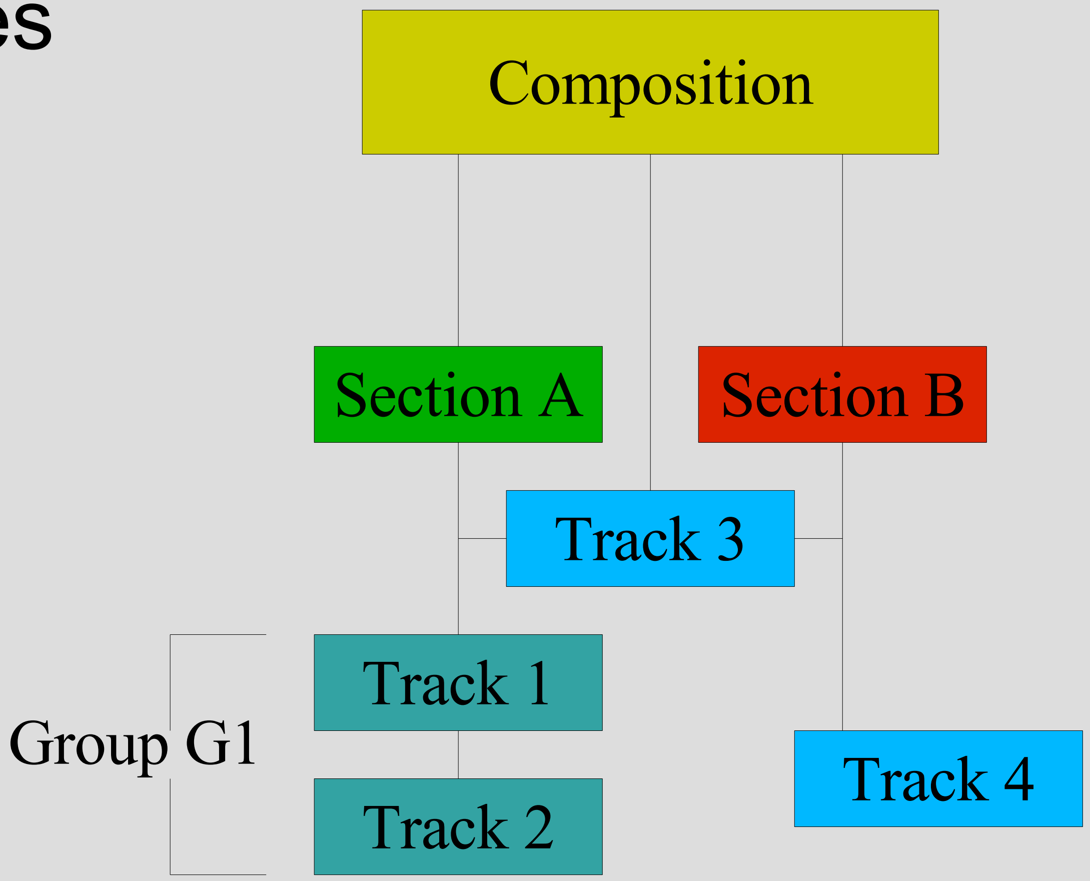
- What's the point?
  - A lot of music is repetitive
    - Mozart
    - Britney Spears
  - Why not let a computer handle the repetition?
    - predefined loops
    - MARS language
    - Creativity
- Problem with professional software?
  - too confusing!
  - grad students don't have any money :(

Overview

- Data Structures vs Music Compositions
  - Complex data structures
    - Collection of primitives
    - Executed in certain order or concurrently
  - Complex music composition
    - Collection of music tracks or parts
    - Played in certain order or concurrently

Overview

- Define music entities
  - Composition
    - Sections
    - Groups
    - Tracks
- Add behaviors
  - Play
  - Delay
  - Loop
  - Mix
  - Volume control



Your first composition

Everything is wrapped in a composition

```
def composition HelloOpus
  track swapneel = “HelloWorld.wav”
  track ritika = “Nifty.wav”

  def section MainSection
    swapneel.play()
    ritika.play()
    mix(swapneel.play(), ritika.play())
  end

  playOrder(MainSection)
end
```

Your first composition

- `def composition / end`
  - Defines new composition
  - Each composition is its own file
    - Like Java!
  - Compositions have
    - Sections, groups, and tracks

Your first composition

- `track swapneel = “HelloWorld.wav”`
  - Tracks are the building block – sound files
- `def section MainSection / end`
  - Sections are defined subdivisions of a song
  - Used to represent Musical Form
    - Mozart – A B A form
    - Britney Spears - Intro, Bridge, Chorus
  - We only have one section here
  - More complex song, more repeated sections

Your first composition

System Commands

- `play(), play(double)`
  - Plays a given track at that moment
  - Optional parameter to only play to certain portion
- `mix(track, track..)`
  - Mixes tracks at same time (super-impose)
    - Mix on play command

- Many other advanced commands
  - fadeIn
  - setVolume
  - delay
  - getLength

Your first composition

- `playOrder(section...)`
  - Acts as a “main” for the composition
  - Will play defined sections in a certain order
  - Mandatory, even if there is only one section
  - Sections can be repeated, common for songs
    - Demo song has 4 sections, played in this order:
      - Verse
      - Bridge
      - Chorus
      - Guitar Solo
      - Verse
      - Bridge
      - Chorus
      - Rap Ending

More advanced features

- For loops
  - The ability to loop tracks with traditional for loops
    - Used to accomplish something on each iteration
      - (i.e. Volume change)
- Groups
  - Used to “Group” tracks
    - Rhythm group consists of bass drum, cymbal, snare
  - Useful for coding group once, used repeatedly

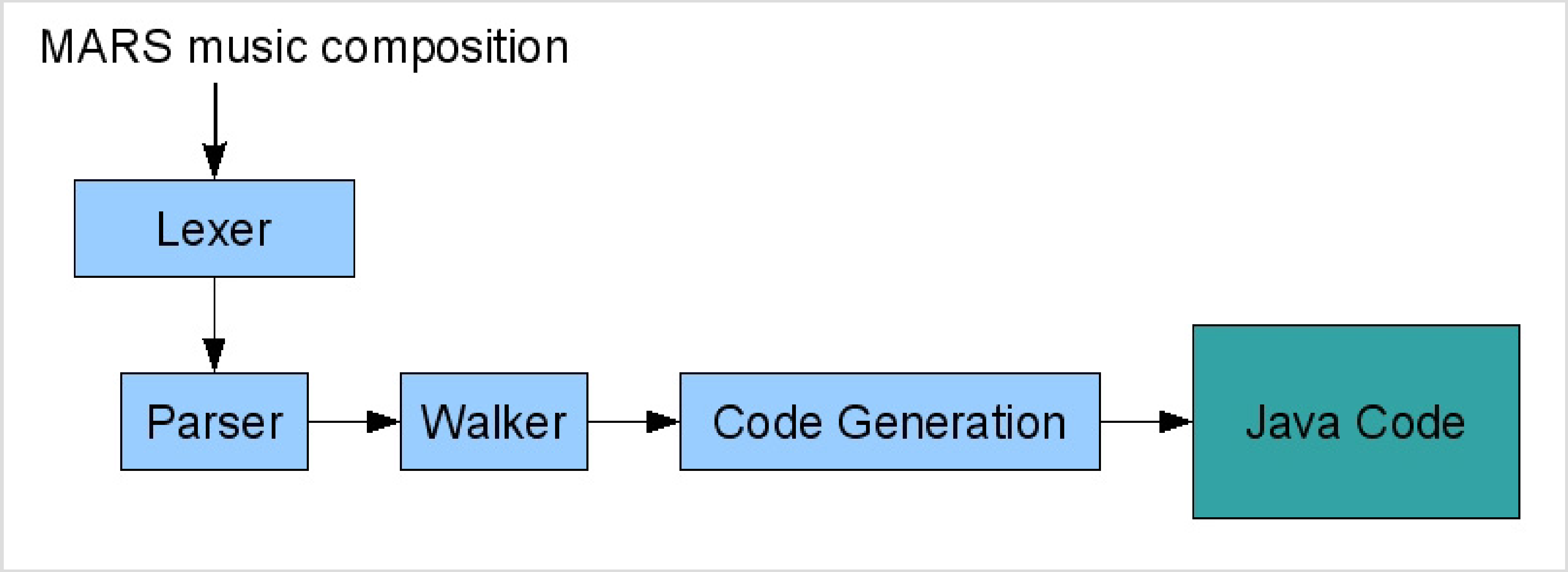
Even More advanced features

- Conditionals
  - If statements, and If/Else statements
  - Used to only play under certain conditions
    - Perhaps only every other iteration?

- Scope
  - Static scoping
  - def / end defines scope
  - Global scope
    - Things not defined inside sections or groups

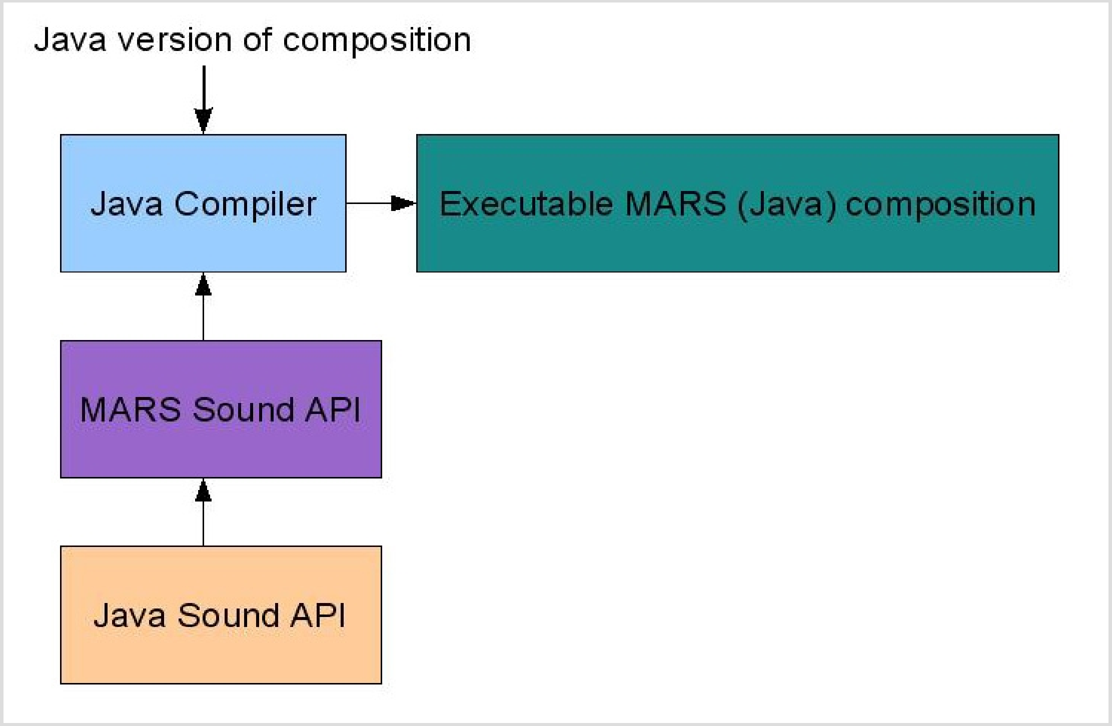
Implementation

- First half of process
  - Generate Java code from MARS (*.mars) composition



Implementation

- Second half of process
  - Generate executable Java composition
  - Interface with MARS Sound API
    - Abstracted version of Java Sound API


Java version of composition
Java Compiler
Executable MARS (Java) composition
MARS Sound API
Java Sound API

Lessons Learned

- Unique language
  - Had a hard time defining semantics
  - Had a hard time understanding control flow
- Start with small subset of language
  - It is easy to overshoot features
  - Start small, then add...NOT the other way!
- Start earlier
  - Easy to put PLT as last priority. Not a good idea!

Summary

- Fun language
  - Creativity in language
  - Fun to test because of sound
  - Harder to test because of sound
- Makes life easier
  - Easy to repeat thing like sections and groups
  - Easy to iterate on your composition
  - Easy to see relationship between tracks/sections
- Not very practical
  - Very hard to “debug” a music language
  - You need a good ear and a lot of patience