

On The Design of Csound5

LAC2005

John fitch

Introduction

Introduction

Csound written by Barry Vercoe *circa* 1985 as a member of the MusicV family. It provides a wide variety of sound synthesis and analysis techniques.

Introduction

Csound written by Barry Vercoe *circa* 1985 as a member of the MusicV family. It provides a wide variety of sound synthesis and analysis techniques.

I started working on Csound about 1990 when I ported it to a DOS machine as a wet-weekend activity

Introduction

Csound written by Barry Vercoe *circa* 1985 as a member of the MusicV family. It provides a wide variety of sound synthesis and analysis techniques.

I started working on Csound about 1990 when I ported it to a DOS machine as a wet-weekend activity

A large (?) and growing, diverse, user community, supported by a small group of programmers.

end

Problems over Time I: Legal

Problems over Time I: Legal

- **Licensed under the M.I.T. licence that allows educational and research purposes only For any other uses...including... distribution...specific prior permission...must be obtained.**

Problems over Time I: Legal

- Licenced under the M.I.T. licence that allows **educational and research purposes only For any other uses...including... distribution...specific prior permission...must be obtained.**
- So, assuming creating music is research, download sites were in effect a monopoly, and actually granted to just two of us.

Problems over Time I: Legal

- Licenced under the M.I.T. licence that allows **educational and research purposes only For any other uses...including... distribution...specific prior permission...must be obtained.**
- So, assuming creating music is research, download sites were in effect a monopoly, and actually granted to just two of us.
- Conflicts with other innovations, and isolated the Csound software from other developments.

end

Problems over Time II: Use

Problems over Time II: Use

- Users started demanding real-time output

Problems over Time II: Use

- Users started demanding real-time output
- Demands for embedding Csound in other programs and/or other programs in Csound

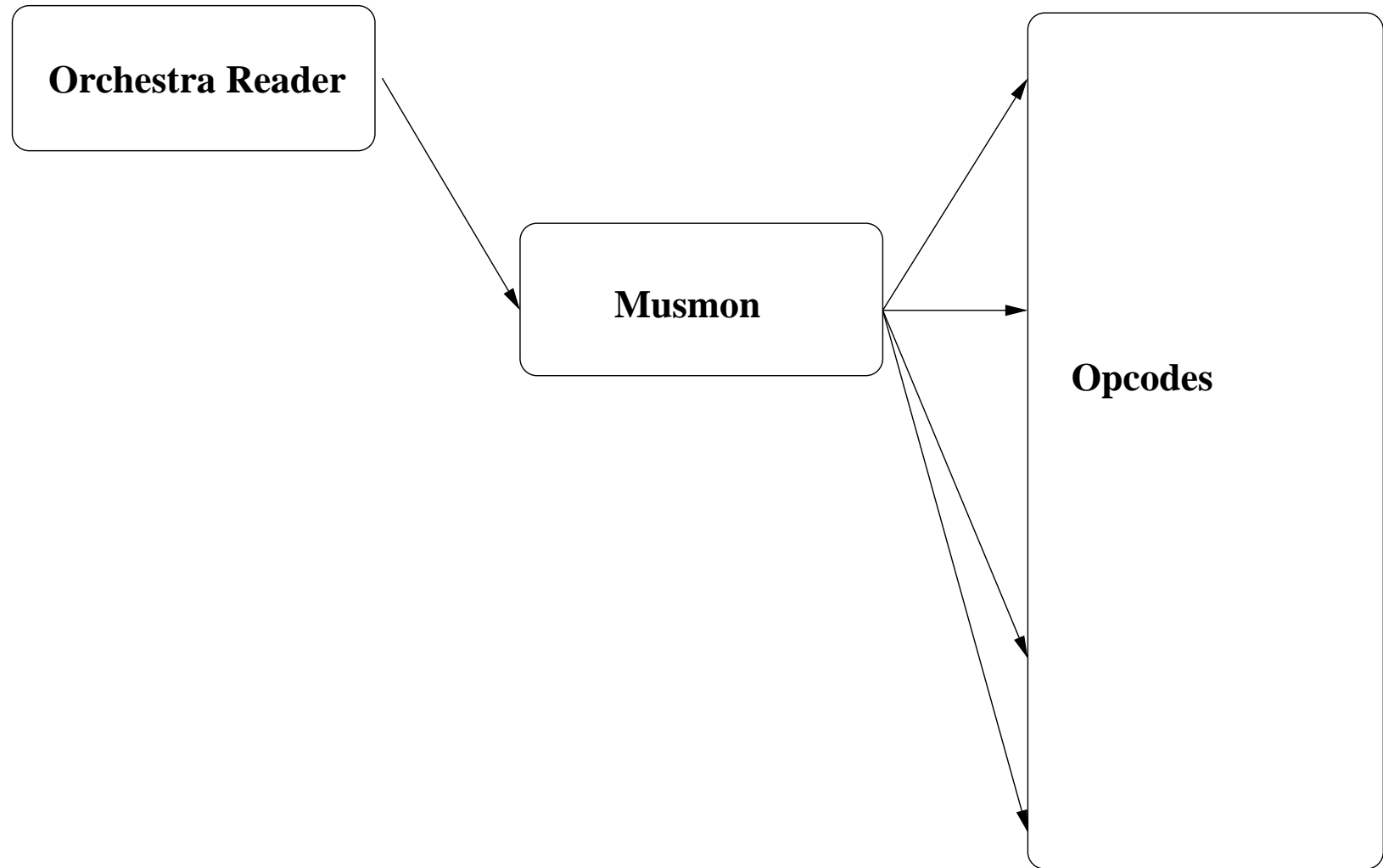
end

Towards The Big Freeze

Csound was written in K&R C, and converted to ANSI in the 1990s. The code style is 1970's, with *ad hoc* parsing processes and few comments.

Eventually MIT's lawyers agreed to a licence change to LGPL, and all authors agreed to the new licence.

Architecture of Csound4



The Big Freeze

In 2001, following false starts, we began the **big freeze**.

The Big Freeze

In 2001, following false starts, we began the **big freeze**.

- No new opcodes or language developments.

The Big Freeze

In 2001, following false starts, we began the **big freeze**.

- No new opcodes or language developments.
- Bug fixing would continue as necessary

The Big Freeze

In 2001, following false starts, we began the **big freeze**.

- No new opcodes or language developments.
- Bug fixing would continue as necessary
- A separate code tree would be started

The Big Freeze

In 2001, following false starts, we began the **big freeze**.

- No new opcodes or language developments.
- Bug fixing would continue as necessary
- A separate code tree would be started
- A small plugin system was made available for Csound for users to create new operations if they must.

end

Requirements

Requirements

- Make the system embeddable by creating an API

Requirements

- Make the system embeddable by creating an API
- Use of other people's libraries as much as possible

Requirements

- Make the system embeddable by creating an API
- Use of other people's libraries as much as possible
- Re-enterable and multi-instantiation of code

Requirements

- Make the system embeddable by creating an API
- Use of other people's libraries as much as possible
- Re-enterable and multi-instantiation of code
- User interface code to be separated from the engine

Requirements

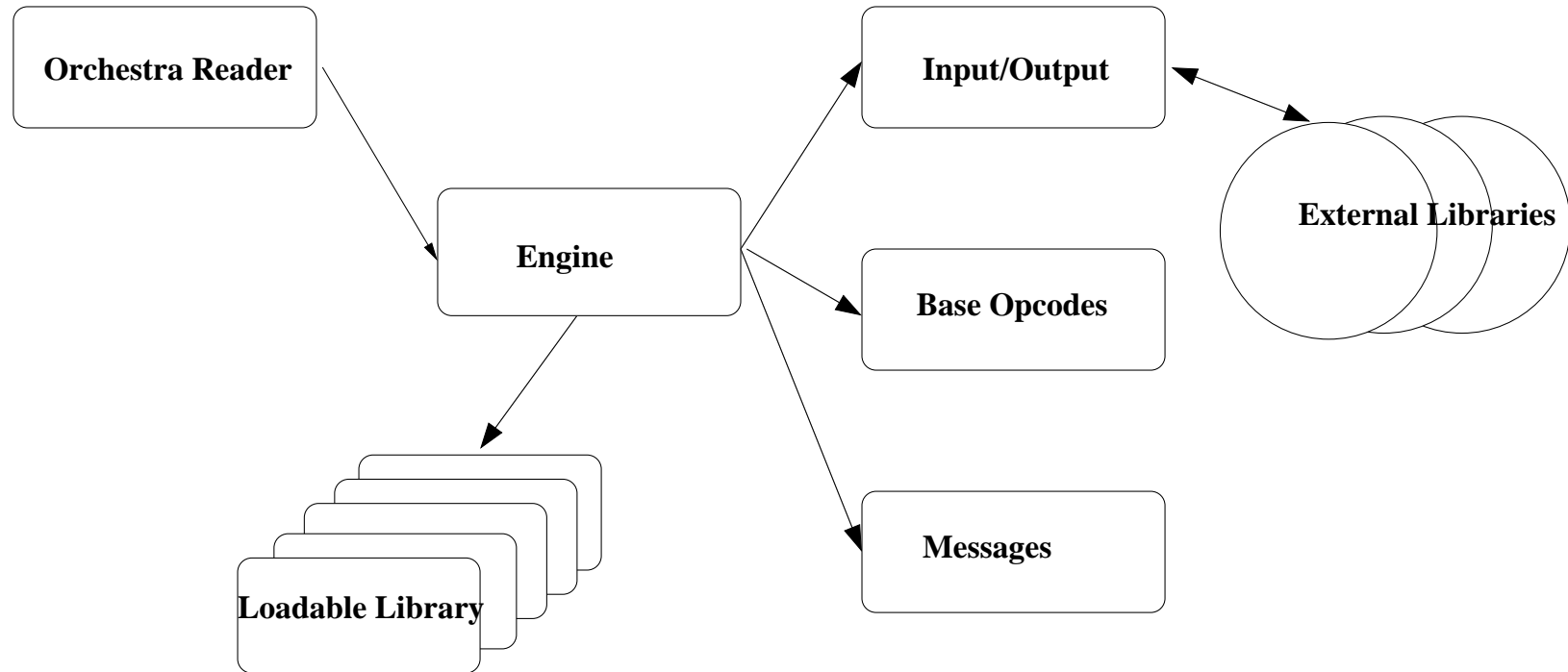
- Make the system embeddable by creating an API
- Use of other people's libraries as much as possible
- Re-enterable and multi-instantiation of code
- User interface code to be separated from the engine
- A new parser that is maintainable

Requirements

- Make the system embeddable by creating an API
- Use of other people's libraries as much as possible
- Re-enterable and multi-instantiation of code
- User interface code to be separated from the engine
- A new parser that is maintainable
- Cross platform and ALL OLD PIECES MUST RUN UNCHANGED

end

Architecture of Csound5



Implementation

Implementation

- API largely the work of Michael Gogins, and he has used it in VST plugin Csound.

Implementation

- API largely the work of Michael Gogins, and he has used it in VST plugin Csound.
- Libraries for
 - Audio files (libsndfile)
 - Realtime audio I/O (portaudio)
 - MIDI (portMIDI)
 - Graphics (FLTK)

Implementation

- API largely the work of Michael Gogins, and he has used it in VST plugin Csound.
- Libraries for
 - Audio files (libsndfile)
 - Realtime audio I/O (portaudio)
 - MIDI (portMIDI)
 - Graphics (FLTK)
- A Static and Global hunt continues, but most now encapsulated in a single `struct` that maintains the instances' state.

Implementation (continued)

- Use of **SCons** as the controlled builder; multi-platform and has proved robust and a good decision after many problems with autoconf *etc..*

Implementation (continued)

- Use of **SCons** as the controlled builder; multi-platform and has proved robust and a good decision after many problems with *autoconf etc..*
- Running on Linux, OSX, Windows and OS9, and probably others.

Implementation (continued)

- Use of **SCons** as the controlled builder; multi-platform and has proved robust and a good decision after many problems with autoconf *etc..*
- Running on Linux, OSX, Windows and OS9, and probably others.
- And still no parser.....

end

Unplanned Developments

Unplanned Developments

Initially the idea of plugins was suggested to allow commercial or patented opcodes to be deployed. I moved many existing opcodes to plugins as a demonstration that it could be done.

Unplanned Developments

Initially the idea of plugins was suggested to allow commercial or patented opcodes to be deployed. I moved many existing opcodes to plugins as a demonstration that it could be done.

Now most opcodes are loaded from libraries.

Unplanned Developments

Initially the idea of plugins was suggested to allow commercial or patented opcodes to be deployed. I moved many existing opcodes to plugins as a demonstration that it could be done.

Now most opcodes are loaded from libraries.

Extended to allow plugin table generators

Unplanned Developments

Initially the idea of plugins was suggested to allow commercial or patented opcodes to be deployed. I moved many existing opcodes to plugins as a demonstration that it could be done.

Now most opcodes are loaded from libraries.

Extended to allow plugin table generators

Extended to allow plugin realtime audio and MIDI

Unplanned Developments

Initially the idea of plugins was suggested to allow commercial or patented opcodes to be deployed. I moved many existing opcodes to plugins as a demonstration that it could be done.

Now most opcodes are loaded from libraries.

Extended to allow plugin table generators

Extended to allow plugin realtime audio and MIDI

Extended to allow plugin utilities.....(as of Monday)

end

Experience

Experience

- The whole process has taken much longer than expected, and caused much more aggravation.

Experience

- The whole process has taken much longer than expected, and caused much more aggravation.
- Libsndfile was an excellent decision and has worked.

Experience

- The whole process has taken much longer than expected, and caused much more aggravation.
- Libsndfile was an excellent decision and has worked.
- Portaudio has been less successful; we started with v19 expecting it to be finished well before us. It works on Linux, but the direct ALSA plugin seems better. On OSX we have had much trouble. Windows just seems confusing.

Experience

- The whole process has taken much longer than expected, and caused much more aggravation.
- Libsndfile was an excellent decision and has worked.
- Portaudio has been less successful; we started with v19 expecting it to be finished well before us. It works on Linux, but the direct ALSA plugin seems better. On OSX we have had much trouble. Windows just seems confusing.
- PortMIDI is OK, but does not do as much as hoped.

Experience

- The whole process has taken much longer than expected, and caused much more aggravation.
- Libsndfile was an excellent decision and has worked.
- Portaudio has been less successful; we started with v19 expecting it to be finished well before us. It works on Linux, but the direct ALSA plugin seems better. On OSX we have had much trouble. Windows just seems confusing.
- PortMIDI is OK, but does not do as much as hoped.
- Much code cleaning has been done, but lost localisation.

end

Current State

Current State

- On Linux it works well, reported as fast and robust. Has been used in live performance. I would not use any other than CS5 now.
- On Windows is almost acceptable.
- Manual now under control.
- Still no parser.....

end

Current State

- On Linux it works well, reported as fast and robust. Has been used in live performance. I would not use any other than CS5 now.
- On Windows is almost acceptable.
- Manual now under control.
- Still no parser.....

end

Current State

- On Linux it works well, reported as fast and robust. Has been used in live performance. I would not use any other than CS5 now.
- On Windows is almost acceptable.
- Manual now under control.
- Still no parser.....

end

Conclusions

This was an exercise in evolution not revolution. We are close to release, and already have users.

I also wish to acknowledge the significant efforts of the community in saying what they want; and in particular **Michael Gogins** for the API, and Windows development. The influence of **Richard Boulanger** is always with me to encourage when things go badly.

And especially **Istvan Varga**, who has done so much in the last 3 months, especially while I was disabled with RSI.