

# **Csound5: The Design, The Outcome, The Experience**

*Electric Sounds '05*

John ffitch

# 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, changing group of programmers.

# 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, changing group of programmers. Note that the developers are composers, performers and musicians, are want to use the system themselves.

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 three of us. CDP asked early; Rick Boulanger and myself were the others.

# 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 three of us. CDP asked early; Rick Boulanger and myself were the others.
- Conflicts with other innovations, and isolated the Csound software from other developments.  
We were getting left behind.

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  
Interesting to note the Extended Csound project of ADI. We were attempting real-time performance with special DSP processors. Now any laptop can do this.

# Problems over Time II: Use

- Users started demanding real-time output  
Interesting to note the Extended Csound project of ADI. We were attempting real-time performance with special DSP processors. Now any laptop can do this.
- Demands for embedding Csound in other programs and/or other programs in Csound.

# Problems over Time II: Use

- Users started demanding real-time output  
Interesting to note the Extended Csound project of ADI. We were attempting real-time performance with special DSP processors. Now any laptop can do this.
- Demands for embedding Csound in other programs and/or other programs in Csound.  
Including me, with research in use of performance agents.

end

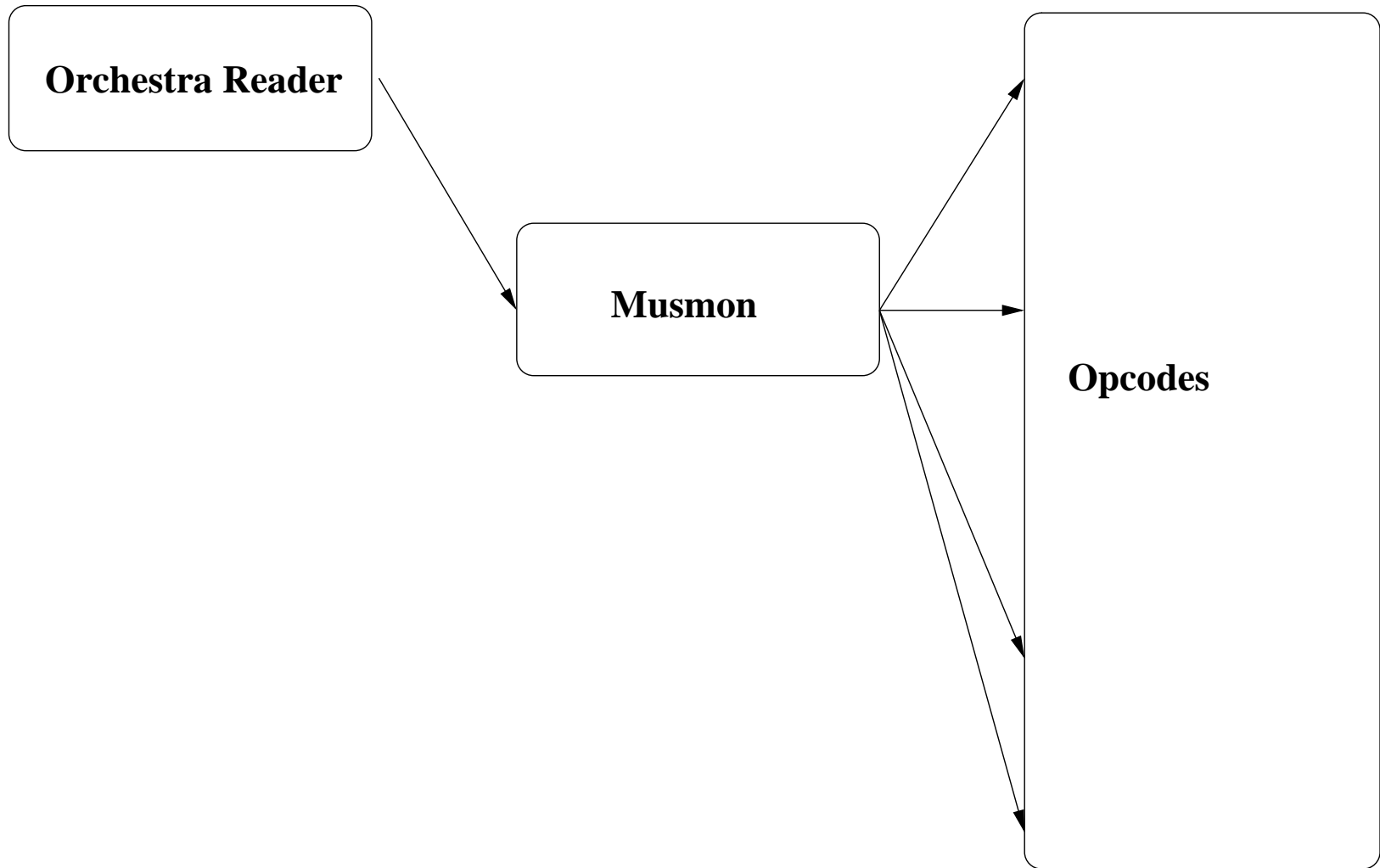
# Towards The Big Freeze

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 as they must.

end

# Requirements for Csound5

# Requirements for Csound5

- Make the system embeddable by creating an API.



# Requirements for Csound5

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

# Requirements for Csound5

- 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 for Csound5

- 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 for Csound5

- 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.

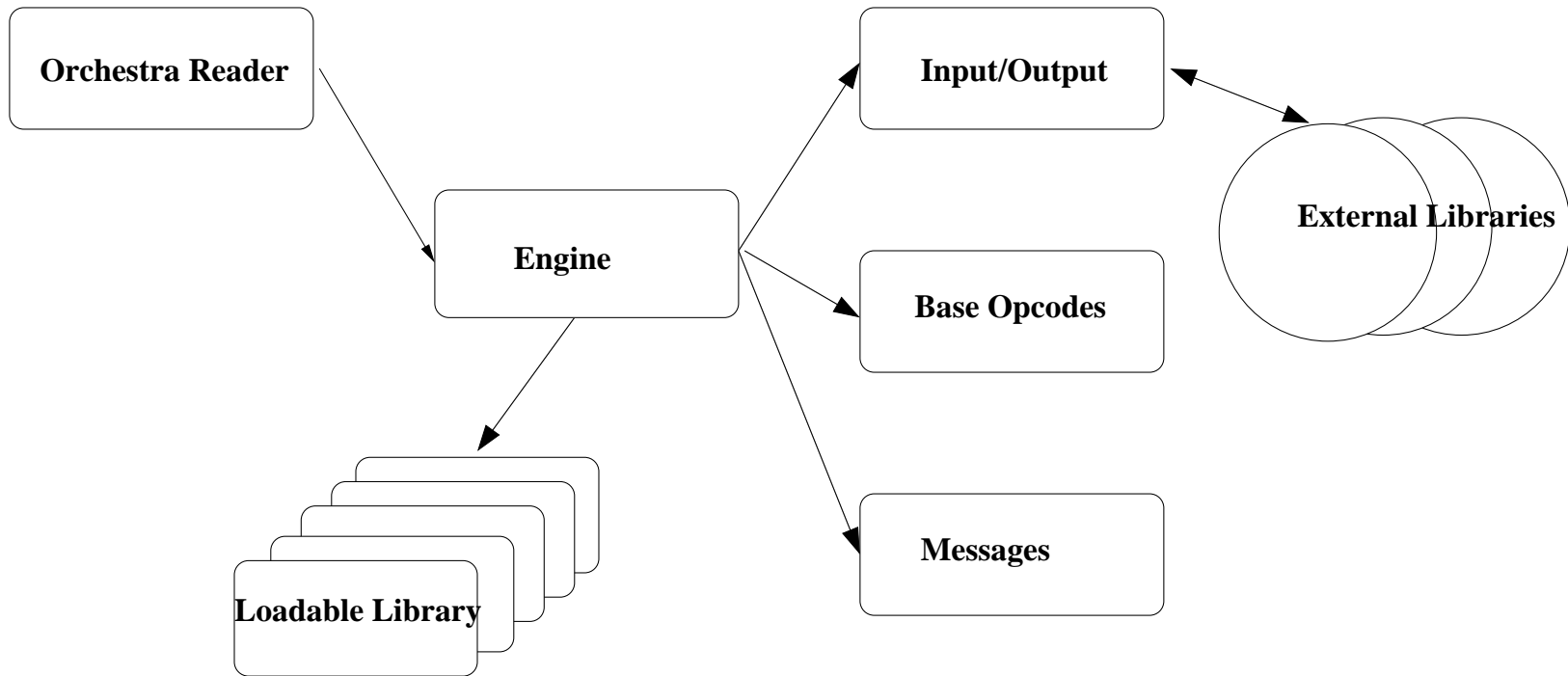
# Requirements for Csound5

- 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.
- Software bus for communication to interfaces.

# Requirements for Csound5

- 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.
- Software bus for communication to interfaces.
- Cross platform and **ALL OLD PIECES MUST RUN UNCHANGED**

# Architecture of Csound5



# Implementation



# Implementation

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

# Implementation

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

# Implementation

- API largely the work of Michael Gogins, and he has used it in VST plugin Csound and similar systems.
- Libraries for
  1. Audio files (libsndfile)
  2. Realtime audio I/O (portaudio)
  3. MIDI (portMIDI)
  4. Graphics (FLTK)
  5. *etc*
- A Static and Global hunt ensued, 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.
- Basic rendering engine is unchanged — only code cleansing. So still a MusicV system, with the same execution model.

# 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.
- Basic rendering engine is unchanged — only code cleansing. So still a MusicV system, with the same execution model.
- 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 modules for 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 modules for realtime audio and MIDI

Extended to allow plugin-based utilities

end

# Internal Developments

- Coding changes to use results of functions rather than globals.

# Internal Developments

- Coding changes to use results of functions rather than globals.
- Better error reporting and recovery

# Internal Developments

- Coding changes to use results of functions rather than globals.
- Better error reporting and recovery
- Re-entrance and reuse considered

end



# Experience I

# Experience I

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

# Experience I

- The whole process has taken much longer than expected, and caused much more aggravation.
- Libsndfile was an excellent decision and has worked.  
*There is an outstanding issue related to looping files on which I am working.*

# Experience I

- The whole process has taken much longer than expected, and caused much more aggravation.
- Libsndfile was an excellent decision and has worked. *There is an outstanding issue related to looping files on which I am working.*
- 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. We have looked at alternative libraries.

# Experience I

- The whole process has taken much longer than expected, and caused much more aggravation.
- Libsndfile was an excellent decision and has worked. *There is an outstanding issue related to looping files on which I am working.*
- 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. We have looked at alternative libraries.
- PortMIDI is OK, but does not do as much as hoped.

# Experience II

- Much code cleaning has been done, but lost localisation.

# Experience II

- Much code cleaning has been done, but lost localisation.
- OSC integration via the liblo library. Not yet exploited but potentially important.

# Experience II

- Much code cleaning has been done, but lost localisation.
- OSC integration via the liblo library. Not yet exploited but potentially important.
- We have a JACK interface, and this is possibly an important way to develop.



# Experience II

- Much code cleaning has been done, but lost localisation.
- OSC integration via the liblo library. Not yet exploited but potentially important.
- We have a JACK interface, and this is possibly an important way to develop.
- I am considering how to use additional features, such as libaubio which is good at beat extraction.

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.
- On OSX very close to acceptable.

# 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.
- On OSX very close to acceptable.
- Manual now under control, and indeed excellent..

# 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.
- On OSX very close to acceptable.
- Manual now under control, and indeed excellent..
- Still no parser.....

end

# What Now?

# What Now?

I would like to announce that Csound5 is finished and I will be moving on to Csound6.....

# What Now?

I would like to announce that Csound5 is finished and I will be moving on to Csound6.....

.....

.....

But I cannot! There are a small number of things I would like to address before that can be the case.



# What Now (really)?

# What Now (really)?

The most important issue facing the developers now is HOW to make this all available to the wider community. Csound5 is in many ways a simpler system, but the dependency on external libraries make installation harder.

# What Now (really)?

The most important issue facing the developers now is HOW to make this all available to the wider community. Csound5 is in many ways a simpler system, but the dependency on external libraries make installation harder.

(Axiom) Most users do want to install from source

# What Now (really)?

The most important issue facing the developers now is HOW to make this all available to the wider community. Csound5 is in many ways a simpler system, but the dependency on external libraries make installation harder.

(Axiom) Most users do want to install from source ,so

- We need simple binary installation system(s).

# What Now (really)?

The most important issue facing the developers now is HOW to make this all available to the wider community. Csound5 is in many ways a simpler system, but the dependency on external libraries make installation harder.

(Axiom) Most users do want to install from source ,so

- We need simple binary installation system(s).
- We need to decide on WHAT to distribute (how many utilities, which language wrappers, which manual(s), which libraries)

# What Now (really)?

The most important issue facing the developers now is HOW to make this all available to the wider community. Csound5 is in many ways a simpler system, but the dependency on external libraries make installation harder.

(Axiom) Most users do want to install from source ,so

- We need simple binary installation system(s).
- We need to decide on WHAT to distribute (how many utilities, which language wrappers, which manual(s), which libraries)
- What mechanism for distribution should we use (zip, tar, rpm, custom installer, apt,...)

# What Now (personal)?

# What Now (personal)?

This section is my current personal views and may have nothing to do with the needs, desires or plans of others.



# What Now (personal)?

This section is my current personal views and may have nothing to do with the needs, desires or plans of others.

- A new parser is necessary to maintain progress. It cleans up bugs, adds desirable features, and opens up the possibility of different user language interfaces.

# What Now (personal)?

This section is my current personal views and may have nothing to do with the needs, desires or plans of others.

- A new parser is necessary to maintain progress. It cleans up bugs, adds desirable features, and opens up the possibility of different user language interfaces.
- Recent work in Bath on the Discrete Fourier Transform opens up a range of new ideas. Integrating this into the current streaming pvoc is a high(ish) priority for me.

# What Now (personal)?

This section is my current personal views and may have nothing to do with the needs, desires or plans of others.

- A new parser is necessary to maintain progress. It cleans up bugs, adds desirable features, and opens up the possibility of different user language interfaces.
- Recent work in Bath on the Discrete Fourier Transform opens up a range of new ideas. Integrating this into the current streaming pvoc is a high(ish) priority for me.
- The anarchy of Sourceforge is not always helpful in fulfilling our goals.

# What Now (personal)?

This section is my current personal views and may have nothing to do with the needs, desires or plans of others.

- A new parser is necessary to maintain progress. It cleans up bugs, adds desirable features, and opens up the possibility of different user language interfaces.
- Recent work in Bath on the Discrete Fourier Transform opens up a range of new ideas. Integrating this into the current streaming pvoc is a high(ish) priority for me.
- The anarchy of Sourceforge is not always helpful in fulfilling our goals.
- The FrontEnd/BackEnd separation is still not as clean as I would like.

# Conclusions

This was an exercise in evolution not revolution. We are close to release, and already have users. We must have a proper binary release of a simple to install system this year.

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.

I would also like to acknowledge the parts played by **Victor Lazzarini**, and of **Istvan Varga**, who has done so much in the last year, especially while I was disabled with RSI and teaching.