Software Architecture Survey Form (CS 746)

Name of system: Audacity

Reviewer: Xuan Choo

Date: 2 Oct 2011

Author of software: Dominic Mazzoni (and Roger Dannenberg)

Author of book chapter: James Crook

Five star rating of book chapter: *****

Purpose of system:

Audacity is a free, multi-platform, open-source audio recording and editor tool. It was developed by Dominic Mazzoni (then a graduate student at Carnegie Mellon University) in May 2000 as a system on which he could develop and test audio processing algorithms. Since its inception, Audacity has grown in popularity, having been downloaded more than 78 million times from sourceforge.net.

Audacity is primarily a program in which you can record, playback, and edit audio files on your computer. Unlike other audio editing software in the market, Audacity is designed to be as lightweight as possible without compromising functionality. It is also designed with the user in mind, providing the user a simple yet intuitive interface from which the user can learn the different features of Audacity through exploration and without ever needing to refer to a manual.

Some of the key features of Audacity include it's multi-platform support, being able to run on Windows, Mac's, Linux, and Unix (and any other operating system it can be compiled on). Other key features include:

- Importing and exporting of multiple audio formats (WAV, AIFF, MP3, OGG, FLAC, WMA, AAC, AC3).
- Audio editing - cutting, copying and pasting of audio clips.
- Applying effects (e.g. amplification, pitch changes, noise removal, fade in/out).
- Changing the sampling rate of audio clips.
- Additional audio effects implemented as plugins.

Basic metrics -

KLOC: Analysis of the code reveals that audacity (and included libraries) comprises roughly 310,000 lines of code

```
----------------------------------------------------------------------
Language            files          blank        comment           code
----------------------------------------------------------------------
C                     434          29945          29967         139076
Bourne Shell           26           9341          12159          75139
C++                   147          13483           8329          48581
C/C++ Header          392           7983          10865          20985
m4                      8           1837            180          14531
HTML                   36            644            300           6074
Lisp                   30           1069           2186           3772
make                   34            371            398           1010
Teamcenter def         14             83              0            462
MATLAB                  3             43              0            142
Python                  1             17             42             93
CSS                     1              0              0             81
DOS Batch               6             10              2             64
Bourne Again Shell      1              1              2              7
----------------------------------------------------------------------
SUM:                 1133          64827          64430         310017
----------------------------------------------------------------------
```

Analyzed using CLOC (count lines of code). [http://cloc.sourceforge.net/]

Project start-up: 28 May 2000

Number of major releases: 3 (1.0, 1.2, 1.3 - although, 1.3 is still in beta)

Number of developers:

Number of core developers: 17

Number of contributors: 19

Number of inactive developers: 7

Size of user community or number of installations:

Total number of downloads for all versions since inception: 78,722,671

Peak number of downloads for any single version (this can be used as a rough estimate on the number of installations there are at any one time): 43,292,360 [http://sourceforge.net/projects/audacity/files/audacity/1.2.6/stats/timeline?dates=2006-10-01+to+2011-10-02]

Major stakeholders:

As this is an open source project, apart from the developers, the other major stakeholders are the users of the software.

Use of concurrency:

Audacity spawns separate threads to manage different aspects of the audio recording / playback process. For example, a thread is used to record / playback directly to the system's sound card, while another thread handles reading / writing the data to disk, while yet another thread handles the updating of information to the display.

Implementation language:

Mainly C/C++. Other languages are also used to program some of the audio effects, namely LISP for the Nyquist effects.

Supporting software:

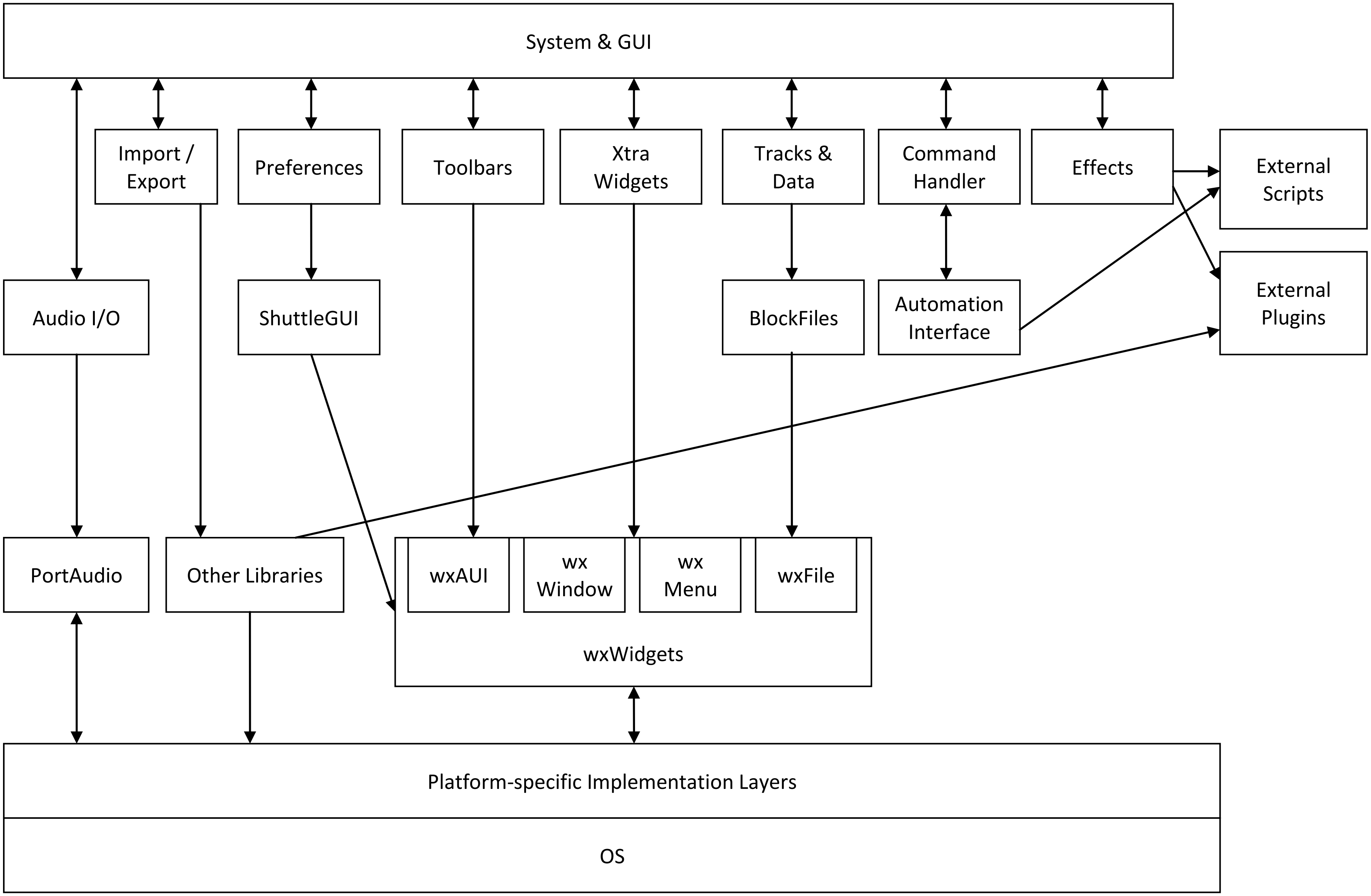
The following libraries are required for Audacity to run. Most of these libraries (with the exception of wxWidgets) are included in the source file tarball.

[http://wiki.audacityteam.org/wiki/AudacityLibraries]

- PortAudio: audio I/O library
- PortMixer: audio mixer library
- SoundTouch: pitch and tempo audio effect library
- libMad: MP3 uncompression library
- libFlac: FLAC support library
- libOgg: Ogg support library
- libVorbis: Vorbis support library
- libsndfile: importing and exporting of uncompressed audio library
- libnyquist: Nyquist audio effects library
- libresample / libsamplerate: audio resampling effect library
- LADSPA - audio plugin architecture library
- wxWidgets: GUI and other utilities library

High level architecture -

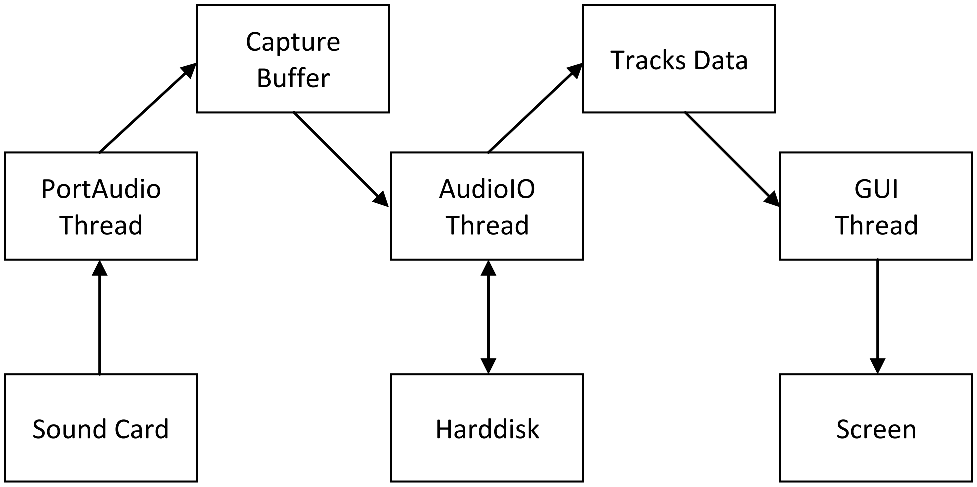
Diagram of software architecture:



High level scenarios:

Scenario: Recording audio

In this scenario, the PortAudio thread records audio data from the sound card, and places it in the capture buffer. From here, the AudioIO thread reads the capture buffer and appends the information to the audio tracks data. The AudioIO thread is also responsible for writing the audio data to the harddisk. Every GUI timer tick, and when there are free CPU cycles, the GUI thread reads this data and updates the GUI with the necessary information.



Data structures or algorithms:

Audacity uses a unique data structure / storage system (called BlockFiles) to manage audio recordings that could easily exceed the maximum memory capacity of the host system's RAM. Instead of storing the entire audio recording in RAM, the audio recording is broken down into 1MB chunks, and stored on the harddrive, and only loaded into RAM when it is needed.

Control flow and/or data key to the architecture (if any):

As mentioned previously, Audacity utilizes multiple threads to handle data during audio recording and playback. Each thread read and writes to shared buffers. However each thread is carefully controlled to avoid reading / writing to the shared buffers at the wrong time.

Architectural style:

Audacity has a layered architecture with pipelines to handle minor aspects of the program (e.g. application of audio effects).

Major evolutionary changes:

There have not been any major changes to the architecture of Audacity. The architecture of Audacity is designed with plugins in mind, so changes to the architecture are not necessary to add functionality to the program.

Performance bottlenecks:

Audacity is designed to be a lightweight program, and with the computer power available nowadays, there are no foreseeable bottlenecks.

Real time:

There are no components that require real-time response support. However, in order to record audio without loss of information, the PortAudio thread has to be as responsive as possible (real-time is good, but it is not system-crucial if real-time performance is not met).

Notation for architecture: None

Methodology:

The development style of Audacity resembles that of most open-source software - which typically employ an agile-like methodology. Although, with the current 1.3 beta build, the development has shifted away from the core modules of Audacity, to a more "modular" development of plug-in modules. [http://wiki.audacityteam.org/wiki/User:James]

Appendix:

Kruchten's eight context attributes applied to Brown/Wilson systems -

1. **Size**: M (~310KLOC)
2. **Criticality**: Lo (multimedia software)
3. **Age of system**: L (11 yrs)
4. **Rate of change**: Lo (Major release every couple of years)
5. **Business model**: Open source
6. **Stable architecture**: Lo
7. **Team distribution**: Hi (Development team meets entirely on the internet)
8. **Goverance**: Med