

# Chris Jones

## Curriculum Vitae

### Work Experience

September 2010 – Present

#### BAE Systems Detica

##### Software Engineer

- Worked on a distributed, scalable data analysis tool based on Hadoop.
- Developed audio transcription software using the Qt application framework and GStreamer media framework.

July 2008 – September 2009

#### Netcraft

##### Internet Services Developer

- Performed penetration tests against web services and applications.
- Reviewed automated vulnerability scan results and developed the scanning framework.
- Worked on the support software for Netcraft surveys and bank fraud detection service.

Summer 2007

#### IT-Innovation

##### Summer Internship

- Developed generic data access Web Service for industrial problem solving environments in the automotive industry.
- Fixed bugs and wrote JUnit based integration tests for the aforementioned and other components of the GRIA package.

Summer 2006

#### Google Summer of Code project for Ubuntu Linux

##### Summer of Code Student

- Developed Onboard - a lightweight on-screen keyboard that now ships with the Ubuntu operating system.

Address	2 Moss Court 51 Portsmouth Road Liphook Hampshire England GU30 7GG
Date of birth	17-12-1986
☎	+44 7719 901814 +44 1428 288795
✉	<a href="mailto:chris@chrisejones.com">chris@chrisejones.com</a>

### Education

2005 – 2010

#### Southampton University

##### MEng Digital Systems Engineering

- Graduated in 2010 with first class honours.
- Projects involved computer vision, robotics and mobile Linux.
- My transcript is available at <http://chrisejones.com/transcript.html>.

2003 – 2005

#### Worthing College

##### A-Levels

Maths	A
Further Maths	A
Physics	A
Chemistry	B

### Interests and Activities

- A general interest in Linux and open source software.
- Indoor wall climbing at weekends.

### References

Available on demand

## Selected Project Experience

---

### Hadoop Analytics and Storage Project

#### *Detica Investment Project - Java*

I am part of a small team working on distributed and scalable data analysis. I administrate a Hadoop test cluster (10 nodes) that we use for performance testing. We have a data analysis pipeline, written in Java, consisting of several Hadoop jobs that produce an output that is inserted into a MySQL database using Apache Sqoop.

In a real deployment our analysis pipeline would require more storage and faster ingest rates than a conventional relational database can provide. We are trialling the Apache Cassandra key-value store and have prototyped our own storage solution built on the Hadoop distributed file system (HDFS).

Our solution consists of a master daemon, slave daemons and a custom Hadoop output format that writes data and indices to HDFS directly from a Hadoop MapReduce job.

The master daemon holds a digest of each index in memory, which is used to dispatch requests to slaves that hold a local copy of matching index and data blocks. We use the ZeroMQ asynchronous messaging library for communication between the client and daemons and use Google Protocol Buffers for message serialization.

### Transcription Tool

#### *Detica Project - C++*

Detica needed a Transcription Tool to integrate with Detica NetReveal Analyzer. I was asked to create a tool allowing streaming playback of multiple voice tracks and the submission of a rich text transcription back to Analyzer.

I wrote the application in C++ using the GStreamer media framework and the Qt application framework. It has a smoothly zooming waveform visualisation of the audio, which scales to multiple tracks that are hours long. The visualisation can be used to control playback, create bookmarks and extract snippets of audio that can be saved to disk. The application is thoroughly unit tested using Qt's integrated unit test suite and is fully internationalisable. On Windows it uses the DirectInput api to allow playback control using a foot pedal peripheral.

The tool allows playback of many different audio codecs and container formats; I improved the GStreamer .au plugin and submitted a patch that was incorporated into a recent GStreamer release.

The transcription tool has been deployed to a customer and I have received good feedback.

### Onboard Onscreen Keyboard

#### *Open Source Project - Python and C*

During my first year of university I applied to the Google Summer of Code. Google sponsors students to work on an open source project during the summer holidays. I sent my proposal and design for a simple onscreen keyboard to the Ubuntu Linux project, which was accepted over many competing proposals. I made the initial release during the summer and was sponsored to attend the Ubuntu Developer's Summit in Mountain View, California

I made further releases with new features and reviewed patches for several years, before passing maintainership of the project onto a contributor. The application is written in Python and uses the GTK+ GUI toolkit. I also created a Python C extension for emulating keypresses and retrieving keyboard geometry from the xserver. Onboard is Ubuntu's default onscreen keyboard. <http://www.launchpad.net/onboard>

### Twigger Robot

#### *University Group Robotics Project - C++ and Python*

We combined a Roomba vacuum cleaner with a webcam equipped netbook to make a robot that retrieved blue tins, but avoided red tins (rice pudding & baked beans). We determined the position of a tin in the robot's field of view by calibrating the webcam with a square chessboard to generate a matrix that transformed webcam to real-world coordinates. We estimated the position of the robot relative to its starting point by integrating the readings from the rotary encoders on the Roomba's wheels.

I contributed the computer vision component of the robot's software; it was implemented in C++ and used the OpenCV computer vision library. The route finding and Roomba communications components were written in Python, so I used the Cython code generation tool to wrap my library into a Python module.

## Pragmap

### University Group Design Project - C++

We implemented an application in C++, leveraging the OpenCV and Orfeo Toolbox computer vision libraries, which processed a sequence of aerial photographs to generate a map showing the relative accessibility of an area. The application transformed and stitched the photos together to create a single image with a consistent perspective. We used three methods to determine terrain accessibility:

- Stereo correspondence between overlapping images to determine steep inclines and buildings
- Pixel colour classification using a support vector machine (SVM), a supervised machine learning technique.
- A road detection algorithm

We developed a user interface using the Qt framework that allows a user to select two points on the map. The application used the A\* search algorithm to find and display a path between the two points. I contributed the terrain classification component of the application. Our report is available at <http://www.chrisejones.com/pragmap.pdf>.

### Libraries

Hadoop  
Qt Application Framework  
GTK+  
GStreamer Media Framework  
GTK GUI Toolkit  
OpenCV  
Orfeo Toolbox  
Django  
ZeroMQ  
Protocol Buffers  
Guava  
JUnit  
JDBC  
Tika  
OpenGL  
Cython

### Languages

C/C++  
Java  
Python  
Perl  
Bash  
HTML  
XML  
LaTeX

### Software

#### IDE/Editors:

Eclipse  
Qt Creator  
Visual Studio  
Vim

#### Build Tools:

Maven  
Ant  
CMake

#### Continuous Integration:

Jenkins/Hudson

#### Version Control:

Perforce  
Git  
CVS  
Subversion  
Bazaar

#### Databases:

MySQL

#### Operating Systems:

Linux  
Windows  
Mac OSX  
Android

#### Office Suites:

Microsoft Office  
OpenOffice  
Google Docs