

Andrew Cooke

This CV is available on the web — as a “general” CV it describes a wide range of projects. If you are interested in a particular skill, please contact me for more specific details.

Personal Details

Address: Europa 2189 Depto 22,
Providencia,
Santiago,
Chile.

Phone: (56) 6 038 070
Email: andrew@acooke.org
Web: <http://www.acooke.org>

Place of birth: Harrogate, Yorkshire, UK.
Nationality: British (permanent Chilean resident).
Languages: English; Spanish.

Summary

I have broad engineering experience and a strong background in physics. Several of my previous jobs have combined these — I have written numerical software in Java, C and Fortran — but I also enjoy technical challenges. I am very self-motivated, reliable, independent and productive. I also have many years’ experience in telecommuting.

- Professional Interests**
- Domain-specific and ‘little’ languages/parsing/code-as-data/flexible configuration. These ideas often provide a good abstraction layer for building adaptable, maintainable systems.
 - Efficient numerical and semi-numerical algorithms. For example, I have developed new, efficient approaches for filtering data in one and two dimensions; more recently I worked on generating correlated, uniformly distributed random numbers.
 - How to involve the client in “lightweight” development — balancing iterative, adaptable development with clear estimates and a useful development history.
- Skills**
- Experience with Agile, Requirements-Driven and Iconix (UML) processes.
 - Over a decade of OO design experience.
 - Strong mathematical and statistical background.
 - Self-motivated problem solver.
 - Educated to PhD level (Astronomy, Cambridge University).
- Languages:** Java, Python, C, SQL (some OpenCL, Fortran, Clojure).
Platforms: Unix (Linux, (Open)Solaris), Windows 7, XP, 2000.
Web: Django, Spring, GAE, JSP, Javascript, YUI, jQuery, SVG.
J2EE, SOA: Mule, Spring, JBoss, EJBs (not entity), JMS.
Databases: PostgreSQL, Oracle, MySQL; JDBC, SQLAlchemy, Spring, IBatis.

Work Experience

2008— Senior Software Engineer. ISTI, USA (Telecommute).

ISTI develop custom software for the geophysical research community; they are based in the USA but have engineers in several countries. On each project below I was responsible for design and implementation and usually interacted with the client (typically in parallel with one of the company founders).

Hardware / Numerical A set of loosely coupled C programs that calibrate seismic detectors. These can be run separately, by hand, or under the control of a scheduler for automated calibration.

GPU / Numerical Optimisation of numerical Matlab/Octave code using OpenCL. Reduced calculation from 12m (Xeon CPU) to 10s (low cost NVidia GPU), shifting work from “batch processing” to “interactive data exploration”.

Web / Database Several projects constructing database representations of complex systems and then providing a variety of ways to access and manipulate that data — both directly (HTML, Ajax) and via additional services (REST, XMLRPC). Implemented with Java (Spring/JSP) and Python (Django/YUI).

One case called for an extensible system that could support large data volumes. The solution used Java’s Service Provider mechanism to support dynamic deployment of new data types, which were automatically mapped to the database. The schema was carefully designed to support application-level sharding.

Client Application Python (WXWidgets) GUI to simplify management of remote data processing system, including a “map” of interconnected components (auto-layout via simulated annealing).

Other I have helped introduce a variety of ideas to the company, including the use of continuous integration, lightweight progress tracking, and wider test use. Most recently I launched <http://isti.bitbucket.org> — a static site generated with jekyll — as a way to share more of our work and raise the company profile.

2007—2008 Software Engineer. MuleSource, San Francisco, USA (Telecommute).

MuleSource is the company formed to support and develop Mule, an open source Enterprise Service Bus (ESB). I was part of a geographically-disperse team maintaining the core system, particularly TCP related transports.

XML Schema I was also responsible for the main user-visible change in Mule 2.0: an XML-based configuration system using Spring’s extensible schema.

2003—2007 Scientific Programmer. CTIO, La Serena, Chile.

The team in La Serena was part of a larger development group based in Tucson, USA, that developed software for NOAO observatories.

SOA / ESB The NOAO Science Archive was developed to store and retrieve astronomical data. I worked on analysis, design, implementation, testing and documentation of the system.

This included assessing ESB systems and selecting Mule as a solution that provided good scalability, wide compatibility with existing transports, and support for rapid development with Java-based messages — a good, future-proof balance for a SOA that was still largely internal.

I also developed a design approach that isolated business logic in POJOs. This separated messaging from the main code and allowed services to be ‘plugged together’ for simple, automated integration tests.

Numerical I designed and implemented the Gemini/IRAF GNIRS Package, to process spectral data. This was based on the existing (but incomplete) NIRI package and implemented in IRAF CL/SPP (Fortran).

Work Experience (cont.)

- 2002—2003** **Head of development / Consultant. Webtron Finance, Santiago, Chile.**
At Webtron I implemented a system to receive and process financial data. That involved learning, over 7 months, how to develop J2EE-based web applications, in a new language and culture. I started as a single Java programmer, writing to a dictated design, but finished leading a small team (two programmers and web designer) to beat an impossible deadline with shifting requirements.
- 1998—2001** **Software Engineer. Intertrader Ltd, Edinburgh / Leicester, UK (Telecommute).**
For the Intertrader CashBox System I designed and implemented most of the server-side application, combining standard Java components (to become ‘J2EE’) within a dynamically configurable framework (similar to the ‘Spring’ framework, although I was unaware of that at the time) to give the flexibility necessary when working for different clients with conflicting requirements.
- 1997—1998** **Software Engineer. Concept Systems, Edinburgh, UK.**
Responsible for algorithms to calculate the position of long (5km) cables towed behind boats prospecting for oil. I developed a novel, fast algorithm for median filtering (using a sorted tree for the data within the window) and helped start an internal discussion group to encourage movement from C to C++.
- 1995—1997** **Postdoc. Institute for Astronomy, Edinburgh, UK.**
Numerical analysis (Fortran 77; maximum likelihood estimates; integration; optimisation) of the distribution of Lyman- α absorption lines to estimate the evolution of the ionizing background at high redshifts.
- 1994** **Research Assistant. CTIO, La Serena, Chile.**
Analysis of Hubble and ground-based long-slit and Fabry-Perot observations. Fortran (fitting models of gas flow to 3D spectral data) and IRAF.
- 1988—1993** **PhD in Astronomy. Institute of Astronomy, Cambridge, UK.**
Voigt profile fitting in Fortran. Observed (mainly echelle spectroscopy) at AAT, CTIO 4m, WHT. Wrote software in Fortran with IRAF / Imfort to do optimal data extraction (not supported in IRAF for echelle spectra at the time) with automatic cosmic ray rejection.
- 1985—1988** **BA in Natural Sciences (Maths and Physics); Christ’s College, Cambridge, UK.**
First class honours (final result and all intermediate examinations); received various scholarships.

Personal Work

Some recent projects:

- LEPL Parser A recursive descent parser, written in Python, which uses co-routines to avoid blowing the stack on recursion. It also compiles simple “leaf” matchers to regular expressions. Current (rather slow) development is based around extending the regular expression support to be non-exponential (see Russ Cox, 2007). <http://acooke.org/lepl>
- Dependency Arcs In-browser SVG-based display of dependencies in a library of code. <http://practi.cl>
- User Icons A service that ran on Google App Engine, to provide more attractive user icons. Died from a lack of market interest, but I learnt about GAE and the Data Store in the process.