

# Candidate 10551

## Curriculum Vitae

### **Academic Qualifications, Memberships, Awards, Courses**

| <b>Qualification, Award or Course</b>                  | <b>Institution</b>          | <b>Year</b> |
|--|-----------------------------|-------------|
| Member University of Wollongong Computer Users Society | University of Wollongong    | 1993-       |
| Life Member University of Wollongong Union             | University of Wollongong    |             |
| Unix Client/Server Network Programming                 | Softway                     | 1994        |
| Advanced C++ for C Programmers                         | Object Oriented             | 1993        |
| Masters Computer Science                               | University of Wollongong    | 1993        |
| Bachelor Mathematics                                   | University of Wollongong    | 1992        |
| Best Faculty Result - Mathematical Sciences            | Weerona Residential College | 1989        |
| Higher School Certificate                              | NSW Board of Education      | 1987        |

### **Positions Held**

| <b>What</b>                           | <b>Where</b>  | <b>When</b>          |
|---------------------------------------|---|----------------------|
| Senior Analyst Programmer             | Day 8 Technology Pty Ltd                              | Jan 2005 – Jan 2006  |
| Senior Analyst Programmer             | Vivitec Pty Ltd                                       | Aug 2004 – Nov 2004  |
| Senior Analyst Programmer / Architect | Development Infostructure                             | Jan 2002 – July 2004 |
| Senior Analyst Programmer             | Confidential  | Jul 2001 – Sep 2001  |
| Senior Architect                      | Confidential  | Dec 2000 – Jul 2001  |
| Project Leader                        | Softworks Pty Ltd                                     | Sep 2000 – Dec 2000  |
| System Architect                      | International Strategies                              | Jul 1999 – Jul 2000  |
| Senior Software Engineer              | Brisbane City Council - Travel Demand Mgmt            | Nov 1998 – Jul 1999  |
| Software Development Manager          | Advercom Pty Ltd                                      | Apr 1998 - Oct 1998  |
| Director                              | The Internet (Aust) Pty Ltd                           | 1998-                |
| Director                              | Sneaker Net Internet Service Provider                 | 1996-                |
| Senior Software Engineer              | University of Western Sydney                          | Nov 1997 - Jan 1998  |
| Senior Software Engineer              | NEC Australia, New Product Development                | Sep 1996 - Oct 1997  |
| Project Leader                        | Computer Telephony Integration Pty Ltd.               | Jan 1996 - Sep 1996  |
| Software Engineer                     | Telstar Systems                                       | May 1994 - Jan 1996  |
| Software Engineer                     | Telstra (Customised Software Solution Centre)         | Aug 1993 - May 1994  |
| Associate Lecturer                    | University of Wollongong, Computer Science Department | Jan 1993 - July 1993 |

## Skills Summary

| Rating | Explanation                           |
|--------|---------------------------------------|
| 1      | Brief Exposure                        |
| 2      | Passing Knowledge                     |
| 3      | Can Work With Some Guidance           |
| 4      | Can Work Unaided Using Own Initiative |
| 5      | Capable Of Teaching In This Area      |

| Skill                    | 1 | 2 | 3 | 4 | 5 | Last Used In |
|--------------------------|---|---|---|---|---|--------------|
| <b>Languages</b>         |   |   |   |   |   |              |
| C++                      |   |   |   | ✖ |   | 2005         |
| C                        |   |   |   |   | ✖ | 2005         |
| Python                   |   |   |   | ✖ |   | 2006         |
| ZOPE                     |   |   |   | ✖ |   | 2006         |
| Java                     |   |   | ✖ |   |   | 2002         |
| Modula-2                 |   | ✖ |   |   |   | 1995         |
| HTML                     |   |   |   | ✖ |   | 2006         |
| WAP/WML                  |   |   | ✖ |   |   | 2000         |
| ASP                      |   | ✖ |   |   |   | 2000         |
| PERL                     |   | ✖ |   |   |   | 1999         |
| Prolog                   |   | ✖ |   |   |   | 1995         |
| Occam2                   |   | ✖ |   |   |   | 1993         |
| Lisp                     | ✖ |   |   |   |   | 1993         |
| Sather                   | ✖ |   |   |   |   | 1995         |
| SQL/Embedded SQL         |   |   | ✖ | ✖ |   | 2005         |
| <b>Operating Systems</b> |   |   |   |   |   |              |
| SunOS 4.1.x              |   |   |   | ✖ |   | 1997         |
| Solaris 2.x              |   |   |   | ✖ |   | 2001         |
| SCO UNIX                 |   |   | ✖ | ✖ |   | 1999         |
| HPUX 9.x/10.x            |   |   | ✖ |   |   | 1997         |
| X Windows                |   |   |   | ✖ |   | 2006         |
| FreeBSD                  |   |   |   | ✖ |   | 2006         |
| Linux                    |   |   | ✖ |   |   | 2006         |
| MS DOS                   |   |   | ✖ |   |   | 1999         |
| MS Windows               |   |   | ✖ |   |   | 1997         |
| Windows 95/98            |   |   |   | ✖ |   | 2001         |
| Windows NT               |   |   | ✖ |   |   | 2001         |
| Windows XP               |   |   | ✖ |   |   | 2006         |
| OS/2 Warp                |   |   | ✖ |   |   | 1996         |
| MacOS 6.x                |   |   | ✖ |   |   | 1995         |
| MacOS 7.x                |   |   | ✖ |   |   | 1996         |
| <b>Databases</b>         |   |   |   |   |   |              |
| Oracle V.7               |   |   | ✖ |   |   | 1998         |
| Oracle V.8               |   |   | ✖ | ✖ |   | 2001         |
| Ingres                   |   | ✖ |   |   |   | 1997         |
| Mini SQL                 |   |   | ✖ | ✖ |   | 1999         |
| MySQL                    |   |   |   | ✖ |   | 2005         |
| MS-SQL                   |   |   |   | ✖ |   | 2000         |
| Sybase                   |   |   | ✖ |   |   | 2000         |
| Postgres                 |   |   |   | ✖ |   | 2005         |

### **Industrial/Commercial/Professional Activities**

|                              |   |
|------------------------------|---|
| Senior Analyst<br>Programmer | Day Eight Technology Pty Ltd<br>Mosman NSW Australia<br>Jan 2005 – Jan 2006   |
| Duties:                      | <p>To develop internet and web based applications using python, COM, ActiveX, Zope, Flash. Various widget sets were used for development, including extensive use of QT3 and wxWidgets.</p> <p>Many projects were developed during this time, some of the major ones are;</p> <p><b>MediaWise:</b><br/>A networked back-office server for processing jobs from multiple clients. Included multiple job queues for allowing processing by multiple CPUs. Utilised twisted framework.</p> <p><b>AccuReach:</b><br/>A windows application for the processing of ratings data to assist with television advertising placements. First revision used the QT widget toolkit. Other versions used wxWidgets toolkit.</p> <p><b>Updater:</b><br/>An application to automatically download and install updates to software packages installed. The updates were housed on a remote server and XMLRPC to a Zope server was used to determine what versions of the software were available to each individual client. Updates were downloaded and installed as required.</p> <p>During investigation of various widget sets, a scripting interface for python to Flash was written, to enable the scripting of MX or Flex applications. I also wrote python extensions in C, and wrapped some C++ libraries using boost.</p> <p>Technologies used; Python, C, C++, boost, py2exe, Zope, Flash, COM, ActiveX, twisted, QT, wxWidgets.</p> |
| Senior Analyst<br>Programmer | Vivitec Pty Ltd<br>Southbank Victoria Australia<br>Aug 2004 – Nov 2004  |
| Duties:                      | <p>To develop an Active Directory -&gt; Zope/Plone interface for use in an intranet redevelopment.</p> <p>An object-relational layer was developed that mapped Active Directory entities into Zope Objects along with hierarchical relationships. E.g. managers had direct reports as child objects.</p> <p>Various Plone services were developed, including a Phone Book application, and an Organisational Chart.</p> <p>Some ASP code was migrated to Python objects and ZPT using ZSQL methods to a MS-SQL instance.</p> <p>Technologies used; Python, Zope, Plone, ZPT, MS-SQL, ASP</p>  |

|  |   |
|--|---|
| Senior Analyst<br>Programmer, Architect. | Development Infostructure<br>Clarendon VA USA<br>Jan 2002 – July 2004   |
| Duties:                                  | <p>Development Infostructure (AKA devIS <a href="http://www.devis.com/">http://www.devis.com/</a>) is a top ten US Government contractor, providing large and small, primarily internet based software solutions to US Government departments. Some projects are utilised by departments located in every corner of the globe.</p> <p>They believe in open source software, and promote it within government departments and to high-level government meetings. They have been written up in various Open Source magazines including LinuxWorld.</p> <p>I worked remotely for devIS from Australia for this entire period, with annual trips to the US for technology retreats. I created my own development environment that mirrored their environment and used this to do development. These are by no means all the projects I worked on, but, are the significant ones.</p> <p>I can not go into too many specific details about projects, since these are mainly US Government projects, and subject to NDA, although as may be obvious from below a lot effort went into developing infrastructure, and then deploying applications using that infrastructure.</p> <p>Changing attitudes internal to the US Government resulted in them specifying J2EE platforms for further work, which meant the Zope work dried up.</p> <p><b>Development Environment Migration:</b><br/>The primary mode of development for devIS was ZODB development via web-browser, with some supporting development done as products. They had no source control system in place. My first major task was to push development into product based development so that a source control system could be adequately deployed.</p> <p>After a major project efforts were undertaken to standardise the environment particularly the layout of CVS.</p> <p><b>Credit Management System v1 (CMS):</b><br/>Environment: Zope 2.4, ZSQL, Postgresql<br/>.<br/>CMS is an application used by banks all over the world to report to the USAID department. It is used to report and manage loans to entities in developing countries in the form of aid.</p> <p>This project was initially ZODB based, my primary job was to migrate (in parallel to the ZODB development) this to a Python product, and for development to continue using this there after.</p> <p>There was no finesse in the migration, the on-disk source tree mirrored the in-ZODB tree to make transition for the developers easier. Primary templating was done in DTML, with lots of ZSQL methods.</p> <p><b>DBFactory:</b><br/>One of the obvious things that came out of the CMS migration was that nearly everything was repetitive. Create, Replace, Update</p> |

|  |   |
|--|---|
|  | <p>Delete (CRUD), methods were done for most objects in the system.</p> <p>During a brainstorming session, I mentioned I could simply turn database tables into Folders and rows into Zope objects.</p> <p>A small demo was created showing the concept at work.</p> <p>A refactoring effort then commenced as all the common code was removed into a base class that automatically generated the SQL for CRUD, and provided Zope overrides for objectValues, and objectIds. Subclassing this then created a Folder-like instance that Zope thought was a normal Folder.</p> <p>Code was then generated from ERWIN, (which seemed like a good idea at the time). Code consisted of 4 classes, 2 base classes that should never be hand-edited, and 2 that could (two for the Folder/Factory and two for the actual Object).</p> <p>Code was also created to do form validation so that incoming form data could be automatically checked against basic rules, including required status, range checking, length checking, type checking.</p> <p>DBFactory understands hierarchical relationships between tables, and allows composite factories to be generated (one Factory/Folder with different object types). It should be noted here that even existing objects come out of a Factory, so from that point of view it is probably misnamed (depending on your view, my view was that existing objects are really clones of database data and so are really created, not just provided).</p> <p>DBFactory also wraps new objects so that creating an object the ZPT are the same, the attributes are pushed into the current context, you never have to fetch things from the REQUEST object. It is totally transparent from the ZPT whether this is an SQL backed object or a ZODB object.</p> <p>This is known as DBFactory v1, and was the primary technology underlying ACMIS.</p> <p>During work on ACMIS (described below) it turned out that the 4-class system was very unwieldy and there was still a lot of duplicated code to deal with calls from the web into the class.</p> <p>During a weekend I created some layer classes that reduced the code duplication down to miniscule amounts, and reduced the classes down to two. The two missing classes were folded into the middle layer classes that automatically generated the methods present in the missing classes. Also the Zope SQL classes were stripped bare (performance reasons), and used internally to provide higher data throughput.</p> <p>DBFactory v2 is completely meta-data driven, and I created (as my own project), a Zope Product that generates the code for any given table including hierarchical relationships.</p> <p>DBFactory v2 was used for subsequent SQL-backed projects. This methodology was also used as the basis for driving the Java development that came later.</p> <p>DevIS intends to release DBFactory v2 as an Open Source tool.</p> |
|--|---|

|  |  |
|--|--|
|  | <p><b>Report Builder:</b><br/>ACMIS also had a large number of reports that had to be generated from the SQL data. I created a set of classes that reduced the generation of a report to meta-data declaration. This reduced the development time, at the expense of running speed.</p> <p>The Report Builder code was reused in CMS v2.</p> <p><b>Acquisition Career Management Information System (ACMIS):</b><br/>Environment: Zope 2.4, python, ZPT, Postgresql, C (postgresql functions). Later migrated to Zope 2.6.</p> <p>ACMIS is a US federal system of records to track acquisition workforce training and education. Managers use the information collected to monitor the availability of staff and warrants and specific skill sets and to respond to information requests of the Administration, Congress and other parties.</p> <p>This is a big system, it's about 34K python SLOC. It utilises DBFactory v1 so the code size could have been somewhat smaller. It tracks information for (currently) 160,000 US Government employees, with more departments signing on (they have a legal requirement to).<br/>Every employee is responsible for their own data, so there are a lot of concurrent users of this system at any one time.</p> <p>This project like most devIS ones heavily utilised Extensible User Folder for user management and session facilities.</p> <p>This used Zope 2.4, and this had some performance issues, details of what performance enhancements were done are detailed below (also there were stability issues w.r.t. Zope 2.4 and ZEO).</p> <p><b>VISA Compliance System:</b><br/>Environment: Zope 2.4 python, Postgresql, ZPT</p> <p>This system tracks all J-Visa applications (foreign students), for entry in the US in the wake of the September 11 terrorist action. It has a complicated workflow that restricts access to data depending on the user, and including interfacing to other government systems. It is used by USAID to enter details from all over the world of potential students.</p> <p>This system is to be extended to also track B-Visa (business travel) at some future point.</p> <p>This was the first project deployed using the DBFactory v2. It's a smaller project consisting of around 10K SLOC. Most of the work involved in this project was in the complicated workflow, and business rules.</p> <p><b>Credit Management System v2:</b><br/>Environment: Zope 2.4 python, Postgresql, ZPT</p> <p>CMS v2, was a rewrite of CMS v1 to include much more functionality. It was deployed utilising the DBFactory v2 code base. Even with the reduced code base, it's still 32K SLOC.</p> <p>This was a much cleaner implementation, not being restricted by the</p> |
|--|--|

|  |  |
|--|--|
|  | <p>initial v1 hierarchy.</p> <p>Environment:<br/>Zope 2.6, python, Postgresql.</p> <p><b>EGovOS:</b><br/>Environment: Zope 2.6 python, Postgresql, ZPT</p> <p>This is a support product for the Electronic Government initiative. DevIS has it's own in-house (now-released as Open Source) product called EZRO which is a ZODB-backed CMS product. EGovOS is a hybrid product that uses both EZRO for content management and DBFactory v2 code to manage what is called "reference book data". Reference book data is data collected from all participants about what Open Source products they use, promote, support or provide and is compiled into a book for distribution to participants at EGovOS conferences.</p> <p><b>Performance Tuning:</b><br/>During the deployment of ACMIS the sheer amount of data started to cause significant performance issues.</p> <p>I utilised different monitoring tools to highlight bottlenecks and remove them.</p> <p>PCGI is a huge performance hole. However, many projects relied on one particular feature of PCGI, which is the ability to pass through variables. To this end, I extended the apache module mod_proxy to create mod_proxy_env that allowed variables to be passed through (as HTTP variables) into Zope.</p> <p>One major bottleneck is the use of the Results class by the Zope SQL class. The creating of these classes is very expensive. The SQL class was refactored to only provide dictionaries and DBFactory rewritten to use the dictionaries() method. This provided a significant speedup in most cases.</p> <p>An elaborate caching scheme was implemented, ACMIS was fortunate in that most of a user's data was cascaded under one common (per user) object. A global cache was set up with this top-level object, and child objects set up to use volatile caching internally.</p> <p>ZPT rendering time was also proving to be fairly long (although this was exacerbated in some cases by a design flaw where result sets were not bounded, but, rather the entire result set was returned to be rendered, this was rectified in subsequent projects). I implemented a new ZPT class that turned each ZPT into python code and then cached and executed this python code. This was determined to be a little too risky to deploy, but, did result in 2-3x speed-up in ZPT rendering (out-performs DTML and raw TALES). Some of this code (HTML parser) did get reused in some web-scraping features of another project.</p> <p><b>Other Extensions:</b><br/>Web sites developed for the US government must be what is called Section 508 compliant, which is a set of standards for access for visually impaired users.</p> <p>In order to facilitate this compliance I developed a ZPT plug-in that</p> |
|--|--|

|  |  |
|--|--|
|  | <p>evaluated each page's compliance as it was viewed and included a report embedded in comments on the page and also into the Zope log (so you could crawl a site using a script and then search the log for non-compliance).</p> <p><b>Other Duties:</b><br/>I was also called on to do other things, including assisting with proposal writing, requirements analysis, quality assurance, reviewing résumés, attending client meetings (when in the US), mentoring, and of course the obligatory testing and debugging of other projects.</p> <p>Environment:<br/>FreeBSD, Linux, Postgresql, Zope 2.x python, C, SQL, DTML, ZPT, CVS.</p> |
|--|--|

|                                      |   |
|--------------------------------------|---|
| <b>Senior Analyst<br/>Programmer</b> | Confidential<br>Jul 2001 – Sep 2001   |
| Duties:                              | <p>A mix of work involving C and C++ (primarily C++) on UNIX platforms.</p> <p>Non-disclosure agreements prevent me from discussing any project specifics for this company, or the company name. The projects were mixed, and focussed primarily on the Internet domain. Core technologies included programming of large applications using CGI, internet services such as SMTP, IMAP and POP servers in C++, as well as smaller utilities and Apache modules in C.</p> <p>Work involved new development as well as maintenance of existing projects.</p>   |
| <b>Senior Architect</b>              | Confidential<br>Dec 2000 – Jul 2001   |
| Duties:                              | <p>To design, implement, and test a web based internet billing system for ISPs, DSL Providers and Peer Network Routing companies.</p> <p>Non-disclosure agreements prevent me from discussing in detail the work or the company for which the work was done. This was quite a large project, and utilised some leading edge application server development.</p> <p>The environment was a mix of UNIX variants, mostly Open Source, and Open Source tools. Key languages were C and Python.</p> <p>Other technologies include Netflow, a network usage protocol primarily used in Cisco routers.</p> |

|                |  |
|----------------|--|
| Project Leader | Softworks Pty Ltd<br>Lvl 1 31 Park Road<br>Milton QLD<br>Sep 2000 – Dec 2000   |
| Duties:        | <p>Softworks required someone to oversee their product migration from an NT ASP+COM infrastructure to a Solaris Java+EJB infrastructure.</p> <p>Softworks were new to web development and this was their first major web project.</p> <p>Having not received the required hardware, and with an Application Server being chosen, but, not delivered, I attempted to implement a Java and EJB infrastructure to meet their missing requirements, using the available tools.</p> <p>There was a large amount of instability in the team with the Project Manager changing multiple times. It didn't seem like this project would ever get off the ground, so I decided not to renew my contract.</p> <p>Environment:<br/>FreeBSD, NT, ASP, COM, Solaris (eventually), Java.<br/>Application Servers:<br/>BEA Web Logic, iPlanet, Enhydra, Tomcat, Websphere.</p> |

|                  |   |
|------------------|---|
| System Architect | <p>International Strategies<br/> Lvl 6 269 Wickham Street<br/> Fortitude Valley QLD<br/> Jul 1999 – Jul 2000</p>  |
| Duties:          | <p>International Strategies was a pre-IPO start-up company that create business directories with localised content. Their primary site is <a href="http://www.ezyfind.com/">http://www.ezyfind.com/</a></p> <p>To investigate, implement, and evaluate internet technology to give International Strategies a strategic advantage in the marketplace. To design systems that enhanced the reliability, and performance of the ezyfind.com system.</p> <p>During the time I was at International Strategies I investigated a number of technologies to help them migrate their systems from Windows NT to a UNIX environment. This included new network designs and application redevelopment.</p> <p>I developed numerous prototype and proof of concept systems, including a complete Yahoo! System (that went live) using the ODP project data. Most of these projects were awaiting deployment when funding ran out.</p> <p>I was responsible for new product development for UNIX projects, including a billing system interface to the 1Disk web-disk project, and providing a WAP/WML interface to the existing ezyfind.com data.</p> <p>I developed a web-based bug-tracking system for their development needs, and created an Intranet Site for the Technical Team. The bug-tracking system was integrated with the Intranet Site.</p> <p>I also provided general solutions to internet problems that were hard to implement in NT environment, including automatically fetching files, processing and database insertion of records.</p> <p>Before the funding ran out I was working on two large projects with overseas companies. I was essentially the project manager for these projects, and I was responsible for the liasing with the client on technical matters.</p> |

|                          |   |
|--------------------------|---|
| Senior Software Engineer | Brisbane City Council - Travel Demand Management<br>Lvl 17 Brisbane Administration Centre<br>Brisbane QLD<br>Nov 1998- Jul 1999   |
| Duties:                  | <p>To design and implement a new system for the Brisbane City Council “Brisbane Linked Intersection Signal System” (BLISS).</p> <p>The project was required to run on legacy hardware and networking infrastructure, so the terms of the redesign were somewhat limiting.</p> <p>The project was designed to run on SCO Servers, although the new system had an eye towards portability in order to facilitate the migration to a new platform if that was required.</p> <p>I also investigated a new “Local Control Module” (LCM) to run under an embedded UNIX environment. This would enable TCP/IP networking to each intersection and simplify some of the replication and management issues involved with the current system.</p> <p>The project will be deployed on SCO Servers, although SCO has ceased support for OpenServer, and so it is likely that a platform migration might be necessary.</p> <p>I reimplemented their monolithic support library into a more modular form to enhance the maintainability of the project. I also redesigned their build environment to facilitate better source control and to lower the learning curve for new developers on their projects.</p> <p>Due to the impending merger of Main Roads and Brisbane City Council the decision was taken to enhance the Main Roads system rather than to continue to enhance BLISS.</p> <p>Recently work has commenced on redeveloping their Bus Priority system RAPID. This includes a full re-design including User Requirements analysis.</p> |

|                              |   |
|------------------------------|---|
| Software Development Manager | Advercom Pty<br>80 Stamford Road Indooroopilly<br>QLD 4068<br>April 1998 - October 1998   |
| Duties:                      | <p>To manage and assist in the development of Internet related marketing and gaming products. To maintain and protect the LAN network, and to administer machines on that network. To set-up a web server and prototype company web pages.</p> <p><b>Multi-media Push Advertising System.</b><br/>Environment: C++ Windows, C Solaris, Oracle 7.3</p> <p>I designed and implemented the UNIX server side of this project, and managed a team of developers writing the client side of the project. This was to be an internet foil to a Fly-Buys loyalty type program.</p> <p>During this project I also redesigned their internal network, installed a firewall to protect their software, and built and installed a Sun Sparc workstation. I also had to install Oracle V7.3 on the Sun.</p> <p><b>Internet Trade Promotion Game.</b><br/>Environment: Java,C,Mini SQL 2.0, Solaris, FreeBSD, HTML, CGI.</p> <p>This was a 'matching faces' game written in Java, that has a product quiz component. It stores information about every player and generates useful marketing information as well as statistics for viewing by customers purchasing the service.</p> <p>It networks to a UNIX server that administers the actual games played as well as the questions asked. The system is set-up to be as automatic as possible, with the Java side not needing to rebuilt for each customer, and the Server not having to be reconfigured when new customer are added.</p> <p>The system can be fully distributed with each component housed on a separate computer.</p> <p>I designed the system, and prototypes the Java game, implemented the UNIX side servers, and implemented the registration and authentication process.</p> <p>I interviewed candidates for the final Java programmer and to assist in the server side development.</p> <p><b>Business Exchange</b><br/>Environment: C, C++, Mini SQL 2.0, HTML, CGI</p> <p>This is an online system for exchanging quotes between vendors and the public. This is a series of interlocking CGIs that present a fairly sophisticated interface to the quotation process. Users are authenticated when they enter the site, and depending on whether they have registered as a vendor, a wholesaler or as a member of the public they are presented with different options.</p> |

|                          |   |
|--------------------------|---|
| Senior Software Engineer | University of Western Sydney<br>College St Richmond<br>NSW 2753<br><br>November 1997 - January 1998   |
| Duties:                  | <p>To design and implement a system to allow the billing of staff, students, and departments for their Internet usage.</p> <p>The system was required to be portable across multiple platforms, to be quick, and to be easily extendable.</p> <p>A number of existing log files were used as the basis for the system. Squid web proxy logs, sendmail entries from syslogs, and router logs. These were used in conjunction with login records from Novell, and DHCP records to identify the owner of a transaction.</p> <p>Each log file is parsed and processed separately, and there is an intermediate log file type, to allow other filters to be easily added. Each of these individual log files were to be combined and processed into a final transaction database that forms the basis for any reporting.</p> <p>Due to the nature of the billing from the bandwidth provider, each transaction has to be categorised into one of six different types of traffic based on the source (local, RNO, AARNET, domestic, international, and unknown). Each one of the six types is also billed at a different rate depending on what type of traffic it is (web proxy traffic is charged at a lower rate since it has already been fetched).</p> <p>All of this is controlled from configuration files that can be modified before processing or report running.</p> <p>The target platform was Solaris 2.x, but, all of the development was carried out under FreeBSD. All code compiled and ran correctly without modification, fulfilling the portability requirement of the project.</p> |

|                          |   |
|--------------------------|---|
| Senior Software Engineer | <p>NEC New Product Development<br/> 244 Beecroft Road<br/> Epping NSW<br/> 2121<br/> (02) 9930 2230<br/> September 1996 - October 1997.</p>   |
| Duties:                  | <p>Initial duties were to port Superview, a network management product, from HPUX to SCO UNIX, fixing discovered bugs as it went along.</p> <p>Having completed this, the agent subsystem was assigned to me to be enhanced in a number of ways. When I inherited this code, it was ten years old and suffering from an extreme amount of bitrot, and of course came with no documentation.</p> <p>Features added including pooling modems for use with multiple PABX as well as proper session management There were also nearly 100 bug fixes related to this communications that were fixed during the enhancement effort by me.</p> <p>During this time a significant flaw in the functionality of the NMS was discovered. To cope with this flaw a method of using the existing NEC PABX MAT systems was needed.</p> <p>I suggested using a freeware DOS emulator and porting it to HPUX.</p> <p>This assignment was of course delegated to me. Most of the source available PC Emulators had restrictive licensing agreements attached with them, and so the only one that truly free lacked a lot of basic elements needed. To this emulator I added, hard drive support, serial communications support, UART emulation, redirected serial communications to UNIX pipes. This was then extended to communicate with remote agents to talk to actual PABXs. This single addition saved the company the cost of a commercial DOS emulator, or the cost of placing a PC with DOS on it at every site, and allowed the product to actually be released. This project was completed under adverse conditions, and on an extremely unrealistic deadline.</p> <p>My final project was 'internationalising' the existing system for use in other languages. With a Windows NT port also happening, this code had to co-exist under all platforms without major changes. When my contract was finished the groundwork for this had been laid, using UNICODE as a base (it's native to Windows NT). I researched all of the relevant standards, obtained standards, and designed and implemented a scheme to allow this to occur.</p> <p>I was also responsible for hardware maintenance of the SCO PCs as well as responsible for installations of NT on new platforms.</p> <p>I set-up an internal web server for the distribution of documentation, as well as configuring the HP workstation to share disk space to the Windows boxen.</p> |

|                |   |
|----------------|---|
| Project Leader | <p>Computer Telephone Integration Pty Ltd.<br/> 5/7 Park Road,<br/> Sans Souci NSW,<br/> 2219<br/> (02) 9529 2156<br/> Jan. 1996 - September 1996</p>   |
| Duties:        | <p>To lead a team of eight software engineers through the full development lifecycle of IVR and CTI projects. To meet with customers and liaise with 3rd parties to report on project status and to suggest possible solutions to sales representatives.<br/> Customers for CTI included, Macquarie Bank, BHP, Microsoft, Datacraft, MMI, ABL, NAB, Qantas, Servcorp.</p> <p>Although much of my time was spent in managerial duties, a fair portion of my time was also spent in developing software for these projects, and investigating new technologies to better implement projects. I was also involved in evaluating tenders for a (voice) VPN for MMI, which involved meeting with all vendors and evaluating the technology.</p> <p>During this time, I developed a drop in replacement for the NEC AAC system that would run on inexpensive PCs, as well as a generic IVR parser that would enable IVR applications to be developed once and then easily migrated to any platform.</p> <p>I collaborated with a team from the then ISSC to create a secure internet brokerage product for MLC Brokerage department called IBIS.<br/> Programming was on Sparc/Axil SCO UNIX, in C, some Visual Basic, some Visual C++ under windows 3.1.</p> |

## Zope Projects

I have been working with Zope for over 5 years, and I have created several major projects with Zope for different clients. To date I have also created and released seven Zope projects to the Zope Community. Amongst these are;

- A simple search engine and indexer
- A system for enabling page translations without requiring multiple source trees
- DTML Tags for content retrieval and display including: polling, RSS channels and METAR weather reports.
- A real time IRC client for Zope that features two-way communication between Zope and the client.
- ZButtons for dynamic graphical buttons in Zope.
- An extendable Authentication System that includes property management, each system is “mix ‘n’ match” so you have the choice of Authentication models as well as property repositories.

More information on these projects can be found at  
<http://www.zope.org/Members/TheJester/>

I have built for a content filtering client an enterprise IP billing system for billing and managing their bandwidth clients.

I have created a larger billing system for another client, that includes real time netflow capture from Cisco routers, asset management, customer management, bandwidth management, in an extensible framework that makes it simple to add further billing modules (RADIUS e.g.) without requiring changes to the front end. This system also generated PDF invoices using ReportLab to email to customers.



## TECHNICAL PROFICIENCY PROFILE (™)

### ANSI C PROGRAMMING - Secure

**Name:** Andrew Milton  
**ID Number:** 511110701  
**Date:** 1110712000

| <u>CONSOLIDATED RESULTS</u> | <u>SCORE</u> | <u>PERCENTILE</u> |
|-----------------------------|--------------|-------------------|
| Overall Rating              | 35           | 100               |
| Necessary Knowledge         | 36           | 99                |
| Understanding of Subject    | 32           | 99                |
| Practical Experience        | 32           | 99                |
| Work Speed                  | 37           | 100               |

### RESULTS EXCLUDING LOWEST 25% OF CATEGORY SCORES

|                          |    |    |
|--------------------------|----|----|
| Understanding of Subject | 37 | 99 |
| Practical Experience     | 36 | 99 |

### CATEGORIES INDICATING STRENGTH

Data Types  
 Arithmetic Operators  
 Logical and Relational Operators  
 Operator Precedence  
 Lexical Scope of Identifiers  
 Control Flow Instructions  
 The C Preprocessor  
 Standard Library Functions

### CATEGORIES INDICATING WEAKNESS

none

### OTHER CATEGORIES EVALUATED

C Language Syntax  
 Data Definitions - Constants and Initialisation  
 Data Declarations - Properties of Variables  
 Function Usage  
 Pointers and Arrays  
 Data Structures and Unions