

**Daniel Grigg****Mobile: +61 425 781 597****E-mail: [mail@danielgrigg.com](mailto:mail@danielgrigg.com)****LinkedIn: [Daniel Grigg](#)****Github: [Daniel Grigg](#)****405 Seventeen Mile Rocks, Seventeen Mile Rocks, QLD 4073****HIGHLIGHTS**

Extremely capable and enthusiastic software engineer with an extensive background delivering advanced software solutions. Particular expertise developing rigorous embedded and autonomous systems. Harnesses best practice engineering processes and methodologies throughout the full software development lifecycle in both agile and formal-driven environments.

I have acquired considerable experience across the software development spectrum, in both permanent and contracting roles and team and independent environments. I am highly proficient in traditional object-oriented analysis and design methods as well as modern, functionally oriented techniques. I have acquired a rich vertical knowledge in multiple domains, particularly enterprise mining systems, aerospace and fleet management. My flexibility, motivation and relentless thirst for knowledge enables me to confidently approach new tasks and domains with success.

I continuously seek to excel the client's expectations, improve my abilities as a professional engineer and build stronger relationships with stakeholders and peers. I thrive operating in stimulating and communicative environments where my colleagues and I can grow, mentor each other and develop innovative, world-class products together.

I simply adore embracing novel technologies and methodologies and have a genuine passion for creating and sharing software with the world professionally and personally.

**Proficiencies**

Professional	Experience (years)	Personal
C++/STL/Boost	5	Clojure, ClojureScript
Linux	4	Common Lisp
Embedded Systems	5	Haskell
Object-oriented design and analysis	5	Java
Software Testing (Unit, integration, system)	4	OpenCL
Scripting (Bash, Ruby, Python, Lua, DSL)	4	Python
Parallel computing	3	Web applications
Network programming	2	iPhone/iPad
Databases (PostgreSQL, MySQL, SQLite, SQL Server)	4	3d graphics
Editors/IDE (Emacs, VIM, VS, Xcode, Qt Creator)	5	Robotics
Configuration Management (GIT, SVN, Perforce)	5	Interaction Design
Change Management (JIRA, Rational Change)	5	
Requirements analysis and specification	3	
Continuous Integration (TeamCity, CruiseControl)	3	
Build automation (CMake, autotools)	3	
Knowledge Management (Confluence, MediaWiki)	2	
Qt	1	
Windows	4	
C#	3	
Mac OS X	2	
Objective-C	2	
UML	3	
Graphics (OpenGL, Direct X)	2	

## Employment History

**April 2010 – November 2011**      **Leica Geosystems**      **Permanent**  
[mining.leica-geosystems.com](http://mining.leica-geosystems.com)

C++, STL, Boost, Linux, Apple, Objective-C, Ruby, JIRA, SVN, GIT, Enterprise, Embedded, SQL, Confluence

Full-time software engineering role within Leica Geosystems Mining - onboard division.

### Responsibilities

I was recruited into a team of 12 engineers, product owners and QA, collaborating with Leica Tucson on the Jigsaw and High-Precision (HP) product range. The Jigsaw Fleet Management System is a complete, tier-1 enterprise suite of on/off-board systems for managing, optimizing and reporting of equipment operations in open and underground mines. The HP product augmented Jigsaw with vital, high-precision functions on shovel, drill and dozer equipment.

I was first tasked with enhancing the embedded HP software. I relished this opportunity to acquire new domain knowledge and utilize my expertise in mission-critical software. Designed in UML it operated as a standalone, embedded-Linux, C++ app with Cairo frontend. My 2010 HP work focused on adding and improving various sensor interfaces, including the Locata Positioning System and improving core infrastructure. The project used STL, Boost and ACE extensively. CppUnit and Google Test were used for unit/integration testing. Python, shell scripts and TeamCity were used for tooling and continuous integration.

I worked on Jigsaw for the remainder of 2010 - mid 2011. The tasks were diverse, covering feature enhancements, new hardware interfaces and continuous stability and performance improvements. The Jigsaw platform was challenging in its architecture. A distributed platform written in Obj-C, C and Ruby, it reused the same, configurable software on-board (embedded Linux) and off-board (Ubuntu, Mac OS X and Windows). Synchronization was managed with a proprietary replication service, exposed as a proprietary ActiveRecord abstraction over PostgreSQL, SQLite and SQL Server. I was also involved in preparing a new on-board platform and porting Jigsaw and its dependencies to it during that period.

All software was developed using SCRUM with fortnightly sprints during 2010. Tasks were drawn from a running backlog, orchestrated using JIRA and MediaWiki with regular code and design reviews. Development process varied according to task complexity, but due to short sprint cycles mostly forwent big upfront designs. Instead, pair-programming was used as a hybrid design, review and knowledge transfer technique. SVN was employed for revision control, whilst MediaWiki was used for both internal and external documentation with project engineers, technicians and client stakeholders.

Many changes occurred in the business space of 2011, including the rebranding of Jigsaw and Leica HP into the cohesive 'J' range. The on/off-board teams were also merged into a single unit. Nonetheless, I was afforded additional responsibilities, including refactoring and integrating the complete HP product into an algorithms library within Jigsaw. I also managed the redevelopment of a legacy .NET project, PitOps to upgrade its deprecated Managed DirectX backend to the XNA framework – leveraging my .NET and graphics experience. Finally, I eagerly undertook numerous side projects, including a Jigsaw iPhone client, 3d-animated demos and expo marketing projects.

My swan song at Leica was working in a team of five on a new, cutting-edge product, the J<sup>2</sup> Dragline, a high-precision guidance system for dragline operations. The product replaced the ageing, Series 3 Dragline offered by Leica and integrated seamlessly into the 'J' hegemony. Built from scratch to illustrate best-practice, it was built as a parallelised, reliable, message-driven system with soft-real-time constraints. The cross-platform software was written in C++ and made voracious use of the Boost libraries and modern template-driven design methods. Unit and integration testing was conducted via Google Test. Git was adopted for revision control. Finally, development was managed and documented in JIRA and Confluence respectively.

### Major Achievements

Compared to earlier roles, I believe my greatest accomplishment was flourishing in the extremely challenging and diverse technical and business landscape of a tier-1 enterprise solution provider. Secondly, I gained a wealth of domain knowledge in mining systems and operations.

**April 2008 –  
December 2010**

**BAE Systems Australia**  
[www.baesystems.com/australia](http://www.baesystems.com/australia)

**Permanent**

C++, UML, Linux, Perl, Synergy, CVS, DOORS, Boost, XML, Real-Time, Object Oriented, Safety-critical

Full-time software engineering role in the Autonomous Mission Systems (AMS) division.

### **Responsibilities**

I was first recruited into a software engineering role on phase-two of the HERTI OC-1B project. A team of 30+ engineers was assembled to develop the flight and ground control systems for a world-first – a UAV certified for civilian airspace. I was initially tasked with requirements analysis of various sub-system interfaces. However, my competency in capturing and formally defining requirements using Rational DOORS and Rhapsody for UML modeling led to being requisitioned for client-side requirements elicitation. All HERTI work was performed in accordance to DO-178B certification requirements using MIL-STD-498.

I was later afforded a sought-after position in the emerging Mantis UAV program, a larger, combat oriented aircraft. Competing with next-generation rivals, the success of the project was crucial for establishing BAE Australia's presence in the global UAV market. A dual BAE UK and Australia effort, I was in a team of 10 engineers tasked with developing phase-one of the Flight Control Computer (FCC). Continuation to phase-two was dependent on our team completing all materials within eight months then proving capability through live flight trials.

My primary responsibility was engineering real-time, embedded, object-oriented software solutions for the support and test equipment, most notably the System Test Unit (STU). The STU was a Linux-based, aircraft and environmental simulator. The unit interconnected with the FCC and ground control, allowing all capabilities to be tested within a lab environment. I designed and implemented on schedule, a per-vehicle reusable, multi-threaded, plugin-based system using an XML data layer in collaboration with other software, algorithm and system engineers. The design addressed risks in changing requirements and limited resources. The software was implemented in C++, with Perl and Bash for tooling and written in Eclipse and VIM using an agile, test-driven development approach via CppUnit. For configuration management, CVS then later Rational Synergy were used.

Following Mantis phase-one trials I was reassigned back to the escalating HERTI project and made responsible for expanding STU to support all HERTI systems. In addition I pioneered rigorous solutions for automating integration, deployment, documentation and verification activities.

### **Major Achievements**

I'm very pleased with all I accomplished working at BAE and the enriching experience I gained engineering safety critical systems and prospering in a rigorous, formally driven sector. However, I'm particularly thrilled knowing the solutions I presented encouraged a continuing culture shift towards more modern, extensible and automated systems and processes.

**August 2007 – March  
2008**

**Industrea (AMT)**  
[www.industrea.com.au](http://www.industrea.com.au)

**Contract**

C#, VS 2005, .NET, UML, SVN, Gemini, Enterprise Architect, Infragistics .NET Advantage, WiX

Contracted to develop on/off-board software and tools for a proprietary vehicle collision avoidance system.

### **Responsibilities**

I was the preliminary developer responsible for the on and off-board software of Industrea's next-generation of vehicle collision avoidance system (CAS). The system defined, detected and alerted on hazardous collision scenarios between suitably equipped vehicles. Additionally, I constructed visual configuration, playback and simulation tools for programming and validating CAS equipment and personnel training.

I was responsible for all aspects of the SDLC until phase one was completed on schedule and handed over for phase two. I adopted an agile development methodology with 3-4 week release cycles. Project management was primarily carried out using Microsoft Project and Countersoft's Gemini Project Issue Tracking software. Finally I liaised regularly with the software development manager, the CTO and BHP stakeholders.

Requirements' gathering was elicited through interviews and workshops in conjunction with Enterprise Architect's (EA) requirements gathering tools. Specification, analysis and design of the business, data and client tiers were communicated through plain-text documents accompanied by UML 2.0 and database diagrams built with EA and MS Visio 2007. The software itself was written as a desktop .NET 2.0 C# application using Visual Studio 2005 and Managed Direct X with NUnit for unit testing. The backend was written using CodePlex's Repository Factory against an SQL Server database. Installation and setup was managed with the WiX and Votive toolset.

### Major Achievements

The interactive tools I developed were declared an immense improvement over the existing text-based tools. Users were able to specify collision avoidance scenarios in minutes, immediately visualize feedback and deploy units without expensive and time-consuming iterations of field installations and tests.

**August 2007 –  
January 2008**

**Industrea (GPS Online Pty Ltd)**  
[www.industrea.com.au](http://www.industrea.com.au)

**Contract**

C++, C#, Cygwin, Eclipse 3.2, Visual Studio 2005, UML 2.0, Wavecom Wireless CPU (M1306B, Q2686)

### Responsibilities

Contracted in addition to my current role to develop a new GPRS communications infrastructure for asset tracking systems. I designed and constructed the firmware and developed all support tools and documentation.

### Major Achievements

I'm extremely proud of the success of the Wavecom project – on completion, Industrea was employing GPRS as its primary communications interface on a rollout of thousands of its GPS tracking units. Being my first commercial embedded system project I learnt a great amount about writing robust firmware and comms systems.

**June 2006 –  
January 2008**

**Industrea (GPS Online Pty Ltd)**  
[www.industrea.com.au](http://www.industrea.com.au)

**Contract**

C#, VS 2005, .NET 2.0, SVN, NUnit, Infragistics .NET Advantage, SQL Server 2005, WiX, OOP

### Responsibilities

Designer and developer for a 3-tier desktop-based GPS asset tracking and driver monitoring system geared towards the mining industry. Began work as part of a paid 60 day university internship before being contracted as a full developer. Software was designed in UML and written in C# using VS 2005 against an SQL Server database.

### Major Achievements

I'm most proud of my work on the journey replay tools and the knowledge I gained on geodesy and mapping systems.

### Education

**July 2007      Bachelor of Engineering, Software (Hons)**  
**University of Queensland**

**Personal Interests**

Motorcycle riding, social coding, tech meetups, startups, online courses, yoga, running, cycling, swimming, hiking, reading, food and travel.

**References**

References are available upon request.