THOMAS M. OINN – CURRICULUM VITAE

3 Elm Court
Elmdon
Essex
CB11 4NP
Tel : +44 (0) 20 8123 5142
Tel (home) : +44 (0) 1763 836 220
Email : tomoinn@crypticsquid.com
Web: http://www.crypticsquid.com

SUMMARY

I am a freelance software architect specializing in the design and implementation of complex distributed systems, in particular workflow and data integration cases. I have extensive experience with a variety of languages and technologies, in particular the Java language, the Spring enterprise framework and service oriented architecture based middleware.

EDUCATION

- **BA (Hons) Computer Science**
  1996-1999 Queens' College, Cambridge University
- **Physics**, **Maths**, **Further Maths** and **Chemistry** A-Levels, fourteen GCSEs at A grade or above.
  Perse School, Cambridge

EMPLOYMENT

- **2008 - Present** : Self employed freelance consultant, various contracts.
- **2002 - 2008** : European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Industry / eScience group. Initially RA on myGrid project, bio service production moving to instigation, design and implementation of Taverna workflow workbench. Lead architect for myGrid project family.
- **1999 - 2002** : EMBL-EBI, Swiss-Prot group, curation tool developer. Designed and implemented generic framework 'Talisman' for web based curation tools for internal use within the group.
- **1999 - 2003** : Undergraduate supervisor for Cambridge University computer laboratory, small group tutorials on advanced Java, graphics, algorithm design courses. Students from Queens', Darwin and New Hall colleges.

TECHNICAL SKILLS AND EXPERIENCE

I have extensive experience in the design, from scratch, of complex software. I've been involved in the design from high level architecture down to UI design and refinement through testing and interaction with testers and real users as well as implementation of both front and back end code.

I'm most fluent in Java, but have experience with other languages including PERL and various web 'languages' including PHP, CSS, HTML etc. Although not languages as such I also have extensive experience with the various XML related standards including XML Schema. I enjoy learning new technologies, and am currently teaching myself Scala.

I have experience in design and implementation of all levels in a typical entire enterprise software stack; on the front end side I have extensive experience with Swing development as well as web based interface design with regular servlets and JSP as well as GWT. The majority of my work, however, has been server-side including substantial experience with service oriented architectures and associated middleware.

I have worked in teams of varying sizes, including one distributed across eight sites and have also worked solo (although I prefer having someone to talk to!). I've used both classical design methods including UML and extreme programming including rapid prototyping, SCRUM methods, test driven development and pair programming.

FRAMEWORKS

- Java Servlets - I introduced Servlet technology (and the Tomcat application server in particular) into the EBI in 2000, and have used it extensively in projects since.
- Spring - Since 2007 I have used the Spring framework for almost all my projects, including writing custom extensions to the Inversion of Control (IoC) container to allow for on the fly download of code when initializing bean components.
    - Aspect oriented coding - aspects are a powerful element of Spring's IoC container, I have used them to create transparent caches and impose easily verifiable security constraints on server side code.
    - Spring-WS - creating and consuming web services with Spring, through contract-first service definition and integration of Spring into Tomcat.
- Hibernate - Used to provide back end data services and accessed through Spring, including dynamic schema generation through detection of data objects within a set of plugins. This involved patching and building a customised version of Hibernate.
- Google Web Toolkit - Used to generate rich clients within the web browser, including integration of GWT-RPC services into Spring to provide access control and connection to data access layers.
- Web Service Frameworks - Various web service frameworks including Axis 1 & 2, CXF and Spring-WS, experience of both contract first and code first design methods. Experience included inter-operation with arbitrary third party services through dynamic clients, this required a much more in-depth knowledge of the web service technology stack than the more typical case where both service and client are written within the same project, or where clients are compiled specifically against a particular service.
- Java core, all aspects of standard and enterprise Java up to Java 6, including:
    - Swing UI design and implementation, including dynamically generated UIs driven from introspection over model object
    - Reflection based and dynamic code, using Java's ability to meta-program to manage plug-in frameworks (classloader isolation), dynamically generate message definitions and UI elements through introspection etc.

APIS

I am familiar with the full API of both J2SE and J2EE, up to and including Java 6. In particular I have worked extensively with the reflection and introspection APIs, including creation of custom class-loaders and use of reflection to dynamically assemble code on the fly. I've also used the Swing UI framework to build highly complex MVC driven user interfaces, often dynamically generated from domain specific languages or introspection over Java objects.

In addition to the standard Java APIs I've used a large number of third party libraries - as with most Java programmers one of the first things I'll do when implementing a piece of software is search to find whether anyone else has already done so. Other than the APIs used by the frameworks listed above I don't have a comprehensive list of APIs I've worked with for the simple reason that adopting a new API is generally not very much work, so saying that I am familiar with, say, JDOM, doesn't really tell you anything. I would assume that any decent programmer would be able to make use of any properly documented third party API.

I have produced a number of APIs intended for third party use as a deliverable of one of my more recent projects, the Taverna Platform. This involved packaging and documenting the code in the form of comprehensive Javadoc, a substantial developer's manual and a set of Maven archetypes and template projects.

TOOLS

Build tools and environments I've used in past projects. This is not an exclusive list, but representative of the tool set I would use on a new project. (My early development work was mostly done in Emacs and using command line tools to build and verify, these days I prefer to live within Eclipse).

- Apache Ant & Maven - My earlier projects used the Ant build system, moving to Maven 2 six years ago. Experience with Maven in particular includes writing custom plugins and archetypes as part of a framework project, allowing users of that project to generate template consumer code with a couple of simple Maven commands.
- Eclipse IDE - I've used Eclipse as my main development environment since 2003, inluding plugins for Spring, GWT, Maven and many others.
- Version control - CVS and Subversion experience, both used on projects with around twenty developers with commit access.
- Testing frameworks - Primarily testing with JUnit, but some experience with continuous integration tools such as Hudson and CruiseControl (I've used but not configured the latter). Experience with using code coverage tools to ensure solid test coverage and general experience with test driven development methods. One of my main motivations when selecting the Spring framework for past projects was the ease of testing, although Spring is obviously not a testing tool in itself.
- Others - Experience with the usual selection of project management tools such as JIRA and other issue trackers, code patching etc.

NON-TECHNICAL SKILLS

TEAM MANAGEMENT

In addition to technical skills I have a broad range of experience with team management and communication. In various projects throughout my career to date I have ended up taking leadership of software teams, largely because of an ability to create enthusiasm for my ideas with other developers. I take great satisfaction in creating a vision for a project and using my belief in that vision to inspire others.

COMMUNICATION SKILLS

Communication was a key part of my career as an academic software architect. In a formal context this took the form of scientific publications, presentations and talks at conferences and other events, and a role representing the projects and groups I worked with at major international meetings and conferences. Less formally, working primarily with infrastructure projects the communication between our project and our potential user communities was critical and something I was heavily involved with, in particular through the creation and use of training material for those communities. I have run a number of workshops, from single hour tutorials at conferences to multi-day residential events attended by professionals from across the world.

I am a confident speaker - I will happily stand up in front of hundreds of people and present my work and ideas, and have experience presenting to a range of audiences from internal management, through general conference audiences to more exotic groups including in one case much of the Malaysian cabinet (during a DTI sponsored visit to the country to represent the UK eScience programme).

I also have experience producing written material including academic papers of various kinds, press articles and longer pieces of work such as user manuals and in-depth documentation.

In support of these skills I have substantial experience with the software used to create presentations and documents, including various versions of Microsoft Office as well as DocBook related tools and Adobe Acrobat. I have been involved with the publication of substantial books, both through regular publishers and through self-publication for use as training material.

VISUAL DESIGN

While I would not claim to be a professional graphic artist I have worked with branding and website design and have a good sense for what works visually. I find this essential when designing front-end systems, whether rich client or web based. I have experience with the Adobe creative suite tools (Photoshop, Illustrator, InDesign) and in particular their use in web design (this site being an example).

I am also a competent web designer, familiar with the various technologies underpinning modern websites both static and dynamic in nature.

Neither graphic nor web design has ever been an official part of my job description, but one of the features of working with smaller teams is that there are opportunities to take on such roles where necessary. Personally I find the creative element of these activities very satisfying, although they have always been peripheral to my main body of work.

OPEN SOURCE INVOLVEMENT

When working at the EBI all the projects I was involved with were released under open source licenses, and all used a large number of third party open source APIs. This gave me experience interacting with the user and developer communities both for our own project and the APIs we used, including contributing patches and changes back and the use of public (and archived!) communication channels to resolve disputes, solve problems and determine future project directions.

Obviously not all the work I've done since leaving has been released under an open source license, but an ability to work with external open source projects, in particular to fill gaps in the functionality of those projects, has been extremely useful.

SELECTED PUBLICATIONS

Please see http://www.crypticsquid.com/?p0=C.V.&p1=Publications for a list of my academic publications and book chapters.