Curriculum Vitae

Name:	Tim Colgate
Address:	48 Ellesmere Drive, South Croydon, Surrey, CR2 9EJ.
Telephone:	07906 398236
Email:	CV@mettalogic.co.uk
Website:	http://www.mettalogic.co.uk/tim

Education

'O' levels	7 Grade A, 2 Grade B, 1 Grade C
'A' levels	Maths A, Chemistry A, Physics A
'S' level	Chemistry 1

BSc(Hons) MEng in Microelectronic Systems Engineering at UMIST

Career overview

I returned to Credit Suisse as a **technical architect** for 3 years, working in Credit Risk. I worked primarily on **multi-threaded, C++** application servers running on **Solaris**, with **Sybase** databases, as well as technology evaluation, specifying hardware, setting up servers, SOX and configuring DR.

I worked at Royal Bank of Canada as a **contract analyst programmer** on a Market Data project, building prototypes and evaluating technology, using **C++, Python, Windows, Linux, PHP, Apache, PHP** and **MySql**.

I returned to Credit Suisse as a **contract analyst programmer** for 15 months, working in Market Risk. I worked on a VAR system; redeveloping a **multi-threaded, C++** application server running on **Solaris** and using **Orbix** and **Oracle**. I then **migrated** the system to **Linux**.

I returned to Royal Bank of Scotland as a **contract analyst programmer** for a 5 month project in the Group Risk area. I specified, designed, developed, tested and released an enhancement to a **multi-threaded, C++** application server running on **Solaris**.

I worked for 13 months as a **contract analyst programmer** in the **front-office**, **fixed-income Repo** area at Goldman Sachs. I developed, maintained and supported a number of systems written in **C++**, **Java** and **C** on **Solaris** using **Sybase** databases.

I worked for 16 months as a **contract analyst programmer** in the Group Risk area of Royal Bank of Scotland. I used a variety of technologies to build and enhance applications on **Windows** and **Unix**, including **C++**, **MFC**, **ODBC**, **ADO**, **COM**, **VB** and Source Safe on Windows; and **C++**, **Java**, **Sybase**, **C**, **JSP**, **Apache**, **Tomcat**, **perl**, **shell scripts** and **RCS** on **Unix**.

I worked for 6 months as a **contract analyst programmer** in the **E-commerce** area at Barclays Capital: firstly on a **bond** trading system on **NT** written in **C++** using **STL** and **JTC** threading library; then on a multi-tier **FX** trading application, where I used **C++** (with **STL**) on **Solaris**, and **Sybase**. I used **XML** in inter-process communication and **ClearCase** for version control.

I worked for 2½ years as a contract analyst programmer and later **team leader/technical architect** at Credit Suisse First Boston on **3-tier client-server systems** in the back office. I wrote high-volume, **multi-threaded** servers in **C++**, using **RogueWave tools** on **UNIX**, accessing a **Sybase** database. Clients ran on **NT** and were written in **VB**. To interface the VB clients to the servers I wrote **COM** objects using **ATL**. I also wrote **web pages** using **Javascript** and **CGI** (**Perl** and **shell script**).

Employment

Recently I've been enjoying some time at home, but I'm now immediately available for work.

Credit Suisse May '07 to May '10

I worked on the Insight project, which is responsible for calculating Credit Suisse's exposure to Credit Risk. The system consists primarily of several multi-threaded, C++ application servers running on Solaris, some Sybase databases, and a Java front-end running on WebLogic. Corba and MQ are used for communicating with other systems, and PowerCenter for ETL.

I worked as a (hands-on) Technical Architect, responsible for selecting and ordering hardware, setting up development, UAT and production servers, failover and disaster recovery, SOX controls, and technology evaluation (e.g. a feature/performance comparison of Sybase ASE, Sybase IQ and Oracle for a data warehouse). I also: worked on the existing C++ servers; migrated Access databases and forms to C++ and Sybase; migrated the code base from ClearCase to Subversion; interviewed prospective team members and gave presentations and demonstrations.

In addition I worked with the lead Technical Architect in Risk IT, to introduce a common set of Unix/Linux development tools across the department, and went on to demonstrate some of these tools to the Tools Team in Credit Suisse, with a view to rolling them out across Credit Suisse. I also attended meetings with the Head of Department to discuss short and long-term technical strategy in the department.

Royal Bank of Canada Feb. '07 to Apr. '07

I worked on a project to provide market data across the bank globally. Initially I set up 3 Linux servers, a Wiki for collaboration, and Subversion for source code control. I then evaluated a number of technologies. I built a prototype market data web service by using Python scripts to load Bloomberg end-of-day data into a MySql database running on Linux, and using Apache and PHP to provide a query interface to the data. The database interaction was encapsulated, to make it easy to change to a different database. Using the same data, I also wrote a simple distributed market data service using ZeroC (http://www.zeroc.com), and compared the throughput of C++ with Python, and tested clients on Linux and Windows.

Credit Suisse Oct. '05 to Dec. '06

I worked on a project on the MaRS team in the Market Risk area of Credit Suisse. MaRS is responsible for calculating value-at-risk for the bank. The system consists of: some GUIs; a number of C++ application servers communicating via Orbix on Solaris; and Oracle databases. Part of the system caches data from a database in a multi-threaded application and performs calculations on it.

I analysed, redesigned and developed the cache to support larger data volumes and enhanced functionality. I also wrote test programs and documentation. Code was held in ClearCase. I then worked on a project to port the application servers to Linux. This provided significant performance improvements and cost benefits.

Royal Bank of Scotland Apr. '05 to Sep. '05

I worked on a project on the UniVaR team in the Group Risk area of RBS. UniVaR is responsible for calculating value-at-risk for the bank. The main component of the system is a multi-threaded C++ server running on Solaris, using a Sybase database on the back-end. The business required an enhancement to the system to support new risk calculations.

I was responsible for finalising the requirements, and the subsequent design, code, test, release and handover. All the work was C++ on Solaris, apart from some small changes to the Javascript front-end.

Goldman Sachs Feb. '04 to Mar. '05

I worked in the fixed-income Repo area which provides processing, GUIs and reports to Trading, Sales and Operations departments, as well as feeds to and from other areas of the bank. Some of my time was spent supporting front-office staff; maintaining existing applications; introducing new reports etc. This involved working with a variety of technologies including C, C++, Java, Sybase, CVS, perl, CGI, shell scripts etc. I was also involved in the design and build of new systems, and was able to introduce new ideas, for example to use XSLT to convert XML. I also introduced a number of tools (some open source and some that I had developed) and suggested procedural improvements.

The Sarbanes-Oxley Act has affected the way banks control and monitor access to their systems. I was responsible for defining and introducing new procedures for controlling database access. This involved auditing the existing systems, producing a plan, managing other staff and dealing with other areas of the company; to make sure a consistent approach was being adopted and that the project was completed in time for an external audit.

Royal Bank of Scotland Aug. '02 to Dec. '03

I worked on a project called GRD in the Group Risk area of RBS. GRD is the "golden source" of counterparty and related information for "large" customers (non-retail) at RBS. The system provides daily extracts, intra-day and web interfaces, a GUI for maintaining and browsing data and also takes in data from external sources e.g. Moody's.

I was responsible for rewriting the back-end of, and making enhancements to, the MFC C++ GUI application; writing new applications to use GRD data; upgrading production and development hardware and software; defining a disaster recovery environment; and website development; as well as giving presentations, mentoring and providing support. This required working with a range of languages (C++, Java, C, perl, shell script) and technologies (Apache, Tomcat, JSP, Sybase, CTlib, ODBC, ADO, JDBC), on Unix and Windows; and involved a pro-active approach to simplify, and improve the resilience of, the existing production processes.

Extended break

I took some time off, to work on the house and garden, and spend time with my family.

Barclays Capital Nov. '00 to May '01

Barclays were developing software, as part of the Bondclick consortium, to sell bonds on the web. They were also developing an application for the traders to allow them to trade on many different bond markets (e.g. BrokerTec, SWX, EuroMTS). My job was to write the interface software between the two projects. This was written as a plug-in library in C++ on NT, and used COM to interface to the web-side software. Data was formatted in XML.

I then moved onto an FX trading application. This used web-based Java on the front-end and C++ on Solaris with a Sybase database at the back-end. I wrote a number of the back-end components, as well as tools and test software (mostly C++, some shell script). Code was stored in ClearCase. I was also responsible for migrating the existing data to a new data model, which I documented, along with its associated stored procedures, on the intranet - automatically generating HTML from the database.

Credit Suisse First Boston Jan. '98 to Aug. '00

Credit Suisse had a requirement to provide reference data to other systems in the bank. They stored the data in a Sybase database, which was replicated to other sites, and needed a simple-to-use GUI application to allow maintenance of the data, and a convenient and efficient interface for passing the data on to other systems. I co-designed and wrote a multi-threaded UNIX (SunOS/Solaris) server in C++ to cache and manipulate data into a format suitable for other systems. I also wrote a COM layer to make this data available to VB applications. I then worked on the maintenance application, which involved VB GUI work, Visual C++ using ATL for the COM objects, and C++ with RogueWave tools h++ for the UNIX server. Code was stored in PVCS. I used shell scripts and SQL to start, monitor and configure the servers. Configuration was mostly done using meta-data, designed to be Summit compatible.

I then moved to an eighteen-person project (GRD) to consolidate all reference data across the bank globally. Development on the project was split between London and New York, with the majority in London. I was hands-on team leader for the six London developers and responsible for the overall technical architecture (analysis, design and implementation). I was also technical liaison to the New York team, to help them set up the software developed in London, and to advise on how to make enhancements without introducing incompatibilities with the London part of the system. I was able to reuse much of my previous work, and extend it, for example to deal with the much larger volumes of updates and queries (100+/sec). Software was required to: load data from existing systems, converting the data model and validating the content; make the data available to other systems using an ad-hoc query interface; distribute data to other systems using MQ Series messaging; and provide a GUI for the maintenance of the data. I was also responsible for the project's intranet website.