Daniel Sundman

Crisp AB

Name	: Daniel Sundman
Experience	: 14 years
Rolls	: Software Architect/Developer
Education	: M. Sc., Engineering Physics

Summary

Experienced software engineer specializing in Java/J2EE web based applications. He spent the last eight years in California working at Niku Corporation, helping develop its applications as a small startup company, through its acquisition by Computer Associates (a.k.a. CA). For the past few years he worked as a systems architect, where he was responsible for the design and implementation of Niku's flagship product, Clarity (http://www.ca.com/clarity).

Work Experience

2006 -	Crisp AB, Consultant, Stockholm, Sweden
1999 - 2006	Niku Corporation/Computer Associates, Redwood City, California
1998 - 1999	OOCL USA Inc. San Jose, California
1997 - 1998	Enea Data AB, Consultant, Stockholm, Sweden
1994 - 1997	Logica Svenska AB, Consultant, Stockholm, Sweden
1990 - 1992	Future Data AB, Software Engineer

Technical Skills

Operating Systems	Windows NT/2000/XP, Solaris
Languages	Java, SQL (Transact-SQL, PL/SQL), XSLT, C, C++, JavaScript
Databases	Oracle 8i, 9i and 10g, SQL Server 2000 and 2005
Version Control Systems	StarTeam, Subversion
IDEs	IDEA IntelliJ, Eclipse, Emacs
Other Technologies	J2EE, ant, XML, XSL, JDBC, JUnit, SVG, Castor, SOAP, Hibernate

Niku Corporation/Computer Associates (CA), Redwood City, California

Daniel started working at Niku in August, 1999 and held various positions through the years. He mainly worked with Clarity, the web application that eventually became Niku's flagship product, and its predecessor Niku 6. Clarity is based on J2EE technologies (Servlets, JSP, XML, XSL, JDBC, etc). Niku was acquired by CA at the end of 2005.

2006

Daniel had a blank check improving select parts of Clarity. Among other things, this led to an Ajax-based implementation of Clarity's portlet infrastructure. This implementation meant a significant improvement since a portlet can now be updated independently of other portlets on the page. A page can contain an arbitrary number of portlets of arbitrary complexity so the value of not having to reload the whole page is great.

Technologies: Java, Ajax, JavaScript, XML, XSLT

2005

During the first part of 2005 Daniel managed the infrastructure group known as CTG (Common Technology Group). CTG is responsible for Clarity's architecture. As the head of CTG, Daniel was responsible for the CTG projects in Clarity and also, as part of the Clarity steering committee responsible for the Clarity application as a whole. This position also meant a customer contacts.
As a manager Daniel still mantained a significant implementation responsibility for parts of the Clarity infrastructure.

2003 – 2004

During these years Daniel was responsible for major parts of CA Clarity™ Studio (http://www.niku.com/products.asp?id=55). More specifically this meant:

- **Portlet Infrastructure**
  Daniel was head designer for the Clarity portlet infrastructure which allows Clarity developers as well as Clarity customers to build their own portlets. Clarity Studio Portlets are "Excel" like tables or graphs/charts. Daniel "invented" NSQL which is a language built on top of standard SQL allowing meta data to be included in a SQL query. Daniel was also responsible for large parts of the implementation.

- **ODF (Object Description Framework)**
  Daniel was the brain behind ODF, which among other things is an injected layer between the logical abstractions in Clarity (Project, Resurce, Process,...) and their respective implementations. For legacy reasons these implementations use different technologies. ODF enables a generic (object oriented) way of working with them. ODF also makes it possible to configure large parts of the Clarity user interface.
  Note that ODF is referred to as PowerMods in the Clarity marketing material.

Technologies: Java, SQL, SVG, XML, XSLT,...

1999 – 2002

During these first years with the company Daniel helped a relatively immature development organization with a series of problems.

- The fast growing Niku spent time and money developing a code generator which was so poorly conceived and implemented that the only reasonable outcome was to discontinue it. Daniel lead the evaluation.
- Member of the group that designed and developed the architecture that became the foundation for Clarity's predecessor, Niku 6. This architecture can be

thought of as a predecessor to todays multitude of frameworks like e.g. Spring or Struts.
- Designed and implemented version 1.0 and 2.0 of the persistence framework that is still being used in Clarity.
- Introduced JUnit and nightly builds.

Technologies: Java, SQL, JDBC, XML, XSLT

OOCL USA Inc. 1998 – 1999, San Jose, California

Member of a top notch team that designed and implemented an optimization application for container management.

Technologies: Java, RMI, JUnit, iContract, GemStone/J

Enea Data AB, Consultant, Stockholm, Sweden

1998 **RPS (Swedish Police Board)**

Systems architect in a project where Daniel developed a C++ framework to abstract a transaction manager (Tuxedo) och a database (Informix) for business objects.

Technologies: Windows NT, Visual C++, Tuxedo, Informix, HP-UX

1997 **RSV (Swedish Tax Board)**

Mentor in a project which struggled to introduce object oriented methodologies. The assignment also involved a fair amount of implementation. E.g. Daniel implemented a utility that generated embedded SQL given the meta data from the database.

Technologies: HP-UX, Oracle, C (object oriented)

Logica Svenska AB, Consultant, Stockholm, Sweden

1996 – 1997 **Ericsson Radio Systems**

Developed a proof of concept application for Video-on-demand. The application consisted of a Java applet and the client's C++ application that shows the video.

Technologies: Java 1.0.2, Solaris, Netscape, Visual C/C++, OrbixWeb

1995 – 1996 **Internal Project**

Designed and implemented a CTI (Computer Telephone Integration) application. Was responsible for implementing ”Call Control” which formed the intelligence of the system. This system was the foundation of the the system now sold by ClearIT (http://www.clearit.se/).

Technologies: C++, Solaris, Oracle, Pro*C

1994 – 1995 **Första AP-fonden (Pension Fund)**

Developed a system for budget calculation.

Technologies: Oracle Forms, Reports och Graphics

Ericsson Radio Systems, Stockholm, Sweden, Masters Thesis 1994

Investigated the possibilities to produce a C compiler for Digital Signal Processors (DSP) that generates sufficiently optimized code for the demanding applications in which these processors are used. This meant creating a partial port of the GNU C compiler for a specific DSP (Texas Instruments C50).

Technlogies: GNU C Compiler, TI C50

Future Data AB, Developer, 1990 – 1992

During a break in his studies, Daniel designed and implemented an application for bookkeeping. He was responsible for the complete system: UI (Apple Macintosh), business logic and persistence.

Technologies: MPW Object Pascal, MPW C++, MacApp

Education

1988 – 1990, 1992 – 1994 **M. Sc., Engineering Physics (Scientific Computing)**
Course examples: Realtime Systems, Compilers, Computer Communication, Object Oriented Programming, Operating Systems, Computer Graphics, Image Analysis, Algorithms for Parallel Processors