

Profil DI Günter Obiltschnig

Software Engineer und C++ Experte mit 16 Jahren Erfahrung im Bereich Software Engineering und Software Architecture. Hauptinteressen sind C++ mit Schwerpunkt plattformübergreifende, verteilte und embedded Systeme, sowie XML und Middleware Technologien (speziell embedded). Gründer und Geschäftsführer von Applied Informatics. Regelmäßige Vorträge auf Fachkonferenzen in Europa und USA.

Geburtsjahr: 1973
Beschäftigungsverhältnis: selbständig
Position derzeit: Geschäftsführung
Reisebereitschaft: international

Aus- und Weiterbildung

1988 - 1993: Höhere Technische Bundes-Lehr- und Versuchsanstalt in Villach, EDV und Organisation, Reifeprüfung (Matura)

1993 - 1998: Universität Linz, Informatik, Diplom-Ingenieur

Berufserfahrung

seit 2004: Gründer und Geschäftsführer von Applied Informatics Software Engineering GmbH. Entwicklung der POCO Platform; Mitarbeit bei Projekten im Bereich RFID Middleware, Industrie- und Gebäude-Automatisierung, Eintrittskontrollsysteme.

Gründer und Lead Developer im POCO C++ Libraries Open Source Projekt.

2000 - 2004: Mitgründer eines Software-Unternehmens tätig in der Halbleiterindustrie. Verantwortlich für die Entwicklung einer C++ basierten Middleware für verteilte Systeme im Bereich Auswertung von Messdaten aus der Halbleiterfertigung. Consulting in den Bereichen XML Technologien, Software-Architektur und Frameworkentwicklung. Mitarbeit bei Standardisierungsgremien in der Halbleiterindustrie (SEMI).

1998 - 1999: Selbständige Consulting Tätigkeit; Mitarbeit bei diversen Softwareprojekten im Umfeld der Halbleiterfertigung (Testdatenauswertung; Schwerpunkt C++ Entwicklung auf unterschiedlichen Plattformen).

1992 - 1997: Mitarbeit bei diversen Softwareprojekten im Bereich automatisierte Testsysteme für integrierte Schaltungen (C++, SunOS/Solaris)

Fremdsprachen

Englisch (verhandlungssicher in Wort und Schrift)

Dienstleistungen

Softwareentwicklung in C++ mit Schwerpunkt verteilte Systeme, plattformübergreifende (cross-platform) Softwareentwicklung, embedded, Middleware, Embedded Linux Treiber- und Applikationsentwicklung, Schulung in den Bereichen C++ Entwicklung und Embedded Linux

Tools, Methoden, Kenntnisse

Programmiersprachen: C++ (STL, POCO, Boost, ACE), C#, Objective-C, Java, SQL, diverse Scriptsprachen

Betriebssysteme: Windows, Solaris, HP-UX, Mac OS X, (embedded) Linux, FreeBSD, QNX, Windows CE, OpenVMS

Datenbanken: Oracle, SQL Server, MySQL, SQLite

Middleware: Tibco Rendezvous, IBM MQSeries

Tools: Visual C++ (.NET 2003, 2005, 2008), Eclipse, Xcode, GCC, GDB, Rational Rose, Rhapsody, Artisan Studio, Enterprise Architect, ClearCase, Perforce, CVS, Subversion, Mantis, OmniTracker

Weitere Informationen

Ich programmiere seit meinem 12. Lebensjahr und kann auf umfangreiche Erfahrung und entsprechend breites und tiefes Wissen in der Software-Entwicklung für unterschiedlichste Systeme zurückgreifen. Im Laufe meiner Karriere als Entwickler habe ich bereits Software für 8-Bit Systeme, Großrechner (CICS, VM/CMS), OpenVMS, diverse Unix Plattformen, Mac OS (X), iPhone, Windows und embedded Systeme entwickelt. Nicht zuletzt dank meiner Erfahrung kann ich mich sehr schnell in neue Aufgabenstellungen einarbeiten. Mein Wissen über Technologien und Methoden der Softwareentwicklung halte ich stets auf dem neuesten Stand. Besonders interessieren mich herausfordernde, eigentlich "unmögliche" Projekte.

Ich bin Gründer und Lead Developer des POCO C++ Libraries Open Source Projektes (<http://pocoproject.org>). Das Projekt hat sich die Entwicklung allgemein verwendbarer, plattformübergreifender C++ Bibliotheken (ähnlich zur Java Class Library, bzw. dem .NET Framework) zum Ziel gesetzt. Die Bibliotheken wurden bereits in einer Vielzahl unterschiedlicher kommerzieller Projekte eingesetzt (z. B. VoIP Systeme,



Gebäudeautomatisierungssysteme, Steuerungen für holzverarbeitende Maschinen, Gensequenzen, Test- und Diagnosesysteme im Automotive-Bereich, etc.) und erfreuen sich ständig wachsender Beliebtheit unter C++ Entwicklern.

Ich halte regelmäßig Vorträge auf internationalen Fachkonferenzen (z. B. Embedded World, Entwicklerforum Embedded Linux, Embedded Software Engineering Kongress, Embedded Systems Conference, XML Conference).

Ich bin Mitglied bei der Association for Computing Machinery (ACM), der IEEE Computer Society und der Association of C and C++ Users (ACCU).

Referenzen

Auf Anfrage

Projekte

Technologie-Demonstration iPhone/Mindstorms

Zeitraum: 10/2008

Projektbeschreibung: Für eine Forschungsveranstaltung wurde eine Technologie-Demonstration entwickelt, bei der ein Lego Mindstorms Roboter über den Beschleunigungssensor eines iPhones gesteuert wird. Eine iPhone Applikation sendet dabei die Bewegungsdaten über eine WiFi Verbindung an ein Embedded Linux System. Auf dem Embedded Linux System läuft eine C++ Applikation, welche die Bewegungsdaten in Steuerungsdaten für den MindStorms Roboter übersetzt, und über Bluetooth an den Mindstorms Roboter sendet. Auf dem Mindstorms Roboter läuft unter nxtOSEK, einer Portierung des OSEK Systems auf die Mindstorms Steuereinheit, ein C Programm, welches die Steuerkommandos über Bluetooth empfängt und ausführt. Außerdem werden verschiedene Sensordaten über Bluetooth an das Embedded Linux System zurückgesendet, und dort über einen HTTP Server auf einer Webseite visualisiert.

Aufgaben im Projekt: Design des Gesamtsystems und Implementierung der iPhone-Applikation, der Embedded Linux Applikation, und des Roboter-Programms.

Eingesetzte Technologien: Apple Xcode, Cocoa Touch, Objective-C, UDP, HTTP, HTML, JavaScript, Prototype, C++, POCO C++ Libraries, GNU Toolchain, Embedded Linux, Bluetooth, C, OSEK (nxtOSEK)

Universal Plug and Play Stack

Zeitraum: 07/2008 – 12/2008

Projektbeschreibung: Eine bestehende Middleware Plattform wurde um Unterstützung für die Universal Plug and Play (UPnP) Technologie erweitert. Ein kompletter Protokollstack für UPnP (Multicast HTTP, SSDP, GENA) wurde von Grund auf neu entwickelt und eine bestehende SOAP Implementierung adaptiert, um den UPnP-spezifische SOAP Dialekt unterstützen zu können.

Aufgaben im Projekt: Projektleitung, Spezifikation der Software-Architektur, Implementierung und Test

Eingesetzte Technologien: Visual Studio 2008, C++, Linux, GCC, GDB, POCO C++ Libraries, POCO Remoting, UDP, Multicast, HTTP, HTML, XML

Eintrittskontrollgerät

Zeitraum: 02/2008 – 06/2008

Projektbeschreibung: Für den Kunden, einen Anbieter von Ticketing-Systemen in der Schweiz, wurde die Software für ein Eintrittskontrollgerät für Veranstaltungen entwickelt. Das Gerät besitzt einen Barcode-Scanner, einen Touchscreen, sowie verschiedene Netzwerkschnittstellen. Der Barcode-Scanner liest den Barcode von "print-at-home" Tickets, und die Codes werden anhand einer Datenbank auf dem Gerät auf Gültigkeit überprüft. Über eine Netzwerkverbindung ist das Gerät mit dem zentralen Ticketing-Server, sowie mit weiteren Kontrollgeräten verbunden. Die Datenbanken der Geräte werden ständig untereinander abgeglichen, bzw. über den Ticketing-Server aktualisiert. Ein integrierter Web Server ermöglicht die Fernwartung und Konfiguration der Geräte.

Aufgaben im Projekt: Definition der Software-Architektur, Implementierung (C++) und Test der Geräte-Software (automatisierte Tests) bis zur Einsatzreife

Eingesetzte Technologien und Tools: Toradex Colibri XScale Boards, Linux, OpenEmbedded, GCC, C++, GNU Make, GDB, POCO C++ Libraries, POCO Remoting, GTK+, gtkmm, SQLite, OpenSSL, ALSA, UDP, HTTP, HTML, JavaScript, XML, Doxygen, Subversion, Jira

Steuergerät für Industrie-Waagensysteme

Zeitraum: 11/2007 – 03/2008

Projektbeschreibung: Für den Kunden, einen Hersteller von Band- und Spezialwaagen, wurde die Software für ein neues Steuergerät für Waagensysteme und die dazugehörige Bedieneinheit entwickelt. Die Besonderheit stellt dabei die Bedieneinheit dar. Es handelt sich um ein ARM9-basiertes System mit Touchscreen, auf welchem ein Web Browser im Vollbild-Modus läuft. Die gesamte Benutzerschnittstelle wird über HTML und JavaScript (Ajax), sowie über den im Steuergerät laufenden Webserver realisiert. Dies ermöglicht es unter anderem, anstatt der Bedieneinheit einen normalen PC mit einem Web Browser zur Konfiguration und Überwachung des Systems zu verwenden.

Aufgaben im Projekt: Erstellung der Software-Architektur, Implementierung der Software für das Steuergerät und die Bedieneinheit (Web Server, Kernelmodule, Web Browser Engine) in enger Zusammenarbeit mit dem Kundenteam.

Eingesetzte Technologien und Tools: Digi Embedded Linux, ARM9, Anybus, Eclipse, GCC, GDB, C, C++, POCO C++ Libraries, XML, HTTP Server, HTML, JavaScript, Ajax, Prototype, WebKit, Qt, Perforce, Mantis

Level 2 Automatisierung im Stahlwerk

Zeitraum: 08/2007 – 11/2007

Projektbeschreibung: Für den Kunden, einen Stahl-Produzenten in den USA, wurde das Level 2 Automatisierungssystem einer Strangguss-Anlage überarbeitet. Das Level 2 Automatisierungssystem erstellt auf Basis von Produktionsplänen die Steuerungsdaten für die SPS Systeme der Level 1 Automatisierung. Das bisherige System, welches aus Visual Basic Applikationen, Oracle Stored Procedures und C++ Applikationen besteht, wurde in ein rein C++-basiertes System basierend auf der POCO Open Service Plattform umgestellt. Die Operator-Benutzerschnittstellen wurden von Visual Basic Applikationen auf Browser-basierte Web-Applikationen umgestellt.

Aufgaben im Projekt: Reverse Engineering der bestehenden Software, Neu-Implementierung und Test der Software bis zur Einsatzreife

Eingesetzte Technologien und Tools: Visual Studio 2008, C++, SQL, Oracle, ODBC, POCO C++ Libraries, POCO Open Service Platform, HTTP, HTML, ExtJS, Ajax, JavaScript, Subversion, Mantis

RFID Middleware

Zeitraum: 01/2005 – 06/2006

Projektbeschreibung: Der Kunde, ein führender Hersteller von RFID Software aus Österreich, entwickelt und vertreibt eine Middleware für die Integration und Verwaltung von RFID Systemen, sowie eine darauf aufbauende Entwicklungsumgebung für die rasche Erstellung von RFID Applikationen.

Aufgaben im Projekt: Beratung bei Software-Architektur, Erstellung von Geräte-Treibern zur Anbindung von RFID Readern, Barcode Scannern, Druckern und Automatisierungstechnik (OPC, MODBUS TCP), Anbindung an SAP, Beratung bei Auswahl von Frameworks und Toolkits.

Eingesetzte Technologien und Tools: Microsoft Visual Studio 2005, Microsoft .NET, C#, NUnit, SQL Server, XML, PML, Soft-SPS, OPC, MODBUS, SAP Auto-ID Infrastructure (All), HTTP Server und Client, diverse (teilweise proprietäre) Netzwerkprotokolle, Subversion, TestTrack Pro, Bugzilla

POCO C++ Libraries

Zeitraum: 10/2004 – heute

Projektbeschreibung: Entwicklung einer general-purpose Open Source C++ Klassenbibliothek welche auf unterschiedlichen Plattformen (Windows, HP-UX, Solaris, (embedded) Linux, Windows CE, QNX) verfügbar ist.

Aufgaben im Projekt: Projektleitung, Spezifikation, Design und Implementierung eines großen Teils der Library, Koordinierung eines internationalen Entwicklerteams; Consulting und Training zu POCO bei diversen internationalen Unternehmen, z. B. TAC, Nucor Steel, Adobe Systems

Eingesetzte Technologien: C++, XML, TCP/IP Sockets, HTTP, SMTP, FTP, POP3, OpenSSL, ODBC, MySQL, SQLite, Visual Studio, GNU Toolchain, Windows 2000/XP, Linux, HP-UX, HP aCC, Solaris, QNX, etc.

POCO Platform

Zeitraum: 03/2006 – heute

Projektbeschreibung: Entwicklung mehrerer kommerzieller C++ Toolkits basierend auf den POCO C++ Libraries (verteilte Objektservices, SOAP/WSDL Web Services, Zeroconf, NETCONF, Universal Plug and Play, Komponenten-Management und Device Management angelehnt an die OSGi Technologie)

Aufgaben im Projekt: Projektleitung, Spezifikation, Design und Implementierung

Eingesetzte Technologien: C++, XML, SOAP/WSDL, TCP/IP Sockets, Universal Plug and Play, NETCONF, Zeroconf, etc.

Leitende Entwicklung einer Middleware Plattform

Zeitraum: 01/2000 – 06/2004

Projektbeschreibung: Leitende Entwicklung einer Middleware Plattform für verteilte Applikationen im Bereich der automatischen Analyse von Qualitätsdaten aus der Halbleiterfertigung. Beinhaltet eine C++ Klassenbibliothek für plattformunabhängige Programmierung (Windows, Linux, HP-UX, Solaris, OpenVMS), verteilte Objektdienste, SOAP/WSDL Web Services, Applikationsserver (Web Server, Servlets, Load Balancing/Clustering), Integration mit Enterprise Messaging-Systemen (Tibco Rendezvous, IBM MQSeries), Entwicklung eines Interpreters für eine JavaScript-ähnliche Skriptsprache

Aufgaben im Projekt: Spezifikation, Design und Implementierung aller wesentlichen Systemkomponenten

Eingesetzte Technologien: C++, XML, SOAP/WSDL, XML Schema, TCP/IP Sockets, Tibco Rendezvous, IBM MQSeries, UML, Rational Rose, HTTP, Oracle (OCI), Visual Studio, MFC, GNU Toolchain, Windows NT/2000, Linux, HP-UX, Solaris, OpenVMS

Analyse von Qualitätsdaten aus der Halbleiterfertigung

Zeitraum: 04/2002 – 06/2004

Projektbeschreibung: Implementierung diverser Applikationen zur Analyse von Qualitätsdaten aus der Halbleiterfertigung.

Aufgaben im Projekt: Design, Implementierung und Test der Applikationen, Entwicklung eines Interpreters für eine domainspezifische Regelsprache

Eingesetzte Technologien: C++, XML, Windows NT, OpenVMS

Diverse Projekte im Bereich automatisierter Test von ICs

Zeitraum: 07/1993 – 09/1997

Projektbeschreibung: Weiterentwicklung eines bestehenden Software-Systems zur Generierung von Testdaten für den automatisierten Test von Halbleitern.

Aufgaben im Projekt: Implementierung eines Interpreters für eine projektspezifische Regelsprache, GUI Entwicklung

Eingesetzte Technologien: C++, xview/OpenWindows, SunOS/Solaris

Benutzerschnittstelle für eine interaktive Musik-Installation

Zeitraum: 10/1996 – 05/1997

Projektbeschreibung: Für das Ars Electronica Center in Linz wurde eine interaktive Installation entwickelt, welche Besuchern spielerisch demonstriert, wie Computer für die Erzeugung von Musik eingesetzt werden können.

Aufgaben im Projekt: Implementierung einer speziellen Benutzerschnittstelle, welche über ein Buchla Lightning II Eingabegerät (eine Art "Infrarot-Taktstock") bedient wird. Die UI Komponente ist eine Erweiterung für MAX (eine graphische Programmiersprache für Applikationen im Musikbereich) und in C programmiert. Benutzerschnittstellen werden über eine spezielle Definitionssprache definiert. Weiters wurde eine Erweiterung für MAX programmiert, welche die Integration von AppleScript in eine MAX Applikation ermöglicht.

Eingesetzte Technologien: Mac OS, C, AppleScript, MAX, MIDI

Persistierung von C++ Objekten

Zeitraum: 08/1992 – 09/1992

Projektbeschreibung: Im Rahmen eines Projektes zum automatisierten Testen von integrierten Schaltungen wurde ein Persistenzframework für C++ Objekte, basierend auf einer B-Tree Struktur implementiert.

Aufgaben im Projekt: Design, Implementierung und Test des Frameworks.

Eingesetzte Technologien: C++, SunOS

Verteiltes Build System

Zeitraum: 07/1992

Projektbeschreibung: Es wurde ein verteiltes Build System entwickelt, mit dem eine umfangreiche C++ Applikation in einem Netzwerk aus Sun Workstations verteilt gebaut werden kann.

Aufgaben im Projekt: Design, Implementierung und Test des Build Systems.

Eingesetzte Technologien: csh, OpenWindows, SunOS