

Profil

Sven Göthel

January 25, 2024

Ich verfüge über mehr als 30 Jahre Erfahrung in der Softwareentwicklung und bietet Unterstützung für den gesamten [QA](#), [Prozess](#) & [Lebenszyklus](#), sowie in C, C++ und Java, einschließlich deren plattformübergreifender Anwendung an. Kundenspezifische Felder & Sprachen werden auf Wunsch erlernt und angewendet.

Ich war bei Sun Microsystems (jetzt Oracle) und ATI (jetzt AMD) in deren Hauptsitz fest angestellt, wonach ich als Auftragnehmer weiterarbeitete. Ich war für ein breites Spektrum von Unternehmen tätig, darunter Zafena, Mathworks, Google, Harman/Becker, Siemens AG München und IBM Frankfurt.

Neben der Arbeit an proprietären Produkten [veröffentliche](#) ich auch [Open Source Artefakte](#).

Ich verfüge umfassende Kenntnisse in dem Embedded Bereich, [Computergrafik & Multimedia](#), [Compiler-Bau](#), Netzwerke & [Bluetooth](#), [Kryptographie](#), Treiberentwicklung, Algorithmen und Hochleistungslösungen.

Mit folgenden System habe ich u.a. gearbeitet, GNU/Linux, BSD, Darwin/OSX, Windows, QNX, Android, iOS, Zephyr und bare-metal Embedded-Devices via VxWorks, Keil, etc.

Das bekannteste Projekt an dem ich arbeite ist [JogAmp](#), bei dem ich Kunden helfe JogAmp einzusetzen und beauftragte Funktionen implementiere. Innerhalb von JogAmp arbeite ich auch an [GraphUI](#), einer plattformübergreifenden Benutzeroberfläche, welche unseren eigenen [auflösungsunabhängigen GPU-Kurven-Renderer](#) nutzt und für Desktop- und Embedded-Systeme geeignet ist.

Präsentationen: [Latest Blogs](#), [JavaOne 2008](#), [Tegra-1 Video Demo](#), [Siggraph\[2010, 2011, 2012, 2013, 2014\]](#), [Fosdem\[2013, 2014\]](#).

Hier finden Sie einige Anmerkungen zu einigen meiner aktuellen Projekte:

- [Direct-BT](#) ermöglicht die direkte Programmierung von [Bluetooth LE](#) und [BREDR](#) und bietet eine robuste und leistungsstarke Unterstützung für Embedded-System & Desktop ohne Overhead über C++17 und Java. Es wurde bereits erfolgreich in einer medizinischen Studie und in einem [Terminal für angeschlossene medizinische Geräte](#) eingesetzt.
Präsentationen: [Latest Blogs](#), [BLE Programming with C++ & Java](#)
- [Cipherpack](#) bietet verschlüsseltes und authentifiziertes (OTA) Streaming und atomare Datei-Operationen, um ein sicheres System- und Anwendungs-Upgrade über eine breite Palette von Medien für Geräte im medizinische Bereich zu implementieren.

Außerdem gebe ich einen [kleinen Informatikkurs](#), in dem ich derzeit [C++-Grundlagen](#), [gfxbox2](#), [hanoi](#), [pacman](#) and [gentech](#), einen genetischen Algorithmus-Löser, behandelt.

Ich nutze eine Vielzahl von Werkzeugen, um den gesamten Projektlebenszyklus abzudecken: [GIT](#) zur [decentralized source control](#); [Bugzilla](#) for [requirement management](#), [bug triage](#) and [release definition](#); [Jenkins](#) for continuous integration including running our more than [1300 unit tests](#) on supported [test platforms](#); [Mediawiki](#) and [API doc](#) for seamless documentation and last but not least a [forum](#) for transient communication.

Um das Jahr 1981, lernte ich mit 13 Jahren das Programmieren auf einem [TI-57](#), [VIC-20](#), [Commodore 64 \(C64\)](#) und später [Amiga 500](#) mit Basic, 6502 & 68k Assembler und C mit dem [Lattice C Compiler](#). Mein erstes Hacken auf dem C64 in 6510-Assembler umfasste Kopierprogramme, eine Indexdatenbank und berüchtigte Sound- & Grafikeffekte. Ich programmierte auch einen funktionierenden IRQ-basierten präemptiven Multitasking-Scheduler auf dem C64 für eine nie fertiggestellte [core wars](#) Programmierspielvariante, die 1984 von [A.K. Dewdney](#) inspiriert wurde.

Bis ich mir meinen eigenen Computer, einen C64, leisten konnte, programmierte ich unfreiwillig mit Gleichgesinnten in den örtlichen Kaufhäusern. Letzteres war eine Quelle der Inspiration und es ermöglichte uns den Austausch von Informationen im Zeitalter vor dem Internet, in dem Stift & Papier, Kassetten und Disketten dominierten.

Meine Erfahrung mit C++ reicht bis ins Jahr 1993 zurück, wo ich mit Bjarne Stroustrups [Design and Evolution of C++](#) begann. Zuerst verwendete ich C++ in einem CAD/CAM-Projekt, das einen DSL-Compiler für die benutzerdefinierte CNC-Programmierung beinhaltete. Ich nutze g++ mit sanitizer, valgrind, clang, clang-format, clang-tidy und anderen Tools zur Code-Validierung und performance Analyse.

Ich lernte Java von den Anfängen in 1996 und verwendete es zunächst in einem Projekt für MVC Business-Logic-Trennung und automatisierte UI-Generierung.

Bei Sun leitete ich 2007 das Refactoring des [New Java Plug-In](#) mit [JNLP-Unterstützung](#) und demonstrierte später ein AWT-loses Proof-of-Concept *Plugin 3*, das für einen Relaunch von Java auf dem Desktop und mobilen Geräten gedacht war. Ich trug auch zu einer neuen experimentellen JavaFX-Plattform bei, die hardwarebeschleunigtes Rendering (OpenGL, OpenMax, ...) nutzt. Infolgedessen fügte ich [JOGL](#) eine plattformunabhängige [Windowing-Architektur](#) mit Unterstützung für [mobile Geräte](#) und [OpenGL-Profilen](#) hinzu, die später Teil des unabhängigen [JogAmp-Projekts](#) wurde.

Bei ATI und AMD leitete ich 2005 die Entwicklung des Embedded Display OpenGL-Treibers, wo ich verschiedene Kundenfunktionen & Leistungsanforderungen verwaltete und implementierte: Performance OpenGL-Erweiterung, Hardware-Overlay, asynchrones ReadPixel, GLSL/DMA-Texturübertragungen, Page-Flip, VSync und Multi-DMA-Page-Locks. Ich verbesserte die Leistung des Treibers um den Faktor 2,5, wobei ich auf funktionale Treiberblöcke wie TLS, Inlining, Caching, Dispatch Table, Stream Copy usw. abzielte.

Ich habe einen Abschluss in Elektrotechnik mit Schwerpunkt Informatik, Dipl.-Ing. (FH) - Diplomingenieur, Angewandte Wissenschaft - MASc-Äquivalent, der mir 1998 von der Fachhochschule Bielefeld, Deutschland, verliehen wurde.

Contents

1	Lebenslauf / CV	4
2	Erfahrungen	5
3	Skills	20

1 Lebenslauf / CV

Sven Göthel

Born 1968 in Bremerhaven, Germany

Married, three children

Goethestrasse 1, 27576 Bremerhaven, Germany

Career

- 11/09 - Software Engineering / Consulting, Sole Proprietor
<http://jausoft.com>
- 9/08 - 11/09 Sun Microsystems, Inc., Santa Clara, CA, USA
Staff Engineer
- 8/07 - 09/08 Sun Microsystems of Canada, Inc., Edmonton, AB, Canada
Staff Engineer
- 6/05 - 7/07 Advanced Micro Devices (AMD),
formerly ATI, Markham, ON, Canada
Senior Software Developer and Team Lead
- 9/91 - 6/05 Software Engineering / Consulting, Sole Proprietor
<http://jausoft.com>
- 9/91 - 1/98 Study of electrical engineering with a focus on computer science
at the Bielefeld University of Applied Sciences, Germany, Europe.
Dipl.-Ing. (FH); Diplom Engineer, Applied Science (Equivalent to MASc)
Graduation in January 1998
- 8/90 - 6/91 A-Level certificate.
Senne-College, Bielefeld, Germany, Europe.
Graduation in June 1991.
- 8/88 - 3/90 Civilian Services.
Communications Center Dodesheide, Osnabrueck, Germany, Europe.
- 8/87 - 6/88 High School examination at
SZ Buergerpark, Bremerhaven, Germany, Europe.
- 9/85 - 6/87 Education as an industrial electrician
SZ Buergerpark, Bremerhaven, Germany, Europe.
Journeyman examination at the Bremerhaven Chamber of Commerce (IHK).

2 Erfahrungen

11/2022 - today Open Source: JogAmp incl. Commercial Support

Project: JogAmp Support, GraphUI, compiler features, high dpi, video encoding & decoding and misc. features

Activities: Supporting clients adopting JogAmp for their commercial projects and implementing commissioned features and platform support.

Activities: Realizing GraphUI, a cross-platform UI utilizing our own resolution-independent GPU curve renderer. GraphUI is suitable for desktop and embedded devices, it also runs on top of console kernel DRM driver without a windowing toolkit. Notable entries are Reimagine Java on Desktop & Bare-Metal Devices, FontView App, Type Animation 2, Type Animation at your Fingertips and others.

Presentations: Latest Blogs, JavaOne 2008, Tegra-1 Video Demo, Siggraph[2010, 2011, 2012, 2013, 2014], Fosdem[2013, 2014].

IT-Environment: Windows, Linux, Android, MacOS, C++, C, Objective-C, Java, Vulkan, OpenGL, OpenCL, OpenMax, OpenKode, Mesa 3D and Linux DRM/DRI driver, ARM32, ARM64, Amd64, BCM, Omap3, Tegra, GCC, MingW, UML, git, Eclipse, vim, ..

10/2023 - 11/2023 Zafena & Göthel Software

Project: Adding Direct-BT GATT-Server while being in BT-Client mode and hardening Passkey Authentication.

Activities: Certain devices acting as master/server require the client to provide a GATT-server for successful authentication. This capability is essential to have these operating successful with the clients POC-Workstation.

Activities: General passkey authentication within Direct-BT and the client's application had been hardened and resolved.

IT-Environment: Bluetooth Core Spec v5.2, Linux Kernel BlueZ driver, C++17, gcc, llvm, valgrind, gdb, Java, Linux, X86 and X86_64, ARM32 and ARM64, Raspberry, cmake, Doxygen, UML, git, ..

08/2023 - 09/2023 Zafena

Project: Detecting Rapid Test Results From a USB attached Camera.

Activities: Design & implementation of an embedded device software module to detect rapid tests results from live streamed images of a USB camera.

Image filter as well as appropriate detection algorithms had been chosen, used and partially implemented.

IT-Environment: OpenCV, C++17, gcc, Java, Linux, X86 and X86_64, ARM64, Raspberry, cmake, git, ..

04/2023 - 07/2023 **Ankama Studio**

Activities: Supporting JogAmp for a specific range of MacOS platforms

Activities: Resolving outstanding HiDPI issues with AWT

Activities: Adding GlueGen compiler support to generate Java callbacks for native toolkits as well as completing struct- and type-mappings.

Activities: General support adopting JogAmp and [GraphUI](#).

IT-Environment: Linux, Windows, MacOS, C++, C, Objective-C, Java, OpenGL ..

12/2022 - 01/2023 **GeoGebra**

Activities: Supporting JogAmp for MacOS M1 aarch64

IT-Environment: MacOS M1, aarch64, C++, C, Objective-C, Java, OpenGL ..

04/2020 - 10/2022 **Zafena & Göthel Software**

Project: [Direct-BT](#) and [POC-Workstation](#) Support.

Activities: Design & realization of Direct-BT, a new Bluetooth stack using C++17 inclusive Java binding.

[Direct-BT](#) provides direct [Bluetooth LE](#) and [BREDR](#) programming, offering robust high-performance support for embedded & desktop with zero overhead via C++ and Java.

It supports a fully event driven workflow from adapter management and device discovery to GATT programming, using its platform agnostic HCI, L2CAP, SMP and GATT protocol implementation.

Multiple Bluetooth adapter are handled, as well as multiple concurrent connections per adapter. Peripheral server device programming is supported as well as the central client, which is also used for self unit testing across two or more Bluetooth adapter.

Further, the provided repeater application allows to connect between a Bluetooth client and server to analyze their protocol.

Direct-BT has been used successfully in a medical trial, as well as in a [connected medical device terminal](#). The [C++ and Java support library](#) has been extracted to encapsulate its generic use-cases.

To become feature complete using the underlying *BlueZ* Linux kernel host Bluetooth driver, the kernel implementation had to be analyzed via code review, manual instrumentation and debug traces. This resulted in full access of the *Security Manager Protocol (SMP)* via L2CAP and helped achieving Bluetooth 5 compatibility.

Activities: Support adoption of Direct-BT and participate in certain life-cycle tasks, code review, testing and production.

The POC-Workstation is a connected medical device terminal, a widely deployed embedded system. Direct-BT enables the station to handle multiple parallel connections to medical devices as well as real-time data streaming and status monitoring.

SMP LE Secure Connections and LE legacy pairing is fully supported. Additionally, a BLE protocol analyzer has been developed.

Presentations: [Latest Blogs](#), [BLE Programming with C++ & Java](#)

IT-Environment: [Bluetooth Core Spec v5.2](#), Linux Kernel *BlueZ* driver, C++17, gcc, llvm, valgrind, gdb, Java, Linux, X86 and X86_64, ARM32 and ARM64, Raspberry, cmake, Doxygen, UML, git, gitlab, Bugzilla, Eclipse, vim, ..

06/2022 - 08/2022 [Zafena](#)

Project: Secure System and Application Upgrade for a Connected Medical Device Terminal

Activities: Adopting [Cipherpack](#) (OTA) streaming and atomic file operations and implementing a secure system and application upgrade.

Upgrade the deployed embedded system's read-only OS and application squashfs image using a wide range of media, e.g. http, smb, usb memory stick, etc.

The implementation follows the 'copy once' pattern, the sizeable OS image is put in place on the filesystem during Cipherpack's decryption and authentication streaming.

A factory default and last version fallback boot facility is provided while managing multiple OS versions. Robust instantiation of a new OS instance is realized via secure hash validated OS level file copy and atomic rename operations.

A full set of POSIX file system operations inclusive a privilege separated filesystem image mount has been [exposed to C++ and Java](#).

IT-Environment: C++17, Java, Linux, X86 and X86_64, ARM32 and ARM64, Raspberry, cmake, Doxygen, UML, git, gitlab, Bugzilla, Eclipse, vim, ..

2022 [Zafena](#) and [Göthel Software](#)

Project: [Cipherpack](#), authenticated secure stream processor.

Activities: Design & realization of Cipherpack using C++17 inclusive Java binding.

Cipherpack is a transport stream agnostic authenticated secure stream processor. It utilizes public-key signatures to authenticate the sender and public-key encryption of a symmetric-key to multiple parties ensuring their privacy while allowing high-performance authenticated message encryption.

Primary use case in this project was the OTA application and OS upgrade process for a deployed embedded system, a terminal for connected medical devices.

The API utilizes a listener event model, allowing the user to interact during stream [de]cipher , e.g. rejecting the stream altogether based on header information.

IT-Environment: C++17, Java, Linux, X86 and X86_64, ARM32 and ARM64, Raspberry, cmake, Doxygen, UML, git, gitlab, Bugzilla, Eclipse, vim, ..

2021 Wyss Center, Neurotechnology Research Foundation

Project: [Direct-BT](#) adoption for a Medical Clinical Trial.

Activities: Support adoption of Direct-BT in conjunction with custom BLE sensors, enhance Bluetooth 5 support and help create an intermediate software layer.

IT-Environment: [Bluetooth Core Spec v5.2](#), C++17, Linux, BLE sensors, ..

11/2009 - 06/2020 Open Source: JogAmp incl. Commercial Support

Project: [JogAmp](#), high performance Java libraries for 3D Graphics ([OpenGL](#)), multimedia ([OpenAL](#), [ffmpeg](#)) and processing ([OpenCL](#)) across platforms (Windows, MacOSX, Linux, Android) for [desktop](#), [mobile](#) & [embedded systems](#).

Activities: Supporting clients adopting JogAmp for their commercial projects and implementing commissioned features and platform support.

To manage the project, I utilize a diverse set of tools to cover the project life-cycle: [GIT](#) for [decentralized source control](#); [Bugzilla](#) for [requirements](#), [bug triage](#) and [release definition](#); [Jenkins](#) for continuous integration including running our more than [1300 unit tests](#) on supported [test platforms](#); [Mediawiki](#) and [API doc](#) for seamless documentation and last but not least [a forum](#) for transient communication.

Besides project management, I have resolved a few [notable tasks](#) from 2019 until [May 2020](#) for the upcoming [2.4.0 release](#): [iOS support](#), [DRM/GBM 'bare metal' support for embedded systems](#), [JavaFX support](#), [Java11 support](#), [SWT and HiDPI issues](#), [Aarch64 support](#) (Linux, Android), compliance with [Android \$\iota=10\$ \(SDK 29, NDK 20\)](#) and with critical MacOS API changes.

From 2010 until 2019, we fixed 589 bugs and realized numerous cross-platform features, some notable are: GPU based curve rendering ([Graph](#)); [Android support](#); [Native windowing toolkit \(NEWT\)](#) with cross-toolkit re-parenting, multi-monitor and [HiDPI support](#); [FFMpeg video decoding](#) and camera input with GPU pixel format conversion; [Smart JAR file deployment support](#); [Runtime binary format detection \(ABI\)](#).

Presentations: [Latest Blogs](#), [JavaOne 2008](#), [Tegra-1 Video Demo](#), [Siggraph\[2010, 2011, 2012, 2013, 2014\]](#), [Fosdem\[2013, 2014\]](#).

IT-Environment: Solaris, Windows, Linux, Android, MacOS, C, Objective-C, Java, OpenGL, OpenCL, OpenMax, OpenKode, Mesa 3D and Linux DRM/DRI driver, ARM32, ARM64, Amd64, Sparc, BCM, Omap3, Tegra, GCC, MingW, UML, git, Eclipse, vim, ..

2008 - 2009 Sun Microsystems, Inc. & Open Source

Project: JOGL Redesign & Support

Activities: Redesigned the OpenGL Java language mapping JOGL to support multiple [windowing systems](#) and exposed multiple [OpenGL profiles](#) (ES 1.1, ES 2.0 and GL 1.5 - 4.1) targeting multiple desktop and embedded/mobile devices, technical lead.

Started a simple [native windowing toolkit \(NEWT\)](#), demonstrating efficient JOGL usage on desktop and mobile devices without AWT.

Supported porting JOGL to multiple embedded devices, [APX 2500](#), Broadcom/LG, Intel Can-More and PowerVR-SGX/Omap3.

[Adding OpenMax IL multimedia support access for embedded platforms.](#)
[APX 2500 Demo at JavaOne 2008](#)

IT-Environment: Solaris, Windows, Linux, Android, MacOS, Java, JOGL, OpenGL, JOCL, OpenCL, OpenMax, OpenKode, ARM, IA-32, Amd64, Sparc, Omap3, Tegra, GCC, MingW, ..

09/2008 - 11/2009 Sun Microsystems, Inc.

Project: *Java Plugin3* Development

Activities: Designed & realized *Plugin3*, technical team lead.
Abstracted the windowing layer and modularized Plugin2, forming Plugin3. It featured usage of custom windowing implementations bootstrapped via JNLP.
Improved startup time about factor 3 - 13 (new load, reload). Added support for offscreen drawable to allow browser side compositioning.
Implemented an OpenGL proof of concept module using JOGL and NEWT.

IT-Environment: Solaris, Windows, Linux, Java, OpenGL, GLX, WGL, Networking, UML, IA-32, Amd64, GCC, MingW, ..

Project: [Experimental JavaFX](#) 3D and multimedia cross platform UI API.

Activities: Targeted desktop and mobile devices for a high performance, state of the art user interface (HMI). Utilized hardware acceleration (OpenGL ES1.1, ES2.0, GL 1.5-2.0, OpenMax) for the scenegraph layer.
Adding support for [per pixel lighting \(normal maps\)](#) using [Maya files](#).
Modularized and prepared the scenegraph layer for JOGL's mobile GL profiles.

IT-Environment: Linux, MacOSX, Java, JOGL, OpenGL, IA-32, Amd64, GCC, Maya, ..

08/2007 - 09/2008 Sun Microsystems of Canada, Inc.

Project: *New Java Plug-In* incl. JNLP support

Activities: JNLP integration in the [New Java Plug-In](#). Designed & refactored the Java Plugin and JNLP modules, to become Plugin2. Added Unix Domain Sockets for the client/server IPC on unices. Stabilized the client/server communication and improved startup time. Helped specifying the JNLP extension and extended the test framework covering the new features.
Presented at [JavaOne 2008](#).

IT-Environment: Solaris, Windows, Linux, Java, IPC, Networking, UML, IA-32, Amd64, Sparc, GCC, VS-C, ..

06/2005 - 07/2007 Advanced Micro Devices (AMD), formerly ATI

Project: Embedded Graphics driver; Various embedded driver features and enhancements.

Activities: Managed requirements, designed & realized an embedded graphics driver for OpenGL core, EGL 1.1, ES 1.1/2.0 on various graphic chips R3xx, R5xx, R6xx and their mobile variants. Technical team lead.

Managed & implemented various customer feature & performance requirements: Performance OpenGL extension, hardware overlay, asynchronous ReadPixel, GLSL/DMA texture transfers, page flip, VSync and multi DMA page locks.

Transferred requirements to new technical specifications.

Enhanced performance about factor 2.5, targeting functional driver blocks like TLS, inlining, caching, dispatch table, stream copy, etc.

Participated in the general driver development within the software team and helped with enhancements. Worked on SDK & demo infrastructure and multimedia demos.

IT-Environment: Doxygen, Office, UML, Perforce, Remedy; GNU: GCC, Linux, Valgrind, VTune, ATI Graphic Cards, Linux Kernel driver, OpenGL, GL ES; IA-32 and PPC, MPEG.

04/2005 - 06/2005 Hella

Project: Non Volatile Ram Manager

Activities: Managed requirements and specification for a new NV Ram Manager. The backward compatible module covered new customer requirements.

IT-Environment: MKS, Doors, Innovator, Doxygen, MS Office, NEC, Lotus

12/2004 - 03/2005 Jausoft

Project: Linux Device Driver - Monolithic and User Space

Activities: Investigated user space functionality within Linux device drivers. Learned pros and cons of simple resource handler and utilization of complex user space libraries. Implemented a portion of the Linux DVB API as an user level driver using FUSD and Gelato project knowledge.

IT-Environment: Linux Kernel 2.6.x driver, gcc, g++, dvb API, FUSD, gelato, ...

08/2004 - 11/2004 SiemensVDO

Project: Embedded Multimedia Player

Activities: Evaluated hardware and software solutions.

Wrote proof of concept including evaluation, test and quality audit of mpeg solutions incl. software toolchain.

IT-Environment: Change Synergy, CM Synergy, x86, ESS 6218, ST20, Ansi-C++, DSP, gcc, mpeg2, a52, mp2, lpcm, dvd, ...

09/2003 - 06/2004 Harman/Becker Automotive Systems

Project: OSGI, Multimedia, Embedded Systems, Open Source, ...

Activities: Designed & realized an embedded multimedia platform for QNX/SH4 inclusive custom DSP A/V device driver, which implemented the Linux DVB-API and hence enabled open-source application on the new target platform.

Evaluated, ported and enhanced misc. open source multimedia projects, like vdr, vdr-dvd, xine, mplayer, libdvd, ... to be used on mentioned QNX/SH4 target platform, utilizing the new DSB A/V driver.

Evaluated Linux-SH for the SH4 developing platform, added pci-dma support and implemented PCI-IRQ management. Ported a SH4 bootloader using bootp/nfs for disc-less system.

Evaluated a platform independent GUI/Widget toolkit: QT. Objectives were to develop, test and cross-compile UI projects on different platforms. Quality requirement goals were source code and UI builder availability and existence of some platform ports.

Implemented OSGI bundled applications in an client/server environment for embedded systems (QNX-NTO, XFree86, Java/J2ME, ...).

IT-Environment: Linux Kernel driver, QNX-NTO driver, x86, SH4, Ansi-C, Ansi-C++, Java, OSGI, XFree86, QT, QT Designer, DSP, gcc, gdb, ddd, mpeg2, a52, mp2, lpcm, dvd, ...

06/2003 - 06/2004 Jausoft

Project: Multimedia

Activities: Evaluated & enhanced open source multimedia projects, like VDR, vdr-dvd, ...

IT-Environment: GNU/Linux, ANSI-C++, mpeg2, a52, dvd, ...

03/2002 - 03/2002 JavaOne 2002, San Francisco, CA

Project: Speaker: [OpenGL\(R\), and New I/O - High-Performance 3D Graphics for the Desktop Client](#).

Activities: Presented utilization of the JVM's high performance New I/O using my OpenGL(R) language mapping for Java(TM) or GL4Java.

IT-Environment: GNU/Linux, Windows, Java, OpenGL

01/2000 - 11/2005 Various customers

Project: System Administration

Activities: Planned, setup and maintained heterogeneous IT infrastructure using GNU/Linux, iptables, ppp, apache, sendmail, fetchmail, procmail, samba, nfs, nis, ssl, ssh, ... for the DMZ to serve GNU/Linux and Windows clients within different secure and insecure subnets. Customers were public schools, training centers and internet shops.

IT-Environment: GNU/Linux, Windows, MacOSX; iptables, ppp, apache, sendmail, fetchmail, procmail, samba, nfs, nis, ssl, ssh, popper, imapd, VPN, ..

02/2002 - 08/2003 Harman/Becker Automotive Systems

Project: OSGI - Car-Remote-Control

Activities: Implemented ClientServer OSGI bundles. The server, installed into the car, provided misc. remote maintenance services, like door locking, camera snapshots and retrieving the GPS position. The client could connect to the server and use said services in the role of the main user or as an observer with read-only access. The network access was secured via SSL & RSA encryption and authorized using RSA signatures.

The goal was to demonstrate the feasibility of a remote car control via Java and OSGI bundles, which could be installed onto the client through the internet.

IT-Environment: Java (Sun J2SE, IBM's J2ME J9), OSGI (IBM, Oscar, Acunia), HTTPS/SSL (JSSE), RSA (JCE) Encryption + Signature, Serial+Parallel (JavaComm), MOST, GPS

02/2002 - 08/2003 Harman/Becker Automotive Systems

Project: XFree86 Integration and Driver Modules

Activities: Implemented an XFree86 driver module for the Scarlet Fujitsu MB86291 graphics device running under Linux & QNX (X86, SH4).

Enhanced and matured the XFree86 driver module for the Siliconmotion Lynx3DM graphic device running under Linux & QNX (X86,SH4).

The X11-Server was intended to be used within embedded systems and therefor should have provided a graphics interface with network abilities.

Using an X11-Server in the Unix (QNX) environment opened up the embedded platform to a sizable amount of available applications, e.g. media players, Java AWT implementations etc.

I added hardware accelerated functions for the graphic device like blitting (XAA) and YUV overlay (Xvideo) and exposed them to the X11R6 interface. High 2D and video playback performance with the Scarlet and the Lynx3DM chips were realized.

Miscellaneous adaptations of the XFree86 internals for QNX-NTO using the X86 and SH4 processor were made.

IT-Environment: Linux driver, QNX-NTO driver, ANSI-C, ANSI-C++ XFree86, Scarlet Gfx Chip, Lynx3DM Gfx Chip, gcc, gdb, ddd, X86, SH4

04/2002 - 05/2002 Harman/Becker Automotive Systems

Project: X11-Server AWT Implementation for Java

Activities: Completion & Bugfixing of the AWT Implementation of Kaffe. This AWT implementation did not require Motif (low Memory Footprint), but rendered the GUI Widgets exclusively using the XLib. With the assistance of this AWT solution we were able to use standard Java GUI's within the J2ME environment on a Unix and X11R6 based platform.

A QT solution was evaluated as well, but because of the limited availability of a native QT implementation for the embedded device, this approach was discarded.

IT-Environment: Java (Sun J2SE, IBM's J2ME J9), Linux, QNX-NTO, X11-Server, QT, Kaffe

11/2001 - 01/2002 Harman/Becker Automotive Systems

Project: Location Based Premium Services (Usability)

Activities: Evaluated an application server based upon XML. The application logic was provided via XML-Tags to the web-author, which were implemented using XSP application-sheets. These XSP application-sheets could be implemented with Java, as well as with other programming languages. The XML engine Cocoon provided a framework for said purpose. XML documents could be filtered (piped) and serialized in such a manner, that the application logic (XSP), multilingual support and several output formats (PDF, HTML) were interoperable.

Database objects could be used through the J2EE interface.

The purpose of this server was to provide location based services. Premium information like traffic control was provided to the subscriber, which was connected to the network (GSM) and provided its GPS location.

IT-Environment: Linux, Java, Servlets (Tomcat), Cocoon (XML/XSP/XSL), J2EE (JBoss), Apache, GPS

08/2001 - 10/2001 Software Consulter

Project: Visualization of Organized Structures in 3D

Activities: Designed & implemented a 3D visualization of organized structures, likely organigrams. The hyperbolic 3D space was used, so the objects and their correlated links were represented within a sphere. Interactions via mouse-over, mouse-drag and selection were used to navigate through the structure.

IT-Environment: GNU/Linux, UML, Java, OpenGL For Java (GL4Java), OpenGL, UML(ArgoUML).

05/2001 - 07/2001 EDS, Continental

Project: Client/Server Protocol via XMLRPC

Activities: Designed & implemented a communication solution between a native server application running on AIX incl. DBMS and many Java clients.

XMLRPC was used for this purpose as a lightweight protocol.

XMLRPC was implemented in native ANSI-C++ on the server side as well as in Java for the client side.

IT-Environment: AIX, Java, XMLRPC, UML(ArgoUML).

08/2000 - 01/2001 Siemens AG

Project: Home Location Registers (HLR) - Innovation

Activities: Worked on HLR's DB replication setup and schematas. HLR was part of the new infrastructure for mobile telecommunication.

IT-Environment: Unix (Solaris), UML, C++, Java, TCP/IP, MAP/TCAP/CSS7, RTC.

08/2000 - 08/2000 Campus Part 2k, Valencia, Spain

Project: Speaker: Introduction to the OpenGL(R) language mapping for Java(TM)

Activities: Presented the OpenGL language mapping for Java (GL4Java): Architecture, implementation and its usage.

IT-Environment: GNU/Linux, Windows, Java, OpenGL

07/2000 - 08/2000 Dekra GmbH

Project: Tutoring on the subjects networking and Java

Activities: Taught basics and praxis of client-server network techniques, as well as programming of tcp/ip based protocols in Java.

IT-Environment: Unix (Linux), Java, TCP/IP

06/2000 - 07/2000 Software/Internet Service Provider

Project: Reassurance auction platform in the internet

Activities: Architected & implemented the user session management and other modules of an internet auction platform. Direct insurers were able to offer partial layers of the overall risk to the reassurance companies. The whole handling including customer management (booker, direct insurer and reassurance) was supposed to take place in an HTML interface (Web-browser).

IT-Environment: Application server (Websphere, Apache, XML, Java Servlets), Java, EJB, DB2, PowerDesigner (DB-Design), Win32, Solaris.

06/1999 - 05/2000 IFS GmbH (An RWE Company)

Project: Cheops

Activities: Modeled, implemented and tested theme objects based upon Use Cases. The product discussed the complete energy economy (device management, customer services, etc.). The application was implemented in Java as a client/server solution.

IT-Environment: Java2, Paradigm Plus, PVCS, Win32, Oracle, Framework incl. ODBMSj-
JRBMS Mapper, GUI (View, Controller, Business Object), Java Generator for Paradigm Plus, JDBC, Corba, RMI, Email.

06/2000 - 10/2000 Desys GmbH

Project: 3D Scenegraph-API for Java.

Activities: Implemented a native interface to the scenegraph library. Using the scenegraph API and GL4Java, an application was be able to manage complex 3d structures and access native 3D techniques. Maintenance and adaption of the scenegraph API is also part of this work.

IT-Environment: Unix (Linux), Windows, VRML, OpenGL, GL4Java, Java

05/1999 - 06/1999 Jausoft

Project: [GLMame32](#)

Activities: Implemented a [M.A.M.E. OpenGL](#) display drivers for for unix and windows. This work based on Mike Oliphant's [GLMame XMame](#) . [GLMame32](#) is licensed under the [GNU Library General Public License](#) !

IT-Environment: Unix (GNU/Linux), Windows NT/95, OpenGL, MSVC++ 5.0

02/1999 - 06/1999 Lufthansa Airplus

Project: [Easy Travel Online \(ETO\)](#) der [Lufthansa Airplus](#) .

Activities: My tasks included the support of the C++, IIOP and Java environments, as well as the networking area. I managed to create a solution for the IIOP tunneling via HTTP incl. proxy server and passing a firewall.

This solution was implemented in Java and used the Java-Servlet-API (JSDK). This new middleware was accessed by the ETO clients using HTTP, connecting to the web server within the server domain. The middleware also provided secure transaction via SSL.

IT-Environment: Unix (Aix, GNU/Linux), Windows NT, , JDK 1.1.X, Java2, JSKD 2.1, Apache, Iona Orbix (-Web), Cygwin32, TCP/IP, HTTP, SSL, Firewall, Proxy-Server

01/1999 - 07/1999 Jausoft

Project: [GL4Java](#) Version 2.X (see 04/1997 - 01/1998).

Activities: Designed & implemented a new objectmodel to separate the [OpenGL](#), GLU and window handler objects. Java2 and Netscape were also supported for Win32. GL4Java is licensed under the [GNU Library General Public License](#) !

IT-Environment: Unix (Aix, Solaris, GNU/Linux), Windows NT/95, Macintosh, OpenGL, JDK 1.1.X, Java2, Java-Native-Interface, Cygwin32, TCP/IP, HTTP, GNU-C++, MSVC++ 5.0

11/1998 - 12/1998 Training center

Activities: Tutored in the areas of ANSI-C++, Java and networking. Example projects were used to deepen newly taught theories.

IT-Environment: Unix (GNU/Linux), Windows NT, , GNU-C++, JDK 1.1.X, RCS, SNMP, TCP/IP, Apache, UML

06/1997 - 10/1998 Siemens AG, Munich

Project: Realization of an *upgrade* process in SDL(UML) and C++

Activities: Designed & realized a multithreaded *upgrade* process in SDL(UML) and C++ for the embedded target platform VxWorks/680xx, XpressLink DSL project. The latter pushed new firmware to the home based xDSL modems.

I introduced & enabled independent module development & test, utilizing synthetic input data, which also untangled teams developing modules of the system.

I took part in the black-box review process, documentation and realization of module & (regression) unit tests on the host and target system.

The *emphupgrade* process was developed for the project XpressLink of the [Siemens AG, Munich](#).

IT-Environment: Unix (Sun - Solaris), Windows NT, [SDT 3.11 \(Telelogic SDL\)](#), GNU-C++, ClearCase (ClearTool), VxWorks (Target), SNMP (MIB)

04/1997 - 01/1998 University of Applied Sciences - FH Bielefeld

Project: [GL4Java](#), OpenGL language mapping for Java

Activities: Created a Java extension, exposing OpenGL and GLU natively under Java. The OpenGL interface was implemented using the Java Native Interface (JNI), connecting the Java-Methods to the native OpenGL functions.

I setup a platform independent development environment under Unix (AIX, GNU/Linux) and Windows NT (Cygwin-GNU-Tools, ...) for using Java, ANSI-C, Makefile, etc.

This works started under the supervision of [Prof. Dr. math. Bunse FH-Bielefeld](#) for my diploma

thesis.

GL4Java is available under the URL: <http://www.jausoft.com/gl4java/>.

IT-Environment: Java Developer Kit 1.1.X (JDK), JavaCC, Latex, HTML, Make, RCS, Unix (AIX, GNU/Linux, Solaris), Windows NT, Cygnus-GNU for Win-NT 4.0 (bash, vi, sed, grep, awk, ...)

11/1996 - 04/1997 IBM Frankfurt

Project: Technical Prototype (GUI) for a Call-Center Application

Activities: Created and integrated GUI widgets (objects) plus functional extensions in Java for Sun's Java-WorkShop:

- Auto-Alignment/Styling for all Windows
- Statuslines for all Windows
- Online-, Bubble- and Statushelp, Tabulator-Focus, Cursor-Focus and Hotkeys for all Objects
- Browser/Tree integration of the JMAPI (sun) packages
- Integration of new Java-Applications/Applets at runtime
- Configuration of all features incl. text via runtime parsed config-files also covering national language (NLS) support at runtime. Config DSL written using JavaCC.
- ...

Created prototypes, installed Source Control System (SCCS) on AIX, supported GUI-Team with Java know-how and Web-Publishing.

IT-Environment: Sun's Java WorkShop 1.0, Java Developer Kit 1.02 (JDK), Symantec Visual-Cafe, JavaCC (Java Compiler Compiler), HTML, Netscape (WWW), Make, SCCS, Unix (AIX), Windows NT, Cygnus-GNU for Win-NT 4.0 (bash, vi, sed, grep, awk, ...)

06/1996 - 09/1996 Software Consulter

Project: Platform independent graphical user interfaces and applications.

Activities: Usability test of Tcl/Tk and Java, to provide OS independent applications with a GUI. Examples with multi-threading, sound, animation, tables, menus and system interfaces are created for testing purposes. The Java libraries, (applet, awt, io, ...) as well as the TK-Library were evaluated for this purpose. The development environment was assembled with the JDK 1.02, GNU/Linux and Java Workshop 1.0.

IT-Environment: Unix (GNU/Linux), MS-Windows 95, ANSI-C, ANSI-C++, Tcl/Tk, Sun's Java WorkShop 1.0, Java Developer Kit 1.02 (JDK), HTML, Netscape (WWW), Make, RCS

03/1996 - 09/1996 Duerkopp-Adler AG

Projects: CAD/CAM DIN66025 for 80166-Controller (Sewing machine), OS services for 8051 and 80166 controller-cards.

Activities: Implemented modules in ANSI-C to support OO development inclusive Run-Time-Type-Information (RTTI) and virtual methods for user defined data types. Implementation of a dynamic polymorph list type. All modules were supporting the bare-metal target platform 80166-Controller, KEIL-C compiler.

Created a Unix like file system for a PCMCIA ram-card, and an IPC communication protocol for two 80166-Controller.

Ported the DIN66025 compiler written in C++ to C, which was created for the Windows CAD/CAM application (see earlier project below). This effort allowed the DIN66025 CNC description to be visualized and editable on the 80166-Controller as well.

IT-Environment: Controller hardware 80166 and 8051, MS-Dos, ANSI-C (KEIL compiler for 80166 and 8051 Controller), Unix-Environment for MS-Dos (MKS), RCS, Make, ANSI-C++ (GNU)

03/1996 - 08/1996 Internet provider

Activities: SQL-DB GUI for a web-server. Queries and db-management were offered by a created http/cgi interface. Filters to process the non-uniform data for the db import were realized with Unix scripts (awk/sed/...).

IT-Environment: Unix (GNU/Linux), HTTP-Server (Apache), mSQL (DBMS), ANSI-C, ANSI-C++ (GNU), RCS, Make, awk, sed, grep

08/1995 - 03/1996 Duerkopp-Adler AG

Project: CAD/CAM DIN66025 under MS-Windows (3.1 u. 95)

Activities: Developed a CAD/CAM application. The applications exposed two DIN66025 CNC views simultaneously, one for the graphical and one for the text based representation of the geometric and the technical data. Both views were editable and the data & views synchronized. The text representation was the source data compiled into an OO data structure, capable to be drawn in the graphical 2D view and to be printed into a DIN66025 text file simultaneously.

IT-Environment: MS-Windows/Windows95, ANSI-C++ (MS VisualC++), MFC, Word 5 (Onlinehelp), Lex & Yacc (MKS), RCS (GNU), Make (GNU), Unix env. (MKS/GNU/Linux)

12/1994 - 06/1995 Software distributor

Project: DB interface to AutoCad (visualization and graphical editing) for analysis and manipulation.

Activities: Data sets from a Street- and Drain-DB were visualized and edited with AutoCad.

IT-Environment: MS-Dos, ANSI-C++ (Watcom), dBase DB, AutoCad (MS-Dos), RCS (MKS), Make (MKS), Unix env. (MKS/GNU/Linux)

08/1994 - 12/1994 Industry

Project: C++ Training

Activities: ANSI-C++ Training 1-2 times weekly.

IT-Environment: ANSI-C++, Teaching with MS-Dos and Borland-C++ 4.0

04/1994 - 08/1995 Prekwinkel AG

Projects: CAM tools, compiler construction, CNC-Format conversion, geometry & process CNC editing, DB interface.

Activities: Designed & realized a DSL compiler for custom programming of CNC machines including geometry & process optimizations. Used generic hash-lists incl. quick-sort pass, dereferencing symbolic objects allowed complex high performance scripts. The compiler generated byte-code structured in an Abstract Syntax Tree (AST).

Wrote an CNC crosscompiler based on above DSL compiler. Scripts written for this DSL module processed a universal CNC-Format (FXM) and translated them to machine specific CNC-Formats.

DSL compiler was also used to implement different CNC filters for analysis and geometry enhancements, as well as to optimize machine-cycles.

Demonstrated superior Genetic Algorithm for route & tool-change optimizations.

IT-Environment: MS-Dos, Unix, ANSI-C++ (Borland 4.0, Watcom, GNU-C++), Lex Yacc (MKS), RCS (MKS), Make (MKS), Unix-Env. (MKS/GNU/Linux)

02/1994 - 05/1994 S&P Media

Project: Cross compiler from SET-PR (SDL) to SDT-PR (SDL)

Activities: Developed a Specification-Description-Language (SDL now UML) Cross Compiler from SET-PR to SDT-PR, as well as of SDL-PR to ANSI-C. The task was solved by using LEX, YACC and an ANSI-C Compiler of a SUN- and HP-Workstation.

IT-Environment: Sun- u. HP-Workstation, ANSI-C, LEX u. YACC, Make, Korn-Shell

01/1994 - 03/1994 University of Applied Sciences

Project: Solution- and optimizing system

Activities: Development of an OO Genetic Algorithm to solve P and NP problems. Sample applications *Missionary and cannibals* (game simulation), as well as distance (travel-salesman) optimization.

IT-Environment: Unix, MS-Dos, ANSI-C++ (Borland 4.0), RCS, Make

03/1993 - 07/1993 University of Applied Sciences

Project: Specification and documentation of parallel Processes.

Activities: Translation of SDL-PR to SDL-GR (Graphic Display) for MS-Windows. Development of protocols for ISDN and Ethernet using SDL.

IT-Environment: MS-Windows, Unix, C++ (Borland 3.1 - Windows), ISDN, Ethernet, SDL

06/1992 - 02/1993 University of Applied Sciences

Project: ScanArts - GUI for manipulation of scanned mathematic 2D formulas (described in c't 1993/09).

Activities: Performed graphic programming on MS-DOS and MS-Windows. Implemented general Drag and Drop, Pixel-chain scanning, fill algorithms, drawing of Lines and circles, Matrix rotation and transformation as well as Techniques for reversible graphical editing (undo). Co-author of publication *Mathematic User Interface with Recognition of two-dimensional Formulas*¹. My additions here were the chapters *Graphical Editing of Graphic Formulas* as well as *User Interface for ScanArts to Maple Algebra Program*.

IT-Environment: MS-Dos u. -Windows, ANSI-C (Borland 3.1, BGI), MapleV, Word 5

06/1992 - 09/1992 Industry

Project: Porting a quick-Basic application to Turbo-Basic

Activities: Build translator converting quick-Basic to Turbo-Basic. Implemented own lexical state machine, similar to LEX.

IT-Environment: MS-Dos, ANSI-C, Basic

3 Skills

1. Domains

- 3D, Device Connectivity, Security
- Architecture, design and development

¹ISBN 3-923216-33-5

- Embedded and platform independent development
- High performance and parallel computing
- Project management and maintenance
- Training

2. Industries

- Medical Device Industry
- Automotive Supply Industry
- CAD/CAM - Woodworks and sewing machines
- Economics
- Electronic Industry
- Energy supply
- Engineering
- Graphics, video and multimedia product manufacturer
- Internet provider
- Software development
- Training center

3. Topics

- 3D, CAD/CAM
- Device Connectivity and Security
- Automation techniques
- Application integration (EAI, J2EE, XML, XSLT, SOAP, ..)
- Communication protocols (ISDN, TCP/IP, MOST, and special protocols e.g. IPC)
- Compiler and interpreter construction
- Controller programming ANSI-C(++) and Assembler
- DB applications and interfaces (DB2, SQL, JDBC)
- Graphic User Interfaces (GUI)
- Multimedia systems (Player software, decoder, hardware driver (mpeg2, mp2, pcm, a52))
- Network techniques (DB interfaces for HTML Server, WWW, Proxy, Ftp, News, Email clients, Firewall installation etc.)
- OO-Design (UML, Rational Rose, Paradigm Plus, Java, C++)
- Optimization automates (Genetic algorithms, Neuronal Networks etc.)
- Organizing and planning
- Parallel Computing, Concurrency (SDL / UML, SIMD / MIMD, OpenMP, OpenCL)
- System analysis and design
- System software

- Communication Protocols
- Filesystem
- Graphic Card Driver (XFree86, Proprietary, Console)
- Linux Kernel Development (device driver and porting)
- OpenGL, GL ES, driver development
- QNX Device Driver Development
- Training (C++, Java, Networks)
- Unix administration

4. Hardware

- AIC RS/6000
- ARM 32bit and 64bit
- Bluetooth LE and BREDR (Programming per Specification)
- CNC-Controlling
- Controller (8051, 80166, ARM, SH4)
- DSP's (TMS..)
- Graphic Chips (ATI R3xx-R6xx, Siliconmotion, Fujitsu MB86291)
- HP
- IBM PC + Compatible
- PowerPC 7xx
- SUN Sparc

5. Operating Systems

- Unix (AIX, BSD, HP, GNU/Linux (Debian, Ubuntu, Red Hat, OpenSuSE, Android), QNX, Solaris)
- VxWorks
- MacOS
- MS-Windows (95/98/NT/2000/XP,7), MS-Dos

6. Programming Languages

- C / C++ (STL, C++98, C++11, C++17, C++20)
- Java (1.0.2 - 12.x, J2ME)
- GLSL (OpenGL), OpenCL
- Scripting languages as necessary (Korn, Bash, Perl, ECMCA, ..)
- Assembler (80x86, 68000, 650x)
- ...

7. Development Environment and Tools

- API: OpenGL/GLSL, OpenCL, OpenMP, ODBC/JDBC, pthreads, STL, ...
- Bugs/Requirements: Bugzilla, ..., Doors
- DB: MariaDB/MySQL, DB2, dBase, Oracle, ...
- Compiler: ANTLR, Lex & Yacc (Berkley, MKS, GNU), JavaCC
- Continuous Integration (CI): Jenkins/Hudson, JUnit, CppUnit, ...
- IDE: Eclipse, Netbeans, MS-Visual C++, ...
- OO/UML: Bouml, Rational Rose, Paradigm Plus, Innovator, Telelogic SDT/SDL, ArgoUML, Thorn, Umbrello, ...
- SCM: GIT, Perforce, Remedy, ClearCase, Change Synergy, CM Synergy; SCCS, RCS, CVS

8. Methodology

- Requirement Management
- UML Analysis and Design (OO, Structured, States, ..)
- SDL specification for multiprocessing / multithreading systems
- Top-Down interface specification
- Bottom-Up implementation
- Review and audit procedures
- Object oriented programming (Java, C++ and either C)
- Configuration-management, project maintenance
- Usage and implementation of platform independent standard libraries
- Debugcode with inductive proof
- OO-Design using UML iteratively
- Tests: BlackBox, WhiteBox, Unit, Continuous Integration, Regression
- Extreme- and team-programming, KnowHow Transfer

9. Server Environment

- Apache, NGINX, Samba, Squid, inn(news), ftpd, routed, ipp, iptables, sendmail, VPN, squid, mysql, git, bugzilla, jenkins, ...
- GNU/Linux Server, Windows clients