

A Comparative Study and Critical Analysis of Various Integrated Development Environments of C, C++, and Java Languages for Optimum Development*

Sampada K. Satav

M.Tech.(CSE) Scholar, Dept. of CSE,
Rungta College of Engg. & Tech.,
Bhilai – 490 024(C.G.), INDIA
Email: sampada.satav@gmail.com

Prof. S. K. Satpathy

Head, Dept. of CSE,
Rungta College of Engg. & Tech.,
Bhilai – 490 024(C.G.), INDIA
Email: sks_sarita@yahoo.com

Prof. K. J. Satao

Head, Dept. of IT & MCA,
Rungta College of Engg. & Tech.,
Bhilai – 490 024(C.G.), INDIA
Email: kjsatao@rediffmail.com

*Orally presented in the Conference on IT - Contemporary & Future Technologies for Social Change, hosted by ITM UNIVERSE, Gwalior(M.P.), India in association with Computer Society of India, held during March 11-12, 2011.

Abstract - This paper deals with a comparative study and critical analysis of various Integrated Development Environments of C, C++, and Java languages so that a comprehensive list of specifications could be drawn for the development of an all inclusive and optimum future IDE. An Integrated Development Environment(IDE), also known as an Integrated Design Environment or an Integrated Debugging Environment, is a software application that provides comprehensive facilities to computer users for software development viz. program entry, editing, compilation, debugging, etc. Various IDEs are studied and compared based on their features, and are critically analyzed. The results show that the IDEs which are more popular usually offer better support for users while the more complex ones, which are not so popular, offer only few features than the other IDEs whereas no existing IDE is all inclusive and optimum.

Keywords - Graphical User Interface(GUI); Integrated Development Environment(IDE); Java Virtual Machine(JVM); KDevelop; Qt Creator; Code::Blocks; Dev-C++; CodeLite; Pelles C; Eclipse; BlueJ; NetBeans; JCreator; JDeveloper.

I. INTRODUCTION

An Integrated Development Environment(IDE) is a kind of enhanced editor for a programming language with built in compiler, edit screen, and tracing or debugging utilities, which makes the program development easier. An IDE does several more things such as:

- A user can browse through files to find the one he wants;
- There is “help” available, so that a user can quickly find out what arguments a particular method needs or what an error means;
- A user can see changed values of variables and objects, etc.

IDEs are programs to enter source codes. They are generally editing environments with tools to help programmers write source code quickly and efficiently. As an example, we can create a PHP-driven web applications using a combination of Eclipse and PHP Eclipse.

Core features of an IDE typically include[6]:

- Code insight: This is an ability of an IDE to know a language's keywords and functions. The IDE may use this knowledge to do such things as highlight typographic errors, suggest a list of available functions based on the appropriate situation, or offer a function's definition, and use different colors for keywords and functions.
- Resource management: When creating applications, languages often rely on certain resources, like library or header files, to be at specific location/directory. IDEs manage these resources. An IDE knows the required resources so that errors can be spotted in the development stage, in the compile or build stage.
- Debugging tools: In an IDE, a user is given an opportunity to thoroughly test his application before release. The IDE may be able to give variable values at certain points, connect to different data repositories, or accept different run-time parameters.
- Compile and build: For a language that requires a compile or build stage, IDE translates code from high level language to the object code of the targeted platform. The requirements for this feature vary substantially from language to language.

Thus, traditionally, an IDE specializes in one language or a set of similar languages. Some famous IDEs and their languages include: JBuilder for Java; Metrowerks CodeWarrior suite for Java, C, and C++; and Microsoft's

Visual Studio for its Visual Basic and C# family of languages.

In this paper we shall compare and analyze some existing IDEs on their features.

II. VARIOUS IDEs

There are many IDEs used by normal users and professionals. Some of the IDEs provide the facility of executing the codes for multiple languages like C, C++, QML, Java, Ada, Bash, FORTRAN, Pascal, PERL, PHP, Python, Ruby, JavaScript, Haskell, HTML, CSS, etc. and there are some which support only one or two languages. In this paper, we are dealing with the IDEs for C, C++, and Java languages.

A. IDEs for C and C++ languages

There are many IDEs for C and C++ languages viz. Anjuta, C++ Builder, Code::Blocks, CodeLite, Dev-C++, Eclipse CDT, Geany, GNAT Programming Studio, GNUstep ProjectCenter, KDevelop, LccWin32, MonoDevelop, NetBeans C/C++ pack, Open Windows, OpenWatcom, Pelles C, Qt Creator, Sun Studio Rational Software Architect, Turbo C++ Explorer, Turbo C++ Professional, Ultimate++ TheIDE, Microsoft Visual Studio, Microsoft Visual Studio Express, wxDev-C++, XCode, etc. Comparing and analyzing all these IDEs is a tough job, so, here we are dealing with only a few IDEs. We shall describe their features, compare and analyze them.

1) *KDevelop* : KDevelop[7,9,14] is developed by KDE(K Desktop Environment, K originally stood for Kool) and is useful on Linux system to provide a Windows like appearance to the user. It is written in C++. It uses cross-platform operating system and works on KDE platform. It is a free software. KDE includes no compiler. It uses the external compiler like gcc/g++ to produce executable code. It uses an embedded text editor and the default editor is KDE Advanced Text Editor. This default editor can be replaced with Qt designer-based editor. It includes the features like Source code editor with syntax highlighting and automatic indentation, Class browser, GUI Designer, Front-end for the GNU Compiler Collection and GNU Debugger, Wizards to generate and update class definitions and application framework, Automatic code completion, Built-in Doxygen support. It uses plugin based architecture. It is programming language independent and system-independent. It supports other technologies like Qt, GTK+, and wxWidgets.

2) *Qt Creator* : Qt Creator[22] is developed by Qt Development Frameworks. It is written in C++. It is a cross platform IDE and works only for C++ language. It includes a visual debugger, an integrated GUI layout, and a forms designer. The IDE features include Syntax highlighting, Auto Completion, Code editor for writing a well formatted

code, User Interface designer, etc. It provides a debugger plugin which acts as an interface between the Qt Creator core and external native debugger to debug the C++ language. Qt Creator provides a support for building and running Qt applications for desktop environments(Windows, Linux, FreeBSD, and Mac OS) and mobile devices(Symbian, Maemo, and MeeGo).

3) *Code::Blocks* : Code::Blocks[15, 23] is developed by the Code::Block developer team. It is written in C++. It uses cross platform operating system. It is free and open source IDE. It can be used for both C & C++ languages. It supports multiple compilers, including MinGW/gcc, Digital Mars, Microsoft Visual C++, Watcom, LCC, Borland C++ and the Intel C++ Compiler, etc. The IDE features include Syntax highlighting and Code folding through the use of the Scintilla editor component, C++ code completion and class browser, an integrated to-do list, and a closely integrated debugger frontend which supports GDB(GNU Debugger) and, to some extent, Microsoft console debugger-CDB(Console Debugger). There is an integrated rapid application development plugin for the wxWidgets toolkit, called wxSmith.

4) *Dev-C++* : Dev-C++[11] is developed by Bloodshed Software. It is written in Delphi. It is used in operating systems like Microsoft Windows, Linux(alpha only). Dev-C++ is a free IDE distributed under the GNU General Public License for programming in C and C++. It is bundled with MinGW, a free compiler. The project is hosted by SourceForge. Dev-C++ was originally developed by programmer Colin Laplace. Dev-C++ can also be used in combination with Cygwin or any other gcc-based compiler. One additional aspect of Dev-C++ is its use of DevPaks, packaged extensions on the programming environment with additional libraries, templates, and utilities. DevPaks often contain, but are not limited to, GUI utilities, including popular toolkits such as GTK+, wxWidgets, and FLTK. Users of Dev-C++ can download additional libraries, or packages of code that increase the scope and functionality of Dev-C++, such as graphics, compression, animation, sound support, and many more.

5) *CodeLite* : CodeLite[12] is developed by Eran Ibrah. It is written in C++. It uses cross-platform operating system. It is a free, open source IDE for the C and C++ languages. It uses wxWidget toolkit for programming. The program is compiled and debugged using free tools(MinGW and GDB) for Mac OS X, Windows, Linux, and FreeBSD. It executes any 3rd party compiler or tool that has a command line interface. Its features include Project management(workspace/projects), Code completion, Code refactoring, Source browsing, Syntax highlighting, Subversion integration, cscope integration, Unittest++

integration, an interactive debugger built over GDB, and a source code editor.

6) *Pelles C* : Pelles C[13] is developed by Pelle Orinius. It is for only C language and works only in Windows operating system. It's features include Integrated source code editor with Win32 API call tips, Auto indenting, Integrated source-level debugger, Project management, Inline MASM-compatible assembler(POASM) with Intel syntax and linker (POLINK), Support for delayed loading of DLLs, Most C99 features, Pocket PC, Fully customizable keyboard actions, and syntax highlighting.

B. IDEs for Java language

There are many IDEs for the Java language viz. BlueJ, DrJava, Eclipse JDT, Geany, GreenFoot, IntelliJ IDEA, JBuilder, JCreator, Jdeveloper, jGRASP, KDevelop, MyEclipse, NetBeans, Rational Application Developer, Servoy, Xcode, etc. For comparison and analysis we shall take some of the IDEs listed above.

1) *Eclipse* : Eclipse[5, 8, 20, 21] is written in Java language. Eclipse is a multi-language software development environment comprising of an Integrated Development Environment(IDE) and an extensible plug-in system. The Eclipse SDK(Software Development Kit) includes the Eclipse Java Development Tools(JDT), offering an IDE with a built-in incremental Java compiler and a full model of the Java source files. This allows for advanced refactoring techniques and code analysis. The IDE also makes use of a workspace. Eclipse implements widgets through a widget toolkit for Java called SWT, unlike most Java applications, which use the Java standard Abstract Window Toolkit (AWT) or Swing. Eclipse's user interface also uses an intermediate GUI layer called JFace, which simplifies the construction of applications based on SWT.

2) *BlueJ* : BlueJ[2] is developed by BlueJ team. It's original author is Michael Kolling. It is written in Java language. It uses cross platform operating system and uses Java platform. It is developed mainly for educational purpose, but is suitable for small-scale software development. BlueJ was developed to support the teaching and learning of object-oriented programming. It's features include representation of object orientation, simplicity of the interface, object interaction, the code pad, regression testing, group work support, Flexible extension system, Jar files and applets, translations, etc. Working with the BlueJ environment provides concrete experience for abstract concepts, such as the class/object relationship, object instantiation, method calling, parameter passing, etc. These

abstract concepts are traditionally hard to understand for novice, and providing concrete representations for these is intended to help the learning process. Although it is efficient software, it misses some of the features like live code checking and error detection, suggested correction for warnings/errors, code folding, etc.

3) *NetBeans* : NetBeans[1, 3, 16, 18] is a product of Oracle Corporation. It is written in Java for cross platform operating system. It uses Java SE platform for executing the programs. The NetBeans IDE runs wherever a JVM is installed, including Windows, Mac OS, Linux, and Solaris. The NetBeans platform allows applications to be developed from a set of modular software components called modules. Applications based on the NetBeans platform(including the NetBeans IDE) can be extended by third party developers. The NetBeans IDE is an open-source integrated development environment. NetBeans IDE supports development of all Java application types viz. Java SE including JavaFX, Java ME, EJB, etc. It supports the feature of modularity. Each module provides a well defined function, such as support for the Java language, editing, or support for the CVS(Concurrent Versioning System), and SVN(Sub Version Number). It also includes a profiler which helps developers to find memory leaks and optimize speed. The GUI design tools enable to prototype and design Swing GUIs by dragging and positioning GUI components. The JavaScript editor provides extended support for JavaScript, Ajax, and CSS. JavaScript editor features comprise of syntax highlighting, refactoring, code completion for native objects and functions, generation of JavaScript class skeletons, generation of Ajax callbacks from a template, and automatic browser compatibility checks.

4) *JCreator* : JCreator[4, 10] is a product of Xinox Software. It is written in C++ for Microsoft Windows operating system. JCreator has two editions. The first one is Lite Edition(LE) which is a freeware, and the second one is Pro Edition(Pro). However, both the LE and Pro versions of JCreator run adequately on Linux(using Wine). The features set of the Pro version is comparable to that of other languages IDEs with respect to project management and editing features, but the set lacks advanced features such as automated refactoring, support for common frameworks, etc, which can be found in the dominant Java IDEs such as Eclipse and IntelliJ IDEA. The free LE version further lacks in some features such as code completion that is included in other free IDEs. Unlike the dominant Java IDEs today, JCreator also lacks the level of extensibility through third-party plugins that is common in popular Java IDEs. It includes the features like Custom color schemes, Wrapping around existing projects, Different JDK profiles, Quick code writing via project templates, Easy project viewing with the class browser, Debugging with an intuitive interface, etc.

No command-line prompts are necessary. Wizards help in writing project, quickly and easily. The run-time environment can run application as an applet, in a JUnit environment or in a command-line window. JCreator's IDE does not require a Java Runtime Environment to execute.

5) *JDeveloper* : JDeveloper [19, 26, 27] is an Oracle Corporation product. It is written in Java for cross platform operating system. JDeveloper covers the full development lifecycle including Design, Coding, Debugging, Optimization, Profiling, and Deploying. With JDeveloper, Oracle has aimed to simplify application development by focusing on providing a visual and declarative approach to application development in addition to building an advanced coding environment. JDeveloper integrates with the Oracle Application Development Framework(Oracle ADF) - an end-to-end Java EE-based framework that further simplifies application development. JDeveloper comes in three editions: Java Edition, J2EE Edition, and Studio Edition. Each one offers more features on top of the other and all of them are free. JDeveloper 11g only has two editions: Studio Edition and Java Edition. In JDeveloper 11g, J2EE Edition features are rolled into the Studio Edition. With the Java Edition the features are Java SE 5 Support, Code Editor, Code Navigation, Refactoring, Swing, Unit Testing, Version Control, Audit & Metrics, Debugging, Profiling, Ant Support, XML Support, Open API & Extensions, User Assistance, etc.

III. COMPARISON AND CRITICAL ANALYSIS

The features of the IDEs on which they are compared and analyzed are given below. They include all the features on which the IDEs of C, C++, and Java get evaluated. It is not mandatory that all the features will suit all the IDEs —

- 1) *Platform used* : This tells the operating system(s) supported by a particular IDE which includes Windows, Linux, Solaris, Mac OS X operating system, and the other operating systems.
- 2) *Profiler* : Profiler is a dynamic programming analysis which is used to determine which section of program to optimize to increase its overall speed, decrease its memory requirement or both.
- 3) *Debugger* : Debugger is a program which runs source program in such a way as to let us see every step of program execution. A debugger will let us stop the program while it is running, change the program or program variables, and start the program running again.
- 4) *Static Code Analysis* : Static code analysis[28] is a methodology of detecting errors in program code based on the programmer's reviewing the code marked by him in those places where potential errors may occur.

- 5) *GUI Builder* : A GUI builder is a software development tool that simplifies the creation of GUIs by allowing the designer to arrange widgets using a drag-and-drop WYSIWYG(What You See Is What You Get) editor. It is also known as GUI designer.
- 6) *AutoComplete* : AutoComplete[24] shows a complete list of functions and variables bound to the scope of the object being typed.

IV. ANALYSIS RESULTS

In this paper we are dealing with IDEs of languages viz. C, C++, and Java. The analysis results are given for the IDEs according to their features.

1) *Platform used* : The various operating systems used by the above IDEs are given below :

Table I : Platforms used by the C/C++ IDEs

	Windows	Linux	Mac OS X	Other Platform
KDevelop	Yes	Yes	Yes	Solaris
Qt Creator	Yes	Yes	Yes	Solaris
Code::Blocks	Yes	Yes	Yes	FreeBSD
CodeLite	Yes	Yes	Yes	FreeBSD
Dev-C++	Yes	Yes	No	-
Pelles C	Yes	No	No	Windows CE

We can see from table I that most IDEs support multiple operating systems with the exception of Pelles C which supports Windows and its mobile versions. From this we can say that most of the IDEs support Windows operating system and then the Linux operating system.

Table II : Platforms used by Java IDEs

	Windows	Linux	Mac OS X	Other Platform
BlueJ	Yes	Yes	Yes	Solaris
Eclipse	Yes	Yes	Yes	Solaris
JCreator	Yes	No	No	-
JDeveloper	Yes	Yes	Yes	generic JVM
NetBeans	Yes	Yes	Yes	Solaris

2) *Profiler* : Profilers are usually very complex and have to be accurate to make any kind of difference in practice. As such they are found mostly in the larger, more popular IDEs.

Table III : C/C++ IDEs with/without Profilers

	KDevelop	Qt Creator	Code::Blocks	CodeLite	Dev-C++	Pelles C
Profiler	Yes	No	Yes	No	Yes	No

Table IV : Java IDEs with/without Profilers

	BlueJ	Eclipse JDT	JCreator	JDeveloper	NetBeans
Profiler	No	Yes	No	Yes	Yes

Profiler is a programming tool incorporated in an IDE which tracks the performance of the program. Table III shows that IDEs like KDevelop, Code::Blocks, Dev-C++ have the profiler feature.

3) *Debugger* : A debugger is an essential component for any IDE. Without this the purpose of an IDE remains largely unfulfilled as simple coding can also be done using text editors like Notepad. The debugger is a tool that differentiates an IDE from a command line compiler.

Table V : C/C++ IDEs with Debuggers

	KDevelop	Qt Creator	Code::Blocks	Code Lite	Dev-C++	Pelles C
Debugger	Yes	Yes	Yes	Yes	Yes	Yes

Table VI : Java IDEs with/without Debuggers

	BlueJ	Eclipse JDT	JCreator	JDeveloper	NetBeans
Debugger	Yes	Yes	No	No	Yes

Debuggers are found in all IDEs as they are usually the first component that is worked on. The importance of a good debugger cannot be overlooked. Indeed, the existence and quality of such a tool for a given language and platform can often be the deciding factor in its use, even if another language/platform is better-suited to the task.

4) *Static Code Analysis* : This feature is used to augment the operation of a debugger. More commonly known as adding breakpoints, this is one of the most important features of an IDE. Unfortunately, this also is very complex and requires a massive amount of coding. As such this feature is found only in the major IDEs.

Table VII : IDEs with Static Code Analysis feature

	KDevelop	Qt Creator	Code::Blocks	CodeLite	Dev-C++	Pelles C
Static Code Analysis	No	No	Yes	Yes	No	No

From the table VII, it can be seen that the static code analysis tool feature is found in only two IDEs i.e. Code::Blocks and CodeLite. As the Static analysis tools are usually rather expensive, they demand knowledge in using them and have rather complicated subsystems of setting and suppressing false messages. In return of complicated usage, static code analyzers allow developers to detect a lot of errors on the early steps of developing program code. So this feature increases the value of the IDE and it should be incorporated by the other IDEs also. The main advantage of static code analyzer lies in the possibility of considerable cost saving of defects elimination in a program.

5) *GUI Builder* : GUI builders are becoming more and more commonplace with almost all the major IDEs supporting GUI design. These greatly speed up the coding process. These depend on automatic code generation and conversion to match the code written by the programmer.

These are also fairly complex and while they greatly speed up the development process, they are very large and require a long time to implement. As such these are rapidly being adopted by most IDEs but now they are limited to the more popular ones.

Table VIII : GUI builders for C/C++ IDEs

	KDevelop	Qt Creator	Code::Blocks	Code Lite	Dev-C++	Pelles C
GUI Builder	Yes	Yes	Yes	No	No	Yes

Table IX : GUI Builders for Java IDEs

	BlueJ	Eclipse JDT	JCreator	JDeveloper	NetBeans
GUI Builder	No	No	No	No	Yes

6) *AutoComplete* : This feature is used by IDEs to offer ease of coding to the developers. The feature creates an index of all the available keywords and literals(function names, variable names, etc.) and whenever the developer starts typing a word, this feature provides suggestions as to what the word might be. This reduces complexity greatly and allows the developer to focus on the coding aspect rather than trying to remember the long list of keywords.

Table X : C/C++ IDEs with/without AutoComplete

	KDevelop	Qt Creator	Code::Blocks	CodeLite	Dev-C++	Pelles C
AutoComplete	Yes	No	Yes	Yes	Yes	Yes

Table XI : Java IDEs with/without AutoComplete

	BlueJ	Eclipse JDT	JCreator	JDeveloper	NetBeans
AutoComplete	Yes	No	Yes	No	Yes

The AutoComplete feature is incorporated in the BlueJ, NetBeans, and in the Lite Edition of JCreator IDE.

V. CONCLUSION

This paper has focused on the various features provided by IDEs available in the market. It provides a general outline on how the various features that are provided in these IDEs make development easier and faster. It also provides a summary of the most common features as well as their availability in the various IDEs under study. The development of future/ optimum IDEs has to include as many of the features listed in this paper.

ACKNOWLEDGMENT

The authors wish to express their heartfelt gratitude to Hon'ble Shri Santoshji Rungta, Chairman, GDR Educational Society, Bhilai; Respected Shri Sourabh Rungta, Director Systems, RCET, Bhilai; Respected Shri Sonal Rungta, Director Finance & Administration, RCET, Bhilai; Respected Dr. S. M. Prasanna Kumar, Principal, RCET, Bhilai for providing the facilities for the research and development work and for constant encouragement.

REFERENCES

- [1] "A Brief History of NetBeans", Netbeans.org. <http://www.netbeans.org/about/history.html>.
- [2] Barnes, David J.; Kolling, Michael (September 1, 2008), *Objects First with Java: A Practical Introduction using BlueJ* (4th ed.), Prentice Hall, p.p. 560, ISBN 0136060862.
- [3] Boeck, Heiko (July 01, 2009), *The Definitive Guide to NetBeans Platform* (First ed.), Apress, p.p. 450, ISBN 1430224177. <http://www.apress.com/book/view/1430224177>.
- [4] CodeGear Announces New Developer-focused Release of Award-winning JBuilder 2007, Integrated Development Environment.
- [5] Darryl K. Taft (2005-05-20), "Eclipse: Behind the Name", *eWeek.com*, Ziff Davis Enterprise Holdings. <http://www.eweek.com/c/a/Application-Development/Eclipse-Behind-the-Name>
- [6] Dana Nourie (2005-03-24), "Getting Started with an Integrated Development Environment".
- [7] David Sweet, et al. "KDE 2.0 Development".
- [8] "Eclipse Platform Technical Overview", Object Technology International Inc., February 2003.
- [9] "Finally KDevelop 4.0 final published", 1 May 2010, <http://apaku.wordpress.com/2010/05/01/finally-kdevelop-4-0-final-published>.
- [10] <http://www.jcreator.com>.
- [11] <http://sourceforge.net/projects/dev-cpp/forums/forum/33286/topic/1334635>.
- [12] <http://sourceforge.net/projects/codelite>.
- [13] <http://www.smorgasbordet.com/pellesec>.
- [14] Ivan Ruchkin, Vladimir Prus", "Single-window integrated development environment".
- [15] JN-UG-3028 Code::Blocks IDE User Guide.
- [16] Keegan, Patrick; Champenois, Ludovic; Crawley, Gregory; Hunt, Charlie; Webster, Christopher (May 9, 2006). *NetBeans IDE Field Guide: Developing Desktop, Web, Enterprise, and Mobile Applications* (Second ed.), Prentice Hall, p.p. 424, ISBN 9780132395526.
- [17] Masamitsu, Jon (28 November 2006), "Presenting the Permanent Generation", http://blogs.sun.com/jonthe_collector/entry/presenting_the_permanent_generation.
- [18] Myatt, Adam (February 21, 2008), *Pro Netbeans IDE 6 Rich Client Platform Edition* (First ed.), Apress, p.p. 491, ISBN 1590598954. <http://www.apress.com/book/view/9781590598955>.
- [19] Mills, Duncan; Koletzke, Peter; Roy-Faderman, Avrom (October 20, 2009), *Oracle JDeveloper 11g Handbook: A Guide to Fusion Web Development* (1st ed.), McGraw-Hill, p.p. 928, ISBN 0071602380. <http://www.mhprofessional.com/product.php?isbn=0071602380>.
- [20] "Oracle JDeveloper 11g handbook: a guide to Oracle fusion web development", Duncan Mills, Peter Koletzke, Avrom Roy-Faderman.
- [21] "OSGi — the footings of the foundation of the platform", The Eclipse Foundation. <http://www.eclipse.org/osgi>.
- [22] Peter Galli (2005-11-02), "Moglen: GPL 3.0 Rewrite Drive Is NoDemocracy", *eWeek*. <http://www.eweek.com/article2/0,1895,1881088,00.asp>.
- [23] "Qt Licensing — Qt - A cross-platform application and UI framework", Qt.nokia.com.2009-11-30, <http://qt.nokia.com/products/licensing>.
- [24] Steve McConnell, "Code Complete, 2nd Edition", Microsoft Press, Paperback, 2nd edition, Published June 2004, 914 pages, ISBN: 0-7356-1967-0.
- [25] "The Verification Process", *The Java Virtual Machine Specification*, Sun Microsystems. 1999, http://java.sun.com/docs/books/jvms/second_edition/html/ClassFile.doc.html#9766.

- [26] "Various Licenses and Comments about Them", Free Software Foundation. <http://www.gnu.org/licenses/license-list.html>.
- [27] Vohra, Deepak (April 16, 2008), *Ajax in Oracle JDeveloper* (1st ed.), Springer, p.p. 224, ISBN 3540775951, <http://www.springer.com/computer/programming/book/978-3-540-77595-9>.
- [28] Walter W. Schilling, Jr. and Mansoor Alam , "Integrate static analysis into a software development process".