



A Various Software Refactoring Tools & Its Information

Ganesh B. Regulwar

Assistant Professor, CSE Dept.
BNCOE, Pusad, Maharashtra, India

Dr. R. M. Tugnayat

Principal, SSPACE, Wardha
Maharashtra, India

Abstract: *Task of managing duplicated or “cloned” code has occupied the minds of programmers for the past 50 years. During this time, researchers and practitioners have developed a variety of techniques for removing or avoiding it by employing functions, macros and other programming abstractions. Functional abstraction was designed into early programming languages, such as Fortran and Lisp. Object-oriented programming, originating with Simula-67, has provided further mechanisms for parameterized reuse to avoid duplication. Aspect-oriented programming has allowed cross-cutting duplication to be abstracted. Engineering practices like Refactoring and Extreme Programming have promoted specific methodologies of abstracting duplicated code. In the last decade, a multitude of tools have been developed (both in research and in industry) that help programmers semi-automatically find and refactor existing duplication into functions, macros and methods. Given this long-term commitment to programming abstractions as a solution use “duplicated code” and “cloned code” synonymously to mean two or more multi-line code fragments that are either identical or similar, particularly in their structure. Duplicated code, it stands to reason that there should be little duplication left in practice.*

Keywords: *Duplicate code; reuse to avoid duplication; abstracting duplicated code; Refactoring; techniques for removing or avoiding.*

I. INTRODUCTION

IntelliJ IDEA Community Edition is the open source version of IntelliJ IDEA, a premier IDE (Integrated Development Environment) for Java, Groovy and other programming languages such as Scala or Clojure.

The Community Edition includes:

An Intelligent code editor that has all the smarts for understanding Java, XML and Groovy code Refactorings, code inspections and intentions, super-fast navigation and search

- Testing frameworks integration: JUnit and TestNG
- Build tools support: Ant and Maven
- Popular version control systems integration: CVS, Subversion and Git
- Swing UI designer

IntelliJ IDEA Community Edition can be downloaded at no cost and is open for contributions by community members. The product is built on top of the IntelliJ Platform, which is fully open source.

JetBrains leverages the open source effort on JetBrains.org to build commercial products such as IntelliJ IDEA Ultimate, which includes many Java EE, Web development and deployment-related features that are not available in IntelliJ IDEA Community Edition. JetBrains also builds different specific products such as RubyMine, WebStorm, PhpStorm, PyCharm, AppCode and MPS, which serve the needs of different developer communities. These products are also built on top of the open source IntelliJ Platform. See the complete list of features available in IntelliJ IDEA Community Edition and the differences between IntelliJ IDEA Community Edition and IntelliJ IDEA Ultimate. Open sourcing the IntelliJ Platform and IntelliJ IDEA Community Edition has provoked multiple questions related to the open source concept in general and the JetBrains approach in particular. We've summarized these questions and grouped them by topic, to help you better understand this new development and its subsequent benefits.

Why has JetBrains decided to open source the IntelliJ Platform and IntelliJ IDEA Community Edition?

The answer is simple: we want to significantly grow the number of users of IntelliJ IDEA. Many people know this IDE is great but have not been able to use it because of the cost. The second reason is that we want to see the growth of the ecosystem around IntelliJ IDEA. The availability of the source code and the growth of the user base will make it easier and much more attractive to build more integration for third-party libraries and tools. And finally, the increased level of community participation, in the form of both feedback and code contributions, will help increase the quality of both the open source and commercial products.

What do you think about competition?

We view competition as a very good and healthy thing. We want to create stronger competition for the other open source IDEs in the market. In fact, with this recent change in licensing, all competitors are now nearly on the same playing field. And this is great for users!

Which parts of IntelliJ IDEA are available as open source?

The IntelliJ Platform has been open-sourced completely. The IntelliJ Platform is an IDE platform which provides underlying APIs for all JetBrains IDEs: IntelliJ IDEA, RubyMine, PhpStorm, WebStorm, PyCharm, AppCode, and MPS. As for IntelliJ IDEA itself, only selected parts have been open sourced. For a complete list of features available in the Community Edition, see the comparison matrix for the 2 IntelliJ IDEA editions.

Basically, the main items missing in Community Edition are detailed web and enterprise support features.

Please note that some plug-ins for IntelliJ IDEA Ultimate are also available as open source and will be functional in the Community Edition. Please check individual plug-ins for details.

What license is used for the IntelliJ Platform and IntelliJ IDEA Community Edition, and why was this license chosen?

The Apache 2.0 license was chosen for both the IntelliJ Platform and IntelliJ IDEA Community Edition. For the full text of the license see: <http://www.apache.org/licenses/LICENSE-2.0.html>. We chose the Apache 2.0 license because it is a very open license which lets anyone create both open source and commercial products on top of our platform. The Apache 2.0 license allows developers to mix our code with their code under basically any license, which gives developers great flexibility in how they reuse our code. This enables JetBrains to spread the usage of our platform as well as increase the number of people knowledgeable about IntelliJ APIs.

How will existing customers benefit from open-sourcing the IntelliJ platform and IDEA Community Edition?

Existing customers will benefit greatly from the open source project:

- Wider user base means more plug-ins and more features developed by the community and available for all
- Better quality thanks to community testing and regular quality assurance by JetBrains
- More and earlier community feedback leading to better implementation of features
- More available documentation thanks to community contributions
- More languages and frameworks supported on top of the platform thanks to community contributions
- As more people learn and use the platform and the IDE, it will become easier to find people with a good knowledge of IntelliJ IDEA (both for hiring and for consulting)

For our customers, we see it as a win-win situation. We're not changing the way we build IDEA, so the quality and the number of features will only improve.

What is the pricing for IntelliJ IDEA Ultimate Edition?

To learn about different pricing options (including free licenses for open source projects), please visit the Licensing/Upgrade page of the official JetBrains website.

Is there a book about IntelliJ IDEA?

There's an old book about IntelliJ IDEA 4 and 5. Although the feature set has evolved quite a bit, the concepts are still the same, so you can still benefit from reading this book. Please see: <http://www.manning.com/fields3>

A new book about IntelliJ IDEA is currently under negotiation with the editor, so stay tuned.

Is it possible to use third-party plug-ins in IntelliJ IDEA Community Edition?

The plug-in architectures and APIs are the same for the Community Edition and the Ultimate Edition. Compatibility will depend on the dependencies of the plug-in you want to install. Plug-ins that rely on functionality not available in the Community Edition (for example, JavaScript, CSS or Java EE) will only run on IntelliJ IDEA Ultimate.

Can I take plugins from the IntelliJ IDEA Ultimate distribution and install them in IntelliJ IDEA Community Edition?

No. They will not work.

How much code is currently being developed by JetBrains employees and how much by the community?

Even before going open-source, IntelliJ IDEA had a lot of community contributions. Many features of IntelliJ IDEA began their life as open-source third-party plug-ins, and some of their authors kept working on them after they were bundled into IntelliJ IDEA. As for the core code, almost all of it is being developed by JetBrains employees, and we don't expect that to change in the near future. However, as community members continuously gain trust based on their high-quality contributions, they can gain committer status and get more influence over their respective area of contributions. Thus, we expect community participation to grow over time.

How do I get the IntelliJ IDEA Community Edition sources?

The source files of the Community Edition reside in the public git repository. To get access to the code, please follow the provided instructions.

What is the IntelliJ Platform?

The IntelliJ Platform is a platform for building IDEs, currently used as the foundation for IntelliJ IDEA, RubyMine, PyCharm, WebStorm, PhpStorm, AppCode, MPS and other products currently in development at JetBrains.

The platform provides a comprehensive set of tools for building IDEs, including:

- virtual file system
- UI framework (action system, toolwindows, etc.)
- text editor
- lexing, parsing, abstract syntax trees and other language-specific infrastructure
- frameworks for implementing navigation, code completion, inspections, intentions, refactorings, etc.
- version control integration
- debugger framework
- graphical unit test runner

All of this functionality is not tied to any particular programming language.

Is the IntelliJ Platform a general-purpose platform, suitable for building any desktop applications, similar to Eclipse RCP and the NetBeans Platform?

The IntelliJ Platform is great for building IDEs and other tools geared for developers. It is not, however, a general-purpose platform. We don't recommend building general-purpose desktop applications on top of the platform, since a lot of IDE-related concepts are still present in the platform itself.

Can I build a commercial product on top of the IntelliJ Platform?

Yes, you can, according to the terms of the Apache 2 license. We encourage developers to build both open source and commercial products on top of the platform.

Does the IntelliJ Platform require me to use a specific user interface toolkit?

Yes, the IntelliJ Platform is built on Swing, so your application based on the IntelliJ Platform will be Swing-based as well.

Are there any plans to open-source other JetBrains products?

This definitely is not among our nearest plans. For now, we are going to keep building new affordable products on our platform - and time will show how the situation evolves in the IDE market.

II. REFACTORIT

Makes it possible to continuously identify opportunities and apply efficiency enhancing changes by analysing information from the source code window and/or the class browser. RefactorIT is designed to be a safe, intuitive, fast, and exciting tool to work with - simply:

1. Point the mouse to a context in any navigation or source code window on the screen.
2. Press the right mouse button.
3. Choose from the list of refactoring operations on the context sensitive RefactorIT menu.
4. Sit back and let the RefactorIT do the work. RefactorIT can be installed/used in stand alone mode or seamlessly integrated into Sun™ ONE Studio, Oracle9i JDeveloper, JBuilder, and other development environments.

RefactorIT is being continuously improved to assist developers by enhancing code efficiency as well as alleviating the drudgery of manually changing code and checking references. RefactorIT allows programmers to choose relevant changes in their code that assist functionality and maintainability; thus reducing "book keeping" effort tracking references and allowing them focus their skills on code development.

RefactorIT 2.0 Versions RefactorIT 2.0 is now available for online purchase with three new editions:

1. Personal Edition - includes only basic refactoring functionality but is aggressively priced.
2. Developer Edition - includes the full set of Refactoring, Audits and Metrics excluding only the Command Line Interface (CLI) and Ant integration.
3. Server Edition - includes all of Developer Edition plus CLI and integration with Ant. We recommend the
 - Personal Edition for occasional refactorers.
 - Developer Edition for advanced refactorings and for Audits and Metrics.
 - Server Edition is meant for running nightly reports and statistics.

RefactorIT also offers an Annual Maintenance Package. This guarantees free upgrades to the next major version 3.0 slated for Q1 2005. AMP subscribers also enjoy increased "voting" power when it comes to priorities about developing new features. RefactorIT Personal Edition The RefactorIT Personal Edition includes a subset of refactorings of the Developer Edition. It features elementary and most powerful refactorings:

- Where Used
- Rename
- Move Class
- Go To
- Extract Method
- Introduce Explaining Variable RefactorIT Developer Edition

The RefactorIT Developer Edition includes the full feature-set of Refactoring, Analyzes Tools, Audits and Metrics. Features Speed and flexibility, innovativeness and aesthetics, power and appeal - today's developers and software architects expect a lot from their software development tools. Spontaneity is especially important: we all want to be able to put our plans into action quickly. RefactorIT is the ideal development tool for people who enjoy their freedom - it is the first comprehensive refactoring and code analyses tool that goes everywhere you want to take it - no matter...

- What IDE your team is working with - by choice or force.
- What kind of Java technologies you are developing with.
- How daring your refactoring project may seem.

RefactorIT provides: RefactorIT Manual

- Automatic Refactoring Operations
- Code Searches and Analysis,
- Audits and Corrective Actions,
- Metrics,
- IDE integrations,
- Full JSP support.

RefactorIT Server Edition The RefactorIT Server Edition includes all features of the Developer Edition plus Command Line Interface (CLI) and integration with Ant. The server edition is meant for running reports and statistics on the server. Annual Maintenance Package Subscribe to the Annual Maintenance Package (AMP) at any time to protect your investment for the next major releases. Standard AMP Updates* Y Y Upgrades** N Y E-Mail Support Y Y Priority E-Mail Support N Y Issue Tracking Y Y Online Forums Y Y *) Updates are typically identified by a change in the digit(s) to the right of the decimal digit (i.e. - 1.1, 1.2, 1.3). **) Upgrades are typically identified by a change in the digit(s) to the left of the decimal digit (i.e. - 1.0, 2.0, 3.0).

RefactorIT Manual Getting Started Please note that unlike the Java compiler, RefactorIT does not require your source files to be 100% compileable. The goal is to make RefactorIT a tool used during development, hence it is as tolerant as possible to errors in a source file. Generally, the source code you intend to refactor should be compilable and RefactorIT must know the source path and class path.

III. JREFACTORY TOOL

JRefactory4 is an open source tool provided under the GPL written in Java to allow easy application of semi-automatic refactorings. The tool supports the application of the following refactorings: • Repackage or move class • Rename class • Add an abstract parent class • Add a child class • Removes a class • Push up field • Push down field • Push up method • Push up abstract method • Push down method • Move method JRefactory can be integrated directly into industrial strength IDEs and it also can be used in command line variant. The tool carries through the formatting of the code source (Java) received as entered. On the basis of the supplied Java language grammar, a parse tree representing the code structure is generated. This information is used to generate summaries and metadata from the code. On the basis of this excellent information the graphic user interface (GUI) is created on the basis of UML class diagram [Rumbaugh98]. Template method must not be confused with the C++ template construct, which has a completely different meaning. Refactorings in design level are applied by selecting the target element (field, method, and class) in the UML diagram. Since the appropriate refactoring in the menu is selected, the transformation is automatically applied. Differently, refactorings in code level are applied by the selection of the appropriated code from the IDE screen and the selection of the correspondent refactoring through the standard interface. Both possibilities for interfacing make it possible to deal with refactorings in both design and code levels. In counterpart, the utilization of two different interfaces to apply the refactoring transformations introduces the need to keep both artifacts (class diagram and source code) in sync. Furthermore, functionalities of JRefactory tool provide the ability to format and print both UML class diagrams and Java code. UML class diagrams can be resized and are useful for navigating through lots of code. In particular, the GUI allows zooming (in certain steps), moving classes and changing association lines.

IV. CONCLUSION

Through this paper provide the information of refactoring tools like IntelliJ IDEA, RefactorIT, JREFACTORY TOOL. How it works, what are different functionalities of each refactoring tool. Also provide method how to remove the duplicate code from the existing software.

REFERENCES

- [1] Brenda S. Baker, A Program for Identifying Duplicated Code Computer Science and Statistics: Proc. Symp. On the Interface, page 49--57. (March 1992).
- [2] J. H. Johnson, Substring Matching for Clone Detection and Change Tracking Software Engineering Laboratory September, 1994 "Proceedings of the International Conference on Software Maintenance (ICSM)," Victoria, British Columbia, September 19–23, 1994, (pp. 120–126), Print ISBN: 0-8186-6330-8
- [3] K. Kontogiannis, Evaluation on the Detection of Programming Patterns Using Software Metrics, Reverse Engineering, 1997. Proceedings of the Fourth Working Conference on 6-8 Oct 1997(pp 44 – 54) Print ISBN: 0-8186-8162-4
- [4] B. S. Baker, On finding duplication and near-duplication in large software systems, Reverse Engineering, 1995., Proceedings of 2nd Working Conference on 14-16 Jul 1995 (pp 86 – 95) Print ISBN:0-8186-711-43
- [5] S. Ducasse, M. Rieger, and S. Demeyer , A language independent approach for detecting duplicated code, Software Maintenance, 1999. (ICSM '99) Proceedings. IEEE International Conference on 30 Aug 1999-03 Sep 1999(pp 109 – 118) ISSN :1063-6773,Print ISBN:0-7695-0016-1.

- [7] J. Mayland, C. Leblanc, and E. M. Merlo. "Experiment on the Automatic Detection of Function Clones in a Software System Using Metrics", Proc. IEEE Int'l Conf. on Software Maintenance (ICSM) '96, pp. 244-253, Monterey, California, Nov. 1996
- [8] Marcelo Sant Anna, Lorraine Bier. Clone Detection Using Abstract Syntax Trees, ICSM '98 Proceedings of the International Conference on Software Maintenance, page 368, 1998-03-16, IEEE Computer Society Washington, DC, USA ©1998, ISBN: 0-8186-8779-7
- [9] <http://jrefactory.sourceforge.net> VOL. 5, NO. 8 JOURNAL OF OBJECT TECHNOLOGY 105
- [10] JOURNAL OF OBJECT TECHNOLOGY Online at <http://www.jot.fm>. Published by ETH Zurich, Chair of Software Engineering ©JOT, 2006 Vol. 5, No. 8. November-December 2006, Mariela Cortés, Computer Science Department. State University of Ceará, Av. Paranjana 1700, Itaperi – 60.740-020, Fortaleza, CE, Brazil Marcus Fontoura, Computer Science Department. IBM Almaden Research Center, 650 Harry Road, 8CC/B1, San Jose, CA, 95120, USA. Carlos Lucena, Computer Science Department. PUC-Rio, Rua Marquês de São Vicente, 225 – 22453-900, Rio de Janeiro, RJ, Brazil.