IDEA VS. EXPRESSION: UNPACKING COPYRIGHT CHALLENGES IN COMPUTER PROGRAMS

Vishakha Burnwal, CHRIST (Deemed to be University) Pune Lavasa

ABSTRACT

The idea-expression dichotomy is the most basic principle guiding copyright-when an idea is expressed, and this expression is within the exclusive rights that can be protected, but the idea itself, being unprotectable, forms the basis. This is a really important distinction especially in computer programs where elements such as algorithms and data structures functionally maintain a relationship with their expressible representations-the source code and user interfaces. This paper follows the evolution and application of the notion of the idea-expression dichotomy in the protection of software, from the early jurisprudence to the establishment and development under important legal frameworks such as the U.S. Copyright Act, the TRIPS Agreement, and the EU Software Directive. This paper looks at the problems which arise in line drawing between functional and creative elements in software by reviewing landmark cases like Whelan v. Jaslow and Oracle v. Google. It further looks into the implications that this legal framework was going to have on innovation, detailing a balance between incentivizing creativity as well as preventing monopolies over basic concepts of software. The paper also discusses the challenges emerging technologies like artificial intelligence and open-source software pose to the implementing of the idea-expression dichotomy. The lack of clearer legal guidelines and international harmonization leads to the view that reform in copyright law is necessary as it needs to be effectively adapted to the new developing technological world. Finally, this paper advocates for a balanced approach that encourages innovation while ensuring just usage and competition in the software industry.

Introduction

The dichotomy on idea-expression is the basic tenet of copyright law that separates an idea from its expression. An idea cannot be copyrighted, but the way such an idea may be expressed is entitled to protection. This principle arises from the landmark U.S. case *Baker v. Selden* (1879), which held that the expression of a method was capable of protection, but the idea or system itself would not be protected. The tension seeks to balance the encouragement of innovation with a prohibition against monopolies of abstract concepts in which knowledge is freely available but creative expressions are protected.[1]

The difference, as it concerns computer programs, is essential. Software is both the abstract ideas of algorithms and logical structures and their concrete forms of source code. Protecting only the expression prevents developers from claiming exclusive rights over fundamental programming concepts or functionality, thus encouraging competition and innovation.[2] However, the line between idea and expression in software is often blurred, especially with complex structures like user interfaces and application programming interfaces (APIs). This ambiguity has led to significant legal challenges and disputes, raising questions about where functional ideas end and expressive elements begin.[3]

It discusses some of these challenges, key legal cases and policy considerations that shape the application of the idea-expression dichotomy in software. It analyzes how courts interpret this principle, the difficulties inherent in distinguishing ideas from expressions, and the global variations in legal frameworks. Based on case studies and analysis of emerging trends, the paper aims to unpack complexities and introduce potential reforms that can better align intellectual property laws with the realities of modern software development.

[1] A.B. Cohen, *Copyright Law and the Myth of Objectivity: The Idea-Expression Dichotomy and the Inevitability of Artistic Value Judgments*, 66 Ind. L.J. 175 (1990).
[2] P.G. Spivack, *Does Form Follow Function—The Idea/Expression Dichotomy in Copyright Protection of Computer Software*, 35 UCLA L. Rev. 723 (1987).
[3] J.H. Pilarski, User Interfaces and the Idea-Expression Dichotomy, or, Are the Copyright Laws User Friendly, 15 AIPLA QJ 325 (1987).

Historical and Legal Background

In the history of copyright, the idea-expression dichotomy has long been profoundly hammered in for protection of creative works, yet still promising free accessibility to the core ideas. This doctrine, studied here, takes its origin from a landmark case in U.S. Supreme Court, *Baker v. Selden (1879)*. In this case, Charles Selden wished to copyright a book explaining a system of bookkeeping, including its forms and tables. The court held that the particular way in which Selden described the method in his book was copyrightable, but that the method or system per se is not.[4] This established a bright line that copyright protects expression, not ideas themselves. This principle was designed to prevent the monopolization of ideas so others can establish their own implementations without fear of lawsuits.

The idea-expression distinction is codified within numerous different countries' legal codes and continues to be a cornerstone of current intellectual property rights. In the United States, it is found within the U.S. Copyright Act of 1976, which specifies that copyright protection extends to original works of authorship fixed in a tangible medium of expression but does not cover ideas, procedures, or methods of operation. This distinction ensures that the creative expression of an idea—whether in literature, music, or software—receives protection, but the underlying idea or concept remains open for public use.[5]

Internationally, this dichotomy is reinforced through the Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS), which establishes minimum levels of copyright protection in member countries of the World Trade Organization (WTO). TRIPS, in fact, stipulates that copyright protection shall extend to expressions but shall not extend to any ideas, procedures, methods of operation, or mathematical concepts as such.[6] This global framework has harmonized copyright laws, ensuring a fair and consistent approach to the protection of expressions while still allowing free use of ideas. The Berne Convention for the Protection of Literary and Artistic

[4] Baker v. Selden | 101 U.S. 99 (1879)
[5] E. Samuels, *The Idea-Expression Dichotomy in Copyright Law*, 56 Tenn. L. Rev. 321 (1988).
[6] C.M. Correa, *Intellectual Property Rights, the WTO and Developing Countries: The TRIPS Agreement and Policy Options* (Zed Books 2000).

Works similarly underscores this principle, reinforcing that only the specific form of an idea's expression is protected.[7]

Different jurisdictions interpret and apply the dichotomy in different ways. The European Union, for instance follows a broadly similar approach but frequently places much greater emphasis on the concept of originality. The EU Software Directive (91/250/EEC) explicitly applies the dichotomy in the context of computer programs, stipulating that protection extends to the expression of a programme but that this does not extend to ideas and principles underlying any element of a programme, including the interfaces.[8]

The application of the idea-expression dichotomy to software has been one of the most difficult and contentious questions in copyright law. Software embodies ideas, such as algorithms, functionalities, and data structures, and also expresses them in source code. As software became more integrally interwoven into technological advance, courts and lawmakers had to figure out how much of the binary code containing the expression of those ideas should be protected and how much should remain unprotectable as bare ideas.

The 1986 *Whelan Associates Inc. v. Jaslow Dental Laboratory Inc*[9]. case had the U.S. court advance the idea that a computer program's "structure, sequence, and organization" (SSO) is protectable expression-that is to say, anything that could be so construed would increase the scope of expression. This new, expansionist view was roundly criticized for stunting innovation on the count of over-protecting aspects of functional software. The court did this in *Computer Associates International Inc. v. Altai Inc.*[10] (1992), embracing the "abstraction-filtration-comparison" test, which breaks the software into different levels of abstraction, filters out unprotectable elements such as ideas, functional aspects, and public domain information, and compares the remaining elements to determine whether infringement exists. This test has since been a basis used critically to distinguish between idea and expression in software. Another important case is *Oracle v. Google*, which surrounded the question of whether Google's use of declaring code in Java API constituted copyright infringement. The U.S.

[7] T. Einhorn, T*he Impact of the WTO Agreement on TRIPS (Trade-Related Aspects of Intellectual Property Rights) on EC Law: A Challenge to Regionalism*, 35 Common Mkt. L. Rev. 1129 (1998).
[8] R.E. Carvalho, *The Protection of Software Between Patent and Copyright Law: Comparing the US and EU Regulatory Approaches* (2023).
[9] Whelan Associates Inc. v. Jaslow Dental Laboratory Inc. 609 F. Supp. 1325 (1985)
[10] Computer Associates International Inc. v. Altai Inc. 982 F.2d 693

Supreme Court eventually ruled in favor of Google in 2021, impressing upon the importance of fair use in functional aspects of software.[11] This case reinforced the perpetual tension between protecting expression and ensuring the necessary building blocks for innovation stay open.

Understanding the Idea-Expression Dichotomy in Software

The idea-expression dichotomy in software is a subtle and complex issue of copyright law, which separates abstract ideas from their concrete expressions. In simple words, an idea is the underlying concept, method, or functionality of a program and expression refers to specific code or a design used for translating the idea. This is a crucial distinction to ensure that innovative and unique forms of software are protected as intellectual property while its underlying ideas and functional principles are allowed to be used by others with the ability to better them. This dichotomy is much more difficult to apply to software than it would be in traditional creative works such as literature or art, with computer programs being strongly functional in nature.

In software, the idea can include many things: algorithms, data structures, and even core functionalities. For example, an algorithm that sorts a list of numbers is an idea; it describes a method or process for getting something done. The expression of that idea is simply code written to describe one way to realize that sorting algorithm. This functionality can be achieved in many ways by writing code by different programmers, and it is the specific implementations that have developed the protection of copyright laws. This would allow others to come up with their versions of the algorithm without infringing on someone else's copyrights, creating an innovation-boosting competition.

Landmark cases Several landmark cases demonstrate how courts have struggled in applying the idea-expression dichotomy to software. One of the initial, most impactful cases was *Whelan Associates Inc. v. Jaslow Dental Laboratory Inc.* (1986). In *Whelan v. Jaslow*, Whelan argued that Jaslow had copied the SSO of their dental lab management software. The court held that the SSO actually constituted protectable expression, thereby opening up copyright to more than just literal code but also the structure of a program as a whole. This decision was criticized for

[11] Oracle v. Google 593 U.S. 1 (2021)

potentially stifling innovation by granting overly broad protection to functional elements of software, which are more appropriately considered ideas rather than expressions.[12]

A more refined approach emerged in *Computer Associates International Inc. v. Altai Inc*. (1992). This case introduced the abstraction-filtration-comparison test, a framework designed to differentiate between idea and expression in software. It requires the software to be broken down into different levels of abstraction; then, those elements that would be unprotectable - such as ideas, functional elements, and code in the public domain - are filtered out, and the comparison is done with what's left of the protectable elements to decide if there's infringement.[13] It brought balance and structure to the test, focusing on what is really original and expressive of courts to navigate the complexities in software copyright.

Another significant case demonstrating the problems of the idea-expression dichotomy in software is *Apple Computer, Inc. v. Microsoft Corporation*[14] (1994). Apple claimed that Microsoft had copied the "look and feel" of its Macintosh operating system in formulating Windows. The court finally ruled in favor of Microsoft because it held that a number of the elements Apple sought to protect were functional rather than expressive. This case highlighted the distinction between ideas from functionality and creative expression in software, especially if the applications in question contain GUIs or other design elements that meld functionality with aesthetic choices.

The *Oracle America, Inc. v. Google Inc.* case is another example. In this case, Oracle sued Google for copyright infringement due to its unauthorized copying of Java API declarations for the Android operating system. The central question was whether the structure, sequence, and organization of APIs constituted protectable expression or unprotectable functional ideas. In 2021, the U.S. Supreme Court ruled in favor of Google, emphasizing the importance of fair use in this context and recognizing that APIs, as building blocks for interoperability, often straddle the line between idea and expression. This case especially underscores the dynamic nature of software development and placed a lot of emphasis on the need for legal frameworks to tailor their prescriptions to existing realities in modern technology.[15]

[12] Whelan Associates Inc. v. Jaslow Dental Laboratory Inc. 609 F. Supp. 1325 (1985)
[13] Computer Associates International Inc. v. Altai Inc. 982 F.2d 693
[14] Apple Computer, Inc. v. Microsoft Corp., 799 F. Supp. 1006
[15] Oracle v. Google 593 U.S. 1 (2021)

While programming is inherently functional, it further complicates the distinction between idea and expression. Ideas in novel or paintings do not offer purposeful functions, which thusly renders distinctions between creative expression and utilitarian purposes of expression much more difficult to separate. For example, user interfaces may be found to include not just functional elements (like buttons and menus) but creative elements (like layout and design). But this will depend on careful analysis and a nuanced understanding of both the law and the technology.[16]

Increasingly, the speed of the pace of technical evolution puts further challenges to the issues. Innovations in artificial intelligence and machine learning continue to blur the lines because a clear human author or creative intent may not be evident in AI-generated code. Similarly, the further development of open-source software blurs the lines drawn by traditional notions of ownership and protection because developers often collaborate and share the same code across different projects and platforms[17].

Legal and Policy Challenges

The applicability of this idea-expression dichotomy in software law throws up considerable legal and policy challenges due to inherent difficulties in distinguishing functional ideas and creative expression. The areas such as user interfaces, application programming interfaces, and other components of software in which these lines are often blurred create legal uncertainty, impacting the developers and the broader software industry.

Overlap in functionality and creativity in software is a major challenge because practical works, such as software, create a challenge between separable protectable expression and what should be left unprotected or ideas or methods of operation. User interfaces (UIs), for example, are created to make user interaction easier; although some aspects of a UI-its color scheme or layout design, for example-may depend on choices that are creative in nature, much of the structure of

[16] S.R. Englund, *Idea, Process, or Protected Expression?: Determining the Scope of Copyright Protection of the Structure of Computer Programs*, 88 Mich. L. Rev. 866 (1990).
[17] D.J. Fetterman, *The Scope of Copyright Protection for Computer Programs: Exploring the Idea/Expression Dichotomy*, 43 Wash. & Lee L. Rev. 1373 (1986).

the interface will be determined by considerations of functionality. Courts need to decide what elements are purely functional and what have enough creativity to be protected. The analysis is highly subjective and often inconsistent and there is an unpredictable legal outcome.[18]

Similarly, APIs pose a seemingly intricate challenge for the idea-expression dichotomy. APIs enable various software programs to talk with each other-essential building blocks for interoperability and innovation.[19] However, the legal status of APIs has been contentious. In Oracle America, Inc. v. Google Inc., for example, Oracle had claimed that Google's use of Java API declarations in the Android operating system would serve as copyright infringement. In fact, the U.S. Supreme Court vindicated Google in 2021 by underlining the need for fair use and the fact that APIs function rather than copy. This judgment has rekindled debate over how far, if at all, a monopoly over critical building blocks should be tolerated, or on the other hand, how much innovation through software should be protected[20]. The ruling came as a boon to developers but underlined debates regarding the extent of copyright protection in the software arena.

International Variation in Interpretation

The interpretation and enforcement of the idea-expression dichotomy vary significantly across jurisdictions, adding another layer of complexity. In the United States, courts generally follow the abstraction-filtration-comparison test established in Computer Associates v. Altai to differentiate between idea and expression. This type of test involves breaking down the software into levels of abstraction, filtering out the unprotectable elements such as ideas, algorithms, and standard practices, and comparing the remaining expressive elements for evidence of infringement. The approach of the United States tends to rely more on functionality and set about preventing monopolies over essential software concepts.[21]

[18] P. Samuelson & R.J. Glushko, *Comparing the Views of Lawyers and User Interface Designers on the Software Copyright Look and Feel Lawsuits*, 30 Jurimetrics J. 121 (1989).
[19] P. Singh & A. Siwal, *Protection of Application Programming Interfaces and the Idea-Expression Dichotomy: The Google-Oracle Dispute Through a Competition Law and Economics Perspective*, 29 J. Intell. Prop. Rts. 569 (2024).
[20] A.D.N.A.N. El-Nasan, Answering Question One in Google v. Oracle: The Creativity of Computer Programmers.
[21] D.S. Karjala, *Copyright Protection of Computer Software in the United States and Japan*, 13 Eur. Intell. Prop. Rev. 231 (1991).

By contrast, the European Union puts a much heavier emphasis on the concept of originality. The EU Software Directive (91/250/EEC) clearly states that copyright protection extends only to the expression of a computer program but not to its ideas and principles. European courts, however often take a more liberal view on what must be classified as expression, with certain cases dealing with graphical user interfaces and other aspects of design having created significant repercussions.[22] For instance, although a certain feature or design could be considered protectable expression under the EU legal framework, it could be held to be a functional idea under the U.S. law, thus creating uncertainty in the development and business levels.

Impact on Innovation

Incentivizing creativity, innovation, and investment in the software industry is related to several uncertainties and inconsistencies in the application of the idea-expression dichotomy. On one hand, robust copyright protection is quite important for the encouragement of creativity and investment. Developers and companies must be assured that their unique implementations and creative works would be protected against unauthorized copying. Otherwise, there would be little incentive to invest in time and resources in the development of new software.[23]

Overly broad copyright protection can stifle innovation by placing barriers to entry and restricting the availability of fundamental ideas and building blocks. If the functional aspects of software-software APIs, for example, or basic algorithms-were treated as if they were protected by copyrights, this could have the effect of preventing other developers from creating compatible or improved versions of the same software, thereby stifling competition and technological advance. But this case, Oracle v. Google, brought home this message: that such tools, if essential, should be made widely available.[24]

Another relevant challenge is the rapid velocity of technological advancement. New technologies, such as artificial intelligence (AI) and machine learning, raise new questions about

[22] J. Reinbothe, *Commentary on the Implementation and Effects of Directive 91/250/EEC on the Legal Protections of Computer Programs*, 6 Int'l Intell. Prop. L. & Pol'y 1 (2001).
[23] A. Usher, *Incentivizing Technological Growth: A Symbiotic Relationship in the Computer Software Industry* (2010).
[24]C. Price, *Google v. Oracle: The Recent Supreme Court Decision, How It Highlights the Inadequacies of Shoehorning New Technology into Intellectual Property Law, and Possible Solutions*, 32 DePaul J. Art & Tech. & Intell. Prop. L. 93 (2022).

authorship, originality, and the scope of copyright protection. At the same time, as AI machines are able to produce code, the traditional notion of human authorship is challenged, and creating the environment for the application of the idea-expression dichotomy is complicated.[25]

Current Trends and Future Implications

The technological landscape is continually evolving, bringing new challenges to the idea-expression dichotomy, especially with the growth of emerging technologies, including artificial intelligence and the increasing importance of open-source software. These developments blur the lines established between ideas and expressions and create new problems regarding authorship, originality, and the scope of copyright.

The code generated by AI is particularly disruptive. A fundamental question arises regarding authorship where traditionally, as in software production, the level of humanity contributes to what is finally produced. AI systems can now autonomously generate code themselves based on some input data or natural language prompts. This may raise questions about authorship such as can AI-generated code be copyrighted, and if so, who owns the rights? Today, copyright is rooted on the premise of human authorship, making it ambiguous over work created by machines. Determining whether AI-generated code constitutes an original expression or merely a functional implementation of an idea is especially difficult. Courts and policymakers will have to wrestle with those questions, perhaps redefining concepts of originality and authorship to fit the facts of AI-driven development.[26]

The open-source movement presents another challenge to the traditional dichotomy: open-source software encourages collaboration and sharing, often blurring the lines between individual contributions and collective works. OPEN SOURCE introduces complexities with regard to ownership and copyright as it is designed such that software should always be freely accessible. For instance, when many developers work on a project, each contributing their own code to

[25] C.J. Craig, *The AI-Copyright Challenge: Tech-Neutrality, Authorship, and the Public Interest*, in Research Handbook on Intellectual Property and Artificial Intelligence 134-55 (Edward Elgar Publishing 2022).
[26] R.M. Ballardini, K. He & T. Roos, *AI-Generated Content: Authorship and Inventorship in the Age of Artificial Intelligence*, in Online Distribution of Content in the EU 117-35 (Edward Elgar Publishing 2019).

implement ideas shared among them, the difficulty in distinguishing between an idea and its expression grows greater. This openness and collaborative nature immediately raise issues of whether copyright can even be enforced-or how contributors can claim ownership over certain aspects of the code.[27]

Policy Proposals

As these are truly modern challenges, there are a range of policy reforms and clarifications required to address them. One critical recommendation would be an evolution of legal definitions of "idea" and "expression" under the rubric of software. Clearer guidelines on what constitutes a protectable expression and what is not, especially regarding functional elements like APIs or user interfaces, would reduce ambiguity and could lead to more uniform legal results. Specific provisions regarding AI-generated works may also cut short questions about authorship and ownership, making copyright law speak properly to the machine-generated content of this new technological age.[28]

Another significant reform is to be made in fair use doctrine or equivalent protections available under other jurisdictions' laws. Fair use provides for limited, transformative use of copyright materials without permission for purposes such as research, education or innovation. Extending fair use protections of functional software elements, like APIs, can be significantly beneficial in protecting developers' rights while promoting technological advancement.[29]

International Harmonization

Harmonization of global legal standards is crucial to effectively tackle the challenges facing such new technologies and to consistently apply the same understanding of the idea-expression dichotomy. Different interpretations of copyright law among jurisdictions create confusion, especially among global software companies. While functionality is heavily emphasized under the analysis of U.S. copyright law, the European Union attaches greater emphasis on originality

[27]N.T. Horne, Open Source Software Licensing: Using Copyright Law to Encourage Free Use, 17 Ga. St. UL Rev. 863 (2000).
[28] E. Rosati, The Idea/Expression Dichotomy: Friend or Foe?, in Handbook on the Economics of Copyright 51-76 (Edward Elgar Publishing 2014).
[29] P. Kimani, Interpreting the Idea/Expression Dichotomy for Enhanced Creativity in the Information Age, 27 Intell. Prop. & Tech. LJ 101 (2022).

and creative expression. These differences may sometimes result in conflicting law outcomes and restrict cross-border innovation.[30]

For instance, international harmonization efforts such as amendments in the TRIPS Agreement or the Berne Convention can promote harmonization between systems. Moreover, defining common standards for distinguishing ideas and expressions in software, to name a few of the emerging issues in this area such as AI-generated content, would provide more legal certainty for developers and companies operating in various jurisdictions.[31]

Conclusion

The idea-expression dichotomy lies at the heart of copyright law, especially where software is concerned. This dichotomy presents the distinction between the ideas underlying the software and the expressions found within the code that actually implements the ideas. While this distinction aims to foster innovation by preventing monopolies on basic concepts, its application in the realm of software has proven complex, particularly with the rise of emerging technologies like artificial intelligence (AI) and the growth of open-source software. For example, landmark cases like Whelan v. Jaslow, Apple v. Microsoft, and Oracle v. Google have highlighted the difficulty that courts experience in trying to reconcile such a dichotomy, particularly with respect to functional elements of a product, such as APIs, user interfaces, or algorithms. These cases reveal the tension between protection of creative works and availability of essential tools for innovation.

With the new evolution of the software industry, so too do law implications that guide it. With the rising use of AI-generated code and open-source software, the collaborative approach asks new questions about ownership as well as the scope of rights into the traditional understanding of copyright protection and authorship. To address these concerns, policymakers may need to define more clearly what constitutes an idea versus an expression in the software arena and insert provisions for work produced by AI. Reforms that focus on substantive improvements in the fair

[30] C.A. Camarce, *Harmonization of International Copyright Protection in the Internet Age*, 19 Pac. McGeorge Global Bus. & Dev. LJ 435 (2006).
[31] T. Margoni, *The Harmonisation of EU Copyright Law: The Originality Standard*, in Global Governance of Intellectual Property in the 21st Century: Reflecting Policy Through Change 85-105 (2016).

use doctrine-including its application to functional elements such as API access-will likely find a better balance between protection and innovation.

International harmonization of copyright laws is also important because there may be differences in jurisdictions that can have an effect on U.S. developers, for instance, and European Union developers creating cross-border issues. A clearer uniform global policy would increase legal consistency and facilitate more innovation across borders. It is obvious, then, that copyright should further evolve in relation to technological changes because, between protection of innovation and the prevention of monopolies over ideas, the real challenge is to find a perfect balance between these extremes in order to definitely not let innovation among creators and developers be stifled while at the same time not letting competition and access be strangled. Future research should focus on the implications of AI in software development, the consequences of open-source licensing on copyright enforcement, and the possibility of developing international frameworks that address such emerging issues. These topics will be significant for the development of future software law as long as they continue to allow for a vibrant, innovative, and dynamic digital economy to thrive.