APPLICATION OF DOCTRINE OF MERGER ON SYSTEM USER INTERFACE

Siddhant Sehrawat, O.P. Jindal Global Law School

ABSTRACT:

The merger doctrine is a fundamental concept in copyright law that addresses the intersection where an idea and its expression are so closely intertwined that separating them would render the idea unexpressed. This doctrine plays a crucial role in the field of software development, particularly in the design and implementation of system user interfaces (UIs). This paper explores the application of the merger doctrine to UIs, emphasizing its significance for intellectual property rights and technological innovation. The theoretical foundation of the idea-expression dichotomy is examined, and how the merger doctrine serves as an exception to this principle, ensuring that when an idea can only be expressed in a limited number of ways, those expressions cannot be monopolized through copyright protection. The implications of the merger doctrine for UIs are significant, as certain elements of a UI are essential for its functionality. If these functional elements were protected by copyright, it would hinder the development of new software and stifle innovation. By preventing the monopolization of essential functional elements, the merger doctrine promotes a competitive and innovative environment in the software industry. This paper argues that the merger doctrine is vital for maintaining a balance between protecting intellectual property and fostering innovation. It ensures that fundamental building blocks for technological advancement remain accessible while allowing creators to protect their unique contributions. The ongoing evolution of the merger doctrine in response to technological advancements is crucial for adapting copyright law to the modern digital landscape, thereby promoting both innovation and creativity.

The theory of idea-expression dichotomy is an age-old concept that has prevailed in Intellectual Property Rights laws all over the world. It is a fundamental principle of copyright law that shows what can be protected by copyright laws. This is vital because its very important to know what is original and worthy of being protected. However, there are certain exceptions that explain as to what will not fall under the protection of copyright laws as well. The doctrine of merger is an exception to the dichotomy theory. When there are circumstances where there are no differences between an idea and its expression, the merger doctrine would infer that a copyright won't be applicable even to the expression of the idea. In recent times, it has become increasingly difficult for courts to assimilate modern era technology into this doctrine. So, **This paper aims to explore the application of the merger doctrine in cases that include system user interfaces by relying upon previous cases in India and abroad.**

There were no rules regarding what constitutes infringement in India until the case of *R.G. Anand v. M/s Delux Films*.[1] It said that there would be no infringement if the same idea is expressed in a different manner (Idea- Expression Dichotomy theory). The merger doctrine mainly comes to play in scenarios where the expression is intrinsically linked to the idea. In the case of *Mattel Inc. v. Jayant Agarwalla*, Indian courts included the doctrine of merger while deciding this case and said that when the idea can only be expressed through the one expression, or vice versa, copyright infringement will not be provided because it would create a monopoly on the idea in its entirety.[2] This is how the doctrine entered in India's jurisdiction.

We shall take the example of a very famous case that involved the elements of doctrine of merger in software user interface, *Google v. Oracle*.[3] This began in 2010 when Oracle had bought the company that owned Java, which made them the new owners of Java. They then sued Google for copyright infringement because google used 37 Application Programming Interface (Herein referred to as API) and wrote their own implementing code to create the software system Android.[4] This led to the majority of Oracle's customers leaving because Android was free while Java was licensed, and the remaining asking for discounts. Soon after, Oracle sued Google in a Californian district court, which ruled that the APIs are not copyrightable but could not whether it was within ambit of fair use. The federal court then

[1] *R.G Anand v. M/S. Delux Films & Ors* 1978 AIR 1613.
[2] *Mattel, Inc. & Ors. vs Mr. Jayant Agarwalla & Ors.* 2008 (153) DLT 548.
[3] *Google LLC v. Oracle America, Inc.* 141 S. Ct. 1183.
[4] Caballar RD, 'Google v. Oracle Explained: The Fight for Interoperable Software' (IEEE Spectrum, 24 June 2021) <https://spectrum.ieee.org/google-v-oracle-explained-supreme-court-news-apis-software> accessed 22 September 2023

reversed the judgement by saying that APIs are also entitled to protection under the copyright act, and google was required to purchase the license from the owning company. When the federal court sent back the case to the district court to discuss the fair use question, the jury ruled in favour of Google. The federal court once again reversed this by saying that Google did not meet all the requirements for fair use.

Google had argued that there was no infringement on Java since the doctrine of merger under its ambit, because the codes that were used from Java were declaring codes and they were the ONLY declaring codes that would be able to perform a function they wanted. The argument used was that they only borrowed the declaring functions that were essential to develop Android and not past that extent, and even after that they input their own code. Oracle had countered this by saying that they had actually offered google several agreements to license the usage of Java SE. They also claimed that Google had abundant resources to create their own programming platform from scratch for its developers. Another contention was that the "necessary" part that Google should have used in Java would have been significantly lesser. Google also argues that since they used the "minimally expressive" declaring codes required, they did not commit infringement.[5] My perspective on this is that the Federal court are not necessarily in consonance with copyright law, since the APIs/ declaring codes are, after all, functional elements. So, if Oracle would have full control over the use of a functional element of programming, then it would lead to a massive monopoly, which means that it would violate copyright law. Another argument that can be given to support Google is that if the fair use doctrine is enforced so extensively, then developers would have an extremely difficult time to develop anything. This is because software is not like literature or cinema, where any idea can be used for expression. This would stop innovation by developers since they would be extremely constrained. On the other hand, it can also be said that if copyright protection is not given, it would disincentivise developers to innovate since no company would like to invest money in development because they won't receive any profit for the same. However, while going to legal discussion, Google had argued that the only way that they could perform programming functions was, by borrowing Oracles declaring codes. This essentially means that it would come under the ambit of the doctrine of merger. Oracle had said that the doctrine would not be applicable because there were several other ways for Google to express their idea.

[5] Shannan T and Kingsbury A, 'Google LLC v. Oracle America, Inc..' (*Legal Information Institute*, 1 July 2022) <https://www.law.cornell.edu/supct/cert/18-956> accessed 23 September 2023

In the end, Google won the case by using the merger doctrine and the doctrine of fair use. So, this is a real life example of the doctrine of merger in software user interfaces.

While looking at the implication of the doctrine of merger on user interfaces, we can see that certain elements of an interface are absolutely necessary in order for the interface to function properly. If these aforementioned elements are protected by a copyright, it would stop people from using the most basic and functional of elements to build complex UIs, hence stifling innovation. Scholars and several courts have seen that the doctrine of merger is extremely important in order to maintain a balance between intellectual property rights and encouraging innovation. It plays a pivotal rule in ensuring that a creator can protect their unique inventions while the basic elements, that may be considered as building blocks for the advancement of technology, remains available for use by others.

Another takeaway from all these observations is that the doctrine of merger works both ways, it has a very important role to protect the ideas of people, but it also has a lot of challenges to face when in cases of system user interface. The doctrine requires the combination of idea and expression, but then the function and creativity of UI will be nearly undifferentiable. So, in conclusion, the exploration of the doctrine of merger in software user interfaces is one that will still take time to properly solidify, and copyright law will continue to evolve with the enhancements of system user interfaces. It is very important that the user experience is balanced well with encouraging people to innovate. It is clear that the doctrine will prevail to play an essential role in development of technology, fostering an environment that ensures the availability of the resources that are required to encourage and cause innovation, while maintaining the protection given to a copyright. This shows the interplay between copyright law and technology in the modern era, with the evolution of technology, the application of the doctrine of merger also evolves to adapt to the new challenges.