Copyright protection of computer programs in the European Union after three decades

The European rules concerning copyright protection of computer programs have not changed since the early 1990s. In the meantime, the software itself evolved. It used to be a distinct product purchased and installed on computers. In the early 2020s, computer programs are everywhere as parts of our everyday devices. Moreover, nowadays programs are not developed the way they used to be. Not only more people are involved in development now than decades ago, but also – more recently – the development generative artificial intelligence allows to, at least to some degree, replace humans in writing code.

In the late 1980s, a decision was made that copyright should be the "go-to" regime of protection in Europe. Consequently, computer programs are treated as literary works. Other developed economies made the same decision. As expected, this led to problems with defining what precisely copyright protects in a computer program. For example, when can one software developer make a program that is, in general, similar to another, but without copying any of its code? When is an owner of a device allowed to modify it by changing the program that runs on such a device? These problems were not resolved in a satisfactory way either when the European directive on the legal protection of computer programs was first enacted or during the subsequent three decades.

Traditional copyright research focuses on strictly legal analysis. However, a more empirical approach is becoming more popular. So far, even when scholars incorporated computer science concepts in their copyright research, they were focused on academic examples and textbook approaches. As a result, the actual practices of developers were neglected. This is a gap that a qualitative study can fill. The research in this project will rely on data obtained from first-hand observation, interviews, and documents. The goal is to improve the understanding of the prerequisites and scope of copyright protection of software by collecting and analysing empirical evidence.

The first aim of the study is to help understand when constraints stemming from technical or functional considerations are sufficient to conclude that copyright protection should not apply. What are scènes à faire – or widely accepted standards – when it comes to software development. Can we really show conventions, standards or design patterns that, for computer programs, work in the same way as standard motives and cliches work for traditional works (e.g. cowboys and shooters with revolvers in western movies)?

Secondly, the study will deal with code-generating AI tools, such as AlphaCode, GPT, Codex or Code Llama, that are already affecting software development. We already know that developers use them to increase efficiency or to explore the options when unsure what to do next. The question remains what these novel approaches to software development mean for copyright protection. Where do generative AI tools leave space for free and creative choices in software development? Is this freedom actually exercised? What creative decisions are made at the design and final "redaction" stages? In short: will the "end of programming" proclaimed by some computer scientists make copyright protection of software obsolete, or merely shift the moment of making creative choices that lead to copyright protection?

The third problem that the study will try to solve concerns the right to modify a computer program. Economic copyright to a program includes an exclusive right to alter the program. There is no doubt that this "monopoly" includes changes in the source code or object code. What about other elements? How should we treat changes in configuration, user-defined data structures or database structures? Such modification will influence the behaviour of the program. However, it is not clear to what extent interfering with these elements must be authorized by the copyright holder. This issue has significant implications even for everyday electronic devices or vehicles, which nowadays are universally equipped with software.