Kluwer Copyright Blog

Prompts as code?

Nuno Sousa e Silva (Universidade Católica Portuguesa) · Tuesday, November 5th, 2024

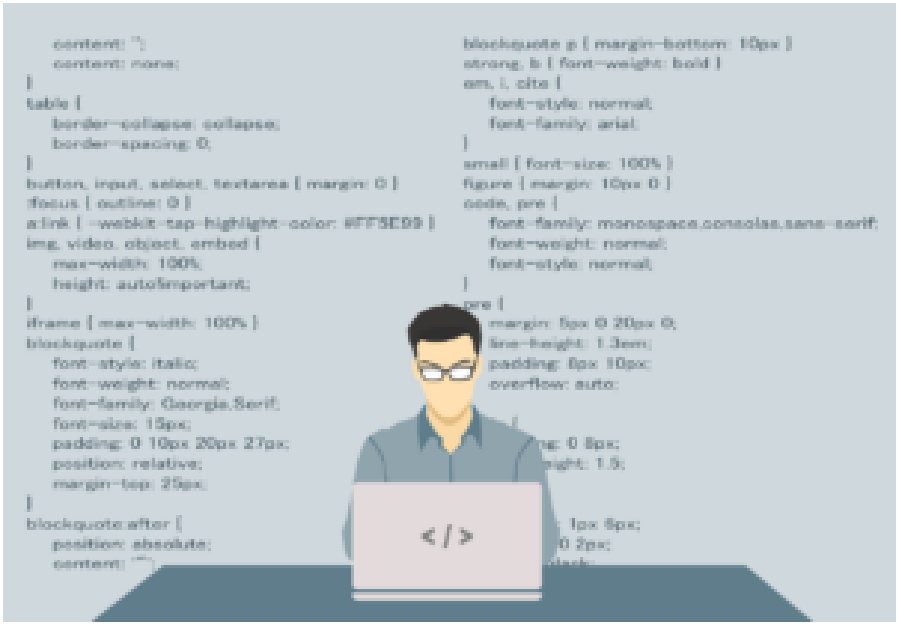

Image by 200 Degrees from Pixabay

Code as a literary work

Following lengthy discussion in the 1970s and 1980s, by 1991 in the EU and 1994 at the WTO level, the legal status of computer programs was a settled matter: software was to be treated under copyright as a literary work.

Source code and object code are protected by copyright. As established in the seminal case C-406/10, SAS: "*…the source code and the object code of a computer program are forms of expression thereof which, consequently, are entitled to be protected by copyright as computer programs, by virtue of Article 1(2) of Directive 91/250*"(§38) and "*neither the functionality of a computer program nor the programming language and the format of data files used in a computer program in order to exploit certain of*

its functions constitute a form of expression of that program for the purposes of Article 1(2) of Directive 91/250." (§39).

AI-generated output and the role of prompts

In the 2020s, we are once again faced with a thorny issue involving computers and copyright: what is the legal status of AI creations?

There is an important distinction to be made between AI-assisted and AI-created output. Using WIPO's words:

> "*AI-generated" and "generated autonomously by AI" are terms that are used interchangeably and refer to the generation of an output by AI without human intervention. In this scenario, AI can change its behavior during operation to respond to unanticipated information or events. This is to be distinguished from "AI-assisted" outputs that are generated with material human intervention and/or direction.*

It seems clear that AI-assisted output is protectable by copyright (see e.g. Hugenholtz&Quintais).

However, even with AI-created works, machines are not creating of their own volition. The ignition or initial input comes from a human. Those instructions for the computer to perform a function, such as generating text, image, video, or code, are given not in a traditional formal programming language such as Python, Java or C. Rather, the users now "control" the computer using natural language, the so-called "prompts". Prompt Engineering is defined in the Stable Diffusion Prompt book as "*the process of structuring words that can be interpreted and understood by a text-to-image model. Think of it as the language you need to speak in order to tell an AI model what to draw.*"

Are prompts code?

Not all prompts are the same. Some are simple questions and look like a typical query in a search engine ("What is the capital of Myanmar?"), others are very simple ("cat drinking cocktail" – here), while a subset are particularly complex. For example, prompt #91: "*Chinese illustration on a blue background, auspicious clouds, 16k, 3d, pastel, ravine stream, boat, pine tree, mountain range, Many houses, Many ancient buildings, multi-dimensional paper quilling, Tunnel composition, warm light, UHD, elevation perspective? ultra-realistic, Morandi color, exquisite details, an expansive view of, epic detail*" generated the image you can see here. Users can also add numbers, certain words, and characters (called parameters) to better control the outcome of the generation process.

Some of the more complex prompts would probably be protectable as a "normal" literary text, but it is also true that these are instructions for a generative AI system, i.e., a computer, to perform a function. Advanced prompts often involve specific syntax, parameters, and even logical operators

to refine the AI's output. This structured approach mirrors the logic and organization found in traditional programming languages.

It is true that when referring to code, the Computer Programs Directive was thinking of a "classic" programming language, i.e., a system of notation using formal rules to provide instructions to computers. However, is there any obstacle to considering these prompts as a programming language? After all, recital 7 of the EU Directive states that "computer programs shall include programs *in any form*, including those which are incorporated into hardware." The choice not to define computer programs was intended precisely to futureproof the law.

At the EU level, there is no definition of a computer program. US copyright law defines it as "*a set of statements or instructions to be used directly or indirectly in a computer in order to bring about a certain result*." To quote a classic work of engineering: "*Computational processes are abstract beings that inhabit computers. As they evolve, processes manipulate other abstract things called data. The evolution of a process is directed by a pattern of rules called a program. People create programs to direct processes. In effect, we conjure the spirits of the computer with our spells*." Prompts seem to be those kind of spells.

Possible objections

Traditional code is deterministic; code will always behave the same way – for the same input, one can expect to receive the same output. This generates some challenges of its own for the copyright protection of software, given that as the Court put it in C-393/09, *BSA*, §49: "…*where the expression of those components is dictated by their technical function, the criterion of originality is not met, since the different methods of implementing an idea are so limited that the idea and the expression become indissociable*."

Under copyright for software, compiled code (a.k.a. "machine-code" or "object code"), i.e., the electric instructions that are read by processors, is protected as a change of format from source code, the text which was actually written by human programmers.

This is not the case with prompts. The same prompt to the same model will not deliver the same results.

It is true that for copyright to arise, absolute control of the outcome is not required. Randomness is allowed if there is still some degree of human control or connection to the result. It's clear that the outcome of a text or image generation model is more than a change of format of the prompt.

Would it, nevertheless, be possible to consider the prompt as a computer program but reject the classification of the output as a compiled program? The Computer Programs Directive protects "*preparatory design work leading to the development of a computer program provided that the nature of the preparatory work is such that a computer program can result from it at a later stage*" (recital 7). So, protecting prompts as software does not necessarily imply that the output is even protected by copyright.

This is not to say that prompts should qualify as preparatory design work. As held by the Dutch Supreme Court in 2018, the difference between normal literary works and preparatory design work is the amount of work required to get to a computer program. If there is a further programming

phase requiring creative choices there isn't yet preparatory design work.

One could choose to look at prompts as normal literary works rather than software. However, the difference is relevant. The applicable rules, namely in terms of ownership, contracts, exceptions and limitations, are not the same.

Conclusion

One year ago, Matt Welsh gave a lecture and proposed the "natural language computer" (see here), in which software would be expressed in natural language. This is, in part, already the case. Programming and development, especially the tasks of code generation, improvement, and testing are increasingly done with the help and support of large language models, with relevant impact. Natural language is expected to become the standard way in which professional users and consumers interact with computers and get them to perform tasks. To an extent, it will replace programming languages.

As a US court put it: "*Applying copyright law to computer programs is like assembling a jigsaw puzzle whose pieces do not quite fit.*" About 30 years ago, we equated this "strange language" of software (the source code) to human language and protected software as a literary work. As the technology develops and prompts become more relevant, it may turn out this was a prescient choice. The challenge, at least at EU level, is having different rules for copyright for software and "normal" copyright.

This entry was posted on Tuesday, November 5th, 2024 at 12:38 pm and is filed under Artificial Intelligence (AI), Authorship, European Union, Originality, USA
You can follow any responses to this entry through the Comments (RSS) feed. You can leave a response, or trackback from your own site.