What, Exactly, Is Software Plagiarism?

BY BOB ZEIDMAN OF ZEIDMAN CONSULTING

Bob Zeidman is the president of Zeidman Consulting, a hardware and software contract R&D firm. He is the developer of CodeMatch™ and CodeSuite™ software for helping to quickly and rigorously locate plagiarism in software source code. He can be reached via email at Bob@ZeidmanConsulting.com.

For several years now I have been working as an expert witness in intellectual property disputes, and for the last four years I have specialized in software copyright infringement and trade secret theft cases. When I began working in this area, I found that most experts used a combination of off-the-shelf computer code analysis tools (to avoid too much confusion I will refer to computer programs that analyze computer software as "tools"), home-grown analysis tools, and lots of long hours and late nights poring over thousands of lines of computer code. There are some tools available from academia that are billed as "software plagiarism detection tools" that were also used by some experts. Expert reports were then written and rebutted. When code from one program looked similar to code from another program, questions arose about whether the code was actually copied or whether there were other reasons for the similarities. Arguments often got very technical and detailed and could easily confuse a non-technical judge or jury. Different experts often had different definitions of plagiarism or found different signs that they considered markers for copied code. It is important to note that code is rarely copied verbatim, but often copied and then modified, sometimes in order to make feature changes and sometimes to hide detection. Some clients, and some experts they hired I am sad to say, seemed to purposely cloud the issue in order to justify illicit or at least questionable behavior.

When I began developing my CodeMatch program, I realized that "plagiarism detection" was really the wrong label for the tools coming out of universities. According to Wikipedia (www.wikipedia.org), plagiarism is "the practice of claiming, or implying, original authorship, or incorporating material from someone else's written or creative work in whole or in part, into ones own, without adequate acknowledgment." Essentially, plagiarism means unauthorized copying, and no software tool can determine whether authorization was given. Also, plagiarism can occur when only parts of a work have been copied – in some cases with numerous changes. It occurred to me that my tool is really a "correlation determination" tool that can perform a quantitative analysis and produce a number representing the amount of correlation between the source code files for trewo different programs. An expert is still needed to determine the reason for the correlation.

Finding a correlation between the source code files for two different programs does not necessarily mean that illicit behavior occurred (not being a lawyer, I will not get into legal definitions of copyright infringement or trade secret theft, but will stick to technical definitions instead). Over the years I have determined that there are exactly six reasons for correlation between two different programs. These reasons can be summarized as follows.

- **Third-Party Source Code.** It is possible that widely available open source code is used in both programs. Also, libraries of source code can be purchased from third-party vendors. If two different programs use the same third-party code, the programs will be correlated.
- **Code Generation Tools.** Automatic code generation tools, such as Microsoft Visual Basic or Adobe Dreamweaver, generate software source code that looks very similar with similar and often identical elements. The structure of the code generated by these tools tends to fit into specific templates with identifiable patterns. Two different programs that were developed using the same code generation tool will be correlated.
- **Commonly Used Identifier Names.** Certain identifier names are commonly taught in schools or commonly used by programmers in certain industries. For example, the identifier "result" is often used to hold the result of an operation. These identifiers will be found in many unrelated programs and will result in these programs being correlated.
- **Common Algorithms.** An algorithm is a procedure or a set of instructions for accomplishing some task. In one programming language there may be an easy or well-understood way of writing a particular algorithm that most programmers use. For example there might be a way to alphabetically sort a list of names. Perhaps this algorithm is taught in most programming classes at universities or is found in a popular programming textbook. These commonly used algorithms will show up in many different programs, resulting in a high degree of correlation between the programs even though there was no direct contact between the programmers.
- **Common Author.** It is possible that one programmer, or "author," will create two programs that have correlation simply because that programmer tends to write code in a certain way. This is the programmer's style of coding. Thus two programs written by the same programmer can be correlated due to the style being similar even though there was no copying and the functionality of each program is different than that of the other.
- **Copied Code (Authorized or Plagiarized).** Code was copied from one program to another, causing the programs to be correlated. The copying may have taken place for only certain sections of the code and may include small or significant changes to the code. When each of the previous reasons for correlation has been eliminated, the reason that remains is copying. If the copying was not authorized by the original owner, then it comprises plagiarism.

This very simple and straightforward categorization can eliminate much controversy and debate among experts and among litigating parties. It is easy to describe to non-technical people and allows for a common definition of terms so that the issue of software plagiarism can be resolved in a more straightforward manner. To date, results of my CodeMatch correlation tool and the six reasons for correlation have been presented in about two dozen software copyright infringement and software trade secret cases and have easily withstood challenges by attorneys and other experts. Most of these cases settled shortly after I had determined that plagiarism had or had not occurred.

I believe that correlation can also be determined for other kinds of documents. The reasons for correlation can be similarly enumerated for those other kinds of documents in order to detect plagiarism in, for example, novels, research papers, and magazine articles. I am currently researching methods for determining correlation of general documents and attempting to enumerate the reasons for correlation as I have done for software source code. It is my hope that by using automated correlation tools and well-specified rules for interpreting the results of these tools, cases of copyright infringement and trade secret theft will be much more deterministic and therefore result in faster and less expensive trials and fairer outcomes that are less open to challenges. I welcome your feedback. IPT