

Rocards arbejdsdokument om softwarepatentdirektivet

EUROPEAN PARLIAMENT

2004



2009

Committee on Legal Affairs

13.4.2005

WORKING DOCUMENT

on the patentability of computer-generated inventions

Committee on Legal Affairs

Rapporteur: Michel Rocard

The Council of Ministers has at last adopted a position on the patentability of computer-implemented inventions, enabling a debate to be held at second reading. Five Member States have voted, pointing out in writing that they were voting to unblock the procedure but wanted the text changed by Parliament. So our disagreement in the first round has been heeded.

This directive is essential not only for economic reasons (several thousand million euros a year are at stake) but also for political or philosophical reasons: the issue is the dissemination of knowledge and ideas in society.

It is a short text but deals with an extremely complex subject. During the two years in which it has been discussed, there have been clear signs that in the difficulty of finding consensual solutions, disagreement on definitions and misunderstandings loom much larger than disagreements on the substance.

I have commissioned a precise and detailed analysis of the subject. It is lengthy. At the time of writing this letter, I am not sure if I can have it translated into English. But I hope to give it to all in French and English.

However, for the debate without the text in Brussels on 21 April, and before officially tabling my proposed amendments, I prefer to suggest that we reflect together on the problem facing us and on the best approach for dealing with the underlying ideas.

For in this short text we in fact only have two serious problems that are likely to encourage conflict with the Commission and Council: the problem of delimiting what is patentable and what is not, and interoperability. If Parliament, and ultimately the Council, follow the guidelines that we put to them, the problem of interoperability will be settled. So we must start by looking at delimitation. What is the issue? It derives from the contradiction between the legal system and inherited tradition, on the one hand, and the needs to reward investment and the security of major industry, underlined by the recent abuses of patentability in the United States, and to a lesser degree the European Patent Office, on the other.

All our legal systems, especially in the Convention on the European patent signed in Munich in 1973, clearly establish the fact that software is not patentable (Article 52(2) of the Convention). There are some 150 000 patents of this type in United States, without a legal basis, and around 50 000 in the European Patent Office, with an uncertain legal basis and not equally valid in all our national legal systems.

The phenomenal growth of information technology over the past 20 years has spread throughout our manufacturing and service industries. Apart from its use for professional purposes, there is now no consumer durable that does not use integrated software: cars, mobile phones, televisions, video recorders, washing machines, lift controls etc.

All of this is costly to create. It is normal and desirable that industry should be able to patent the results of its investment to ensure that it is rewarded, and protected from counterfeiting and unfair competition. Regulating the physical processes implemented inside inventions is a very old problem: it has taken countless forms, particularly mechanical or pneumatic ones. Devising such regulation, which was itself patentable when it was innovative, was extremely expensive. Replacing them by software, whose cost of development and production is a good deal lower, represents an enormous saving. This has encouraged their proliferation. But software is by nature different. It is immaterial.

Software is in fact the combination in an original work of one or more algorithms, that is to say, a set of mathematical formulae. As Albert Einstein has said, a mathematical formula is not patentable. It is by nature an idea, like a book, a set of words, or a chord in music.

For thousands of years knowledge has been established and disseminated by copying and improvement, that is by free access to ideas. The fact that modern branches of knowledge, at least those that have some relation to logic or quantification, can be more easily expressed in the form of software must in no case lead to our relinquishing the principle of free access, which is the only way of safeguarding humanity's luxuriant ability to constantly create new areas of knowledge.

The compatibility of these two contradictory requirements has long been sought, and that search is the purpose of the present directive. Common sense, and the case law in some patent offices, suggest that what is patentable is the invention and not the software that may be necessary to control it. The reference texts, such as the case law of the EPO, express this difference by referring to a 'technical contribution', and up to that point everyone entirely agrees. To be patentable, an invention must:

- be new,
- not be self-evident,
- be susceptible of industrial application,
- have a technical character.

'Technical character' is defined as the ability to provide a technical solution to a technical problem, that is to say to belong to a technical field and have a technical effect. But the word 'technical' is not defined, except by 'the use of technical means' or worse still, by the mere need for 'technical considerations'. This tautology has led to the patenting of everything involved in creating the invention, whether it is software or not.

In particular, Article 52(2) of the Munich Convention stipulates that software is not patentable 'as such', which has resulted by abuse in the obviously flawed interpretation that there might be a difference between software as such and software incorporated in an invention or software as inventions, in other words software that is usable and hence patentable.

This is where we have a duty to innovate, and did innovate at first reading, and where the five or six Member States who have told the Council they are looking for improvements want us to find a solution.

Parliament's wording at first reading was perhaps rather dry and caused some surprise. But a good deal of discussion and argument, particularly with representatives from industry, has confirmed that the path we were exploring was correct.

Software, formulating an idea, is by nature immaterial. The work that it triggers off inside the computer is internal to the computer and not directly communicable to anyone or anything. For that work to be communicable and have an effect, a part must start to move, an electrical, radio or light signal must be produced, data must appear on a screen, or some physical effect must be unleashed. What is evidently patentable are, firstly, the sensors and, secondly, the effectors that supply the computer with data processable by the software and that obtain from the data ultimately produced by the software in its language a physical effect constituting the technical solution to the technical problem posed. The distinction that we are after thus separates the immaterial world from the material, or rather, from the physical world.

But each of these two words is somewhat inadequate to cover the whole area required. 'Material' refers too much to matter and not to energy, while 'physical' implicitly suggests a palpable quality.

Your rapporteur's preference is rather for the following wording, which would find its place in Article 2 of the directive setting out the definitions:

‘Technical field’ means an industrial field of application requiring the use of controllable forces of nature to obtain predictable results in the physical world.

If we agree that even a simple electrical, radio or light signal is composed of energy, this wording covers every possible way of sensing the immaterial data produced by the computer while the software is running to produce an effect perceptible and usable by a machine or human being.

I believe this definition will cover all the real needs of industry, except of course the need identified by some companies for controlling a chain of patented software programs, each of which depends on the other, to prohibit access by the competition to activities downstream affecting the industry and the invention concerned, which clearly we have a duty to prevent.

All the other problems, and hence all the other amendments, are consequent upon this initial choice. I suggest to my colleagues that we should discuss them only after we have reached agreement, which is the purpose of the debate on the 21st.

To enable the directive to permit the patenting of computer-generated inventions while preventing the patentability of software programs, it will be necessary to address the following points:

- to clarify the scope of the directive, wherever possible replace the term ‘computer-implemented invention’ by ‘computer-generated invention’ or ‘computer-assisted invention’, which much more clearly illustrates that the software cannot form part of the technical characteristics of applications for patents;
- clearly define ‘technical field’ in both positive and negative terms: on the one hand, we must specify that a technical field is ‘an industrial field of application requiring the use of controllable forces of nature to obtain predictable results in the physical world’, thus limiting the technology to the physical world; on the other, we need to specify that the processing of the data should not be regarded as a technical field within the meaning of patent law, and that innovations in the processing of data do not constitute inventions within the meaning of patent law;
- define in an untautological way the concepts of ‘technical contribution’ and ‘inventive activity’, and on the latter specify that only the technical characteristics of inventions can be taken into account when assessing it;
- describe the form of applications in both positive and negative terms, so that, on the one hand, applications for computer-generated inventions can refer only to technical products or processes, and on the other that applications for software programs, in themselves or on any supporting base, are prohibited;
- to secure interoperability, strengthen the confirmation of the rights arising from Articles 5 and 6 of Directive 91/250, by ensuring that when use of a patented technology is needed for the sole purpose of ensuring interoperability between two systems, such use should not be regarded as infringing a patent.

Following the debate on 21 April my amendments will be finalised and available as soon as possible.

The Council of Ministers has at last adopted a position on the patentability of computer-implemented inventions, enabling a debate to be held at second reading. Five Member States have voted, pointing out in writing that they were voting to unblock the procedure but wanted the text changed by Parliament. So our disagreement in the first round has been heeded.

This directive is essential not only for economic reasons (several thousand million euros a year are

at stake) but also for political or philosophical reasons: the issue is the dissemination of knowledge and ideas in society.

It is a short text but deals with an extremely complex subject. During the two years in which it has been discussed, there have been clear signs that in the difficulty of finding consensual solutions, disagreement on definitions and misunderstandings loom much larger than disagreements on the substance.

I have commissioned a precise and detailed analysis of the subject. It is lengthy. At the time of writing this letter, I am not sure if I can have it translated into English. But I hope to give it to all in French and English.

However, for the debate without the text in Brussels on 21 April, and before officially tabling my proposed amendments, I prefer to suggest that we reflect together on the problem facing us and on the best approach for dealing with the underlying ideas.

For in this short text we in fact only have two serious problems that are likely to encourage conflict with the Commission and Council: the problem of delimiting what is patentable and what is not, and interoperability. If Parliament, and ultimately the Council, follow the guidelines that we put to them, the problem of interoperability will be settled. So we must start by looking at delimitation. What is the issue? It derives from the contradiction between the legal system and inherited tradition, on the one hand, and the needs to reward investment and the security of major industry, underlined by the recent abuses of patentability in the United States, and to a lesser degree the European Patent Office, on the other.

All our legal systems, especially in the Convention on the European patent signed in Munich in 1973, clearly establish the fact that software is not patentable (Article 52(2) of the Convention). There are some 150 000 patents of this type in United States, without a legal basis, and around 50 000 in the European Patent Office, with an uncertain legal basis and not equally valid in all our national legal systems.

The phenomenal growth of information technology over the past 20 years has spread throughout our manufacturing and service industries. Apart from its use for professional purposes, there is now no consumer durable that does not use integrated software: cars, mobile phones, televisions, video recorders, washing machines, lift controls etc.

All of this is costly to create. It is normal and desirable that industry should be able to patent the results of its investment to ensure that it is rewarded, and protected from counterfeiting and unfair competition. Regulating the physical processes implemented inside inventions is a very old problem: it has taken countless forms, particularly mechanical or pneumatic ones. Devising such regulation, which was itself patentable when it was innovative, was extremely expensive. Replacing them by software, whose cost of development and production is a good deal lower, represents an enormous saving. This has encouraged their proliferation. But software is by nature different. It is immaterial.

Software is in fact the combination in an original work of one or more algorithms, that is to say, a set of mathematical formulae. As Albert Einstein has said, a mathematical formula is not patentable. It is by nature an idea, like a book, a set of words, or a chord in music.

For thousands of years knowledge has been established and disseminated by copying and improvement, that is by free access to ideas. The fact that modern branches of knowledge, at least those that have some relation to logic or quantification, can be more easily expressed in the form of software must in no case lead to our relinquishing the principle of free access, which is the only way of safeguarding humanity's luxuriant ability to constantly create new areas of knowledge.

The compatibility of these two contradictory requirements has long been sought, and that search is the purpose of the present directive. Common sense, and the case law in some patent offices, suggest that what is patentable is the invention and not the software that may be necessary to control it. The reference texts, such as the case law of the EPO, express this difference by referring to a ‘technical contribution’, and up to that point everyone entirely agrees. To be patentable, an invention must:

- be new,
- not be self-evident,
- be susceptible of industrial application,
- have a technical character.

‘Technical character’ is defined as the ability to provide a technical solution to a technical problem, that is to say to belong to a technical field and have a technical effect. But the word ‘technical’ is not defined, except by ‘the use of technical means’ or worse still, by the mere need for ‘technical considerations’. This tautology has led to the patenting of everything involved in creating the invention, whether it is software or not.

In particular, Article 52(2) of the Munich Convention stipulates that software is not patentable ‘as such’, which has resulted by abuse in the obviously flawed interpretation that there might be a difference between software as such and software incorporated in an invention or software as inventions, in other words software that is usable and hence patentable.

This is where we have a duty to innovate, and did innovate at first reading, and where the five or six Member States who have told the Council they are looking for improvements want us to find a solution.

Parliament’s wording at first reading was perhaps rather dry and caused some surprise. But a good deal of discussion and argument, particularly with representatives from industry, has confirmed that the path we were exploring was correct.

Software, formulating an idea, is by nature immaterial. The work that it triggers off inside the computer is internal to the computer and not directly communicable to anyone or anything. For that work to be communicable and have an effect, a part must start to move, an electrical, radio or light signal must be produced, data must appear on a screen, or some physical effect must be unleashed. What is evidently patentable are, firstly, the sensors and, secondly, the effectors that supply the computer with data processable by the software and that obtain from the data ultimately produced by the software in its language a physical effect constituting the technical solution to the technical problem posed. The distinction that we are after thus separates the immaterial world from the material, or rather, from the physical world.

But each of these two words is somewhat inadequate to cover the whole area required. ‘Material’ refers too much to matter and not to energy, while ‘physical’ implicitly suggests a palpable quality.

Your rapporteur’s preference is rather for the following wording, which would find its place in Article 2 of the directive setting out the definitions:

‘Technical field’ means an industrial field of application requiring the use of controllable forces of nature to obtain predictable results in the physical world.

If we agree that even a simple electrical, radio or light signal is composed of energy, this wording covers every possible way of sensing the immaterial data produced by the computer while the

software is running to produce an effect perceptible and usable by a machine or human being.

I believe this definition will cover all the real needs of industry, except of course the need identified by some companies for controlling a chain of patented software programs, each of which depends on the other, to prohibit access by the competition to activities downstream affecting the industry and the invention concerned, which clearly we have a duty to prevent.

All the other problems, and hence all the other amendments, are consequent upon this initial choice. I suggest to my colleagues that we should discuss them only after we have reached agreement, which is the purpose of the debate on the 21st.

To enable the directive to permit the patenting of computer-generated inventions while preventing the patentability of software programs, it will be necessary to address the following points:

- to clarify the scope of the directive, wherever possible replace the term ‘computer-implemented invention’ by ‘computer-generated invention’ or ‘computer-assisted invention’, which much more clearly illustrates that the software cannot form part of the technical characteristics of applications for patents;
- clearly define ‘technical field’ in both positive and negative terms: on the one hand, we must specify that a technical field is ‘an industrial field of application requiring the use of controllable forces of nature to obtain predictable results in the physical world’, thus limiting the technology to the physical world; on the other, we need to specify that the processing of the data should not be regarded as a technical field within the meaning of patent law, and that innovations in the processing of data do not constitute inventions within the meaning of patent law;
- define in an untautological way the concepts of ‘technical contribution’ and ‘inventive activity’, and on the latter specify that only the technical characteristics of inventions can be taken into account when assessing it;
- describe the form of applications in both positive and negative terms, so that, on the one hand, applications for computer-generated inventions can refer only to technical products or processes, and on the other that applications for software programs, in themselves or on any supporting base, are prohibited;
- to secure interoperability, strengthen the confirmation of the rights arising from Articles 5 and 6 of Directive 91/250, by ensuring that when use of a patented technology is needed for the sole purpose of ensuring interoperability between two systems, such use should not be regarded as infringing a patent.

Following the debate on 21 April my amendments will be finalised and available as soon as possible.