

# Criteria voor patent protection of computerprograms

P. van Schelven (FENIT)  
J. Helberg, M. van der Boom, L. van Dijk (VOSN)

September 7, 2001

## 1 Introduction

This document contains recommendations for testable criteria for the protection of computerprograms under patent law.

On February 15, 2001, members of the Second Chamber committee for Economic Affairs R. Hindriks en M. Udo, requested the Dutch IT Federation (FENIT) and the Dutch Open Source Society (VOSN) to draft these recommendations. On March 14, 2001, the parliamentary committee, in a General Meeting, instructed the secretary of state for economic affairs to take a stand against trivial patents in Europe, and to take initiatives to narrow the gaps in law, based on these recommendations.

## 2 Background

Art. 52 of the European Patent Treaty (EPT) and art. 2 of the Dutch patent law (ROW) of 1995 both contain a condition that excludes "computer programs as such" from protection under patent law. This explicit exclusion has not prevented that in patent law-practice some patents have been granted on software-related inventions.

The reason of this exclusion lies in the distinction the lawmaker and the treaty partners of the EPT wanted to impose between the "technical" domain and the domain of abstract reasonings. The patent system is intended for the first, and not for the second domain, which is already covered by other systems of law.

The lawmaker did recognize, in art 52, 3rd member EPT that a technical invention may contain a computerprogram, and that this does not prevent the patentability of the invention as a whole.

In Europe the practice of patent law uses artificial means, in a way, in the description of the object of patent applications, with the intention to get around the exclusion of "computer programs as such". From various sides there has been justified opposition to the fact that this has caused an unclear practice, and a situation where the status of patent applications involving software raises a number of questions. These unclaritys have contributed to legal uncertainty and have opened the door to patents with a dubious status. These developments are undesirable.

It is a fact that the words "as such" in the EPT and the ROW have led to a very restricted explanation of the exclusion of software. In summary: only if a patent claim is explicitly invoked for a computer program "as such", the program, without regard to its contents and functionality, is excluded from patentability.

For software related inventions that satisfy the other legal criteria, the practice of the EPO seems to allow patent protection. This possibility is now widely exploited.

A computer program is a total of instructions, intended or useful to make an informationprocessing machine fulfil a certain task, posed in a formal, mathematically precise language. That makes a computerprogram practically usefull, and at the same time a mathematical abstraction. (A precise description of a computerprogram *is* that program itself, while a precise description of, say, a lock or a clock is not itself that lock or that clock.) Because of this character of pure information, a

computerprogram, as opposed to a technical invention, can be multiplied without physical effort, and at a very low cost.

The extension of the domain of patent law, by granting in practice patents to computer programs as such, therefore contains the danger that the difference between technical and non-technical is being eroded.

The planned use of controllable natural forces has always been used as a minimal condition to recognize the technical character of an invention. Including human mental powers as such into the domain of natural forces would therefore mean that all results of human reasoning, if they form a causally overseable prescription, would be ascribed technical character. That would mean that the meaning of the word "technical" would disappear.

Patent law is not intended as a catch-all in which all mental efforts otherwise not benefitted by law should find protection, but rather as a special law to protect a definite domain of mental efforts, namely technical ones.

Therefore it has to remain the case that a pure organisational or calculation rule, whose only relation to the realm of technique consists in the planned use of a known information processing machine, does not deserve patent protection.

The relevance of this warning becomes obvious from the fact that a broadening of patent law has led to the granting of patents on computer supported business methods through the mentioned loophole. This is generally considered undesirable.

Apart from blurring the traditional border between the technical and the mental, the practice of the EPO to allow patents on computer programs as such, has led to the situation in which a huge number of elementary processes may nowadays enjoy protection by a patent. Claims on trivial processes and based on functional rather than deterministic descriptions have been granted in some numbers over the past few years.

The goals of the patent system are being missed by these developments with respect to software-related inventions. After all, the intention of patent law is to stimulate innovation by granting the inventor a monopoly in exchange for publication of the knowledge.

If the knowledge contained in the patent application would have entered the public domain even without granting the monopoly, or when the application is formulated in such a way that it contains no information about the workings of the invention, but only about its intended use, society does not gain her rightful increase of knowledge, but does suffer the economic consequences connected with a monopoly. In that case, the patent becomes a market disturbing privilege.

To end this situation, criteria must be put in law to make a sharp distinction between technical and non-technical, and between trivial and non-trivial, before the explicit ban on patents on software "as such" may be struck.

### **3 De jure factors**

The decisive factors which should, in law, apply to any patent-request are:

1. the invention should be new, in other words, should not be part of the current state of technology.
2. the invention should be a product of inventive actions.
3. the invention should be applicable.

Future patent-law for software related inventions should conform also to the above mentioned principles. There are however reasons to evaluate concrete appliance of these factors with respect to computersoftware.

## 4 Technical, computersoftware and Business Methods

Patentlaw protection of 'business methods' is not on anyones wishlist. 'Business Methods' are outside the technical domain; patentlaw is restricted to this technical domain only. This restriction is imposed upon patentlaw by the term 'invention' and the common interpretation of this word.

Patentlaw limits itself to technical subjects. In contrast with patentlaw development in the USA, the EU should confirm the restriction to the technical domain only.

In this context the following principle holds: the bare fact that the presentation of something in computersoftware does something inside a computer, is not enough to make this qualify as being technical. Being technical cannot be assumed in the proces of a patent appliance and must always be tested for.

## 5 Trivial patents

It is beyond discussion that no patentprotection may be reserved for trivial software, i.e. software that has a strongly obvious character or supports a strongly obvious functionality.

To answer the question whether this does or does not apply in a concrete case, it is not a good idea to look for an answer in the definition of 'novelty' as found in the ROW and EPT. After all, the novelty of an invention is determined by the 'state of the technique' and that is a given at the time of application.

## 6 Recommendations

To avoid non-technical and trivial patents, the following reccomendations are made:

### 6.1 Applicability: publishing working source code

A particular computer software program is a set of instructions for a machine, performing a particular task. The original representation of a software program: the work of the programmer, is a readable document. This is called the source code.

In theory, it is possible to describe the functionality or content of a computersoftware program in general, abstract terms. It is also possible to describe this in precise terms.

It should be ensured that patent law protection cannot be granted for computer software described only in general, abstract terms. Or computer software which only list functionality and not solutions. Otherwise the implied monopoly of the patent would be too broad, this is not desirable. If the subject of patent-request is mainly conceptual, this may introduce a worse situation: monopolising of concepts.

It goes without saying that a non-complete or vague specification of a computer software program, or just a list of goals for the program to achieve, cannot be seen as an applicable computer software program. Therefore we should demand that the patent-request should always be accompanied with a working version of the software.

One of the goals of patenting is publication of knowledge. Therefore any software which is part of a patent-request, should be published in human-readable form; the source-code. After the patent has been granted, this source-code should be published together with all other technical details of the invention.

### 6.2 Limitation of scope

For software in particular a distinction has to be made between the physical implementation of a method and the method itself. A patent may be granted to a specific implementation of a certain method or algorithm, not to the method or algorithm itself. The development of software is the

innovative combination process of methods to reach a certain end result. These methods vary from elementary mathematical operations, like addition and multiplication, to more complex methods.

The grant of a patent must be limited to the actual implementation. The methods used are to be explicitly excluded from the patent. The specific manner in which a certain solution has been implemented may be part of the patent, but the patent may not forbid others to use this method in the creation of another end result.

Example: A patent may be granted on a calculator implemented in software; the operations performed by this calculator (addition, division, multiplication) are excluded from the patent. The 'blue prints' of the calculator are an integral part of the patent request. In the case of software these blueprints are the source code of the (relevant part) of the implementation.

To avoid unnecessary demotivation to innovate for third parties, which would ignore the goal of the patent system and stimulate unfair competition, it should not be possible to request a patent for a certain software-method, which is an abstract algorithm. First, an implementation of this method must be developed before a patent request can be made and possibly granted.

### **6.3 Threshold for inventors: necessity of experiments**

Next to the (general) requirement that an invention in an absolute sense must be new, it must be required that a computer program, which is eligible for a patent grant, has enough 'invention-height'.

To prevent that patents will be granted to computer programs which are merely the result of a simple mental discovery or act, the law needs to state explicitly that the work of a programmer (inventor) provides in a surprising and, for the expert, non-trivial, creative solution of the technical problem for which the program was developed.

A minimal requirement is that an experiment was necessary to come to the invention. From the patent-description it must be clear that:

1. the experiment has been performed,
2. the experiment was necessary to come to the invention.

These two conditions should not be sufficient, but necessary for granting a monopoly. In other words: the lack of an experiment shows that the height of the invention was not sufficient, apparently because the invention could be constructed with logical mental reasoning alone.

Vice versa, the proof of experiment is not sufficient to show beyond doubt that a leap forward has been accomplished. It still has to be shown, in the minimum case, that the experiment was absolutely necessary. The complexity of an invention or its description can be indicators for this. However, a needlessly complicated description of the invention clearly violates the second condition of necessity.

An experiment, as described here, is a procedure in which under controlled and repeatable circumstances a theory or hypothesis is tested against the laws of nature. The essence is that a question is asked, which is answered by a physical process and not by mental reasoning.

The rationale behind this requirement is that for obvious inventions, (and for many computer programs "as such"), it can be seen up front that certain steps lead to certain results.

For inventions which are valuable but nonetheless do not comply with the mentioned requirements, it can be assumed that they will be implemented and published anyway, even in the absence of the reward of a monopoly.

### **6.4 Assessment by patent-agencies**

The patent-agencies must be sufficiently equipped to perform their assessments with the appropriate level of quality. For this, enough educated people and sufficient resources must be available for them.

Especially the assessment of the 'state of technology' to judge the innovative element in an invention needs to be addressed, which is even more difficult in the dynamic area of software inventions. A classification system to address this problem is needed.

The patent agency will also need to implement control measures to prevent that in the event of questionable patentability of a computer program a patent is granted lightheartedly. In case of doubt, the preferred decision is to be the refusal of the patent.

Of decisive influence is the infrastructure of the assessment system. It must be assured that no advantage is present for the assessing agency to rather grant a patent than deny it. If these stimulants are off-balance, as is the case now, a situation will evolve in which the "polluter profits". In the current situation, the three criteria of innovation, invention and applicability are applied with regretablely looseness.

## 7 Conclusion

In conclusion, the FENIT and the VOSN propose that the following criteria are applied in the law, to prevent patents on non-technical and trivial inventions:

1. distinction between technical and non-technical in terms of the planned use of natural forces (section 4)
2. in the case that a computer program is part of the invention, a working program must be supplied in a human readable original format (source code) (section 6.1)
3. limitation of protection to the supplied implementation (section 6.2)
4. necessity of experimentation (section 6.3)
5. stimulants in the assessment system for a correct interpretation of the limitations of patent law (section 6.4).

As the parliamentary committee on Economic Affairs has ordered the secretary of state, in the event that the above criiria will not be implemented, the explicit prohibition of patentability of software "as such" must be maintained.

Federatie Nederlandsche IT (FENIT) Postbus 401 3440 AK Woerden tel. 0348 493838 fax. 0348 482444  mr.P.C. van Schelven ( <i>peter.van.schelven@fenit.nl</i> )	Vereniging Open Source Nederland (VOSN) Postbus 72 4180 BB Waardenberg tel. 0418 653336 fax. 0418 653666  J. Helberg ( <i>voorzitter@opensource.nl</i> ), M. van der Boom ( <i>marcel@hsdev.com</i> ), dr.L. van Dijk ( <i>lvd@mndmttr.nl</i> )
---	---