

Kiesewetter-Köbinger 2000: Über die Patentprüfung von Programmen für Datenverarbeitungsanlagen

<http://swpat.ffii.org/papiere/grur-skk01/index.de.html>

Swen Kiesewetter-Köbinger

elisabeth-koebinger@t-online.de

2003-09-18

Ein Patentprüfer zeigt die Ungereimtheiten der Prüfung von Software-Anmeldungen auf. In ihrem Bemühen, ein Gesetz umzuinterpretieren, welches unmissverständlich die Patentierung von Datenverarbeitungsprogrammen verbietet, hat die Rechtsprechung im Laufe der Jahre Funktionsansprüche zugelassen, die es dem Anmelder erlauben, ein Programm zu verkleiden. Aber diese Funktionsansprüche stellen eher Probleme als Lösungen dar, und die Lösung zu diesen Problemen besteht in einem (nicht patentierbaren) Datenverarbeitungsprogramm (als solchem). Probleme zu patentieren ist aber noch weniger zulässig und in seinen Auswirkungen noch bedenklicher als Programme zu patentieren.

Inhaltsverzeichnis

| | | |
|----------|--|----------|
| 1 | A Program as Such | 2 |
| 1.1 | The Objective Subject of Invention | 4 |
| 1.2 | Inventions Must be Solutions | 4 |
| 2 | The Technical Character | 5 |
| 2.1 | The Basic Assumption | 6 |
| 2.2 | The Reversal Conclusion | 6 |
| 2.3 | Exclusion also of Technical Theories | 6 |
| 2.4 | Technicity of Computer Programmes | 7 |

| | | |
|----------|--|-----------|
| 3 | The technicity concept | 8 |
| 3.1 | The “Language Analysis” and “Logic Verification” cases | 8 |
| 3.2 | The Bell | 9 |
| 3.3 | Additional Technical Effect | 10 |
| 3.4 | Programmes for the Description of Hardware | 10 |
| 3.5 | Borders of the Technicity concept | 10 |
| 4 | The Objective Subject of Protection | 11 |
| 5 | The Exclusion Catalogue as Legal Disclaimer | 12 |
| 6 | Source Code in the Patent Claim | 13 |
| 7 | Concluding Remark | 14 |

Titel: Kiesewetter-Köbinger 2000: Über die Patentprüfung von Programmen für Datenverarbeitungsanlagen

Quelle: GRUR¹ GRUR 2001-03

: The author placed his original German text² under an OpenContent license and submitted it to GRUR – Gewerblicher Rechtsschutz und Urheberrecht³ for publication. The English translation is so far only a rough draft.

Dieser Artikel soll einige der Probleme und Ungereimtheiten klären helfen, mit denen sich der Autor als ehemaliger Softwareentwickler und heutiger Patentprüfer am Deutschen Patentamt bei der Prüfung von computerprogrammbezogenen Patentanmeldungen konfrontiert sieht.

1 A Program as Such

In one of its recent decisions⁴ the 17th senate of the German Federal Patent Court⁵ had to decide whether the following items qualify for patenting:

Digital storage medium, in particular diskette, with electronically selectable control signals, that can cooperate with a programmable computer system

⁴<http://swpat.ffii.org/papiere/bpatg17-suche00/index.de.html>

⁵Bundespategericht, BPatG

in such a way that a procedure is operated according to one of the claims 1 to 17.

Computer programme product with program code for the execution of the procedure according to one of the claims 1 to 17 stored on a machine-readable carrier, if the programme product runs on a computer.

Computer programme with program code for the execution of the procedure according to one of the claims 1 to 17 if the programme runs on a computer

In the patent claim 1, to which the patent claims mentioned refer, a claim has been made for

Procedures for the computer-assisted search and/or correction of a defective character string F_i in a digitally stored text, which contains the appropriate error free character string S_i

characterised by the fact that

1. the occurrence frequency $H(S_i)$ of the error free character string S_i is determined
2. the error free character string S_i is changed according to a rule R_j , so that a possible defective character string f_{ij} is produced,
3. the occurrence frequency $H(f_{ij})$ of the character string f_{ij} in the text is determined,
4. the occurrence frequencies $H(f_{ij})$ and $H(S_i)$ are compared and
5. based on the comparison in step d), it is decided whether the possible defective character string f_{ij} is the looked for defective character string F_i .

and has been considered as being worthy of patent in accordance with §§ 1-5 PatG (Patentgesetz, German Patent Law) by the patent examiner. This at least tacitly implies according to present comprehension that the subject of registration has a technical character. The 17th senate has no objections against this.

The claimed subjects are to solve the task to create “an improved procedure and computer system for the search and/or correction of a defective character string in a text”.

The question, where objectively the invention is to be seen, whether in the procedure of the patent claim 1, the programme of the patent claim 24, or in an algorithm on which the patent claim 1 is based, has unfortunately never been posed. Since this question was not at all raised, the BGH (Bundesgerichtshof, German Federal Court) will probably not take explicitly a position regarding this issue. This could be, however, a crucial point for the comprehension of the problem of patenting programmes for data-processing systems in general.

1.1 The Objective Subject of Invention

The 17th senate explained in its decision under 1.3.2:

In the very next paragraph it says:

A processing of the procedure can only be successfully undertaken with a computer system, which is able to quasi completely interpret the individual parts of the recording and thus to cause an execution of the desired process steps.

The “desired process steps” are very clearly not regarded here themselves as solution of the task. Below under 1.4.3 it says:

A potential for the production of a technical effect... is attributed... only to the computer system with the programme loaded from the storage medium into the main memory, which is able actually to implement, if necessary, a technical procedure.

On the basis of these remarks, however, only one conclusion can be drawn: not (the work) procedure represents the solution of the task and thus the objective registration subject, but only the computer system with the programme loaded from the storage medium into the main memory, whose recording must be interpreted and implemented quasi completely. The working procedure is only the effect of the execution of the program code by the computer system that is not in detail characterised in the patent claims 1, 22, 23 and 24. In view of the fact that in practically all cases of programme related registrations these procedures are claimed as effects and are also frequently patented, this judicial insight is of special importance!

1.2 Inventions Must be Solutions

Now §1 PatG reads literally: “Patents are awarded for inventions ...”.

With which wording such a computer system is claimed at present is unfortunately not yet published. From the disclosure writing, however, such a computer system can be deduced. Beside the memory means 1, 3, 4, 10, 12, 14 and 17 apparently the means 5, 6, 7, 8, 9, 11 and 13 are substantial for the execution of the procedure. This is described as follows:

In the processor means 2 the contained means 3, 5, 6, 7, 8, 11 and 13 as well as the bus 16 do not have to be implemented as discrete electronic construction units, but can be produced by an appropriate programming of the processor 2. An appropriate programme suitable for the implementation of the invention related procedure will cooperate in such a way with the operating programme of the computer system in an actually well-known way, that the computer system takes on the configuration shown in Fig. 4.

The claimed working procedure is to be realised therefore with priority by a suitable programme, which is processed by a processor 2 and accesses over the bus 16 its memory 1, 3, 4, 10, 12, 14 and 17. Computer systems with a processor, which accesses over a bus its memory and processes a programme from the memory, have been known at the latest since Von Neumann. The remaining characteristics (allocation of the memory and means to...) are caused therefore - as far as understandably revealed - by the programme in the memory 17. A programme for a data-processing system is however not regarded as an invention according to §1 (2) 3rd PatG. If however the programme itself is based on no inventive activity, how can then its execution on a well-known computer system lead to an inventive effect in form of an inventive procedure?

A second solution variant, however, is also mentioned in above quotation, since the means 3, 5, 6, 7, 8, 11 and 13 can also be implemented as discrete electronic components. No reference about the constitution these components can be found. Even their exact functionality appears unclear to the author. Something could be said on this only after having knowledge of the programme code to be processed. It is well-known how Rapid Silicon Prototyping ASICs can be developed with well-known IP blocks, however the allocation between the desired functional character of the means 3, 5, 6, 7, 8, 11 and 13 and the known logic blocks would need to be recognisable. This is not possible for the author on the basis of the disclosure writing.

Therefore one would like to suppose that according to the wording of the exception catalogue of §1 PatG such a subject would not be patentable for lack of inventive activity.

2 The Technical Character

As a way out of the exception catalogue of §1 PatG the jurisdiction, however, has found ways how, despite the clarity of §1 PatG, explicitly excluded subjects can nevertheless be made accessible for patenting.

In relation to §1 par. 2 and 3 PatG here the 17th senate justifies

This so-called exclusion catalogue has been based in general opinion on the thought that the subjects specified there lack the necessary technical character (see Busse, PatG, aaO, §1 Rdn 37; Schulte, PatG, 5th ed., §1 Rdn 60, in each case mwN; also the law justification, BIPMZ 1976, 332 left frame, points into this direction, if it says there among other things that the negative catalogue excludes only subjects which “in the jurisdiction of the supreme court have not been recognised as inventions up to now”; see also EPA, UC 3.5.1 “computer programme product/IBM”, GRUR Int 1999, 1053, 4.1).

Which group of interest carries this general view is not discussed. The circle of the software developers concerned, however, seems to share this view only to a small percentage. In the case of the computer programmes there are completely contrary opinions even among well-known patent specialists.

2.1 The Basic Assumption

In the basic assumption it is stated that “in general opinion” all subjects specified in the exclusion catalogue of the §1 par. 2 lack the “necessary technical character”.

Instead of judging whether the registration subjects are one of the of the subjects specified in the exclusion catalogue, in simplifying practice it is only examined whether the registration subject demonstrates a “technical character”. In §1 PatG the word technique or technology is not mentioned at all by the legislator. An indirect relation results only from the stipulations on the state of technology in the subsequent paragraphs 3 and 4 PatG, from which it is to be deduced that thereby an invention in the sense of §1 PatG must also necessarily demonstrate a technical character.

In the case of numerous registrations of programme related theories the courts came to the conclusion that these theories demonstrate a technical character nevertheless. Therefore obviously there are subjects in the exclusion catalogue of §1 par. 2 that demonstrate a technical character.

According to all rules of logic one would have to recognise at this point that by this the basic assumption is disproved, since it shows exceptions in this general version. The technical character of the registration subject can represent thus only a necessary, not a sufficient criterion for its patentability.

2.2 The Reversal Conclusion

Instead, however, the reversal conclusion is drawn that also with the subjects specified in §1 par. 2 an invention and thus patentability can be present, if they contain a technical theory . The basic assumption was disproved by numerous decisions of the BGH and the BPatG, as has been pointed out. Each final conclusion, which is based on a defective basic assumption, can however only be defective itself. Suddenly it is not differentiated between necessary and sufficient any more; completely in the sense of art. 27 par. 1 TRIPS, whose applicability is still to be clarified. For a programme developer, who is forced by daily practice to think strictly logically, the argumentation for the sufficient patentability of technical theories is therefore not understandable. The exclusive restriction to technical theories is not stated in this form in §1 PatG, and the argumentation which tries to limit the exclusion catalogue of §1 PatG to this is simply wrong.

2.3 Exclusion also of Technical Theories

The logical consequence from this would be that also technical theories can be excluded from patent protection, if they fall under the exclusion catalogue. [BGH software patentability expert judge] Melullis explains in this regard:

The legal state of affairs is formulated in such a way that it can exclude even a technical theory from patent protection according to a type of fiction, if it falls under the negative catalogue of Art. 52 paragraph 2 EPA (European Patent Agreement). If one proceeds from the pure wording of the law, it appears to be more appropriate to see the reason for the exclusion of the

subjects specified there not in their lacking technical contents, but in the statement that they do not deserve a protection from the point of view of the law by the patent law or are not to be subordinated to this for overriding reasons.

This is exactly the point! The legislator did not aim at the absence of a technical theory as a criterion against patentability, because this could have been formulated very easily. Rather it wanted to exclude these articles just as such from the patent protection. Additionally, the legislator in no word made technical theories accessible for patent protection, because a theory is regarded a rule for a mental activity and thus just like programmes for data-processing systems not an invention.

Melullis however calls this a fiction. After the customary comprehension of a constitutional state it is the business of the legislator to promulgate laws generally accepted; the administration has to use these laws in individual cases; the courts have to decide separately for each individual administrative action whether the respective execution has been according to the law. How the wording of the law in a constitutional state can be regarded as a fiction, and how court decisions can as a consequence transform the sense of word of the law into the opposite, is completely incomprehensible for the author.

But without these constructions computer programmes as such as well as new applications of well-known articles with new programmes would not be patentable.

2.4 Technicity of Computer Programmes

With his view Melullis assumes “in principle the technicity of each computer programme”. More exactly said he differentiates between

- “the mental logical concept..., which is based on the problem to be solved by the software, i.e. the actual programme contents including the programme idea which is the basis for it and”
- “their conversion into the individual steps necessary for the execution.”

“So the computer programme as such is within this interpretation the non-technical concept; this means it is the conception preceding the conversion into a procedural instruction to the computer.”

So far, however, the author has not seen one German or European patent writing about programme related applications, in which it was not precisely this conception in the form of “procedures for ...” achieving a certain effect that was claimed.

For Melullis is in contrast to this

In principle patentable - also because technically- ... the implementation of this first only mental solution into a procedural instruction for the computer and its control, i.e. the technical conception serving the realisation of this programme, as it found its expression in the finished programme.

The finished programme is for him thus no programme as such any more and thus in principle accessible to patent protection. He also seems to regard the conception as a plan for a mental activity. Other authors however define just these coded instruction sequences as a programme as such; have however no doubts to classify as technical and patent thus the underlying programme idea or business idea despite their characteristic as a plan for mental activity.

All these final conclusions seem to be based however again on the incorrect basic assumption on the exclusion catalogue.

3 The technicity concept

Melullis uses an argumentation line frequently to be found, by giving its own definition to the term “computer programme as such” as a “non-technical concept; this means that of the conception preceding conversion into a procedural instruction to the computer”. However, the technical field can in effect not be exactly defined. The respective specialist speaks normally of programming techniques, painting techniques, presentation techniques, football techniques, speech techniques, cultural techniques (reading, writing, counting), running techniques, grasping techniques for the playing of a musical instrument. Even with reference the sexual act one speaks occasionally of special techniques. In Brockhaus Encyclopedia the extensive definition of technique begins with

[from *technica* in *internat. Scholar Latin*, from Greek *techne* >art<, >talent<] in ancient Greece had the significance of art, trade, handicraft, also science and arts and crafts as well as technical skill to reach something determined (e.g. t. of painting). In an extended sense t. today is the constructive creation of products, devices and procedures by use of the materials and forces of nature and with regard to the laws of nature. ...

The BGH defines:

A theory is technical if it is for deliberate action employing controllable natural forces for the achievement of a causally surveyable success, which is the direct result of the employment of controllable natural forces without inserting human understanding activity.

3.1 The “Language Analysis” and “Logic Verification” cases

The fact that also the term technique of the BGH changes, is illustrated by the following: Instructions to the human mind were always rejected so far by the BGH as not technical. Starting with the “language analysis” decision, in which the senate states

the fact that it can be made use of the human understanding without that alone thereby the area of the technical is left, can be deduced already by the fact that theories for deliberate action employing controllable natural forces

for the achievement of a causally surveyable success are accessible to patent protection .

the BGH has called instructions to the human mind not necessarily hindering a patent and has discarded the requirement for immediate causality in producing an effect of controllable natural forces. Also in the “logic verification” decision where it is only demanded that

If a theory for a programme for data-processing systems is coined by a realisation that is based on technical considerations, a delimitation criterion is found that is also accepted elsewhere and is promoting a uniform patent right practice for Europe, which permits to state the necessary technical character of a theory for a programme for data-processing systems.

the BGH has rejected the employment of controllable natural forces as a direct condition of patentability and justified this with the fact that

the term technique of the patent right can not be understood statically, i.e. once and for all it being certain. It is rather accessible to modifications, if the technological development and an effective patent protection adapted to it require this.

With some maliciousness these two decisions can be understood also in such a way that even the instruction to the programmer to write a programme which is based on technical considerations can be regarded by the senate as technical. A interpretation very friendly to claims for the synopsis of both BGH decisions is also found in Hoessle, who sees in “the “non-static technology concept” a worthy and handy generalisation, which will render good services in practice”.

3.2 The Bell

Another malicious example: For Melullis is to be proceeded in principle from the technicality of each computer programme, since the programme controls the computer. But with the help of programmes also technical information is shown , which is based on technical considerations. The difference, whether it concerns a technical CAD design in source code with control statements for an CAD or print programme, or whether it is the description of Schiller’s poem “Die Glocke” (The Bell), with control statements for a browser programme written in HTML, is of rather academic nature, because both contain control statements, which can be run by a computer with the programme designed for the interpretation of the control statements; even the displayed information is in each case technical, even if the metallurgical aspect is not revealed by Schiller in any language customary for the bell caster.

3.3 Additional Technical Effect

With the help of programmes each computer controls its periphery. From the point of view of the programmer it is not important whether this periphery is a screen, a printer, an I/O card, one or more A/D-D/A transducer or an other article designed for the control by a processor. With a programme in assembler language the control of the underlying hardware is described most directly; each individual assembler instruction describes and produces with its execution a directly technical effect . An additional technical effect, which goes beyond the customary cooperation between programme and computer - how it is demanded recently by the technical Boards of Appeal of the EPO (European Patents Office) - can consequently only be understood as an effect which is caused by none of the implemented programme steps, which is therefore independent of the programme related theory and which is not in connection with it. Why such an additional technical effect is then to be brought at all in connection with a programme remains unclear. If the mentioned technical effect is caused however by the running of one or more program steps, it can not be rightfully spoken of an additional effect, since this is the intended effect.

3.4 Programmes for the Description of Hardware

VHDL or VERILOG programmes describe the structure of highly integrated circuits and control the configuration of FPGAs (Field Programmable Gate Arrays) . These circuits are sketched by using these programmes, are verified with the help of further programmes and are tested on appropriate logic simulators. The masks for wafer production are also produced by programmes, which those and/or VERILOG programmes evaluate . The finished IC, processor, ASIC is in the end nothing more than a product of several computer programmes, in which the solution, how the desired functionality is to be established, is finally only based on the programmes written in the selected language for hardware description . Such programmes contain certainly technical considerations, since these programmes describe concrete hardware and concrete semiconductor topographies can be produced with them. But nearly nobody applies for a patent for the code of these programmes - although their clear succession of commands is offered for sale as a VHDL or VERILOG programme and is thus completely exposed - but nearly exclusively for their desired functionality (thus for a problem to be solved somehow, see below).

3.5 Borders of the Technicity concept

Finally everything made by people can therefore be understood as “technical” depending on the spirit of the time . It would only need to be distinguished from the divine creation. About any other delimitation of the term technique is disputable, since by this a painter will understand something else than a physicist, a musician something else than a craftsman, a writer something else than an engineer and a computer scientist finally something else than a lawyer. A purely legally coined technique definition for the delimitation of patentable against non-patentable, which leaves the respective target group of the term technique unconsidered, appears arbitrary before this background. As

a necessary and at the same time sufficient criterion for the decision whether a subject should or not be patentable, the technique term therefore appears unsuitable, because detrimental to the requirement of the judicial security. As a necessary characteristic beside others it appears quite meaningful and in line with the legal tradition of many years of the German patent right.

4 The Objective Subject of Protection

§1 par. 3 itself already points out another way to patenting software, by declaring

Paragraph 2 opposes the patentability only to that extent, as protection is desired for the mentioned subjects or activities as such.

Since the request for protection is specified by the patent claims (§34 par. 1 (2) PatG), the logical reaction is to claim derived subjects instead of these concrete subjects. Instead of the concrete programme only the working procedure to be caused by the programme in a computer is claimed as a requirement for effect.

Also the subject, which was the basis for the decision 17 W (pat) 69/98 of the federal patents court in the patent claim 1, was generated after this pattern. The description of the desired working procedure however always precedes the actual coding in the programme development and is mostly done at the beginning of the design phase. The description of the working procedure therefore corresponds to the “non-technical concept” thus specified by Melullis, to which hr would not like to grant any patent protection. The concept for the preparation of a programme is nothing else like a plan for a mental activity . A patent on that would forbid others to think in the same way.

The claimed working procedure can also be rated as a requirement for effect with exclusive functional characteristics. Schulte explains:

Functional features define parts of the subject of the protection request not by physical characteristics, but by the indication of the effect or by properties. They are to be rated as technical features which can justify the novelty. Such indications of effect or design principles are a generalisation of the theory, because all means of equal effect are placed under protection, while the concretely revealed means are only examples. They are permissible, if by the indication of the effect, the property or the principle: i) a technical theory is sufficiently revealed; ii) the state of technology permits the generalisation of the theory; iii) the clarity of the claim does not suffer; iv) the feature objectively cannot be described more precisely.

For a computer programme however the effect caused by it cannot normally be regarded as sufficient revealing, because the solution of the task “How do I teach that to my computer?” is not revealed. In any case the program code describes the solution of the task objectively and more precisely.

5 The Exclusion Catalogue as Legal Disclaimer

It is also frequently tried to let the term “programme as such” appear unclear or to redefine it in order to minimise the number of the subjects excluded from patent protection thereby. But the following argumentation appears more convincing to the author: If for the subjects or activities mentioned as such no patent protection can be requested, this corresponds to a legally imposed disclaimer. In the case of programme related inventions this would have as a consequence that a programme could never infringe a patent as such, since as such no protection is entitled to a programme. Also a subject well-known from the state of technology, which executes only a new (according to §1 (2) 3rd not inventive) programme, cannot infringe a patent, since “the violation form does not represent any patentable invention vis-a-vis the state of technology”. An infringement of a patent is thereby ruled out. Thus the possibility of a free planning of the programmers would be ensured. Much suggests that the law maker explicitly wanted to protect just this free planning possibility; just like he had it foreseen for plans, rules and procedures for mental activities.

A comparable argumentation develops also Melullis for the mental logical concept as the basis for computer programmes and explains for this:

With this view the programmes for data-processing systems represent only a further manifestation of the facts, in which also conceivable regulations and proposals for the human behaviour can be recorded, with which the justification of an exclusivity right could mean a danger for the freedom of thinking.

He grants however only a subordinated role to the guarantee of this freedom, so that he can justify the patent protection of executable programmes.

However, if one attributes at least equivalent role to the protection of this freedom of thinking and of programming - as does the author - then the exclusion list of the §1 PatG would be understood in such a way that subjects or activities mentioned may not be prevented by the patent right; independent of whether they demonstrate a theory for technical acting or not. This aspect seems to be shared also the majority of the opponents of software patenting who see the booming progress of their economic sector endangered by patenting software - no matter whether in the form of ideas, concepts, algorithms (mathematical methods) or in form of the concrete source code. The high number of signers of the EuroLinux petition confirms this. In computer sciences as well in the computer industry the driving force for development is the disclosure of the own development and the improvement of existing programmes. Without the quick and unhindered distribution and discussion of the “Requests for Comments”, which form the basis of the Internet, today’s boom of the software industry is inconceivable. The substantial character of the scientific development and of the formation of opinion has been described very plasticallt by Wilhelm Busch:

Science is and remains what one copies from the other.⁶

⁶In German this rhymes very nicely.

Programmes in form of source code as linguistic works live on the fact that they can be quickly adapted to the respective requirements. Only with this freedom there can be free lance programme developers - as there are free lance doctors, artists or writers.

But back to the view of the exclusion catalogue as legal disclaimer. It would be certainly interesting to see, how a violation judge would react if the defendant indicated for its defence, that he only programmed a “programme as such”, based on the conviction well-founded by the legal wording that “programmes as such” cannot violate any patents, since on “programmes as such” no patent protection can be granted.

It was and is obvious that for each programming the programme must be stored, tested and thus a working procedure must be executed. Therefore a patent protection aimed at these partial aspects of the programme development can just as little be wanted by the legislator. If the aforementioned argumentation is perhaps not able to convince everybody, it must however raise at least justified doubts. If sometimes in dubio per inventore is demanded, one can demand in this case with at least the same right in dubio per reo, even if doubts in the patent right are not appropriate. Whether the argumentation mentioned is logically perfect, may be elaborated in public discussion or on a concrete individual case.

6 Source Code in the Patent Claim

If one wants to regard programmes for data-processing systems themselves as inventions and to make these accessible for patent protection as such - as this is intended with the planned novel of the EPA - then one should think more precisely what could be the consequences.

If one regards the definitions of a programme quoted above again, one recognises that all instructions understandable for a computer which solve a task represent a programme. The specialist knows programmes in binary form and/or their description in Hexcode, as well as programmes in one of the many programming languages (Java, C++, Tcl/Tk, Perl, Lisp, Phyton, Postscript, HTML, TEX, Ada, Fortran, Pascal, Basic, Cobol,...). Also combinations of graphic programming with textual supplements are known (LabView and others). If the programme is executable on a computer and if it causes the procedure to be implemented, it is to be regarded as solution of the task, all the same whether it is executable by the processor directly - or only with the help of further well-known programmes (operating system, compiler, interpreter, emulator, virtual machine,...). Together with the indication, under which conditions this programme is executable (type of processor, operating system, program libraries or what else may influence executability) the programme code is, in the opinion of the author, even the only unambiguous revealing form of the solution of the task (by Brockhaus it is called “clear cut”, by britannica.com “unambiguous”).

7 Concluding Remark

As one can see at this long enumeration of problems with the present patenting of programmes for data-processing systems, there would be more than sufficient controversy reasons for objection and nullity proceedings. The fact that simultaneously with the embittered discussions between proponents and opponents no true tide of objections and processes against the present practice of the patent administration for programme related theories is happening, can only surprise. It is possible that the pain threshold of the software developers is not exceeded yet. At the latest if the patentability of software is approved by the legislator, however, this pain threshold will be reached.

It is also possible also that none of the sides can understand the other any more, since both proceed from incompatible principles. An indication for the latter may be the following episode: At the helpless attempt of the author to explain to a befriended computer scientist, why programmes would not be patentable as such, programme related theories with an additional technical effect however would be, he reacted bluntly: "You are completely nuts!" This clear pronouncement caused the author to think quite a bit.