union

OF EUROPEAN PRACTITIONERS IN INDUSTRIAL PROPERTY
EUROPÄISCHER BERATER FÜR DEN GEWERBLICHEN RECHTSSCHUTZ
DES PRATICIENS EUROPÉENS EN PROPRIÉTÉ INDUSTRIELLE

Registry of Enlarged Board of Appeal
European Patent Office
Erhardtstrasse 27
80331 Munich
Germany

Brussels, 27 April 2009

Re: ***Case G3/08:***
Referral by the President of the EPO to the Enlarged Board of Appeal
Patentability of programs for computers

Dear Sirs,

The UNION of European Practitioners in Intellectual Property is a long established group who are keen to maintain a system which is well balanced between the interests of inventors and possible owners of intellectual property rights and those of competitors or alleged infringers who may run into difficulties with the intellectual property rights of others. The rules governing the system should also be made as clear as possible.

As practitioners, we believe an effort towards balance and clarification is desirable in the field of software based inventions.

We therefore submit the attached *amicus curiae* brief on the questions asked by the President of the EPO.

Yours faithfully,

Philippe OVERATH
President of Union

Philippe OVERATH, President of UNION
c/o Cabinet BEDE, Bd General Wahis, 15, B-1030 BRUSSELS
Tel.: +32 2 779 03 39 Fax: +32 2 772 47 80 E-mail: union@bede.be

Union is an association of more than 600 (independent and employed) practitioners in the field of intellectual property in Europe. As a major independent European association of practitioners, Union felt it appropriate to deliver an *amicus curiae* paper in the referral G 03/08 to the Enlarged Board of Appeals.

This *amicus curiae* paper has been prepared by Union's Software Commission, whose eight members are experts in the field of intellectual property protection for software.

The present paper is organised into a section A of preliminary observations, which set out useful background and definition on which the answers to the questions are based. After the preliminary observations, the answer to each question is given in section B, with appropriate comments where relevant.

A. PRELIMINARY OBSERVATIONS

A1 – A little history ("un peu d'histoire")

A1.1 - Where does the exclusion of computer programs come from?

Historically, the exclusions appeared initially in the French patent law of 1968. Most of the exclusions were inspired by the idea of preserving certain freedom for the public, by avoiding the existence of a broad monopoly on certain knowledge. Hence the exclusion of

Les méthodes financières et comptables, les règles de jeux, et tous autres systèmes de caractère abstrait, et notamment les programmes ou séries d'instructions pour le déroulement des opérations d'une machine calculatrice.

According to the rumour at that time, the exclusion of computer programs was a last minute addition to the exclusions. After some hesitation, it was added to the "systems of an abstract character" section. In fact, it was to avoid the threat of patent descriptions based on printed program code listings, and having to carry out searching on those strange words (the code) fixed on paper. Hence the idea of classifying programs within "systems having an abstract character [or nature]". In fact, program code is disembodied, but not abstract.

Besides, in the late 1960s, the programs were "seen" only through paper. There came the institution of a copyright protection on the "text" of the program[1]. Thus, the exclusion of computer programs from patentability was also intended to avoid interference with copyright protection of the program code, and also with other IP rights, like *aesthetic creations* and *presentations of information*.

[1] In the professional computers being current at that time (no personal computers existed), the user input interfaces were punched tapes or cards, and the user output interface was a printer.

In this context, the primary scope of the exclusion was "no claim reciting program code". The purpose was to avoid direct conflict between patent protection and copyright (or author's right) protection.

On another hand, excluding computer programs looks much like a foreign body amongst the other exclusions.

Also, the exclusion of e.g. a *business method as such* apply to the method being claimed as a whole. By contrast, the exclusion of computer programs apply to the features contained in the claim.

This indicates that the exclusion of computer programs must be considered separately from the other exclusions. This is exactly what the G 03/08 referral does.

A1.2 - Links with other exclusions?

A1.2.1 - In the text of the EPC, the exclusion of computer programs is simply added to the list of *schemes, rules and methods for performing mental acts, playing games or doing business, [and programs for computers].* There is no more reference to *"systems of an abstract nature"*, as was done in the French patent law 1968.

Attempts were nevertheless made to link the exclusion of computer programs to other exclusions.

A1.2.2 - It has been suggested that programming is a *mental act*. This is irrelevant. All or almost all inventions are the result of thinking, i.e. of *mental acts*.

Furthermore, what is excluded in Art. 52.2.c is not a mental act, but a *method for performing a mental act.* A computer program cannot be excluded as being a *method for performing a mental act* as such.

The exclusion of a *computer program as such* is the only one to be considered.

A1.2.3 - Writing a computer program is not mathematics.

According to other statements, a computer program was understood as a *"mathematical application of a logical series of steps in a process which was no different from a mathematical method".*

This "*mathematical*" approach is wrong at the level of <u>designing</u> software. If it were true, some mathematical robot would be able to write all the software produced in the world. Nobody ever asserted this. What is true is that it is possible to mathematically "verify" what a given <u>existing</u> software does, based on the fact it uses a collection of manmade tools designed to obtain fully predictable results, within certain boundaries. This possibility of verification, however, is irrelevant at the level of <u>designing</u> software.

A1.3 - Practitioners' recollections

In practice, it appears that nobody did file patent claims reciting program code[2].

Also, in early times, many inventions could be implemented both in software form (potentially faced with the exclusion) and in hardware/firmware form (raising no problem of exclusion)[3]. The patent claims were naturally covering both cases. An abrupt exclusion of whatever contains software would have resulted e.g. in:

- Rejecting/invalidating claims simply because they may be implemented through software[4]; or

- Forcing patent owners to supplement such a claim with a disclaimer of whatever is software; or

- Worse, implicitly inferring into such claims an exception in their scope to the effect they do not cover whatever is software[5].

None of these consequences is acceptable. Thus, a number of practitioners came to the conclusion that the exclusion of computers is inappropriate, and tended to ignore it, considering only the other exclusions.

Besides, the exclusion had the harmful effect that many European players did not seek patent protection in Europe for their software based inventions, while non Europeans did.

In this context, Union raised a debate on the subject in 1997[6]. The conclusion was towards deleting the exclusion of computer programs from patentability, or, at least, that the "computer program as such is the program code".

As noted in the Referral, there came thereafter a proposal for revision of the EPC. The idea of deleting the exclusion of computer programs from patentability was considered, however not retained, pending the forthcoming EU Directive. A change was made that inventions could be "in all fields of technology". Union expressed its position during the discussion of the EU directive[7].

While the draft directive did not mature, it raised an extended public debate. A number of European players then realized that they could seek patent protection in Europe for their software based inventions.

In parallel, the field of software based inventions has expanded, up to e.g. bioinformatics.

---

2 Or, if some people did file claims aiming at program code, these claims were rejected without any discussion

3 This situation still exists nowadays.

4 Mobil Oil v/ INPI, Cour d'Appel de Paris, 22th May 1973, later reversed by Schlumberger v/INPI, Cour d'Appel de Paris, 15th june 1981.

5 This tendency re-appeared during the debate on the draft EU Directive

6 Union Roundtable on the patentability of computer programs, EPO, Munich, December 9th/10th 1997.

7 Union's Position on the Proposal for a Directive on the Patentability of Computer-Implemented Inventions, 9th May 2005.

Union also hosted a debate on this related, albeit more specialized, subject[8]. At this occasion, the debate extended to enforceability of bioinformatics patents, as reported in another Union Roundtable[9].

Thus, another crucial aspect is enforceability. In this regard, the claim category is essential, as discussed in section A3 below.

A2 – Are method claims necessary for Software?

A2.1 - Different kinds of method claims for computer programs

When using method claims for computer programs, the claim wording will be functional, thus presumably easier to understand. By contrast, it may become excessively general, thus standing far from the real core of the invention. These claims are now considered by decreasing degree of generality.

The upper degree of generality is a method claim reciting an overview of what the computer "does" when it runs a program[10], however without reference to software objects. In this case, the invention is elsewhere than in the underlying software. Claim 1 in T 258/03 (Hitachi) is largely of that kind.

The lower degree of generality is a method claim reciting how the computer handles software objects when a program is run. Examples may be found in the main claims of T 1173/97 (Computer program product - IBM), T 1177/97 (SYSTRAN), or T 424/03 (Clipboard formats - MICROSOFT).

Claims having that lower degree of generality deserve a special treatment, to determine whether they go beyond a *computer program as such,* i.e. whether they involve a *further technical effect*.

A2.2 - Do computer programs always necessitate method claims?

The Referral seems to admit that method claims are the standard claim category to be used in software based inventions. This is apparently related to the definition of a computer program as a "*series of steps*". From the very beginning, the word "series" has been interpreted to mean that the operation of the computer is essentially sequential. Hence the idea that method claims must be used.

Nowadays, most of modern computer programs are no longer sequential as a whole. They should rather be viewed as a main program or loop managing a collection of tools (individual procedures or functions). This main program or loop has no interest in itself.

Moreover, in an event-driven object-oriented language, there is no real series of operations. Instead, the user decides what to do from the Graphic User Interface (GUI), in the order he/she wants. For example, a word processor program cannot be adequately

---

8 Union Roundtable on bioinformatics and IP, Paris, 16th February 2007.

9 Union Roundtable on IP enforcement in Europe, Munich, 9th March 2007

10 or even merely a method depicting what the user "does" and "sees" when the computer runs a program.

described by a method. It is a collection of tools or functions, which one can use in the computer when the program is executed[11].

In these cases, only the individual procedures are sequential. Often, these procedures will be called repetitively. This results in an extremely complicated method claim. A typical example is claim 1 of T 1177/97 (SYSTRAN). This claim actually refers to a collection of tools, whose interactions would have been more clearly defined in a system and/or computer program claim.

Method claims may still be adequate in certain circumstances. However, they should not be considered as the standard claim category to be used in software based inventions.

Also, since only the individual procedures in a computer program are sequential, today's computer programs are only locally sequential. This should be acknowledged by authorizing steps locally in a computer system or computer program claim, rather than systematically using method claims.

A3 – The consequences of the claim categories

A3.1 - Claim categories and enforcement

Considering first the claims in the categories "Methods..." or "Computer-implemented methods...", the direct infringer of such claims is the end-user[12]. Thus, end-users may be found unconditionally responsible for the infringement, while they often do not even know what happens in their computer.

In the field of software based inventions, it is common that the hardware and the software reach the final user through different commercial channels, and/or are sold separately. Then, claims in the categories "Systems (i.e. computer systems)", or "Apparatuses" or "Devices", or the like are in the same situation as method claims. Again, the first-line infringer is the end-user.

Although end users may happen to be involved in infringement actions, patents are basically a matter between the professionals in the field of the claimed subject matter, i.e. between the patentee and the persons who designed the would-be infringing method or system ("primary designers"), possibly their distributors, normally not good faith end-users.

However, claims in the categories "[Computer-implemented] methods..." or "[computer] systems" are only indirectly infringed by the primary designers and the distributors.

Thus, it would not be sound to restrict patent owners to write claims (method and/or system) that are basically applicable to the end user only. Claims directed to "computer program [products]" are necessary for the correct operation of the patent system considered as a whole, ranging from the acquisition of rights to the exercise of rights.

---

11 in contrast with a series of steps (instructions) which will be carried out by the computer when the program is executed (Referral, section 2, "Definitions")

12 Proceedings of Union Congress, Porto, Portugal 29th-31st May 2008; "Computer-Implemented Inventions; Indirect Infringement Issues" Peter Hanna

A3.2 - Computer program [product] claims

In this regard, T 1173/97 (Computer program product/IBM) represents a major advance, and is likely the most important decision in the field. According to T 1173/97, the difference between the computer program product being claimed and an excluded *computer program as such* resides in the fact that the computer program product being claimed has a *further technical effect*, which lies beyond the intrinsic technical effect of any program[13].

T 1173/97 applied this criterion of *further technical effect* for the purpose of allowing computer program product claims (the only question submitted to the Board).

A key issue is whether the same criterion of a *further technical effect* should also be applied to determine whether claims directed to computer implemented methods or to computer systems/apparatuses fall under the exclusion, or not.

Apparently, this is not the current leading approach of the Boards of Appeals, where method/system claims are considered technical (therefore not excluded) from the time they comprise a technical means, whether novel, or not[14].

This situation generates divergence in case law, and should be resolved. There does not seem to be any objective reason that computer program product claims, and computer implemented methods or computer systems/apparatuses claims be treated differently.

A4 – Technical aspects

A4.1 - What a computer program is

According to the referral's definition, a computer program is a series of *steps (instructions) which will be carried out by the computer when the program is executed.*

This definition corresponds to a low-level view of how a computer program interacts with the hardware when it runs. It may be adequate for lower level programming languages, e.g. assembly language. Manipulating these languages indeed requires a good understanding of how the underlying hardware works.

By contrast, modern high level languages rely on a number of underlying software layers, which take care of most of the interaction with the hardware. Such high level languages may be viewed as a collection of manmade tools designed to obtain fully predictable results (within certain boundaries).

Writing a piece of software using such a high level programming language is work, often hard work.

However, in most cases, writing a piece of software is nothing more than correctly using a small subset of the extensive predefined possibilities offered by the high level language. In

---

13 which goes beyond the "normal" physical interactions between the program (software) and the computer (hardware) on which it is run"

14 In general, a method involving technical means is an invention within the meaning of Article 52(1) EPC. (T 258/03 – Hitachi)

that respect, considering the piece of software itself, there is nothing new. In other words, the piece of software is intrinsically technical, but offers no technical novelty.

There are exceptions, e.g. where the software writer has to:

a. consider what happens underneath the high level programming language, e.g. in the hardware, e.g. to supplement the high level programming language with additional capabilities, and/or
b. consider what happens outside the computer, i.e. look at the interactions with the external world.

The *further technical effect* of T 1173/97 reflects these exceptions.

A4.2 - What is more than a computer program as such?

This is a matter of claim interpretation.

A patent claim may be explicitly referring to computer programs, by reciting "software objects", like the *commit procedure* and *resynchronization* of claim 1 in T 1173/97 (Computer program product - IBM), the *clipboard formats* of claim 1 in T 424/03 (Clipboard formats - MICROSOFT), or the *offset address linkages* of claim 1 in T 1177/97 (SYSTRAN). Clearly, software technicalities are involved in such cases.

On the contrary, a claim may be referring to a computer program only implicitly, when it states it indirectly, e.g. by headings like "Computer implemented method", or "Computer system", or "computer program [product]", with no software object being recited in the claim. Most of these claims are expressed in terms of results obtained using a computer and software. There is no software technicality being involved.

According to this view, the software aspects of the claimed subject matter qualify as software as such when no software technicalities are recited in the claim.

In such a case, there is no teaching as to programming the computer to obtain the desired result. Thus, the programming is deemed to be the suitable use of a high level programming language. There is no *further technical effect* deriving from writing such a program. A *further technical effect* may nevertheless exist for other reasons, e.g. the interactions of the functions being claimed, and their computer implementation.

By contrast, where technicalities of the software are recited in the claim, there are teachings as to programming the computer to obtain the desired result. Thus, there is a *further technical effect*, in the sense of T 1173/97(IBM), at least potentially[15]. In such a case, the novelty and inventive activity analysis is required, as also suggested in reason 8 of T 1173/97.

Prima facie, it seems paradoxical to consider that a claim is not software as such where it involves software technicalities. The appearances are in fact misleading. A patent claim is intended to define the novel and inventive features of an invention, rather than the substance of a computer program.

---

[15] It should be reminded that reason 8 of T 1173/97(IBM) does not require that the further technical effect be novel.

When the claim does not refer to software objects, this means that the substance of the underlying implemented program does not matter. There is no further technical effect due to the software implementation of the invention in itself. The further technical effect, if any, has to be found elsewhere.

By contrast, when the claim does refer to software objects, this means that the invention does involve software considerations, which are related to the underlying software or hardware, and/or to an interaction with the external world. An analysis of the novelty and inventive step of the claimed subject matter is then appropriate.

A5 – The situation

A5.1 - What should be excluded?

The primary scope of the exclusion, i.e. "no claim reciting program code", is of course applicable, however, with no practical effect, since no such claims are in discussion.

This situation clearly induced a tendency to look for a further scope of the exclusion. Some people question whether this tendency is correct. In other words, is it really necessary to interpret the exclusion as meaning more than "no claim reciting program code"?

In using the concept of *further technical effect*, T 1173/97 (Computer program product - IBM) suggests that the exclusion might mean more than "no claim reciting program code".

By contrast, the BOAs now rarely use the exclusion, and prefer to declare a lack of inventive step.

A5.2 - Current practice of the BOAs

Since T 258/03 (Hitachi), the Boards of Appeals now examine novelty (mixing technical and non technical features of the claim) and then restrict the claim to its technical features when turning to inventive step[16].

There is little or no criticism as to whether the final decisions of the BOAs are technically correct.

By contrast, the approach may be found disputable, for several reasons.

First, the mere fact to appreciate novelty on the basis of both the technical and non technical features of the claim seems to work as a *de facto* bar to apply the exclusion. Indeed, the recent BOA decisions rarely apply the exclusion[17].

There is no reason that national practice and case law follow this EPO approach. A tendency outside the EPO is to consider that the exclusions should be appreciated first, before the usual novelty and inventive activity discussion. This seems to be the case in the UK and in France (*ex parte* cases), at least.

---

[16] In its reason 9, T 154/04 states: *an invention meeting all criteria of patentability, must provide a novel and inventive technical contribution to the prior art.* But the rest of the analysis does not really depart from Hitachi; see e.g. Reason 22.

[17] beyond the primary approach of "no claim reciting program code"

After denying the exclusion, the BOAs often find a lack of inventive step, based on a purely affirmative statement like:

"*Still, the Board is convinced* [why ?] *that this way of ranking the bids is a routine programming measure well within the reach of the skilled person* [in which field ?]"[18]

This has several consequences:

- At the EPO, there is no Supreme Court, in charge of revising such affirmative statements on the merits. This is implicitly deferred to the National Courts, which have no real possibility to understand the basis of such merely affirmative statements of an EPO Board of Appeals.

- The Examining divisions will also tend to declare a lack of inventive activity based on similar affirmative statements, which prove to be hard to dispute other than by mere negation on the side of the applicant. Hence a tendency to systematically revert to the competent Board of Appeals.

Also the affirmative statements often mix several excluded matters together. An example from T 258/03 is quoted above. Another example may be found in T 1177/97 (SYSTRAN). After identifying in the claim a "matching principle" different from that of the closest prior art, the Board concludes:

Applying the principles laid down by the Board in its COMVIK-decision cited above (see headnote II), the decision for one or the other matching principle does not seem to solve any technical problem and hence does not fall within the responsibility of a technically skilled person. It is rather a non-technical constraint determined by the linguistic expert and given to the skilled person as part of the framework of his task, namely implementing the known low frequency dictionary look-up process by applying the "longest match principle".

Choosing to apply the one or the other principle has clearly consequences for the technical implementation of the translation process since the computer routines have to work differently and the automated translation process will produce objectively different results, technical differences which establish novelty. <u>These technical differences, nevertheless, are not inventive since they originate from a non-technical constraint to the technical problem, the implementation of which is obvious.</u> (emphasis added).

Since *the automated translation process will produce objectively different results*, it appears that the underlying program is not only novel, but also improved. After having observed these facts, it seems strange to declare obviousness, while discarding the *linguistic expert* from *the skilled persons*.

In other words, the technical novelty is established. After that, the reasoning concerning obviousness is mixing software considerations and linguistic considerations (a would-be excluded matter). However, none of the *exclusions as such* is applied. It would then seem

[18] T 258/03 (Hitachi), reason 5.8.

more logical to consider both the software expert and the linguistic expert in assessing inventive step[19].

A5.3 - The exclusions, seen together

It seems necessary to consider not only the exclusion of computer programs, but also the exclusions as a whole, since many of the quoted decisions were potentially involving several exclusions.

In Art 52.2 EPC, most of the exclusions refer to the overall function or result of the claimed subject matter: is it a mathematical method as such? Is it a method for performing mental acts, playing games or doing business as such? Is it a presentation of information as such?

The exclusion of computer programs is different. This is because it does refer to the content of the claimed subject matter, rather than to its overall function or result. The question is now: does the content of the claimed subject matter qualify as a computer program as such?

This suggests that the exclusion of computer programs should be appreciated differently from the other exclusions, and maybe separately from the other exclusions.

A common situation is a claimed subject matter in which business oriented features are computer implemented, e.g. T 258/03 (Hitachi).

The board stated that the software aspects were "*a routine programming measure well within the reach of the skilled person"* in the field of software. There remains the question whether the subject matter of claim 1 as a whole qualifies as a business method as such[20]. The response depends upon the existing interactions between the business aspects and the software aspects of the auction system being claimed. These interactions may show a further technical effect ("interactive further technical effect") in the sense of T 1173/97.

There were two possibilities:

a. either there is no interactive further technical effect; claim 1 is a business method as such, and the patent was to be rejected under art. 52(2)(c) ["business method"] and (3).

b. or an interactive further technical effect is recognized; claim 1 is not a business method as such, and the traditional novelty and inventive step criteria were to be applied, considering also the contribution of the auction specialist (what T 258/03 did not).

Having considered these preliminary observations, we will now turn to the answers to the specific questions.

---

[19] assuming the non-technical constraint determined by the linguistic expert ... namely implementing the known low frequency dictionary look-up process by applying the "longest match principle" was not publicly given to the software specialist, i.e. part of the state of the art.

[20] The approach taken in T 258/03 ignores this question, on the ground that the programming is "routine".

B. ANSWERS TO THE QUESTIONS

B1 - QUESTION 1

Q1 - CAN A COMPUTER PROGRAM ONLY BE EXCLUDED AS A COMPUTER PROGRAM AS SUCH IF IT IS EXPLICITLY CLAIMED AS A COMPUTER PROGRAM?

In this question, the expression "*is explicitly claimed as a computer program*" is interpreted to mean that the claim heading refers to a computer program. The case where a computer program would be one of the means recited in the body of the claim does not match the condition "*excluded as a computer program as such*".

Our answer is NO, with emphasis on the word "**ONLY"**.

The exclusion should apply to what the subject matter of the patent claim actually is, rather than to what the heading of the claim is.

The national laws of many member states have a provision prohibiting delivery of an essential means to implement the invention. Such a provision has the effect that method claims or computer system claims permit to bring an infringement action against putting a computer program on the market, as an essential means to implement the invention. Accordingly, in respect of the exercise of patent rights, a computer program may be indifferently attacked on the basis of a [computer implemented] method claim or of a [computer] system claim, or of a corresponding computer program claim, as well.

Thus, reverting to patentability issues, the exclusion should refer to the actual contents of the claim, regardless of whether the claim heading is "a computer program", "a [computer implemented] method" or "a [computer] system".

The computer program is frequently put on the market alone. In such a situation, explicit *computer program claims* have the advantage to closely correspond to the "product" being commercialized. This puts the responsibility of the infringement where it actually is, i.e. on the software designer, rather than on the end user.

Comments on the cited case law in question Q1

The claimed subject matter in T 424/03 (Microsoft) refers to an enhancement in a programming language, involving technical considerations as to what happens in the computer, underneath the programming language. This is more than writing code which merely uses the programming language. Accordingly, there is a *further technical effect*. Thus the claims – whether seen as a method or as a computer program product – do not aim at a *computer program as such*.

Up to that point, there is no divergence between T 424/03 and T 1173/97.

In T 424/03, the technical character of the claim is confirmed (reason 5.2). Thereafter, there was no real need to refer to the rather artificial argument that a [computer implemented] method claim avoids the exclusion, but a claim directed to a computer program falls under the exclusion. Thus, the divergence is only apparent.

B2 - QUESTIONS 2

Q2A - CAN A CLAIM IN THE AREA OF COMPUTER PROGRAMS AVOID EXCLUSION UNDER ART. 52(2)(C) AND (3) MERELY BY EXPLICITLY MENTIONING THE USE OF A COMPUTER OR A COMPUTER-READABLE DATA STORAGE MEDIUM?

The answer is NO.

Seeing it differently would amount to examine any software based invention according to the traditional novelty and inventive step criteria. This would tend to deprive the exclusion of *computer programs as such* from any substantial meaning.

This answer refers only to the exclusion of *computer programs as such* under art. 52(2)(c) and (3).

Very often, the claimed subject matter extends beyond software considerations, and therefore does not qualify as a *computer program as such*. If so, reverting to the traditional novelty and inventive step criteria is appropriate, however after having determined whether the claimed subject matter as a whole does overcome the other exclusions.

Comments on the cited case law in question Q2

T 258/03 concludes to the lack of inventive activity, because:

"Still, the Board is convinced [why ?] that this way of ranking the bids is a routine programming measure well within the reach of the skilled person [in which field ?]".

In so saying, the board restricted its analysis to the person skilled in software programming, and implicitly eliminated the question whether the *way of ranking the bids* might be a part of the invention.

The board's statement means that the programming is deemed trivial, i.e. the mere use of a programming language for its predefined possibilities. Indeed, there is no reference to software objects in claim 1 of the main request. In other words, there is no *further technical effect* related to the software aspects of the claim.

According to the approach in T 1173/97, there might be a *further technical effect* related to the interaction between the business aspects of the claim and its software aspects (or even related to its business aspects in themselves). If so, the traditional novelty and inventive step criteria was to be applied, considering also the contribution of the auction specialist.

T 258/03 indeed went to the novelty and inventive step analysis, however on a different ground, namely the fact that mentioning a computer in the claim avoided the exclusion (of *computer programs as such*). Thus, the board did not consider whether the combination of business aspects and software aspects as claimed had a (further) technical effect. It also ignored the contribution of the auction specialist, implicitly applying the exclusion of a business method (as such ?) at the level of inventive activity.

The approach taken in T 1173/97 would have been preferred, examining whether the software aspects and business aspects together had a *further technical effect*.

Q2B - IF QUESTION 2 (A) IS ANSWERED IN THE NEGATIVE, IS A FURTHER TECHNICAL EFFECT NECESSARY TO AVOID EXCLUSION, SAID EFFECT GOING BEYOND THOSE EFFECTS INHERENT IN THE USE OF A COMPUTER OR DATA STORAGE MEDIUM TO RESPECTIVELY EXECUTE OR STORE A COMPUTER PROGRAM?

The answer is YES.

Now, where the claimed subject matter extends beyond software considerations, a *further technical effect* may exist beyond the software aspects of the claim. If so, the traditional novelty and inventive step examination should apply.

B3 - QUESTIONS 3

Q3A - MUST A CLAIMED FEATURE CAUSE A TECHNICAL EFFECT ON A PHYSICAL ENTITY IN THE REAL WORLD IN ORDER TO CONTRIBUTE TO THE TECHNICAL CHARACTER OF THE CLAIM?

In view of question Q3B, the computer on which a computer program is run is seen as a part of the "real world". Subject to this interpretation, the response is YES.

In other words, the *further technical effect* of T 1173/97 may reside in an effect of the real world external to the computer, and/or in an effect within the computer, including an enhancement of a programming language (see e.g. T 424/03).

Comments on the cited case law in question Q3

In both decisions **T 163/85** and **T 190/94**, the claimed subject matter system was found to make a contribution to the art in a field not excluded from patentability because the difference (of the claimed subject-matter over the prior art) manifested itself in the real world in a technical effect on a physical entity. This is sufficient to avoid the exclusions[21].

As already noted, **T 424/03** shows a further technical effect in a software based invention (see comments on Question 1). Thus, there is no problem with the technical character.

On the contrary, the software aspects in **T 125/01** are the mere use of a programming language (with single or multiple tables). By contrast, the application is specific: a control unit for a telecommunications apparatus, e.g. a car radio. The traditional novelty and inventive step examination should apply (as the Board did).

Thus there is no real divergence.

[21] however not necessary, as suggested by the phrase “a technical effect on a physical entity in the real world was required” in the referral

Q3B - IF QUESTION 3 (A) IS ANSWERED IN THE POSITIVE, IS IT SUFFICIENT THAT THE PHYSICAL ENTITY BE AN UNSPECIFIED COMPUTER?

The expression "*unspecified computer*" is somewhat ambiguous. In view of question Q3C, it is understood to refer to a computer that is not precisely defined by its manufacturer/model, and/or by its hardware ingredients, and/or by its hosting capabilities in terms of underlying software, including e.g. the operating system (the " environment").

In other words, this question apparently refers to a software based invention which may work on a variety of different computers, and/or computer structures. This is a rather common situation nowadays. Furthermore, a given same software environment may be hosted in very different computers. For example, a Windows™ environment is available on the so called PCs, but may also be emulated on Apple™ machines. Similarly, Linux™ may be emulated on almost any machine.

In this interpretation of Question 3B, the answer is YES, i.e. the claim does not have to specify the computer. In particular it does not have to specify the computer manufacturer/model, or its hardware ingredients, or its computing capabilities, or its processor, or its operating system.

Q3C - IF QUESTION 3 (A) IS ANSWERED IN THE NEGATIVE, CAN FEATURES CONTRIBUTE TO THE TECHNICAL CHARACTER OF THE CLAIM IF THE ONLY EFFECTS TO WHICH THEY CONTRIBUTE ARE INDEPENDENT OF ANY PARTICULAR HARDWARE THAT MAY BE USED?

Although the response to question Q3A is positive, it is nevertheless felt appropriate to answer Question Q3C.

The answer is YES.

For example, an invention may enhance the amount of working memory (RAM devoted to data) versus the amount of memory devoted to the storage of the program (RAM devoted to program). The effect exists independently of the underlying hardware that is being used.

B4 - QUESTIONS 4

Q4A - DOES THE ACTIVITY OF PROGRAMMING A COMPUTER NECESSARILY INVOLVE TECHNICAL CONSIDERATIONS?

In this question, the expression "technical considerations" may refer to the technicality of programming in general, or to a *further technical effect* beyond the programming technicalities.

As noted in A4.1 above, in most cases, writing software is nothing more than correctly using a small subset of the extensive predefined possibilities offered by a high level language. Such programming technicalities are not "technical considerations" in the sense of T 769/92 (SOHEI). Furthermore, if programming technicalities were being aimed at, the response would be necessarily Yes, and the question would be meaningless.

Accordingly, the expression "technical considerations" is understood not to include the technicality of programming in general, but to aim at "technical considerations" in the sense of T 769/92.

The answer is NO.

The activity of programming a computer may or may not involve technical considerations. This is correlated to the question whether the claimed subject matter shows a further technical effect, or not.

Comments on the cited case law in question Q4

If the claimed subject matter does not involve software objects, it will hardly involve a further technical effect, or technical considerations, in the field of software.

Claim 1 in decision **T 1177/97** clearly involves software objects: see e.g. the *offset address linkages.* In this case, the activity of programming does involve technical considerations.

Older decision **T 769/92** incidentally considers programming to be a mental act, but relies on the existence of "technical considerations" during software design[22] to recognize patentability.

By contrast, the main claims in **T 172/03, T 833/91 and T 204/93** do not show any software object. They may therefore be considered excluded subject matter.

Incidentally, the mental act approach raises several objections:

- All or almost all inventions are the result of thinking, i.e. of mental acts ;

- It is not appropriate to confuse the exclusion of *methods for performing mental acts as such* with the exclusion of computer programs as such ;

- after T 1173/97 has recognized the concept of *further technical effect*, it seems difficult to consider that programming in general is a pure mental act.

Interestingly, **T 769/92** makes a distinction between the level of "computer programming" and the prior level of software design[22], in which it finds the possibility of "technical considerations". These two levels may be merged in practice. So, it seems better to simply look for "technical considerations" in the claimed subject matter.

Q4B - IF QUESTION 4 (A) IS ANSWERED IN THE POSITIVE, DO ALL FEATURES RESULTING FROM PROGRAMMING THUS CONTRIBUTE TO THE TECHNICAL CHARACTER OF A CLAIM?

No need to answer.

[22] “a stage of activities involving technical considerations to be carried out before programming can start”

Q4C - IF QUESTION 4 (A) IS ANSWERED IN THE NEGATIVE, CAN FEATURES RESULTING FROM PROGRAMMING CONTRIBUTE TO THE TECHNICAL CHARACTER OF A CLAIM ONLY WHEN THEY CONTRIBUTE TO A FURTHER TECHNICAL EFFECT WHEN THE PROGRAM IS EXECUTED?

The answer is YES

Features resulting from programming indeed can contribute to the technical character of a claim when they contribute to a *further technical effect*.

Such features resulting from programming may have an effect within the computer. They may also have an effect in the real world external to the computer. Accordingly, the *further technical effect* may happen in the real world external to the computer, or within the computer, or both.