Those comments on the Commission consultation paper, are being e-mailed to:
MARKT-SOFTPAT@cec.eu.int
ASPI agrees that such comments may be published on the Commission's web site "as is".

The "Association des Spécialistes en Propriété Industrielle de l'Industrie" (ASPI, with its seat at 20, rue Vignon, 75009 Paris, France) is an association of almost 500 IP professional individuals employed in all kinds of industries in France (as opposed to those IP professionals working in private practice). Few of them are employed by companies doing business in the information technology industry. ASPI members do not have any financial interest in promoting or denying patentability of software-related inventions. Their opinion is merely that of skilled specialists in the industrial property field. It is as follows:

I - There is no justification for excluding inventions from patentability for the reason that they involve, or are in practice implemented in the form of, a computer program:

a) A computer program is of its nature "technical". It is written by engineers or technicians. There seems to be little reason *a priori* why they should be refused protection by patent. The work of those who create computer programs is a technical activity for which patent protection should not be denied, unless, like any other invention, the computer program is not new, or does not involve an inventive step. A computer program is a product which performs a specific function, task or result, just like any other product.
b) Going back to the reasons for the exclusion of A52(2)c) *in fine*, it appears that, back in 1971 and the drafts to the EPC, they are derived from Rule 39(1) PCT, itself inspired by certain national patent offices who were not comfortable at the prospect of having to, as International Searching Authority, search prior art among computer programs (it is to be noted that in the list of subject-matter excluded from searching under Rule 39(1) PCT, computer programs is excluded only "to the extent that the International Searching Authority is not equipped to search prior art concerning such programs"). In other words, the exclusion in the EPC was mainly, if not exclusively, included in the legislation for administrative reasons. To now represent reasons for the exclusion otherwise, for example to state that it resulted from a deeply-rooted ideological or economical European thinking or belief, the removal of which would be a major change of policy, is misrepresenting the real situation. Further there is no homogeneity, semantically or conceptually, in the list of A52(2) EPC: it indeed includes "programs for computers", a concrete object "causing a machine having information processing capabilities to indicate, perform or achieve a particular function, task or result", in a list of otherwise abstract notions, or activities bringing about no "function, task or result": discoveries, theories, mathematical methods, aesthetic creations, certain schemes, rules, or methods for performing mental acts, and presentation of information.
c) There is no convincing evidence that confirming the patentability of new and inventive computer programs would suppress innovation, or be detrimental to small inventors. On the contrary, it can be conceived that a clear patent protection in the last 30 years would have stimulated software innovation: we could have seen for example the development and the co-existence of a wider and more diverse range of operating systems. In the antitrust trial, there is hardly any mention of Microsoft using or abusing its software patents: monopoly, if any, was established solely through marketing strategies and financial strength. Patents could possibly have allowed much smaller firms to compete on technical grounds and merits, when they do not have the ability to compete on financial grounds,

for the benefit of all software users. Further, It is not uncommon for small inventors to be able to enforce or leverage their software-related patents against big companies. In other words, there is no evidence that patentability in the field of software technology would have an impact on the industry that is any different from the impact that the availability of patent protection has in any other area of technology.

d) Even though software development indeed requires relatively low starting investments, and duplication of software has a marginal manufacturing cost, software producers are just like any other producers, seeking to protect themselves from competitors improperly appropriating the fruits of their research and development. Patents provide the right mechanism that can be used to ensure that their competitors incur comparable development and manufacturing costs in order to enter the market.
e) There is no convincing argument that patents are not a form of protection adapted to software technology. For example, arguments that software is complex, or that a software developer may without knowing it infringe several patents, may be made for many other fields of technology for which the availability of patent protection is never questioned. Such arguments are not relevant to the fundamental question as to whether or not patent protection should be available.

II - It is essential to remove quickly the exclusion of A52(2)c) *in fine*, and corresponding provisions in national laws and the European regulation for the Community patent:

a) It seems very likely that exclusion has caused considerable harm over the last 30 years or so to the European software industry. As already expressed by ASPI in its answer to the Commission over the 1997 Green Paper: « L 'exclusion expresse des programmes d'ordinateur de la brevetabilité a fait par ailleurs insidieusement passer le message aux développeurs en Europe que leurs inventions liées au logiciel ne présentaient aucun caractère brevetable. Les subtilités de la limitation de l' exclusion aux programmes "en tant que tel", et les techniques pointues de rédaction des revendications, leur sont totalement étrangères. Des lors, ils pratiquent une "auto-censure" en renonçant a priori a revendiquer toute forme de protection en rapport avec les brevets. Cela représente une perte pour les inventeurs européens, dont les innovations ne sont pas protégées dans leur pays d'origine, contrairement a ce qui se passe pour les inventeurs américains et japonais,… ».
b) There is no evidence that such removal would create any major upset in the industry. There are already thousands of software-related patents in existence, however the scope of protection they afford is unclear, and probably depends even more strongly than usual on the skill of the patent practitioner that prepared the application. We think this current situation is far "worse" than having a legislative framework that is more legally certain, whereby a program product is clearly patentable provided it is new and involves an inventive step, and in which the scope of the patent protection is precisely defined. The argument that software patentability would only benefit patent practitioners is not sustainable: the current legal uncertainty is far more beneficial for them !
c) One has to wonder about the impact of the different standards of patentability between Europe and the USA: currently, US software developers file patent applications in the USA that they do, to some extent, extend to Europe, obtaining corresponding European patents, whereas European developers, having operated for 30 years in the paradigm created by the exclusion, do not even file a patent application in Europe, and even less so in the USA. Only a quick change of mindset in European developers could level the situation, and such a mindset can only be obtained through strong and clear messages sent by European authorities, such as a quick removal of the exclusion.

III - It seems unduly difficult and unnecessary to attempt to differentiate between computer programs with or without "technical character"*:*

a) There is inherent difficulty in defining what "technical character" is or is not. And indeed, EPO Boards of Appeals have only been able to come up with positive lists of instances when they thought "technical character" was involved, or was not. They have never given any general definition, and further, are unlikely to find any.
b) Subject to adequate documentation describing the prior art, in particular that available to patent offices, it seems much easier and legally certain to focus and argue over familiar notions such as novelty and inventive step, than it is to argue over "technical character".
c) We thus suggest that it may prove unnecessary to examine the "technical character", if the technicality of the computer program product *per se* is admitted, and the only remaining questions are whether it is new and involves an inventive step.
d) Leaving it up to administrative bodies and patent office jurisprudence to define the boundary between what is and is not technical, although understandably practical for regulating the flow of incoming patent applications and issued patents, can only give mixed messages to the software development community, and create increased legal uncertainty.

IV - About the scope of protection of patents issued for computer programs

a) One possible or probable form of issued claim could be: "Computer program product comprising computer instructions implementing the steps of:

a. step 1
b. step 2
c. step 3
d. step 4."

b) The invention would not be patentable if it is not new, ie there is a computer program implementing at least the same steps 1 to 4 in the prior art.
c) The invention would not be patentable if it does not involve an inventive step, ie, for example:
    1) it is obvious for the man skilled in the art to combine two programs, one doing steps 1-2 and one implementing steps 3-4;
    2) it is obvious to add step 4 to to an existing program implementing steps 1-3;
    3) it is obvious to derive the program from an existing method which has all 4 steps (but certain implementation of the program could involve an inventive step even though the method is known, depending on the innovative character of the implementation).
d) A method implementing steps 1-4 above could be patentable, providing it does not fall under the method exclusions of A52(2) EPC, and it is new and inventive.
    1) If the method is patentable, then a computer program is likely to be patentable as well, because as said by the EPO Board of Appeals in T1173/97, "it would be illogical to grant a patent for a method and an apparatus adapted for carrying out the same method, but not for the computer program, which comprises all the features enabling the implementation of the method and which, when loaded in a computer, is indeed able to carry out that method".
    2) But if the computer program is patentable, this does not imply that the method is patentable.
e) We suggest that any product, whether hardware or software, can be described as a tool for implementing a certain method. If the method can be patented because it does not fall under the

method exclusions, and is new and inventive, the tool may be patented. But the tool may be patentable, and the method not patentable, as being a method 1) excluded from patentability, or 2) not new, or 3) not inventive).

f) The scope of protection of a patent issued on the basis of a properly worded claim, such as on the model listed above, is well defined.

g) It is of utmost importance however that issued patents be of the same level of quality as issued patents in other fields of technology, ie that the assessment of novelty and inventive step be as accurate. This requires in particular the patent offices to be able to retrieve prior computer programs. As raised by some authors, it is the very fact that so many blatantly invalid patents have been issued by some patent offices which triggered in part the outraged reaction of the software community.

h) One real practical difficulty is with the description: what is the level of description which is sufficient and supports the claims. Special description requirements, including so as to facilitate subsequent searches by the patent offices, could be laid out in the EPC. After all other technologies have already given rise to creative description solutions (ex: Rule 27a, 28, etc…). Program listings may not be necessary in that respect.

i) We do not believe there is any problem with a particular piece of software being protected both by copyright and by patents. The copyright protects the program code from copying, the patent would give exclusive rights to certain functionality implemented by the code, as defined in the patent laws. The two kinds of protection are very different in nature and are both of significant value to a software developer.

V - How to protect certain forms of software distribution, such as 'open source' from being impacted by software patentability

a) The notion of "open source" or "logiciel libre" relates to the particular form of license agreement under which a piece of software is distributed. Such license agreements are predominantly copyright licenses. This aspect of the dissemination of software technology has nothing to do with the existence of the rights that protect the software in question and we believe should not influence the question of whether patent protection should or should not be available for software technologies. A patented software invention may equally be implemented as an open source program or as a non-open source (or proprietary) program - there is no technical difference between the two.

b) If it is considered necessary to make special provision for open source programs in European patent legislation then this would more appropriately be achieved by limiting the enforceability of patents for computer programs in certain specific cases. For example, it may be possible to provide that if a computer program comes with a license that its author gives a free authorization for others to copy and use it without limitations (“free software”), in such particular case a patent covering such computer program is not enforceable. On the other hand, the patent would be enforceable against the same program but distributed under different terms. We question however whether such legislation is necessary or desirable. The closest analogy we could find to such a limitation to patent enforceability, is in Article L.613-5 of the French “Code de la Propriété Intellectuelle”, which provides that “Les droits conférés par le brevet ne s’étendent pas…aux actes accomplis…à des fins non commerciales…”.

VI - On the question of patentability of certain methods

It seems like the question of patentability or not of certain methods (for doing business or else) was less of a concern when computer software was not around. Yet the methods have been around a long time.

We believe that most of the current difficulties with patentability of such methods stems from the fact, not that nowadays such methods are implemented in software, but rather that we are now attempting to assess whether the software underlying such methods is patentable or not. In other words, we ourselves have created the difficulties: in an awkward attempt to reverse a historical lack of patentability of software, we have generated the problem of patentability of certain methods.

But, when assessing its patentability, one should really focus on the method itself, whether or not implemented using a computer software. Is it a method listed as an excluded method in A52(2) EPC ? is it new ? is it inventive ? without consideration for the underlying software.

If the method is not patentable, then the underlying software may still be, provided only it is new and inventive. The scope of protection is then much more narrow and economically viable, as extending to one tool for implementing the method, and not to some other tools or the method itself.

VII - Bibliography

The Economic Impact of Patentability of Computer Programs, Report to the European Commission by Robert Hart, Peter Holmes and John Reid on behalf of Intellectual Property Institute, London (http://europa.eu.int/comm/internal_market/en/intprop/indprop/study.pdf)

Debunking the Software Patent Myths, a study by Paul Heckel, 1992 (http://www.heckel.org/Heckel/ACM%20Paper/acmpaper.htm)

DOES THE PATENT OFFICE RESPECT THE SOFTWARE COMMUNITY ?, by Greg Aharonian, IEEE Software, Jul/Aug 1999 (http://www.bustpatents.com/)

15 December 2000