

Intangible Patents: The European Patent Office Approach

Dr Myles Jelf, Partner, Bristows

1. The EPO's ostensible approach to patentability

The EPO's approach to the specific question of patentability can appear deceptively simple. The *ratio* from the line of decisions including *Hitachi* is sometimes paraphrased as, 'does the invention involve any hardware'? If the answer is yes, then the analysis moves on beyond the patentability questions raised by Art. 52 of the EPC. The simplicity and certainty of applying this test makes it attractive. It is important, however, to appreciate that it only works as part of a more complex overall approach.

2. The EPO's overall approach to potentially excluded inventions

Many patent applications will readily pass the *Hitachi* patentability test (or can be readily redrafted to do so). Before reaching grant, however, they will also need to be both novel and inventive. It is in the assessment of the inventive step that the current EPO approach balances out the relatively low hurdle of *Hitachi*. Lack of obviousness is considered through applying the well-known problem/solution methodology – however, in assessing that question any 'non-technical' excluded aspects of the invention are stripped out of the characterisation and loaded into the problem to be solved. They cannot, therefore, form part of the inventive solution.

3. Comparison with the UK approach

Current UK case law applies a more complex analysis at the patentability stage. Does this difference in form, however, lead to any eventual difference in substance given the EPO's analysis downstream of the patentability evaluation? If there is no difference of substance, does the difference in form matter?

4. Software Patents: Pending questions before the EPO's Enlarged Board of Appeal

Whilst the specific questions are quite involved, the central issues they tease out are:

- is the computer program exclusion one of form or substance? If the former, can it be overcome by simply reciting the technical nature of a program itself or the standard hardware used in its execution?
- If the latter, then what more is needed from the computer program to avoid exclusion? If simply a 'technical effect', then where must that effect manifest itself? Must it be external to the computer? If not, how does one identify a patentable technical effect beyond that inherent in running any computer program?

5. Why are software patents so controversial?

What is the basis and rationale of the computer program exclusion? Does hardware / software convergence, together with the growing ubiquity of programmable devices make the exclusion increasingly hard to justify?