

What makes a patent claim indefinite? Microprocessor Enhancement Corporation v. Texas Instruments Incorporated and Intel Corporation

By George Jakobsche, Associate

What a patent covers is defined by its claims, and to be valid, a patent claim must be *definite*. That is, a person of ordinary skill in the art must be able to ascertain the metes and bounds of the claim, i.e., determine whether a particular device or method is covered by the claim or not. Significantly, a final court determination that a claim is indefinite is fatal, because an indefinite claim is invalid and can never again be asserted.

In *Microprocessor Enhancement Corporation v. Texas Instruments Incorporated and Intel Corporation* (2007-1249, decided April 1, 2008), the United States Court of Appeals for the Federal Circuit (“the Federal Circuit”) said it is possible to mix apparatus (system) recitations with method steps in a single claim, without necessarily making the claim indefinite. Previously, in *IPXL Holdings, LLC v. Amazon.com, Inc.* (430 F.3d 1377; Fed. Cir., 2005), the Federal Circuit said that a patent claim may not simultaneously cover a system and a method of using the system, because the scope of such a claim can not be reasonably ascertained by one of skill in the art. Can these two cases be reconciled with each other? When should apparatus claims include functional limitations? Are there risks involved in drafting apparatus claims with functional limitations?

In *IPXL Holdings*, claim 25 is directed to a system, but the claim includes method steps that are not performed until after the system is built and installed. The claim recites a “system of claim 2 ... wherein ... the user uses the input means to [perform one of two functions].” The Federal Circuit said the claim is indefinite, because “it is unclear whether infringement ... occurs when one creates a system that allows a user to [perform one of the two functions], or whether infringement occurs when the user actually uses the input means to [perform one of the two functions].” (Emphasis added.)

Citing *IPXL Holdings*, the defendant in *Microprocessor Enhancement* argued that asserted claim 1 is invalid, because the claim includes both apparatus and method step recitations. The claim recites a “method of executing instructions in a pipeline processor comprising: [structural limitations on the processor], said method further comprising the steps of: [steps implemented in the processor, including fetching, performing, etc.]” Thus, on the surface, the claim appears to impermissibly mix two subject matter classes, i.e., apparatus and method steps.

However, according to the Federal Circuit, the *Microprocessor Enhancement* claim is not indefinite for including the two types of recitations, because the structural limitations on the processor simply constrain the type of processor that performs the recited steps. Preambles of method claims often limit the physical structures of systems that perform the recited steps, the Court of Appeals explained, so there is no impropriety in including these structural limitations in the body of a claim. According to the Court, what the claim covers and when the claim is infringed are clear from the language of the claim.

Similarly, functional language in some *Microprocessor Enhancement* apparatus claims does not necessarily make the claims indefinite, according to the appeals panel. An example of functional language appears in claim 7, which recites, in part, “the condition execution decision logic pipeline stage, when specified by the condition execution specifier, determining the enable-write using the Boolean algebraic evaluation.” (Emphasis added.) The underlined language appears to define the condition execution decision logic pipeline stage by what the stage *does*, not by its *structure*. To infringe an apparatus claim, an accused device must include all the *structures* recited in the claim. So why doesn’t the functional language in claim 7 make the claim indefinite?

Functional language is explicitly sanctioned by the patent statute to permit so-called “means-plus-function” claim recitations (35 U.S.C. §112, ¶6), such as “means for urging a first member against a wall.” Such recitations do not define structure. Instead, structures that are described in the remainder of the patent as performing the recited function, and equivalents of such structures, are covered by such recitations.

Although from a casual reading means-plus-function claims may *appear* to be very broad, in fact they should be used with caution because they are limited to the structures disclosed in the patent and equivalents thereto. Furthermore, only equivalents that are known at the time the patent application is filed are considered. After-developed technology is only available for consideration under a different, and confusingly-similarly named, analysis called the “doctrine of equivalents,” which is not discussed here.

Claim elements other than means-plus-function elements may also include functional limitations, without necessarily being indefinite. For example, “a spring” is a purely structural claim element, whereas “a spring for urging a first member against a wall” is narrower than “a spring,” because the “spring for urging...” is a particular type of spring, based on the functional limitation. Similarly, the functional language in Microprocessor Enhancement claim 7 defines a particular type of “condition execution decision logic pipeline stage.”

A claim element may be interpreted as a means-plus-function element, even though it does not include the word “means,” if the element recites a function but does not recite sufficient structure to perform the function. Thus, claims should be drafted to recite enough structure to perform all the recited functions, and unnecessary functional limitations should be omitted, to avoid unwanted means-plus-function interpretation of the claims.

So, one lesson from Microprocessor Enhancement is that mixing apparatus and method step recitations in a claim does not necessarily make the claim indefinite, as long as

what the claim covers and when the claim is infringed are clear. Functional language is permissible, although for patentability, an *apparatus* claim must be distinguishable from the prior art in terms of its *structure*, not function.

In Microprocessor Enhancement, the Federal Circuit also said a single claim term may have different meanings in different parts of a single claim, without the claim being indefinite. How can this be? How badly written must a claim be to be indefinite?

At least one of the Microprocessor Enhancement claims includes the term “condition code,” but this term means two different things in different places in the claim. In some places, the term refers to a storage unit, and in other places the term refers to a value derived from output of the storage unit. A defendant argued that this inconsistent meaning of a single claim term makes the claim indefinite, citing Process Control Corporation v. HydReclaim Corp. (190 F.3d 1350; Fed. Cir., 1999). However, in the Process Control Corporation patent, the claim 1 term “discharge rate” is surrounded by uniform language “from said common hopper to said processing machine” in the two instances at issue, so the claim term can be construed consistently, and the scope of the claim can be ascertained. On the other hand, the Microprocessor Enhancement Corporation term “condition code” is not surrounded by uniform language. The term has two different meanings, but each meaning is readily apparent from its context.

According to the Microprocessor Enhancement ruling, “A claim is presumed valid. Thus, a claim is indefinite only if the claim is ‘insolubly ambiguous, and no narrowing construction can properly be adopted.’” So, if the court can construe the meaning of a claim, the claim is not invalid for indefiniteness. Although the Federal Circuit seems to have rescued the Microprocessor Enhancement claims by distinguishing identical claim terms by their contexts, it seems wise to draft claims that use clearly distinct terms. After all, winning on appeal is nice, but preventing an attack in the first place is better. ✧