United States District Court,
N.D. Georgia, Atlanta Division.

PICIS, INC. and PICIS, S.A,
Plaintiffs.
v.
SURGICAL INFORMATION SYSTEMS, LLC and Capsuel Technologie,
Defendants.

Civil Action No. 1-04-CV-1870 (TCB)

Oct. 12, 2006.

SPECIAL MASTER'S REPORT ON CLAIM CONSTRUCTION UNDER MARKMAN v. WESTVIEW INSTRUMENTS, INC.

SUMNER C. ROSENBERG, Special Master.

I. *Introduction*

Plaintiffs, PICIS, Inc. and PICIS, S.A. (collectively "PICIS") bring this action against Defendants Surgical Information Systems, LLC and Capsule Technologie (collectively "SIS") for infringement of United States Patent No. 5,161,222 (the "'222 patent") FN1, entitled "Software Engine Having an Adaptable Driver for Interpreting Variables Produced by a Plurality of Sensors." Specifically, the ' 222 patent has fourteen claims, each of which is allegedly Infringed.

FN1. References to the text of the '222 patent are referred by herein by column and line number, e.g. "6:49-53".

It is the role of the Court to interpret or construe the meaning of claims as a matter of law. Markman v. Westview Instruments, Inc., 52 F.3d 967, 979 (Fed.Cir.1995) (en banc), *aff'd,* 517 U.S. 370, 116 S.Ct. 1384, 1393 (1996). Courts typically construe disputed claim terms at an intermediate stage of the litigation, and the process is generally referred to as a *Markman* hearing or order.

The Court has appointed the Special Master to preside over the *Markman* hearing in this case kind submit a Report and Recommendation on claim construction to the Court. (See Order, February 2, 2006.) The Special Master has considered the briefs of the parties FN2, and held a hearing on June 14, 2006 where the parties provided further expert testimony and arguments. (A transcript of that hearing is available and referenced herein as "hearing transcript.") This report sets forth the Special Master's recommendation for construction of the disputed claim terms, after due consideration of the applicable law as applied to the evidence and the arguments of the parties.

FN2. References to the parties' briefs shall be as follows: Plaintiffs' Opening Brief Concerning Claim Construction ("PICIS Brief"); Defendants' Opening Claim Construction Brief ("SIS Brief"); Plaintiffs' Responsive Brief Concerning Claim Construction ("PICIS Response"); and Defendants' Reply to Plaintiffs' Opening Brief Concerning Claim Construction ("SIS Reply").

II. *Overview of* '222 Patent *and Disputed Claim Terms*

The subject of the '222 patent is computer software that enables communication with different medical devices to capture data and direct the data to a user-designated storage or display destination (PICIS Brief, p. 1; SIS Brief, pp. 3-5). The '222 patent has fourteen claims. (28:8-34:3), of which claims 1, 13 and 14 are independent claims (that is, claims that are self-contained) and claims 2-12 are dependent claims (that is, claims that refer back to previously numbered claims and, therefore, incorporate the limitations of all of the claims on which they directly and indirectly depend). Each claim defines a distinct invention, which is literally infringed if and only if each element of the claim is in the accused product.

The application that led to the '222 patent was filed in the United States Patent and Trademark Office on August 20, 1990. The communications between the patent applicants' attorney and the patent examiner constitute the prosecution history (or file history) of the patent.FN3 The ' 222 patent was issued on November 3, 1992.

FN3. The prosecution history has been submitted to the Court as Exhibit 2 to PICIS Brief.

The elements of the claims of the '222 patent include numerous terms for which the parties cannot agree on the meaning or construction.FN4 Some disputed claim terms are for specific terms, e.g. "object queue loop", and many disputed terms are "means plus function" elements, where the element specifies the function to be carried out (typically by software).

FN4. The parties have previously submitted to the Court their Joint Claim Construction Statement, Tab A of the Statement is the Construction of Claim Terms, Phrases or Clauses on which the Parties Agree (hereinafter, "Agreed Construction"). Tab B is Plaintiffs' Proposed Construction of Terms, Phrases or Clauses (hereinafter, "PICIS Construction"). Tab E is Defendants' Proposed construction of Terms, Phrases or Clauses (hereinafter, "SIS Construction"). After the *Markman* Hearing, the following were submitted: Supplemental Construction of Claim Terms, Phrases or Clauses on which the Parties Agree (hereinafter, "Agreed Construction Supp."), Parties' Supplemental Proposed Construction of Means-Plus-Function Terms (hereinafter, "MPF Construction Supp."), and PICIS' Supplemental Construction of Terms, Phrases or Clauses (hereinafter, "PICIS Construction Supp.")

During and subsequent to the June 14, 2006 hearing, the parties were able further to agree on the construction of several terms. These terms are addressed in the Supplemental Construction of Claim Terms, Phrases or Clauses on which the Parties Agree, and in the Parties' Supplemental Proposed Construction of Means-Plus-Function Terms, submitted to the Court on June 30, 2006. The Special Master will not address these terms any further.

III. *Law of Claim Construction*

A. *Construction of Terms*

Phillips v. AWH Corp., 415 F.3d 1303 (Fed.Cir.2005) (en banc), sets forth the principles that should guide the claim construction process:

"It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude.' " Id. at 1312 (citations omitted). "We have frequently stated that the words of a claim 'are generally given their ordinary and customary meaning.' We have made clear, moreover, that the ordinary and customary meaning of a claim term is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application." Id. at 1313 (citations omitted).

"Importantly, the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." *Id.*

In some cases, the ordinary meaning of claim language as understood by a person of skill in the art may be readily apparent even to lay judges, and claim construction in such cases involves little more than the application of the widely accepted meaning of commonly understood words. In such circumstances, general purpose dictionaries may be helpful. In many cases that give rise to litigation, however, determining title ordinary and customary meaning of the claim requires examination of terms that have a particular meaning in a field of art. Because the meaning of a claim term as understood by persons of skill in the art is often not immediately apparent, and because patentees frequently use terms idiosyncratically, the court looks to "those sources available to the public that show what a person of skill in the art would have understood disputed claim language to mean." Those sources include "the words of the claims themselves, the remainder of the specification, the prosecution history, and extrinsic evidence concerning relevant scientific principles, the meaning of technical terms, and the state of the art."

Id. at 1314 (citations omitted).

"Quite apart, from the written description and the prosecution history, the claims themselves provide substantial guidance as to the meaning of particular claim terms." *Id.* "The claims, of course, do not stand alone. Rather, they are part of 'a fully integrated written instrument,' consisting principally of a specification that concludes with the claims." *Id.* at 1315 (citations omitted). "For that reason, claims 'must be read in view of the specification, of which they are a part.' As we stated in Vitronics [Vitronics Corp. v. Conceptronic, Inc., 90 F.3d 1576, 1582], the specification 'is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.' " *Id.*

Ultimately, the interpretation to be given a term can only be determined and confirmed with a full understanding of what the inventors actually invented and intended to envelop with the claim. The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction.

Id. at 1316 (citations omitted).

In addition to consulting the specification, we have held that a court "should also consider the patent's prosecution history, if it is in evidence." The prosecution history, which we have designated as part of the "intrinsic evidence," consists of the complete record of the proceedings before the PTO and includes the prior art cited during the examination of the patent. Like the specification, the prosecution history provides evidence of how the PTO and the inventor understood the patent.

Id. at 1317 (citations omitted).

"Although we have emphasized the importance of intrinsic evidence in claim construction, we have also authorized district courts to rely on extrinsic evidence, which 'consists of all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises.' " *Id*. (citations omitted).

Within the class of extrinsic evidence, the court has observed that dictionaries and treatises can be useful in claim construction. We have especially noted the help that technical dictionaries may provide to a court "to better understand the underlying technology" and the way in which one of skill in the art might use the claim terms. Because dictionaries, and especially technical dictionaries, endeavor to collect the accepted meanings of terms used in various fields of science and technology, those resources have been properly recognized as among the many tools that can assist the court in determining the meaning of particular terminology to those of skill in the art of the invention. Such evidence, we have held, may be considered if the court deems it helpful in determining "the true meaning of language used in the patent claims."

We have also held that extrinsic evidence in the form of expert testimony can be useful to a court for a variety of purposes, such as to provide background on the technology at issue, to explain how an invention works, to ensure that the court's understanding of the technical aspects of the patent is consistent with that of a person of skill in the art, or to establish that a particular term in the patent or the prior art has a particular meaning in the pertinent field However, conclusory, unsupported assertions by experts as to the definition of a claim term are not useful to a court.

Id. at 1318 (citations omitted).

In sum, extrinsic evidence may be useful to the court, but it is unlikely to result in a reliable interpretation of patent claim scope unless considered in the context of the intrinsic evidence. Nonetheless, because extrinsic evidence can help educate the court regarding the field of the invention and can help the court determine what a person of ordinary skill in the art would understand claim terms to mean, it is permissible for the district court in its sound discretion to admit and use such evidence. In exercising that discretion, and in weighing all the evidence bearing on claim construction, the court should keep in mind the flaws inherent in each type of evidence and assess that evidence accordingly.

Id. at 1319 (citations omitted).

Moreover, we recognize that the distinction between using the specification to interpret the meaning of a claim and importing limitations from the specification into the claim can be a difficult one to apply in practice. However, the line between construing terms and importing limitations can be discerned with reasonable certainty and predictability if the court's focus remains on understanding how a person of ordinary skill in the art would understand the claim terms. For instance, although the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments.

Id. at 1323 (citations omitted).

B. *Construction of "Means Plus Function" Elements*

"Means plus function" claim elements typically use the phrase "means for [performing a function]." There are specific rules for interpreting means plus function elements in claims, based on 35 U.S .C. s. 112, paragraph 6:

An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

For each means plus function element, the Court "must then construe the function recited in that claim and determine what structures have been disclosed in the specification that correspond to the means for performing that function." Kemco Sales, Inc. v. Control Papers Co., 208 F.3d 1352, 1361 (Fed.Cir.2000). Only structure that is "necessary to perform the claimed function" may be included. Micro Chem., Inc. v. Great Plains Chem. Co., 194 F .3d 1250, 1258 (Fed.Cir.1998). The structure of a microprocessor programmed to carry out an algorithm is limited by the disclosed algorithm. WMS Gaming Inc. v. International Game Tech., 184 F.3d 1339, 1348 (Fed.Cir.1999). A computer-implemented means plus function term is limited to the corresponding structure disclosed in the specification and equivalents thereof, and the corresponding structure is the algorithm. Harris Corp. v. Ericsson Inc., 417 F .3d 1241, 1253 (Fed.Cir.2005).

IV. *Construction of Terms in Dispute*

The role of the Special Master in this *Markman* process is to provide recommendations to the Court for the construction of claim terms on which the parties cannot agree. The following recommended constructions are made after careful consideration of the record in the case and the legal principles described above. First, some general comments are given with respect to the process.

Construction of a specific term or close variations will be construed as having the same meaning if it appears in more than one claim.

Many of the terms in issue are in "means plus function" format (see III.B. *supra*). In construing the means plus function elements, the function must first be construed. In most cases, that is little more that a restatement of the function stated in the claim element. Then, the associated structure described in the patent specification associated with the function must be determined. *Kemco Sales, Inc*. at 1361. That determination is set forth below for the means plus function elements in dispute, but it should also be noted that 35 U.S.C. s. 112, paragraph 6, states that the structure includes "equivalents thereof." The parties agreed that it is not the role of the Special Master to address what are the equivalent structures, and that appears to be a fact question to be answered at the appropriate time. Therefore, the Special Master does not make any recommendations with respect to equivalent structures, except as specifically noted herein.

As noted, the recommendations address only claim terms in dispute, so that the claim terms on which the parties agree are not addressed herein. With regard to the disputed claim terms, the Special Master is not necessarily constrained to selecting one of the parties' proposed constructions over the other, but has used the proposed constructions as guides to modified constructions where appropriate. The Special Master notes that even for claim terms for which the parties agree on construction, the construction may use terms that require a level of understanding of one skilled in the art, which leaves to the Court and the parties the subsequent difficulty of explaining to a lay judge or jury what the claims mean. The Special Master's recommended constructions herein will generally be directed to the same level of skill as the agreed upon constructions or the proposed constructions.

A. *Software engine* (all claims except claim 9)

PICIS proposes a general construction of the term "software engine" as a collection of software routines or modules (PICIS Construction, p. 1). SIS proposes a much more detailed construction, reciting several paragraphs from the specification (SIS Construction, pp. 1-2). "Software engine" is merely the identification of the invention in the preamble of each claim. As such, the invention is defined by the claim and the construction of "software engine" should not otherwise limit the claim. The broadest possible construction of this term is the most appropriate.

Therefore, the following construction is recommended:

"Software engine" means "computer software."

B. *The "Operating System" Terms*

The parties have raised several terms that address the relation of the invention to an "operating system." These terms include "operating system" itself, along with "operating system independent," "operating system dependent," "software engine otherwise being operating system independent," "means for interfacing with an operating system to determine if an appropriate operating system is present," and "means for interfacing with and operating system to execute the necessary memory and disk management functions such that the software engine can operate." These terms appear in the three independent claims 1, 13 and 14. The parties briefed these terms in depth and devoted much of their time in the *Markman* hearing to these issues.

1. *Operating system* (claims 1, 13 and 14)

The parties both agree that an "operating system" is software that acts as an intermediary between the computer hardware and the application software (PICIS Construction, pp. 4-9; SIS Construction, p. 4). Current well known operating systems include Windows and UNIX. In the context of construction of the basic term "operating system," PICIS has proposed, in addition, that different versions of an operating system (e.g. Windows 98 and Windows XP) are "different" operating systems.

It is inappropriate to address what may be different operating, systems in the construction of this basic term, since the distinction is unnecessary to the understanding of the term and the word "different" does not even appear in the claims as a modifier of "operating system."

Therefore, the following construction is recommended:

"Operating system" means "software that acts as an intermediary between the computer hardware and the application software to manage computer resources."

2. *Operating system independent* (claims 1, 13 and 14)

The parties appear to be in agreement that "operating system independent" means, at a minimum, that the software application does not rely on or make calls to an operating system in order for the application software to operate (PICIS Construction, p. 10; SIS Construction, p. 7). SIS' construction goes further in saying that it also means that the program is capable of operating with any operating system. If, by this, language, SIS really is saying that *no* operating system is required in order for the software to operate, that appears to be the same as the agreed portion of the parties' construction. If, on the other hand, SIS is saying that the software utilizes operating systems functions and must be capable of using *any* (i.e.every) operating system, that construction must be rejected. Nothing in the claims or the specification requires such a universal program. Furthermore, as discussed below, the claims say the software first determines if an appropriate operating system is present. If the software was required to be universal, there would be no need for this function.

On the other hand, PICIS supplemented its proposed construction to add the example of "a function call that is not specific to a particular operating system" (PICIS Construction Supp., p. 3). Thus, PICIS is suggesting that operating system independence is when the software is capable of using *some* operating systems rather than a particular operating system. PICIS provides no support for this modified construction, nor does it seem proper to refer to an operating system function call not being specific to a particular operating system, since each operating system (or family of operating systems) has a unique application program interface (or API) for function calls. (See, for example, testimony of Dr. Sussman, Transcript of Markman Proceedings, pp. 20-21.)

Independence with respect to operating systems is the absence of need to access an operating system, not some level of flexibility to which operating system is to be accessed. The patent specification gives no indication that an object or purpose of operating independence of the invention is the ability of a portion of the software engine to work with multiple operating systems while the operating system dependent aspect of the software engine requires a specific operating system.

Therefore, the following construction is recommended:

"Operating system independent" means "does not rely on an operating system and does not use any operating system services, procedures, calls or processes."

3. *Operating system dependent* (claim 14)

The comments with respect to "operating system independent," above, apply similarly to "operating system dependent," which must have the opposite construction.

Therefore, the following construction is recommended:

"Operating system dependent" means "relies on an operating system and uses some operating system services, procedures, calls or processes."

4. *Means for interfacing with an operating system to determine if an appropriate operating system is present* (claim 1)

The parties are not far apart on the construction of this term (MPF Construction Supp, p. 2).FN5 They agree that it is a "means plus function" element. As to the function of the element, SIS is a bit more specific about what constitutes an appropriate operating system (able to execute the necessary memory and disk functions), whereas PICIS states more generally that the operating system meets the minimum requirements of the software engine. It is unnecessary to distinguish between these two options at this point, so I will accept the more general language. As to the corresponding structure, the parties are in basic agreement on one structure, but PICIS adds a second proposed structure: "at least portions of main module and operating system module (Fig.34)." An operating system module is not an element of claim 1, and, in any case, this phrase says little except that the structure is part of the software. The agreed upon structure refers more particularly to the software that carries out the function and nothing more is accomplished by PIGS' proposed additional phrase.

FN5. In the initial proposed constructions and in the briefs, SIS consistently proposed as part of the construction of each "means plus function" claim that the structure specifically required the program to be written in assembly language, and for the program to run in the main module of Figure 34 of the '222 patent, and that the main module carries out the algorithms in Figures I-33." SIS Construction; SIS Brief; SIS Reply. It appears that SIS has dropped these assertions after the *Markman* hearing. See MPF Construction Supp. Biased on this understanding, I will not consider whether the structure of the various "means plus function" elements in the claims should include such limitations.

Therefore, the following construction is recommended:

The function for "means for interfacing with an operating system to determine if an appropriate operating system is present" is "interfacing with an operating system to determine if an appropriate operating system is present that meets the minimum requirements of the software engine," and the structure is "software for performing step 11 of Fig. 1, as described at 9:27-30 ."

5. ***Means for interfacing with an operating system ... to execute necessary memory and disk management function such that the software engine can operate*** **(claim 1)**

This element includes some of the most strongly contested terms to be construed. The issues include what is included in "memory and disk management functions," whether the phrase "necessary ... such that the software engine can operate" should lead to "memory and disk management functions" being construed to include functions that are arguably more than merely those related to the computer memory or disk management, and factual issues of whether the software engine can operate without operating system functions beyond "memory and disk management functions." Also, this element was included in the claim by the patent applicant in response to a prior art rejection. Thus, the claim language, the specification, and the prosecution history will be considered.

Working in reverse, I will first comment on the prosecution history. In the briefs the parties submitted, the prosecution history was discussed with respect to "operating system independent," but the argument was more properly directed to the construction of the "means" language presently under consideration. In any case, SIS argues that the prosecution history shows that the application was amended to add the subject language in response to a prior art rejection by the patent examiner. SIS makes the statement that the claims were narrowed "to specifically identify which functions [sic] calls the claimed software engine could make and excluded all others" (SIS Brief, p. 16). In essence, SIS is asserting its proposed construction, but the prosecution history provides no specific support for this construction. As PICIS points out, the prosecution history cannot be used to alter the meaning derived from the claim wording unless the prosecution history is unequivocal or contains a clear disavowal of an interpretation urged by the patentee (PICIS Response, p. 23, citing Serrano v. Telular Corp., 111 F.3d 1578, 1584 (Fed.Cir.1997)). Interestingly, in its presentation at the *Markman* hearing, PICIS then put forth its own arguments based on the prosecution history. In either case, the only conclusion I can reach in reviewing the prosecution history is that a significant amount of claim language was added in response to the examiner's rejection. However, nothing in the prosecution history sheds any meaningful light on the construction of this, or any, claim language.

The central dispute with regard to the stated function of this element is whether the term "memory and disk management functions" is either limited or expanded by the surrounding phrase "necessary ... such that the software engine can operate." PICIS argues that the latter phrase requires that any operating system functions necessary to operate the software engine are encompassed by "necessary memory and disk management functions," stating that "the plain language of the claims ... states that the software engine relies on the operating system for necessary functions *such as* memory and disk management" (PICIS Response, p. 28 (emphasis added)). But the claim language is not suggestive, as asserted by PICIS; instead, it is specific in stating "memory and disk management functions."

I find that the plain claim language is significant. This limitation becomes relevant because of the subsequent limitation that "the software engine is *otherwise* operating system independent." In accordance with the construction above of "operating system independent," this means that, except for the operating system functions properly included in "necessary memory and disk management functions," the software engine does not rely on an operating system and does not use any operating system services, procedures, calls or processes. PICIS' position that this element includes any operating system function "necessary" to allow the software engine to operate effectively eliminates the specific claim language "memory and disk management functions" from the claim, and further eliminates any meaningful purpose to the following phrase "otherwise being operating system independent." Construction of a claim in a way that "would render [a claim limitation] meaningless is improper. Unique Concepts, Inc. v. Brown, 939 F.2d 1558, 1563 (Fed.Cir.1991); Texas Instruments v.. U.S. Int'l Trade Comm'n, 988 F.2d 1165, 1171 (Fed.Cir.1993) (claim construction cannot "read an express limitation out of the claims").

The specification of the '222 patent does not assist in construing this claim element and specifically the phrase "necessary memory and disk management functions such that the software engine cam operate." The specification does not reference either of the terms "memory management" or "disk management," nor is there any specific use of the word "necessary" in conjunction with "functions." In the *Markman* hearing, SIS pointed to various advantages cited in the specification for operating system independence (2:6-3:5) or a retrofit piece of hardware (3:17-23), but the specification does not tie these advantages to the execution of memory or disk management functions. PICIS also cited the specification at the hearing, which refers to the main module "operat [ing] with various operating systems" (6:60-63) and the library function being "completely dependent on the operating system environment while in use" (7:3-7). But these references also are not directly connected to the execution of memory or disk management functions. Moreover, whatever understanding the parties may argue that the cited specification language may imply, claims define the invention and they are often more narrow than what is described in the specification.

PICIS supported its position in the *Markman* hearing by asserting if the software engine cannot use the operating system to do input and output, the system will not work. Whether or not this is in fact the case, and the parties clearly disagree about this, the fact that a claim may be inoperative should not have a bearing on its construction. Chef Am., Inc. v. Lamb-Weston, Inc., 358 F .3d 1371, 1374 (Fed.Cir.2004) (suiting that the Federal Circuit "repeatedly and consistently has recognized that courts may not redraft claims, whether to make them operable or to sustain their validity").

I find that this and other extrinsic evidence was no more illuminating, and less reliable, than the specification itself. Thus, the construction of the function set forth is based on the claim language, considered in the context of the complete claim. I must rely on the specific claim language over the general language, in order to arrive at the plain meaning.

A separate issue may arise as to whether specific functions are included in "memory and disk management functions." That is, given the construction that functions carried out by the software engine that are not memory and disk management functions must be operating system independent, there may remain an issue as to whether a given function is included in either the category of memory management or disk management. SIS has proposed in its construction of this element a list of functions and services excluded from memory or disk management (SIS Construction, pp. 6-7). It is not clear to me, at this point, that the parties will have a material difference of opinion as to whether specific functions found in the accused product fall within or without the category of "memory or disk management." If so, this may involve consideration of factual evidence not in the record. It seems impractical and inappropriate, at present, to attempt to determine generally what types of functions, such as listed by SIS, do not constitute memory or disk management functions.

Finally, having construed the function, it is a straight forward matter to set forth the structure that carries out the function. The parties are in agreement that software code described at 9:32-46, which refers to steps 13, 15 and 16 of Fig. 1, at least, is the corresponding structure (MPF Construction Supp., pp. 2-3). I find that this is sufficient.

Therefore, the following construction is recommended:

The function for "means for interfacing with an operating system ... to execute necessary memory and disk management functions such that the software engine can operate" is "interfacing with an operating system to execute memory and disk management functions that are necessary for the operation of the software engine," and the structure is "software for performing step 13, 15 and 16 of Fig. 1, as described at 9:32-64."

6. *Operating system interface module which executes the necessary memory and disk management functions such that the software engine can operate* (claim 13)

This element is not in means plus function format, but the discussion in section IV.B.5 above applies similarly to this claim element.

Therefore, the following construction is recommended:

"Operating system interface module which executes the necessary memory and disk management functions such that the software engine can operate" means "computer software that interacts with an operating system to execute memory and disk management functions that are necessary for the operation of the software engine."

7. *Main module is operable upon the request of the external process to determine whether an appropriate operating system is present to execute the necessary memory and disk management functions such that the software engine can operate* (claim 14)

This element is also not in means plus function format, but the discussion in sections IV.B.4 and IV.B.5 above applies similarly to this claim element.

Therefore, the following construction is recommended:

"Main module is operable upon the request of the external process to determine whether an appropriate operating system is present to execute the necessary memory and disk management functions such that the software engine can operate" means "the main module includes computer software that determines whether an appropriate operating system is present in the computer and, if so, interacts with the operating system to execute memory and disk management functions that are necessary for the operation of the software engine."

8. *The software engine being otherwise operating system independent* or *which is otherwise operating system independent* (claims 1, 13 and 14)

The construction of this element is dependent on the discussion above, with the word "otherwise" providing the exclusionary effect that is the crux of the parties' dispute over the construction of the operating system related terms. The construction of this phrase follows from the construction of "operating system independent" and the elements which include the phrase "necessary memory and disk management functions such that the software engine can operate."

Therefore, the following construction is recommended:

"The software engine being otherwise operating system independent" or "which is otherwise operating system independent" means "that, except for execution of disk and management functions, the software engine does not rely on an operating system and does not use any operating system services, procedures, calls or processes."

C. *Means for interpreting an external process request* (claims 1 and 13)

This is a means plus function element, for which the parties agree on the function, and differ only slightly on the structure (MPF Supp. Construction, p. 1). Again, PICIS adds an alternative structure that in "includes at least portions of main module, operating system module, and library functions, as shown in Figure 34." As I previously stated, the agreed upon structure refers more particularly to the software that carries out the function and nothing more is accomplished by PICIS' proposed additional phrase.FN6

FN6. PICIS makes similar proposed constructions for the structure of means plus function elements throughout the claims, so I will not repeat my rejection of such alternative structure that refers to Fig. 34 in subsequently considered means plus function elements.

Therefore, the following construction is recommended:

The function for "means for interpreting an external process request" is "interpreting an external process request for data information" and the structure is "software code for implementing object queue loop as shown in Fig. 2, as described at 9:64-10:2.

D. *Piece of software code which generates a message in a predetermined format requesting from the*

***software engine the delivery of data which is in a specific format and syntax* (claims 1 and 13)**

The parties' differences on this element are not clear (it seems that the dispute instead relates to the term "library function" in claim 14, which I will address below), and the plain claim language does seem clear. PICIS' proposed construction suggests restating the technical words "format" and "syntax" in common language, which may be useful (PICIS Construction, p. 3).

Therefore, the following construction is recommended:

"Piece of software code which generates a message in a predetermined format requesting from the software engine the delivery of data which is in a specific format and syntax" means "computer software that sends requests to the software engine in a form that the software engine can process, which requests specify the form of the response."

E. *Main module* (claims 1 and 14)

PICIS proposes that the main module is a "software routine, or a collection of routines" (PICIS Construction, p. 11). SIS had a construction that was much more specific about the language of the main module and it being a "single executable program" (SIS Construction, p. 9). First, it appears that SIS has retreated from its assertion that the various means claims require the software to be written in assembly language, so I will assume that SIS no longer asserts that in the case of this term. Second, the term "main module" is simply an identifier in claim 1 and serves no limiting purpose. In claim 14, "main module" is an element defined by what it has and what it does. It is clear from the claim that "main module" is nothing more than an identifier of some portion of the software engine and should be construed as such.

Therefore, the following construction is recommended:

"Main module" means "computer software included in the software engine."

F. *Means for creating a mailbox adapted to exchange information with the external process request* (claim 1)

The parties have agreed that the function of this element is "creating a mailbox adapted to exchange information with the external process request, e.g., initializing a data structure, in which messages can be stored, in memory or on disk" (MPF Supp. Construction, p. 4). Both parties also narrowed their proposed structures in the supplemental construction. Yet there still remain some differences between the proposed structures, although these are semantic rather than substantial.

The structure proposed by PICIS is lacking in that it does not set forth the structure of a mailbox, referring only to software code for using a mailbox. The remaining SIS construction seems more appropriate, based on the claim language, and this language was not challenged by PICIS.

Therefore, the following construction is recommended:

The function for "means for creating a mailbox adapted to exchange information with the external process request" is "creating a mailbox adapted to exchange information with the external process request, e.g., initializing a data structure, in which messages can be stored, in memory or on disk," and the structure is "software for creating a temporary data storage area through which data may be indirectly communicated between the external process request and the main module."

G. *Object queue loop* (claims 1, 3, 5-8, 13 and 14)

Again, PICIS proposes a very general construction and objects to SIS importing specific embodiments into this term. This term has been subject to much briefing before and during the hearing, but even now it is not clear which points are in contention.

PICIS objects to the use of a particular language, but, as noted above, SIS has conceded this assertion. A key issue between the parties appears to be whether the term requires that the object queue be processed on a first-in, first-out (FIFO) basis (as opposed to, e.g., last-in, first-out). In fact, in its presentation at the hearing, SIS submitted that its "key construction point is proper: 'Software code which repetitively processes various object types on a first-in, first-out (FIFO) processing basis.' " Although the specification does refer to FIFO processing, the claim language is not limited in that way, so such a limitation is improper.

The claim language itself fills in much of the detail of what the object queue loop does. For instance, claim 1 says that "the object queue loop identifies an object type and initiates a predetermined process." Even SIS pointed out in its hearing presentation that "object queue loop" is not a term of art and does not have any common understanding. Most software is defined by the steps it carries out, not by what it is called. A "loop" is a well known term of art in computer software, so it does seem appropriate to incorporate the repetitive nature of a software loop into the construction, again based on the claim language. See definition of "loop" in *IBM Dictionary of Computing,* p. 255 (Joint Claim Construction Statement, Tab G, page B033). Otherwise, it appears that the claim language associated with "object queue loop" in claims 1, 13 and 14 provides the essential description of the operation of the "object queue loop."

Therefore, the following construction is recommended:

"Object queue loop" means "a computer software routine that repetitively carries out its functions while a certain condition prevails."

H. *Means for opening an object queue loop* (claim 1)

The parties are now essentially in agreement on this means plus function element (MPF Supp. Construction, p. 4). SIS proposes an expanded definition of the function by adding language describing the purpose of the object queue loop, but this is not necessary since the construction of "object queue loop" is addressed above.

Therefore, the following construction is recommended:

The function for "means for opening an object queue loop" is "starting or initializing an object queue loop," and the structure is "software code for performing step 16 of Fig. 1, as described at 9:40-43."

I. *Predetermined adaptable driver* (claims 1, 3, 9, 10, 13 and 14)

The parties appear to agree that "predetermined adaptable driver" is not a term of art and, therefore, that the whole term must be construed with reference to the patent. Neither party argues that the file history has any bearing on the construction of this term. While the parties are in general agreement on what a "driver" is, there has been significant discussion in the briefs and the hearing of the meaning of the term as a whole.

The parties have generally adopted the definition of "driver" set forth in the specification: "discrete pieces of software that decode the information coming from any device, local area network, or modem ... separate executable programs which are resident on the hard disk of the computer" (7:13-16). Beyond this, PICIS argues that the claim language specifies what the driver does, while SIS relies on the patent specification to help construe the complete term "predetermined adaptable driver." Certainly, the claim language helps in setting forth what the predetermined adaptable driver does (e.g., "decodes the variables requested by the external process request" "when adjusted ... to correspond to the data characteristics of the individual sensors" (claim 1; 28:41-45)), but this does not necessarily set forth what the term is. While PICIS' definition of "predetermined adaptable driver" goes no farther, SIS goes so far as to suggest that the definition of the term includes that the driver may be modified (adapted) by a "relatively unskilled programmer at the end user's site" (SIS Brief, p. 30).

I find that the claims must be read in conjunction with the specification in order to properly construe "predetermined adaptable driver," but I do not think all of the limitations proposed by SIS are required. The title of the invention-"Software Engine having an *Adaptable Driver* for Interpreting Variables Produced by a Plurality of Sensors"-highlights the importance of this feature to the invention. Furthermore, one of the stated objects of the invention is to provide a "means by which the operator may rapidly program the invention to receive the information produced by the ... sensor" (3:10-16). "This is achieved by means of an adaptable code overlay or skeleton" (6:8-10) "which permit[s] a hospital technician or a programmer to write a proper interface between the main module and any new medical monitoring device which may not be resident in the library of drivers provided" (7:52-58). Thus, it seems central to the concept of adaptability that the software engine include the capability for an operator or user to modify an existing driver or driver structure (skeleton) to adapt the driver to a new device for which there is not a presently applicable driver. This conclusion is reinforced by claim 10, which, as an example of the means for overlaying the predetermined adaptable driver, sets forth a list of specific information that may be entered or provided to customize the adaptable driver.

However, it is neither practical nor necessary to define this term by the level of skill of the person doing the modification or the location where the modification occurs. I find that an "adaptable driver" is one where the software engine provides the specific capability for an individual to modify or customize the driver to communicate with a desired device. That is, it should not be necessary to obtain the appropriate driver for a device either from the device manufacturer, the provider of the software engine, or a third party.

The term "predetermined" is not very clear and does not seem to factor into either party's arguments in a significant way. SIS suggests that "predetermined" means that "the programmer at the end user's site knows what the device is before adjusting the adaptable driver to it" (SIS Brief, p. 30). However, this definition is directed to the knowledge of an individual, rather than the characteristics of the claimed invention. Perhaps the claim language itself is the best guide when it refers to the driver being "adjusted in a predetermined fashion" (e.g., claim 1). That is, the software engine determines the specific manner in which the driver is to be adapted. This is consistent with the previous finding that the software engine provides the specific capability to modify the driver.

Therefore, the following construction is recommended:

"Predetermined adaptable driver" means "a software driver where the software engine provides the specific capability and the specific manner for an individual to modify or customize the driver to communicate with a desired device, where a 'driver' is a discrete piece of software that decodes the information coming from any device, local area network, or modem, which is a separate executable program resident on the hard disk of a computer."

J. ***Means for overlaying a predetermined adaptable driver*** **(claims 1, 13 and 14)**

Aside from the meaning of the term "predetermined adaptable driver," the parties appear now to be in substantial agreement of the construction of this means plus function element (MPF Construction Supp, p. 4). The parties agree on the function, and both agree that the structure includes either step 80 of Fig. 5 or step 123 of Fig. 12. However, my review of the patent finds that item 80 of Fig. 5 does not involve the overlay of a driver. Rather, Fig. 5 calls Fig. 12 at step 74 if a driver is not already loaded. Thus, it appears that only Fig. 12 is relevant to the structure of this element.

Although not discussed in the parties' briefs, SIS adds the additional limitation to the structure that it requires also the predetermined adaptable driver as described at columns 20-26 of the patent. I believe SIS' point is that the means in question refers to overlaying a "predetermined adaptable driver" rather than simply a "driver," and that the structure otherwise proposed by PICIS merely shows and discusses a driver overlay, without specifying that a predetermined adaptable driver is being overlaid. Although it is not clear why the origin of the driver matters at this stage, the claim language is clear that this means plus function element deals with a predetermined adaptable driver, and the structure should be so limited.

Therefore, the following construction is recommended:

The function for "means for overlaying a predetermined adaptable driver" is "calling up and loading a predetermined adaptable driver into memory and overwriting prior code or data if necessary," and the structure is "software code for performing step 123 of Fig. 12, as described at 13:35-38, where the software acts on a predetermined adaptable driver."

K. ***Means for transmitting the data information generated by the individual sensors to a predetermined destination*** **(claim 1)**

The parties have agreed that the function of this element is "transmitting the data information generated by the individual sensors to a predetermined destination" (MPF Supp. Construction, p. 5). The parties have previously agreed, with respect to claim 13, that "predetermined destination" means "a designated location, *i.e.* a video monitor, printer, modem, local area network, mailbox, serial port, or to a file on a hard disk" (Agreed Construction, p. 4).

The differences between the parties' proposed structures have narrowed considerably since the hearing. The main difference now seems to be that PICIS lists, in the alternative, seven specific steps from the figures of the patent, as well as a general reference to "portions of main module," while SIS excludes this reference to the main module and specifies that the structure is "[s]oftware for performing step 83 of Fig. 6; and at least one of the following: [list of alternative steps]" (MPF Supp. Construction, p. 5). SIS thereby proposes a structure that requires the software to perform two steps-step 83, which was also the first alternative step in PICIS' list, along with one of the other six steps listed by both parties.

As stated with respect to a previously addressed "means" element, I do not find the reference to portions of the main module a meaningful construction in view of the more specific software steps set forth in the specification.

It is not clear why SIS says that step 83 of Fig. 6 is a required step for the "means for transmitting the data," and, indeed, the parties' joint construction of the previous "means for polling or listening to the individual sensors" includes step 83 as one of the alternative structures for that element (MPF Supp. Construction, p. 5). Thus, it seems both unnecessary and inappropriate to include step 83 in the structure of the present "means" element.

Therefore, the following construction is recommended:

The function for "means for transmitting the data information generated by the individual sensors to a predetermined destination" is "transmitting the data information generated by the individual sensors to a predetermined destination," and the structure is "software code for performing at least one of the following steps: step 293 of Fig. 33A; step 295 of Fig. 33A; step 301 of Fig. 33A; step 303 of Fig. 33B; step 305 of Fig. 33B; or step 307 of Fig. 33B."

L. ***Wherein the external process request is an independently executable software program which runs simultaneously and independently with respect to the software engine*** **(claim 13)**

The parties do not differ significantly on the construction of this term, except that SIS makes reference in its proposed construction that "an external process request for data from a sensor is sent to the main module" (SIS Construction, p. 64). However, the plain claim language is clear, and the additional language proposed by SIS is not appropriate.

Therefore, the following construction is recommended:

"Wherein the external process request is an independently executable software program which runs simultaneously and independently with respect to the software engine" means "the external process request is a software program that can run by itself and at the same time as the software engine."

M. ***Wherein the interpreting means includes a module disposed in data receiving relation relative to the external process request*** **(claim 13)**

Since this phrase is not a mean plus function element, SIS' construction unduly limits the claim the specific algorithms of the specification and the main module. Most of the terms in this phrase are previously addressed, and most of the other terms are plain from the claim language. "Module" appears to be the only term that is not plain, and the specification makes clear that a module is a portion of software (e.g. "the present invention [which is software] is made up of several modules," 6:50-51).

Therefore, the following construction is recommended:

"Wherein the interpreting means includes a module disposed in data receiving relation relative to the external process request" means "the interpreting means includes software that receives data from the external process request."

N. ***Which is operable to initiate an object queue loop which identifies different object types including a mailbox through which data information is changed with the external process*** **(claim 13)**

The parties' proposed constructions of this phrase are unclear and inconsistent. PICIS construes the phrase as a "means plus function" element (PICIS Construction, p. 57), while SIS says that it refers to the "data receiving module" (SIS Brief, p. 38). Since the phrase does not appear in the form of a means plus function element, and since it logically refers back to the data receiving module most immediately recited in the claims, SIS' characterization of the phrase is most sensible. Beyond that, the phrase consists of terms either previously construed or on which the parties agree, connected by plain claim language.

Therefore, the following construction is recommended:

"Which is operable to initiate an object queue lop which identifies different object types including a mailbox through which data information is exchanged with the external process" means "the software starts the object queue loop which identifies different object types, including a mailbox object which communicates data with the external process."

O. ***Timer which regulates the frequency of data exchanges with the individual sensors requested by the external process*** **(claim 13)**

A "timer" is a well known computer device (e.g. a register) that measures time. See definition of "timer" in *IBM Dictionary of Computing,* p. 437 (Joint Claim Construction Statement, Tab G, page B033). Otherwise, the plain language of the claim is clear. SIS improperly imparts specifics from the specification that are not necessary for construction of this phrase.

Therefore, the following construction is recommended:

"Timer which regulates the frequency of data exchanges with the individual sensors requested by the external process" means "a computer implemented timer for controlling the frequency of data exchanges with the individual sensors requested by the external process."

P. ***Wherein the module further processes the data exchanges which the individual sensors*** **(claim 13)**

The term "module" is discussed above in section M. The plain claim language determines the construction of this phrase.

Therefore, the following construction is recommended:

"Wherein the module further processes the data exchanges which the individual sensors" means "software that facilitates the communications of data to and from the individual sensors."

Q. ***Driver skeleton*** **(claims 13 and 14)**

The parties agree that a "driver skeleton" is a software framework for a driver. SIS proposes adding to that construction additional specific detail about the language of the driver skeleton and the level of skill of one programming the driver skeleton. There is no reason why the language need be part of the construction, nor does the term require reference to the skill of a programmer. Furthermore, the remainder of claim 13 sets forth several characteristics of the driver that further define it.

Therefore, the following construction is recommended:

"Driver skeleton" means "a software framework for a driver."

R. ***Wherein the driver decoding area decodes the variables requested by the external process request* (claim 13)**

This phrase is listed in the parties' proposed constructions of terms on which they disagree, but it appears that all of the constituent terms are now the subject of agreement by the parties. Thus, no further construction is given to this phrase.

S. ***Means for routing the variables generated by the individual sensors, and which have been decoded by the driver decoding area, to a predetermined destination, or to the requesting external process, as appropriate* (claim 13)**

The parties have agreed that the function of this element is as stated in the element (MPF Supp. Construction, p. 32). The parties differ on the corresponding structure, in that PICIS references two additional corresponding structures at least portions of the main module; and software for performing step 83 of Fig. 6. Reference is made to Section K above, which is a similar element of claim 1. PICIS is consistent in its proposed structure of the two claims, which SIS makes no reference to step 83 here, while stating that step 83 was a required step in all structures for the similar element in claim 1. I adopt the reasoning set forth in Section K for the structure of the present element under consideration.

Therefore, the following construction is recommended:

The function for "means for routing the variables generated by the individual sensors, and which have been decoded by the driver decoding area, to a predetermined destination, or to the requesting external process, as appropriate," is "for routing the variables generated by the individual sensors, and which have been decoded by the driver decoding area, to a predetermined destination, or to the requesting external process, as appropriate," and the structure is "software code for performing at least one of the following steps: step 293 of Fig. 33A; step 295 of Fig. 33A; step 301 of Fig. 33A; step 303 of Fig. 33B; step 305 of Fig. 33B; or step 307 of Fig. 33B."

T. ***A library function which can be accessed from the external process* (claim 14)**

The parties agree to the extent that a library function is software or a program for transforming a user demand (PICIS Construction, p. 62; SIS Construction, p. 75). The parties also both cite the same portion of the specification that discusses the library functions (6:64-7:8). The specification does not provide a definition of "library function," so much as it discusses characteristics that may apply. Since claim 14 specifically sets forth that the library function is accessed from the external process, as well as what the software does, there is no reason that the construction of the term should be further limited, as suggested by SIS. The remainder of the phrase is not subject to dispute.

Therefore, the following construction is recommended:

"A library function which can be accessed from the external process" means "software or a program for transforming a user demand accessed from the external process."

U. ***Where the main module includes a memory which stores a plurality of driver which are operable to interpret the external process request for information* (claim 14)**

This phrase is composed of terms previously construed (Sections E and I above) or for which the parties have agreed to the construction.

V. ***Where in the library function transmits the external process request for information and converts this external process request into a formal which is understood by the main module*** **(claim 14)**

The parties agree to the extent that the library function translates the external process request into a format that the main module can understand (PIGS Construction, p. 64; SIS Construction, p. 77). SIS adds in additional limitations that are not required by the plain language of the claim.

Therefore, the following construction is recommended:

"Wherein the library function transmits the external process request for information and converts this external process request into a format which is understood by the main module" means "the library function translates the external process request into a format that the main module can understand."

W. ***Wherein the main module is operable upon the request of the external process, to ... create a mailbox through which data information is exchanged with the external process, and initiate an object queue loop which identifies predetermined object types, and wherein the main module initiates a process unique to each of the object types, and the particular remote sensor requested by the external process*** **(claim 14)**

This clause is listed in the parties contested claim elements (PICIS Construction, p. 65; SIS Construction, p. 78). However, the parties do not raise any particular terms in the clause that are not previously title subject of agreement or have not been otherwise construed. These constructions and the plain claim language seem to be sufficient to understand this clause. To the extent that SIS proposes more specific details or limitations to the construction of this clause, that is rejected.

X. ***Where the main module is operable upon the request of the external process, to ... initiate an object queue loop which identifies predetermined object types*** **(claim 14)**

It is unclear why the parties have listed this clause since it is a subset of the clause addressed above in Section W. In any case, the terms included in this clause have been addressed in Section N above with respect to claim 13.

Y. ***Means for routing the variables generated to a predetermined destination, or to the requesting external process, as appropriate*** **(claim 14)**

This means plus function element is basically the same as in claim 13, discussed in Section S. That reasoning applies here.

Therefore, the following construction is recommended:

The function for "means for routing the variables generated to a predetermined destination, or to the requesting external process, as appropriate," is "routing the variables generated to a predetermined destination, or to the requesting external process, as appropriate," and the structure is "software code for performing at least one of the following steps: step 293 of Fig. 33A; step 295 of Fig. 33A; step 301 of Fig. 33A; step 303 of Fig. 33B; step 305 of Fig. 33B; or step 307 of Fig. 33B."

Z. ***Means for overlaying and updating the predetermined adaptable driver format to substantially correspond to the characteristics of the sensor*** **(claim 3)**

"Means for overlaying a predetermined adaptable driver" has already been addressed with respect to claim 1 (Section J). Thus, the additional limitation added by this claim element is essentially a "means for updating the predetermined adaptable driver format to substantially correspond to the characteristics of the sensor." SIS states that the term "updating ... driver format" means modifying a driver to decode the variables from a particular sensor (MPF Supp. Construction, p. 7). SIS' point appears to be that adjusting the format of a driver is to be distinguished from updating the values of variables in the driver (such as shown, for example, in Fig. 5, item 80 of the patent). I agree that the word "format" must be given meaning, and, therefore, the function for this element must include a more explicit definition of "updating ... driver format." Moreover, based on this construction, the corresponding structures referenced by PICIS (e.g. step 80 of Fig. 5) that refer to the "updating of the driver overlay with the variables requested" do not relate to the updating of the *format* of the driver overlay.

Therefore, the following construction is recommended:

The function for "means for overlaying and updating the predetermined adaptable driver format to substantially correspond to the characteristics of the sensor" is "overlaying the predetermined adaptable driver, as defined in claim 1, and modifying the format of the predetermined adaptable driver to substantially correspond to the characteristics of the sensor," and the structure is "the skeleton driver code as shown in Cols. 20-26 of the patent."

AA. ***Means for transmitting the decoded information to a predetermined destination direct by the external process*** **(claim 3)**

The function of element is substantially the same as a similar element in claim 1 (addressed in Section K, above), on which claim 3 is indirectly dependent. "Decoded information" is the same as "data information" of claim 1, and although this element is explicit about the transmitting being directed by the external process, that is implicit in claim 1. Thus, there is no reason for the structure to be different, yet the parties both make reference to Subroutine RO of Fig. 32 for their proposed constructions of claim 3, even though neither referred to Fig. 32 when discussing the similar element of claim 1. The briefs and proposed constructions of the parties do not explain any reason why this element should be construed any differently than in claim 1.

Therefore, the recommended construction for this element is the same as the element in claim 1 provided in Section K.

BB. ***Polling timer process*** **(claim 4)**

The "timer process" is one of the "predetermined processes" of claim 2. The parties have agreed that a "predetermined process" is "preset logic (e.g.software) associated with an object type" (Agreed Construction Supp.). The parties have also agreed that "polling" means "interrogating sensors to transmit their data" (MPF Construction Supp, p. 5). "Timer" is discussed above, in Section O, as "a computer device (e.g. a register) that measures time."

Therefore, the following construction is recommended:

"Polling timer process" means "software that is associated with timing of the interrogation of sensors to transmit their data."

CC. ***Delay timer process*** **(claim 4)**

The "timer process" is one of the "predetermined processes" of claim 2. The parties have agreed that a "predetermined process" is "preset logic (e.g.software) associated with an object type" (Agreed Construction Supp.). "Timer" is discussed above, in Section O, as "a computer device (e.g. a register) that measures time." As stated by SIS, a "delay timer is a timer which sets a delay for obtaining an answer from a sensor" (SIS Brief, p. 62). This seems sensible from the plain claim language and the specification, and PICIS has raised no objection to this.

Therefore, the following construction is recommended:

"Delay timer process" means "software that is associated with timing which sets a delay for obtaining an answer from a sensor."

DD. ***Means for determining the characteristics of the driver associated with the polling request*** **(claim 5)**

The parties agree on the function on this "means plus function" element (MPF Construction Supp, p. 10). The parties also agree that the corresponding structure relates to steps 173 and 175 of Fig. 18. The parties differ only on whether these steps define the structure in the conjunctive or the alternative. Since these steps are simply sequential questions in the flow chart that ask if the driver is of a specific type, presumably it is necessary that the polling timer request be capable of making both inquiries.

Therefore, the following construction is recommended:

The function for "means for determining the characteristics of the driver associated with the polling request" is "determining the characteristics of the driver associated with the polling request," and the structure is "software code that carries out steps 173 and 175 of Fig. 18."

EE. ***A predetermined destination which corresponds to the external process request upon deactivation of the delay timer*** **(claim 5)**

The parties have previously agreed, with respect to claim 13, that "predetermined destination" means "a designated location, *i.e.* a video monitor, printer, modem, local area network, mailbox, serial port, or to a file on a hard disk" (Agreed Construction, p. 4). Furthermore, the parties have now agreed to the construction of the "means plus function" element in which this phrase occurs (MPF Construction Supp, p. 12). Thus, there appeals to be net further construction necessary for this phrase.

FF. ***Means for determining if conditions for polling the sensor have been met*** **(claim 6)**

The parties agree that the function of this element is "determining if conditions for polling the sensor have been met" (MPF Construction Supp, p. 13). However, the parties differ on the corresponding structure, with PICIS identifying step 193 of Fig. 21 and SIS identifying steps 170-173 of Fig. 18. Neither party's suggested structure seems clearly related to the function of determining if polling conditions have been met. The clearest exposition of what occurs is given by SIS (SIS Brief, p. 69), which references the text of the patent (13:30-57) as explaining that the delay timer process includes software code that is "flagged" to a handshake that initiates communications.

Therefore, the following construction is recommended:

The function for "means for determining if conditions for polling the sensor have been met" is "determining if conditions for polling the sensor have been met," and the corresponding structure is "software code that recognizes a signal from the sensor that it is prepared to communicate."

GG. ***Means for transmitting the information received from the sensors to a predetermined destination*** **(claim 6)**

The parties are in substantial agreement on the construction of this "mean plus function" element, except that PICIS includes as an alternative structure "at least portions of main module (Fig.34)" (MPF Construction Supp, p. 14). As addressed in Section B.4 above, this phrase does not set forth a structure.

Therefore, the following construction is recommended:

The function for "means for transmitting the information received from the sensors to a predetermined destination" is "transmitting the information received from the sensors to a predetermined destination," and the corresponding structure is "software for performing any one of the steps of Subroutine RO of Fig. 32 as described at 18:54-19:3 and Subroutine Z of Figs. 33A and 33B as described at 19:3-20:5."

HH. ***Wherein the keyboard process includes a status process*** **(claim 7)**

The parties have agreed to the construction of "keyboard process" with respect to claim 2. This phrase adds the term "status process" as a part of the "keyboard process," so that "status process" is also part of the "preset logic" associated with the keyboard (Agreed Construction Supp.). However, since the word "status" is not specific, and since the subsequent elements of claim 7 define what the "status process" includes, there is no reason to separately construe the term "status process" apart from the construction of the elements that fallow in claim 7.

II. ***Wherein the keyboard process includes ... a keyboard interaction process*** **(claim 7)**

As in Section HH above, "keyboard interaction process" is not specific, nor is the term defined in the specification. However, detailed claim elements setting forth what is included in the "keyboard interaction process" appear in claim 8, which is dependent on claim 7. The term "keyboard interaction process" should therefore be given the most general construction in claim 7 based on the literal words.

Therefore, the following construction is recommended:

"Keyboard interaction process" means "software that relates to keyboard input."

JJ. ***Wherein the status process includes means for determining the type of input or driver*** **(claim 7)**

The parties are in agreement about the function of this "means plus function" element, and also agree to the corresponding structure, except that PICIS again adds "Main module (Fig.34)" as an alternative structure (MPF Construction Supp, p. 15). As stated previously, this phrase does not set forth a structure.

Therefore, the following construction is recommended:

The function for "means for determining the type of input or driver" is "determining the type of input or driver," and the corresponding structure is "software for performing step 215 of Fig .24, as described at 16:56-59."

KK. ***Means for selecting an appropriate data format which corresponds with the type of input or driver*** **(c1aim 7)**

The parties agree that the function of this "means plus function" element is "selecting an appropriate data format which corresponds with the type of input or driver," but SIS further states that the term "selecting an appropriate data format means that a user (in step 218) chooses a format for data which correlates with the type of input being received or driver used" (MPF Construction Supp, p. 15). However, nothing in the specification is cited that requires such a further construction of the function.

The parties agree on the corresponding structure, except that PICIS again inappropriately includes "portions of the main module" as an alternative structure. PICIS also includes "video and printer of Fig. 34" as another alternative structure. I fail to understand why PICIS considers either a video terminal or a printer to be an alternative software structure.

Therefore, the following construction is recommended:

The function for "means for selecting an appropriate data format which corresponds with the type of input or driver" is "selecting an appropriate data format which corresponds with the type of input or driver," and the corresponding structure is "software for performing step 218 of Fig.24, as described at 16:62-65, or for performing step 230 of Fig. 25, as described at 17:3-11."

LL. ***Wherein the keyboard interaction process includes ... means for identifying the type of driver which corresponds to the user request*** **(claim 8)**

The parties agree that the function of this "means plus function" element is "identifying the type of driver which corresponds to the user request" (MPF Construction Supp, p. 17). The parties differ somewhat on the corresponding structure, in that SIS cites step 232 of Fig. 26 and all of subroutine R in Fig. 27, whereas PICIS merely cites step 252 of Fig. 27 and corresponding text at 17:57-59. Step 232 is not involved in identifying the type of driver, but rather whether there is a request for a driver. Subroutine R broadly does more than just determine a driver type. Furthermore, SIS gives the same structure for the next "mean plus function" element in the claim, which results in no differentiation between the two elements.

Therefore, the following construction is recommended:

The function for " 'means for identifying the type of driver which corresponds to the user request" is "identifying the type of driver which corresponds to the user request," and the corresponding structure is "software for performing step 252 of Fig. 27, as described at 17:57-59."

MM. ***Means for loading a driver code overlay which corresponds with the type of driver*** **(claim 8)**

The parties agree that the function of this "means plus function" element is "loading a driver code overlay which corresponds with the type of driver" (MPF Construction Supp, p. 17). SIS repeats the same corresponding structure as for the previous element (see Section LL), and, for the same reasons, this broader statement of structure is rejected.

Therefore, the following construction is recommended:

The function for "means for loading a driver code overlay which corresponds with the type of driver" is "loading a driver code overlay which corresponds with the type of driver," and the corresponding structure is "software for performing steps 251 and 254 of Fig. 27, as described at 17:53-65."

NN. ***Means for requesting a direction and format for the output*** **(claim 8)**

The parties agree that the function of this "means plus function" element is "requesting a direction and format for the output" (MPF Construction Supp, p. 18). The parties agree, at least, that the corresponding structure includes steps 234 and 240 of Fig. 26 as described at 17:12-28. SIS states that the structure also includes Subroutines S and T of Figs. 28 and 29. These subroutines are those called by steps 234 and 240 when a request is recognized; however, these routines are not part of the keyboard interaction process, but rather are the responses to such requests.

Therefore, the following construction is recommended:

The function for "means for requesting a direction and format for the output" is "requesting a direction and format for the output," and the corresponding structure is "software for performing steps 234 and 240 of Fig. 26, as described at 17:12-28."

OO. ***Means for loading the adaptable driver*** **(claim 9)**

This "means plus function" element is a further limitation of the means for overlaying the predetermined adaptable driver (see Section J). As such, SIS' suggestion that the function of this element include "overlaying" as well as "loading" the adaptable driver seems redundant to claim 1 (MPF Construction Supp, p. 18). The parties otherwise agree that the function is "loading the adaptable driver,"

The parties agree that the corresponding structure relates to step 173 of Fig. 12, step 145 of Fig. 13 and step 274 of Fig. 30. The parties differ only on whether these steps define the structure in the conjunctive or the alternative. In this case, the cited steps are not directly related to each other, in that each is an alternative situation where a driver is loaded.

Therefore, the following construction is recommended:

The function for "means for loading the adaptable driver" is "loading the adaptable driver," and the corresponding structure is "software for performing step 123 of Fig. 12, as described at 13:35-38, or for performing step 145 of Fig. 13, as described at 14:7-10, or for performing step 274 of Fig. 30, as described at 18:33-37."

PP. ***Means for modifying the predetermined adaptable driver with the variables which substantially correspond with the variables requested by the external process and which correspond with the sensor being polled*** **(claim 9)**

The parties agree that the function of this "means plus function" element is "modifying the predetermined adaptable driver with the variables which substantially correspond with the variables requested by the external process and which correspond with the sensor being polled" (MPF Construction Supp, p. 19).

The parties differ on whether step 80 of Fig. 5 or step 154 of Fig. 13 constitutes corresponding structure. SIS argues that updating the driver is not the same as modifying the driver. The patent specification is not helpful in this regard because the word "modify" does not appear in the text. This issue is similar to that discussed for claim 3 in Section Z, above. However, in that claim the language to be construed was "updating the predetermined adaptable driver *format* to substantially *correspond to the characteristics* of the sensor." Interestingly, in the construction of this phrase in claim 3, the effect was to construe the word "updating" as "modifying." Here, SIS is saying that "modifying" is to be distinguished from "updating." While the choice of words is not very satisfactory in either case, the total context of the claim language is the guide to the most reasonable construction.

Claim 9 and claim 3 both depend from claim 2. Claim differentiation would suggest that the "means for ... updating the predetermined adaptable driver" in claim 3 should be construed differently from the "means for modifying the predetermined adaptable driver" of claim 9. Furthermore, the context of the elements of claim 9 is the communication with a sensor. A construction of the structure of the present "means plus function" element is more consistent with reading the values of the variables requested by the external process and updating those values in the predetermined adaptable driver. It does not seem sensible that the format or structure of the skeleton of the predetermined adaptable driver would be modified in response to a request by the external process.

Therefore, the following construction is recommended:

The function for "means for modifying the predetermined adaptable driver with the variables which substantially correspond with the variables requested by the external process and which correspond with the sensor being polled" is "modifying the predetermined adaptable driver with the variables which substantially correspond with the variables requested by the external process and which correspond with the sensor being polled," and the corresponding structure is "software for performing step 80 of Fig. 5, as described at 11:24-30, or for performing step 154 of Fig. 13, as described at 14:16-22, where the software acts on a predetermined adaptable driver."

QQ. *Claim 10*

Claim 10 is dependent on claim 2, and sets forth eleven "means plus function" elements that farther limit the "means for overlaying the predetermined adaptable driver." In each case, the parties agree that the function is exactly as stated in the claim for each such element (MPF Construction Supp, pp. 21-27). In most cases, the parties are also in agreement that the corresponding structure disclosed in the specification is one or more lines of software code of the example C language driver skeleton shown at columns 19-24 of the patent, along with text in the specification that discusses the associated software code. The parties differ in their exact description of the structure, with PICIS referring to "skeleton driver code" and SIS referring to "code in a predetermined adaptable driver." I do not understand this to be a substantive difference, so I will refer to the structures as "driver skeleton code" since that is specifically how the example is identified in the patent (see, e.g., 20:36-39). The listed code in most cases represents a data identifier (e.g., mon_name), along with the type of data (e.g., "int" for integer; "char" for characters) and an entry for the data (fixed or variable).

Based on this general discussion, the construction on the elements of claim 10 are set forth below. Specific discussion for any particular claim element is also set forth, where appropriate.

1. ***Means for entering the name of the adaptable driver format***

The parties agree on a structure that refers to step 251 of Fig. 27. In addition, PICIS proposes an alternative structure that is the driver skeleton code "mon_name" (MPF Construction Supp, pp. 2). Inclusion of the driver skeleton code as an alternative structure is appropriate since the specification specifically says that the skeleton provides a location where the name of the sensor is entered (25:65-68).

Therefore, the following construction is recommended:

The function for "means for entering the name of the adaptable driver format" is "entering the name of the adaptable driver format," and the corresponding structure is "software for performing step 251 of Fig. 27, as described at 17:53-57, or driver skeleton code 'mon_name' as shown at Cols. 21 and 22, as described at 25:65-68."

2. ***Means for providing a polling timer interval for the sensor***

The following construction is recommended:

The function for "means for providing a polling timer interval for the sensor" is "providing a polling timer interval for the sensor," and the corresponding structure is "driver skeleton code 'mon_poll_int' as shown at Cols. 21 and 22, as described at 25:68-26:22."

3. ***Means for entering a timing delay interval for the polling timer and which is operable to delay the polling requests to the sensor***

The following construction is recommended:

The function for "means for entering a timing delay interval for the polling timer and which is operable to delay the polling requests to the sensor" is "entering a timing delay interval for the polling timer and which is operable to delay the polling requests to the sensor," and the corresponding structure is "driver skeleton code 'mon_delay_int' as shown at Cols. 21 and 22, as described at 26:22-29."

4. ***Means for providing a baud rate and parity for the sensor being polled***

The following construction is recommended:

The function for "means for providing a baud rate and parity for the sensor being polled" is "providing a baud rate and parity for the sensor being polled," and the corresponding structure is "driver skeleton code 'mon_baud' and 'mon_parity' as shown at Cols. 21 and 22, as described at 26:29-34."

5. ***Means for providing the name of the language which is employed by the sensor***

The following construction is recommended:

The function for "means for providing the name of the language which is employed by the sensor" is "providing the name of the language which is employed by the sensor," and the corresponding structure is "driver skeleton code 'mon_language' as shown at Cols. 21 and 22, as described at 26:34-42."

6. ***Means for entering the type of sensor device being polled***

The following construction is recommended:

The function for "means for entering the type of sensor device being polled" is "entering the type of sensor device being polled," and the corresponding structure is "driver skeleton code 'mon_flavor' as shown at Cols. 21 and 22, as described at 26:43-50."

7. ***Means for determining whether the sensor can be polled, or listened to, and providing instructions as appropriate***

The following construction is recommended:

The function for "means for determining whether the sensor can be polled, or listened to, and providing instructions as appropriate" is "determining whether the sensor can be polled, or listened to, and providing instructions as appropriate," and the corresponding structure is "driver skeleton code 'mon_ pollable' as shown at Cols. 21 and 22, as described at 26:50-61."

8. ***Means for providing special communication instructions to the sensor, if necessary***

The following construction is recommended:

The function for "means for providing special communication instructions to the sensor, if necessary" is "providing special communication instructions to the sensor, if necessary," and the corresponding structure is "driver skeleton code 'mon_spec_comdef' as shown at Cols. 21 and 22, as described at 26:61-68."

9. ***Means for entering the number of polling requests which must be completed to ensure receipt of all data information from sensor***

The following construction is recommended:

The function for "means for entering the number of polling requests which must be completed to ensure receipt of all the data information from the sensor" is "entering the number of polling requests which must be completed to ensure receipt of all the data information from the sensor," and the corresponding structure is "driver skeleton code 'mon_step4hello' and 'mon_numof_steps' as shown at Cols, 21 and 22, as described at 26:69-27:21."

10. ***Means for providing the variables which are to be read from the sensor***

The following construction is recommended:

The function for "means for providing the variables which are to be read from the sensor" is "providing the variables which are to be read from the sensor," and the corresponding structure is "driver skeleton code in section "VARIABLES TO READ FROM DEVICE" as shown at Cols. 23 and 24, as described at 27:22-49."

11. ***Means for decoding the variables received from the sensor and delivering same to a predetermined destination***

The parties differ somewhat on the delineation of the corresponding structure for this "means plus function" element, with PICIS adding a reference to a particular portion of the code. However, the code section entitled "BEGIN CODE TO DECODE STRING FROM DEVICE" runs from that title to the end of the code listing at Col. 25/26, which encompasses the "Passing of Argument" code lines.

Therefore, the following construction is recommended:

The function for "means for decoding the variables received from the sensor and delivering same to a predetermined destination" is "decoding the variables received from the sensor and delivering same to a predetermined destination," and the corresponding structure is "driver skeleton code in section "BEGIN CODE TO DECODE STRING FROM DEVICE" as shown at Cols. 23 through 26, as described at 27:49-51."

RR. ***Claim 11***

Claim 11 is directly dependent on claim 10, sets forth three "means plus function" elements that further limit the "means for providing special communications instructions" of claim 10 (see Section QQ.8). Claim 11 seems redundant of two "means plus function" elements of claim 10 (see Sections QQ.8 and QQ.9), since identical structures are proposed by the parties for the "means plus function" elements in this claim. The distinction seems to be that the stated functions in claim 11 are the operation of the software with the driver, whereas the functions of claim 10 are the setting op of the driver by entering the desired characteristics. The general comments for claim 10 also apply to the construction of the elements of claim 11.

1. ***Means for initialization of the senor being polled***

The following construction is recommended:

The function for "means for initialization of the senor being polled" is "initialization of the senor being polled," and the corresponding structure is "driver skeleton code 'mon_spec_comdef' as shown at Cols. 21 and 22, as described at 26:61-68."

2. ***Means for establishing communication with the sensor being polled***

The following construction is recommended:

The function for "means for establishing communication with the sensor being polled" is "establishing communication with the sensor being polled," and the corresponding structure is "driver skeleton code 'mon_spec_comdef' as shown at Cols. 21 and 22, as described at 26:61-68."

3. ***Means for polling the sensor a predetermined number of times to receive all the data information produced by same***

The following construction is recommended:

The function for "means for polling the sensor a predetermined number of times to receive all the data information produced by same" is "polling the sensor a predetermined number of times to receive all the data information produced by same," and the corresponding structure is "driver skeleton code 'mon_ step4hello' and 'mon_numof_steps' as shown at Cols. 21 and 22, as described at 26:69-27:21."

SS. *Claim 12*

Claim 12 is directly dependent on claim 11, sets forth two "means plus function" elements that further limit the "means for providing the variables which are to be read from the sensor" of claim 10 (see Section QQ.10). The general comments for claim 10 also apply to the construction of the elements of claim 12.

1. *Means for providing the number of variables which are to be read from the sensor*

The following construction is recommended:

The function for "means for providing the number of variables which are to be read from the sensor" is "providing the number of variables which are to be read from the sensor," and the corresponding structure is "driver skeleton code 'vars_numof' as shown at Cols. 23 arid 24, as described at 27:43-45."

2. *Means for identifying the variables which are to be read from the sensor*

The following construction is recommended:

The function for "identifying the variables which are to be read from the sensor" is "identifying the variables which are to be read from the sensor," and the corresponding structure is "driver skeleton code 'var_present[2]' as shown at Cols. 23 and 24, as described at 27:45-49."

V. *Conclusion*

The undersigned Special Master hereby submits the above recommendations to the Court.

N.D.Ga.,2006.
PICIS, Inc. v. Surgical Information Systems, LLC

Produced by Sans Paper, LLC.