

relating to communications, such as payment terminal devices and other devices in which message processing and communication comprise a significant proportion of the operation of the device,” *see id.* at 4:51-65, this does not need to be a part of the court’s construction. Accordingly, the court construes “virtual machine means” and “virtual machine” to mean “a computer programmed to emulate a hypothetical computer for applications relating to transport of data.”

b. “emulatable in different computers having incompatible hardwares or operating systems”

Plaintiffs’ Proposed Construction	Defendants’ Proposed Construction
Capable of being implemented on computers having different hardware or operating systems.	The virtual machine means of the claimed communications device processes instructions expressed in a language that is hardware/operating system-independent so that the claimed virtual machine means can also be implemented, without compiling to a hardware/operating system-specific code or otherwise altering the virtual machine means or the instructions it processes, on other computers having hardware that is incompatible with that of the claimed device and on yet other computers having operating systems that are incompatible with that of the claimed device

Claim 1 of the ’945 Patent, which is representative of the use of the phrase “emulatable in different computers having incompatible hardwares or operating systems,” recites as follows:

A communication device which is arranged to process messages for communications, comprising a virtual machine means which includes

a virtual function processor and function processor instructions for controlling operation of the device, and

message induction means including a set of descriptions of message data;

a virtual message processor...

wherein the virtual machine means is emulatable in different computers having incompatible hardwares or operating systems.

Id. at 50:49-67 (emphasis added). CardSoft urges the court to construe the phrase “emulatable in different computers having incompatible hardwares or operating systems” to mean “capable of being implemented on computers having different hardware or operating systems.” Defendants, on the other hand, argue that the court should construe the phrase to mean “the virtual machine means of the claimed communications device processes instructions expressed in a language that is hardware/operating system-independent so that the claimed virtual machine means can also be implemented, without compiling to a hardware/operating system-specific code or otherwise altering the virtual machine means or the instructions it processes, on other computers having hardware that is incompatible with that of the claimed device and on yet other computers having operating systems that are incompatible with that of the claimed device.” The parties’ primary disputes are: (1) whether the virtual machine means must process instructions expressed in “a hardware/operating system-independent language;” and (2) whether the virtual machine must be implemented on various different computers “without compiling to a hardware/operating system-specific code or otherwise altering the virtual machine means or the instructions it processes.” As discussed above, the court rejects Defendants’ argument that the virtual machine must be expressed in “a hardware/operating system-independent language.” Accordingly, in its analysis of this term, the court will address only Defendants’ contention that the virtual machine cannot be compiled directly to the hardware-specific code of a particular processor.

Defendants argue that compiling to the hardware-specific code is outside the claim language because, if such compiling is done, then the virtual machine would be limited to operation on that one particular processor and would no longer be emulatable on a different, incompatible processor. Similarly, Defendants contend that programming the virtual machine in code that is specific to a particular operating system would limit operation of the virtual machine

to that single operating system and preclude its operation on a different, incompatible operating system. Defendants, therefore, urge the court to conclude that the “emulatable” limitation must be construed to recognize that it requires that the virtual machine not be compiled to a hardware/operating system-specific code.

As noted above, however, both Claim 5 and Claim 6 of the '945 Patent require that the virtual message processor and the virtual function processor, respectively, are implemented in the native code of the specific microprocessor in the device. As such, Defendants' proposed limitation is again at odds with the plain language of Claims 5 and 6 of the '945 Patent.³ Furthermore, the common specification teaches that the “message processor 105 and protocol processor 106 are implemented *in native code of the payment terminal* and therefore operate at relatively high speed.” '945 Patent at 10:26-29 (emphasis added). Thus, Defendants' proposed construction would also improperly read embodiments out of the scope of the patents-in-suit. As such, the court rejects Defendants' proposed construction.

Plaintiff's proposed construction,⁴ however, is more consistent with the plain meaning of the words of the claim and with the common specification of the patents-in-suit. For example, the specification states that “[d]ifferent incompatible computers may be programmed to emulate the same hypothetical computer. Any computer programmed to emulate the hypothetical computer will thus be capable of executing programs for the virtual computer.” *See, e.g.*, '945 Patent 3: 40-46. The specification further states that “[t]he virtual machine 101, 102, 103 can be

³ Defendants again rely on prosecution history statements discussed in the court's analysis of the “virtual machine means.” In accordance with the court's previous analysis, the court rejects Defendants' contention that any of the prosecution history statements on which they rely constitute a clear disclaimer of virtual machines that have been compiled down to the hardware-specific code of the processor.

⁴ “Capable of being implemented on computers having different hardware or operating systems.”

adapted for many different hardware 100 arrangements (i.e. many different brands of payment terminal). Different arrangements of hardware 100 can therefore be controlled by the same application software 104.” *See id.* at 10:2-7. Thus, the court construes the phrase “emulatable in different computers having incompatible hardware or operating systems” to mean “capable of executing programs on different computers having incompatible hardware or operating systems.” *See id.* at 3:43-46 (“Any computer programmed to emulate the hypothetical computer will thus be *capable of executing programs* for the virtual computer.”) (emphasis added)).

c. “virtual message processor”

Plaintiffs’ Proposed Construction	Defendants’ Proposed Construction
A program module which processes messages, including assembling, disassembling and/or comparing messages, for communication to and/or from a payment terminal device.	Software that emulates a physical processor on the claimed communications device to handle the claimed messages in accordance with instructions expressed on the communications device in a hardware/operating system-independent language.

The parties’ only dispute regarding the claimed “virtual message processor” is whether the processor must “handle the claimed messages in accordance with instructions expressed on the communications device in a hardware/operating system-independent language.” With regard to this term, Defendants argue that their proposed limitation is required by the following description of the “virtual message processor”:

The message processor means is preferably translated into the native code of the microprocessor in each hardware device on which the virtual machine is to be implemented. The message processor instructions are preferably virtual instructions to be expressed only in the language defined by the message processor means- and thus never requiring translation to any real hardware processor.

’945 Patent at 4:5-11. Furthermore, Defendants contend that the prosecution history confirms that their proposed limitation is necessary. In particular, Defendants argue that when the

applicant amended Claim 1 of the '945 Patent to add the “message instruction means” to the “virtual message processor” limitation, the applicant argued:

As stated in the Specification page 7, providing a separate virtual message processor allows for ‘faster, simpler programming.’ Stern does not teach the provision of the claimed virtual machine with a dedicated virtual message processor. That is, if a Java Virtual Machine as described in Stern is used to perform messaging, each application developed would be required to adjust to the characteristics of the different devices that the application was to execute on, such as screen width and fonts.

The claimed virtual message processor removes this burden from the development of the application and places it on the software platform that resides on the device. This relieves the application developers of the burden of programming to the physical characteristics of the platform that application will execute on.

Ex. G at 13-14, attached to Defendants’ Responsive Claim Construction Brief, Dkt. No. 210; *see also id.*, Ex. D at 2-3.

The specification explains that the “virtual machine processor” includes a “message processor 105” that is “implemented in software code.” ’945 Patent at 10:18-20. The specification then explicitly states that the “message processor 105 ... [is] *implemented in the native code* of the payment terminal and therefore operates at relatively high speed.” *Id.* at 10:26-29. When read in light of the specification, the claimed “virtual message processor” is implemented in the native code of the communications device. The court disagrees with Plaintiffs that the doctrine of claim differentiation requires the court to hold otherwise. Although claim 5 requires that “the message processor be implemented in the native software code of the microprocessor,” claim differentiation does not trump the clear import of the specification. *See Edward Lifesciences LLV v. Cook Inc.*, 582 F.3d 1322, 1332 (Fed. Cir. 2009) (“claim differentiation is a rule of thumb that does not trump the clear import of the specification.”). Here, the specification makes clear that the claimed “virtual message processor” is implemented

in the native code of the communications device. The specification, however, states that the claimed invention is not limited to devices configured to process payment transactions. *See id.* at 3:50-55. The court, therefore, rejects CardSoft’s proposed “payment terminal device” limitation.

In conclusion, the court construes “virtual message processor” to mean “software implemented in the native code of the communications device that processes messages, including assembling, disassembling and/or comparing messages, for communication to and/or from a communications device.”

d. “virtual function processor”

Plaintiffs’ Proposed Construction	Defendants’ Proposed Construction
A program module which controls and/or selects general operations of a payment terminal device.	Software that emulates a physical processor on the claimed communications device to control the operation of the device, and that interfaces with an application running on the device to process instructions from the application that are expressed on the communications device in a hardware/operating system-independent language.

Defendants again attempt to import a limitation, requiring that the “virtual function processor” “interface[] with an application running on the device to process instructions from the application that are expressed on the communications device in a hardware/operating system-independent language.” Defendants’ proposed limitation runs contrary to the language of Claim 6 of the ’945 Patent, which requires that “the function processor is implemented in the native code of the microprocessor.” Considering this, the court rejects Defendants’ proposed construction.

In contrast to Defendants’ proposed construction, CardSoft’s proposed construction is supported by the common specification of the patents-in-suit. In particular, the common specification states that the claimed virtual machine includes “a function processor 107 the

operation of which is to control and select general operations of the device not specially controlled by the message and protocol processors 105, 106.” ’945 Patent at 10:34-37; *see also id.* at 5:15-18. The court, however, again notes that the claimed invention is not limited to “payment terminal” devices. *See id.* at 3:50-55. The court, therefore, construes “virtual function processor” to mean “software which controls and/or selects general operations of a communication device.”

e. “message instruction means”

Plaintiffs’ Proposed Construction	Defendants’ Proposed Construction
Instructions arranged to provide directions for operation of a message processor, which include a description of a field of message data.	<p>Governed by § 112, ¶ 6.</p> <p>Function: Using the hardware/operating system-independent language of the virtual machine means to specify operations that the virtual message processor carries out on the claimed messages.</p> <p>Structure: A set of instructions for processing the claimed messages, issued by the application and written and loaded onto the claimed communications device in a hardware/operating system-independent language.</p>

Claim 1 of the ’945 Patent, which is representative of the patents’ use of the term “message instruction means,” recites as follows:

A communication device which is arranged to process messages for communications, comprising a virtual machine means which includes

a virtual function processor and function processor instructions for controlling operation of the device, and

message induction means [sic] including a set of descriptions of message data;

a virtual message processor, which is arranged to be called by the function processor and which is arranged to carry out the message handling tasks of assembling the messages, disassembling messages and comparing the messages under the direction of the *message instruction means that is arranged to provide directions for operation of the virtual message processor,*

whereby when a message is required to be handled by the communications device the message processor is called to carry out the message handling task,

wherein the virtual machine means is emulatable in different computers having incompatible hardwares or operating systems.

Id. at 50:49-67 (emphasis added). The parties' dispute concerning the claimed "message instruction means" is two-fold: (1) whether the term is governed by 35 U.S.C. § 112, ¶ 6; and (2) whether the claimed message instructions must be "in a hardware/operating system-independent language."

First, Defendants contend that the term "message instruction means" is subject to means-plus-function treatment. It is well settled the use of the word "means" in a claim limitation raises a rebuttable presumption that the limitation is a means-plus-function limitation under § 112, ¶ 6. *Altiris, Inc. v. Symantec Corp.*, 318 F.3d 1363, 1375 (Fed. Cir. 2003). This presumption may be rebutted only if the patentee can demonstrate that the claim language itself recites sufficient structure to perform the claim function in its entirety. *Id.* Because the "message instruction means" limitation uses the word "means," the presumption that this limitation is a means-plus-function limitation applies. The recited function of the "message instruction means" is clear from the plain language of the claims – that is, "[providing] directions for operation of the virtual message processor." CardSoft argues that the independent claims of the patents-in-suit recite sufficient structure to perform this function in its entirety. The court, however, is not persuaded that CardSoft has overcome the presumption that is invoked by the use of the term "means." As such, the court rejects CardSoft's argument that the term "message instruction means" is exempt from means-plus-function treatment.

Defendants argue that the function of the "message instruction means" is "using the hardware/operating system-independent language of the virtual machine means to specify

operations that the virtual message processor carries out on the claimed messages.” Defendants, however, offer no support for their proposed alteration of the function recited in the claims. Furthermore, Defendants’ proposed construction attempts to import a limitation as to the “way” in which the function is performed. Federal Circuit precedent, however, makes clear that the “court must not import unclaimed functions into means-plus-functions limitations.” *Applied Med. Res. Corp. v. U.S. Surgical Corp.*, 312 Fed. Appx. 326, 332 (Fed. Cir. 2009) (citing *JWW Enters., Inc. v. Interact Accessories, Inc.*, 424 F.3d 1324, 1331 (Fed. Cir. 2005)). As such, the court rejects Defendants’ proposed function and concludes that the function of the claimed “message instruction means” is “providing directions for operation of the virtual message processor.” See *Lockheed Martin Corp. v. Space Sys./Loral, Inc.*, 324 F.3d 1308, 1319 (Fed. Cir. 2003) (“The function is properly identified as the language after the ‘means for’ clause and before the ‘whereby’ clause, because a whereby clause that merely states the result of the limitations in the claim adds nothing to the substance of the claim.”).

With regard to the structure corresponding to this function, Defendants argue that the corresponding structure is “a set of instructions for processing the claimed messages, issued by the application and written and loaded onto the claimed communications device in a hardware/operating system-independent language.” As with their other proposed constructions, Defendants again seek to import a limitation, requiring that the claimed message instructions be “in a hardware/operating system-independent language.” Defendants’ proposed construction, however, again runs afoul of the language of the claims. In particular, Claim 7 of the ’945 Patent recites that “the message instruction means do not require translation to the native software code of the microprocessor.” According to the doctrine of claim differentiation, this creates a presumption that Claim 1 (from which Claim 7 depends) must cover both “message instruction

means” that do not require translation to the native software code of the microprocessor and those that do require translation. *See Seachange Intern., Inc. v. C-COR, Inc.*, 413 F.3d 1361, 1368-69 (Fed. Cir. 2005). The court is not convinced that Defendants have overcome this presumption. Furthermore, Defendants’ reliance on the specification for their proposed limitation is misplaced. Although the specification states that the “message processor instructions are preferably virtual instructions to be expressed only in the language defined by the message processor means- and thus never requiring translation to any real hardware processor,” this is merely a embodiment of the claimed “message processor instructions.” It is improper for the court to read such an embodiment into the claims. In summary, the court rejects Defendants’ proposed structure because it is not supported by the claim language, common specification, or prosecution history of the patents-in-suit.

Having carefully reviewed the patents-in-suit, the court concludes that the structures corresponding to the function of “providing directions for operation of the virtual message processor” are: 13:29-14:2; 15:23-34; Figure 11 and Figure 8. The specification states that “FIG. 11 is a schematic diagram illustrating the structure of the message instruction means 109.” ’945 Patent at 13:29-30. It then goes on to explain that structure in detail. *Id.* at 13:30-14:2. Furthermore, the specification states that:

the present invention includes another class of message instruction means, known as a “Form”. Instead of a Data Representation as a message descriptor, a Form includes description of a Location of the data field in the Form. FIG. 8 is a display provided by a development tool enabling the programmer to prepare message instructions for a Form message.

Id. at 15:23-29. The specification also explains the structure of the “form” embodiment of the “message instruction means.” *Id.* at 15:23-34. These are the only two structures identified in the specification that are clearly linked to the function of the “message instruction means.”

In conclusion, the court construes the term “message instruction means” as follows: (1) the function is “providing directions for operation of the virtual message processor;” and (2) the structure is “13:29-14:2; 15:23-34; Figure 11 and Figure 8, and equivalents thereof.”

f. “function processor instructions” (’945 Patent: 1, 12, 14; ’683 Patent: 1)

Plaintiffs’ Proposed Construction	Defendants’ Proposed Construction
Instructions arranged to provide directions for operation of a function processor.	A set of instructions that control operation of the claimed communications device, written and loaded onto the communications device in the hardware/operating system-independent language of the virtual function processor.

The parties’ proposed constructions for the claim term “function processor instructions” differ in two material respects. First, Defendants’ proposed construction requires that the “function processor instructions” control the operation of the claimed communications, and second, Defendants’ proposed construction requires that the “function processor instructions” be written in the hardware/operating system-independent language. As to the first point, CardSoft does not dispute that the “function processor instructions” control the operation of the claimed communications device. Indeed, the claims expressly recite “function processor instructions for controlling operation of the device,” and the specification explains that the “function processor instructions” “control[] operation of the device.” ’945 Patent at 3:60-61; 7:26-27; 7:47. As such, the court agrees with Defendants that the “function processor instructions” is a set of instructions that control operation of the communications device.

With respect the parties’ second dispute, the court rejects Defendants’ contention that the “function processor instructions” must be written in a hardware/operating system-independent language. Defendants’ proposed limitation again runs contrary to the language of the claims. Specifically, Claim 8 of the ’945 Patent recites “wherein the function processor instruction

means are implemented in software defined by the function processor means and do not require translation to the native code of the microprocessor.” As discussed above, this claim creates a presumption that because Claim 8 limits the function processor instruction means to implementation in software defined by the function processor, Claim 1 is not so limited and is broad enough to cover both function processor instructions implemented in software defined by the function processor and function processor instructions not implemented in software defined by the function processor. Furthermore, Defendants’ reliance on statements in the specification indicating that the function processor instructions “preferably” never require translation to any real hardware processor do not overcome this presumption. ’945 Patent at 5:19-25. These statements merely describe an embodiment of the invention claimed by the patents-in-suit and such embodiments cannot be read into the claims.

In conclusion, the court construes “function processor instructions” to mean “a set of instructions that control operation of the communications device.”

V. CONCLUSION

The court adopts the constructions set forth in this opinion for the disputed terms of the patents-in-suit. The parties are ordered that they may not refer, directly or indirectly, to each other’s claim construction positions in the presence of the jury. Likewise, the parties are ordered to refrain from mentioning any portion of this opinion, other than the actual definitions adopted by the court, in the presence of the jury. Any reference to claim construction proceedings is limited to informing the jury of the definitions adopted by the court.

It is so ORDERED.

SIGNED this 23rd day of September, 2011.



CHARLES EVERINGHAM IV
UNITED STATES MAGISTRATE JUDGE