

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC. and AMAZON WEB SERVICES, LLC,
Petitioner,

v.

PERSONALIZED MEDIA COMMUNICATIONS, LLC,
Patent Owner.

Case IPR2014-01528
Patent 7,783,252 B1

Before KARL D. EASTHOM, TRENTON A. WARD, and
GEORGIANNA W. BRADEN, *Administrative Patent Judges*.

EASTHOM, *Administrative Patent Judge*.

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

Petitioner filed a Petition requesting an *inter partes* review of claims 1–3, 5, 9–14, 18–23, 27–32, 36–41, 45, 46, and 50–52 of U.S. Patent No. 7,783,252 B1 (Ex. 1001, “the ’252 patent”). Paper 1 (“Pet.”). Patent Owner filed a Preliminary Response. Paper 6 (“Prelim. Resp.”). The panel instituted an *inter partes* review on all the challenged claims. Paper 7, 2 (“Dec. on Inst.” or “Institution Decision”).

Patent Owner filed a Response (Paper 22, “PO Resp.”) and a contingent Motion to Amend Claims (Paper 23, “Mot. to Amend”). In response, Petitioner filed a Reply (Paper 32, “Pet. Reply”) and an Opposition to the Motion to Amend (Paper 33, “Pet. Opp.”). Patent Owner filed a Reply to Petitioner’s Opposition. Paper 39 (“PO Reply”). The parties filed other papers, each considered but discussed below only where material. The parties presented arguments at an oral hearing before the panel, which was transcribed by a court reporter. See Paper 30 (“Tr.”).

In this Final Written Decision, issued pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73, we determine that Petitioner has shown by a preponderance of the evidence that challenged claims 1–3, 5, 9–14, 18–23, 27–32, 36–41, 45, 46, and 50–52 are unpatentable. We also determine Patent Owner has not met its burden regarding its proposed substitute claims, and therefore, deny its Motion to Amend.

A. Related Proceedings

According to the Petition, the ’252 patent is involved in *Personalized Media Communications, LLC v. Amazon.com, Inc.*, No. 1:13-cv-1608-RGA (D. Del., filed Sept. 23, 2014). Pet. 1. Granting a motion for judgment on the pleadings, the U.S. District Court for the District of Delaware found

claim 13 of the '243 patent invalid as not directed to patentable subject matter. *See* Ex. 1040, 6–8; Pet. Reply 4–5, Paper 35, 1. Petitioner states that Patent Owner appealed that judgment in the Court of Appeals for the Federal Circuit as Appeal No. 15-2008. Paper 35, 1.

Petitioner filed petitions seeking *inter partes* review of related U.S. Patent Nos. 5,887,243 B1 (IPR2014-001527); 7,864,956 B1 (IPR2014-01530); 8,046,791 B1 (IPR2014-01531); 7,801,304 B1 (IPR2014-01532); 7,805,749 B1 (IPR2014-01533); and 7,827,587 B1 (IPR2014-01534).

B. The '252 Patent

The '252 patent discloses a method of reprogramming a receiver station by revising an operating system of a programmable device in the receiver system. Ex. 1001, 265:20–273:25. Examples of an operating system include “PC-DOS” or “MS-DOS” for an IBM personal computer (“IBM PC”). *Id.* at 265:29–35. Receiver stations may include different types of programmable devices, for example, microcomputer 205, controllers 12 and 20 of signal processor 200, the RAMs associated with the processors, 39B and 39D, and decoders 203 and 282. *Id.* at 265:62–266:4.

C. Challenged Independent Claims

Challenged claims 1, 10, 19, 28, 37, and 45 are independent.

Independent claim 1 is representative and follows:

1. A method of reprogramming a receiver station, said receiver station including a programmable device of a specific version having a memory, a signal detector, and a receiver operatively connected to said signal detector, said method comprising the steps of:

- [a] storing information specifying said specific version of said programmable device, wherein said specific version indicates a version of an operating system executing on said

programmable device and controlling the processing capabilities of said programmable device;

[b] receiving an information transmission at said receiver, said information transmission including a control signal which designates a designated version of programmable device;

[c] passing said information transmission to said signal detector and detecting said control signal;

[d] determining whether said specific version is said designated version in response to said control signal;

[e] communicating operating system instructions to said memory only when said step of determining determines that said specific version is said designated version, wherein said communicating comprises erasing any operating system instructions stored within an erasable portion of said memory and then storing said communicated operating system instructions within said erasable portion of said memory; and

[f] executing said communicated operating system instructions to control operation of said programmable device.

Ex. 1001, 286:50–287:10 (bracketed information added).

D. Asserted Prior Art References

Tamaru, U.S. 4,788,637 (issued Nov. 29, 1988) (Ex. 1006).

Schmidt et al., U.S. 4,558,413 (issued Dec. 10, 1985) (Ex. 1007, “Schmidt”).

Daniel Nachbar, *When Network File Systems Aren’t Enough: Automatic Software Distribution Revisited*, USENIX ASSOC. SUMMER CONF. PROCEEDINGS 159–171, © 1986 by The USENIX Association (Ex. 1005).

E. Instituted Claims—35 U.S.C. § 103(a)

Claims	References
1–3, 5, 9–14, 18–23, 27–32, 36–41, 45–46, and 50–52	Nachbar and Schmidt
1–3, 5, 9–14, 18–23, 27–32, 36–39, 41, 45–46, and 50–52	Tamaru, Nachbar, and Schmidt

F. Declarants

Petitioner relies on the Declaration of Michael Lesk, Ex. 1008, and the Rebuttal Declaration of Michael E. Lesk, Ex. 1035. Patent Owner relies on a Declaration of Samuel H. Russ, Ph.D., Ex. 2009, and a Declaration of Gerald Holtzman, Esq., Ex. 2007.

Petitioner alleges that Dr. Russ's testimony should be entitled to no weight. Pet. Reply 1–2. Petitioner contends that Dr. Russ contradicts himself, could not understand several questions in his deposition, lacked sufficient understanding for some terms, and admits that he was not one of skill in the art in 1987. *See id.* at 1–2 (citing Ex. 1033). Petitioner also contends that Dr. Russ “was unable to explain what was known in 1987.” *Id.* at 1 (citing Ex. 1037, 73–117). It appears from the cited testimony that Dr. Russ for the most part attempted to narrow or clarify his understanding of certain questions. *See, e.g.*, Ex. 1035, 63–65 (discussing software update); 193 (discussing software). Regarding Petitioner's allegation that Dr. Russ was unable to testify about the state of the art in 1987, most of the questioning in the cited Exhibit concerns 1981. *See, e.g.*, Ex. 1037, 81, 83, 177, 178. Whether Dr. Russ was one of skill in the art in 1987 is not material.

On balance, Petitioner does not explain persuasively how the cited testimony goes to a material factual issue in the case apart from the credibility of Dr. Russ. After consideration of the arguments in light of the evidence and the context, we determine that Dr. Russ's testimony is entitled to due consideration as seen below. We have weighed the Declarants' testimony based on the merits of the positions taken, including taking into

account deposition testimony and other arguments and record evidence, including Motions for Observation and Responses thereto.

II. ANALYSIS

A. Claim Construction

The claims of an unexpired patent are interpreted using the broadest reasonable interpretation (“BRI”) in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b); *see also In re Cuozzo Speed Techs., LLC*, 778 F.3d 1271, 1279–83 (Fed. Cir. 2015) (“Congress implicitly adopted the broadest reasonable interpretation standard in enacting the AIA,” and “the standard was properly adopted by PTO regulation.”), *cert. granted, Cuozzo Speed Techs. LLC v. Lee*, 136 S. Ct. 890 (mem.) (2016). Under “the best practices for claim construction,” a claim term generally carries its “‘ordinary and customary meaning’”—“‘the meaning that the term would have to a person of ordinary skill in the art in question. . . . in view of the specification.’” *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007) (quoting *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc)). The outcome in this case would not be altered under a BRI or *Phillips* claim construction standard.

Claim 1 recites in step [a] “storing information specifying said specific version of said programmable device, wherein said specific version indicates a version of an operating system executing on said programmable device.” The other challenged independent claims recite a similar “storing information” limitation. Claim 1 and the other challenged independent claims also recite a determining step (step d in claim 1) or a similar step, which essentially require using the stored information to determine whether

to reprogram the specified programmable device. The “storing” and “determining” limitations are the focus of the discussion below.

1. Initial Institution Decision on Claim Construction

We initially determined that “‘said specific version of said programmable device’ includes a variation of a programmable device indicated by the operating system it employs.” Dec. on Inst. 7. We reasoned that stored information about the operating system (“OS”) executing on a specific device specifies said specific version of said programmable device.

Part of our initial reasoning follows:

As the quoted passages indicate, a “version” may indicate a type or latest upgrade of an operating system. Claim 1 specifies “storing information specifying said specific version of said programmable device, wherein said specific version indicates a version of an operating system executing on said programmable device.” According to this claim language, specifying a “version of said hardware device” is broad enough to encompass identifying a version of an operating system used by the programmable device. This construction comports with the example from the Specification cited by Patent Owner: a specific PC-DOS operating system version indicates that an IBM PC uses that version. *See* Ex. 1001, 265:51–55. Stated differently, a generic computer device becomes a specific computer device defined by the specific operating system it employs—i.e., an algorithm defines the structure of the device.

Dec. on Inst. 6–7 (referring to “quoted passages” Ex. 1001, 265:51–55, 286:50–287:10). Petitioner agrees with our initial claim construction. Pet. Reply 5–8. A district court similarly reasoned that claim 1 “involves . . . checking a receiver station to see if it has the current operating instructions, and, if it does not, updating them.” Ex. 1040, 8.

2. Patent Owner's Motion to Amend Shows Agreement with the Panel's Claim Construction

Although Patent Owner's Response argues that "there is no support in the intrinsic record" for our initial claim construction that allows stored OS information to specify said programmable device, PO Resp. 17, Patent Owner's Motion to Amend contradicts Patent Owner's Response, as follows:

The '908 Application discloses that the receiver station stores "information specifying said specific version of said programmable device, wherein said specific version indicates a version of an operating system" on the programmable device. Ex. 2010 at 517:5–16 ("[] many versions usually exist of any given operating system which versions have greater or lesser capacities []"). It was also well known and inherently disclosed in the '908 Application that the programming or re-programming of a programmable device with "operating system instructions" would cause information indicative of the version of the operating system to be recorded in the device. The '908 Application describes the functions of the operating system as "executing on said programmable device and controlling the processing capabilities of said programmable device." Ex. 2010 at 516:15–28 ("So-called 'operating systems' are well known in the art and generally comprise the most basic form of processor control instructions . . . a computer is usually preprogrammed with an operating system that controls such fundamental aspects as, for example, so-called 'input/output' functions").

Mot. to Amend 4 (citing Ex. 2010, 517:5–16, 516:15–28) (emphases added).¹

¹ According to Patent Owner, the '908 Application (App. No. 80/446,908), Ex. 2010, "led to the '252 [p]atent." Petitioner does not challenge the statement.

The passage shows that Patent Owner agrees that the disputed phrase requires, and the '908 application supports, storing OS information (which PCs inherently store according to Patent Owner), thereby “specifying said specific version of said programmable device.” *See also* Mot. to Amend 4, 4–7 (distinguishing the disputed clause at issue here from the newly added clause in its amendment: “a specific apparatus version of said programmable device”), A1 (proposed substitute claim 53 for claim 1).

3. Patent Owner’s Proposed Construction Does Not Support an Exemplary Disclosed Embodiment and Does Not Show Lack of Support for the Panel’s Construction

Patent Owner proposes that

the broadest reasonable construction of the claim term “specific version of said programmable device, wherein said specific version indicates a version of an operating system executing on said programmable device” that is consistent with the claims, specification and prosecution history is “a variation of an apparatus version of a programmable device wherein the apparatus version of the programmable device is also used to indicate a version of an operating system executing on the programmable device.”

PO Resp. 18 (citing Ex. 2009 ¶ 70).

Petitioner replies that “the claims do not say that the programmable device version and the operating system (“OS”) version are separate pieces of information. Rather, the claims recite an OS version in order to define ‘a version of a programmable device.’” Pet. Reply 5 (citing Ex. 1035 ¶ 12).

Patent Owner also argues that “[i]t is . . . significant that the claim recites that there is information ‘*specifying*’ the version of programmable device which in turn ‘*indicates*’ a version of an operating system.” PO Resp. 14. As Petitioner notes, however, claim 1 also recites “designated version” and “specific version” and equates the two, signifying that different

modifiers of “version” do not necessarily signify different things. *See* Pet. Reply 6.

The record supports Petitioner’s arguments and the panel’s initial claim construction. Patent Owner’s construction does not address the critical question of what “information” the system stores to specify “said specific version of said programmable device” to indicate an OS as recited in the disputed clause of claim 1. Patent Owner’s construction seems to require storing mere hardware information: “a variation of an apparatus version.” Alleging support for its construction, Patent Owner lists “IBM PC,” “Apple II,” “decoders 203,” and other “different types of programmable devices, for example microcomputer 205, controllers 12 and 20 of signal processor 200, the RAMs associated with the processors,” and other devices as hardware. *See* PO Resp. 6 (citing Ex. 1001, 265:62–266:4; 68:44–55). Notwithstanding the list of exemplary devices, Patent Owner does not explain, and the record does not show, how a generic processor or decoder using RAM (random access memory) for storage, for example, indicates a specific OS version executing thereon.²

² Furthermore, in its Motion to Amend, as explained more fully below, *see* Section II.C.3, Patent Owner does not argue that a specific apparatus indicates an OS version. Patent Owner lists Apple II, decoders 203, and an IBM PC, as exemplifying support for an “apparatus,” as recited in its additional proposed substitute claim 53 phrase (i.e., which claim 1 does not recite): “information specifying a specific apparatus version of said programmable device.” In other words, newly proposed claim 53 delineates an apparatus version from the OS version at issue here. *See* Mot. to Amend 4–6 (listing Apple II, IBM PC, URS decoders 203, and other devices), A1 (reciting the new apparatus phrase in substitute claim 53).

Even though a generic programmable device and a less generic IBM PC each fail to indicate an OS version executing on that device, Patent Owner contends that the original claim language of challenged claim 1 focuses on specifying a hardware “apparatus,” and that hardware “apparatus” information indicates a version of an OS on that hardware apparatus/device:

A person of ordinary skill in the art would understand that information about the particular version of programmable apparatus (*e.g.*, IBM PC, Apple II) can indeed be indicative of the version of operating system that will execute on the device. (The IBM PC’s operating system would not execute on an Apple II, and vice versa.) Whereas, the converse is not true, as Petitioner[]’[s] own expert acknowledges. One cannot identify the apparatus version of the programmable device of the receiver station simply because one knows that the receiver station is executing a particular version of operating system.

PO Resp. 15 (citing Ex. 2006, 27:15–29:24) (Lesk Deposition); Ex. 2009 ¶ 28 (Russ Declaration)).

The argument that an IBM PC “can indeed be indicative” of its OS is not clear. In another passage, Patent Owner quotes the following as alleged support for the notion that the claimed stored “information specifying said specific version of said programmable device . . . [that indicates its OS]” constitutes mere hardware information:

Automatically, controller, 12, loads at its SPAM-input signal memory the command information of said message and any padding bits immediately following said command information, selects the meter-monitor ***information that identifies a specific preprogrammable apparatus version***—that is, an IBM PC—and inputs to controller, 20, said operating instructions-received-for-specific-apparatus instruction together with said ***information that identifies an apparatus version***.

PO Resp. 8 (quoting Ex. 1001, 268:59–269:3 (emphasis by Patent Owner); citing Ex. 2009 ¶ 45); *see also* Ex. 1001, 269:3–13 (information “identifies specific preprogrammable apparatus of the station . . . in other words, to determine that an IBM PC is the microcomputer 205, of the station”).

Not only does the passage and argument fail to discuss generic programmable PCs or other processor devices, the cited passage refers to “information that identifies a specific preprogrammable apparatus version—that is, *an* IBM PC.” Ex. 1001, 268:65–67 (emphasis added). The term “*an* IBM PC” does not necessarily mean “*any* IBM PC.” The passages do not preclude specifying “*an* IBM PC” by its specific OS. As a specific example described in the ’252 patent Specification, the stored OS information includes “[f]or example, version[] 1.00 . . . of . . . MS DOS,” which execute on “an IBM PC.” Ex. 1001, 265:53–54 (“Each version [of MS-DOS and PC-DOS] has capacity for controlling the operation of an IBM PC.”). Therefore, the ’252 patent Specification indicates that “an IBM PC” that executes MS DOS 1.00 constitutes a specific programmable device: i.e., an IBM PC that executes MS DOS 1.00.

In addition, the passage quoted by Patent Owner identifies sending two items of information, i.e., OS and apparatus information: “operating instructions-received-for-specific-apparatus instruction together with said information that identifies an apparatus version.” Ex. 1001, 269:1–3. The next sentence in the cited passage (not quoted in the Patent Owner Response) states that “[r]eceiving said *instruction and information* causes controller, 20, to determine, in a predetermined fashion,” if the information matches stored EPROM information, “to determine that an IBM PC is the microcomputer 205, of said station.” Ex. 1001, 269:3–12 (emphasis added).

The quoted passages show that the disclosed transmitting portion of the system somehow predicts each specific OS in “*an IBM PC.*”

Reprogramming or updating occurs when the receiver verifies the prediction to determine that its information indicates it is “an IBM PC” specified in the control signal. *See id.* at 269:3–31. Furthermore, the ’252 patent Specification notes that most PCs, including an IBM PC, are “usually preprogrammed with an operating system that controls . . . fundamental aspects” of that specific PC. *See Ex. 1001*, 265:28–29. As indicated above, this analysis comports with the district court’s rationale that claim 1 “involves . . . checking a receiver station to see if it has the current operating instructions, and, if it does not, updating them.” *Ex. 1040*, 8.

Although not entirely clear, Patent Owner may be arguing that the phrase “indicates *a version* of an operating system executing on said programmable device” means a whole class of OS versions, for example, MS-DOS or PC-DOS. This argument ignores the otherwise argued support for more generic programmable devices embraced by the claimed programmable device, including decoders and processors with one or more RAMs, for example, which fail generally to indicate a class of OS versions. The argument also fails to appreciate other considerations, including the surrounding claim language and the ’252 patent Specification, as indicated above and as discussed further below.

Our construction requires an indication of the OS to specify the hardware executing that OS. *See Dec. on Inst. 6–7*. The ’252 patent Specification supports our interpretation by referring to “an IBM PC,” and also, describing generic programmable devices with specific software. The ’252 patent Specification also notes that “a computer is usually

preprogrammed with an operating system that controls . . . fundamental aspects One such system that is commonly known as ‘PC-DOS’ or ‘MS-DOS’ is an operating system of the IBM personal computer, commonly known as the ‘IBM PC.’” Ex. 1001, 265:28–33 (emphasis added). “For example, *versions* 1.00, 1.10, 2.00, etc. exist of PC-DOS and MS-DOS. Each *version* has capacity for controlling the operation of an IBM PC. . . . [L]ater [OS] versions generally have expanded capacities in comparison to earlier versions.” See Ex. 1001, 265:51–55 (emphasis added). These passages show an OS “version” is more specific than a generic class of OS versions, that PC hardware includes its OS version, and that the inventors deemed OS versions 1.00, 1.10, etc., of PC-DOS and MS-DOS to be specific to executing on an IBM PC.

Patent Owner also argues that the invention “allows different types of devices ***with different apparatus versions*** to be updated remotely where desired.” PO Resp. 11 (citing Ex. 2009 ¶ 50). The challenged claims do not recite “different apparatus versions.” More importantly, however, Patent Owner’s construction does not support providing an *update* for the exemplary IBM PC embodiments that include different OS versions. On the other hand, our construction provides for updates, because it requires knowing the specific “programmable device” version by its OS, as required information necessary to update the existing OS version.

In other words, tracking our construction, the ’252 patent Specification touts updating older OS versions stored on RAM. See Ex. 1001, 265:45–57 (describing “[o]ne advantage” of storing an OS in RAM—they “can be conveniently overwritten,” thereby “expanding system functions”). Similarly, “[a]nother objective of the present invention is

flexibility and convenience in *reprogramming* operating systems in order to *expand* system functions.” Ex. 1001, 266:24–26 (emphases added). In context, “*expand*” by “*reprogramming*” means to update an existing OS (or software) with a later version: “For example, *versions* 1.00, 1.10, 2.00, etc. exist of PC-DOS and MS-DOS. . . . [L]ater [OS] versions generally have expanded capacities in comparison to earlier versions.” *Id.* at 265:53–56. (In addition to updating, another “objective of the unified system of *programming* information of the present invention is *standardization* of receiver station operating systems.” *Id.* at 266:15–17.)

Claim 1 calls for “[a] method of reprogramming” by storing “information . . . [that] indicates a version of an operating system executing on said programmable device,” and then “erasing any operating instructions stored.” Reprogramming further implies that the claimed method requires current OS information, so the programmable device can be reprogrammed with a new OS.³ Otherwise, under Patent Owner’s construction, devices could be “reprogrammed” with the same program they currently have—an unreasonably broad and stilted construction of reprogramming (and

³ Claim 1 recites a method that includes “storing information specifying said specific version of said programmable device, wherein said specific version indicates a version of an operating system *executing on said programmable device and controlling the processing capabilities*.” (Emphasis supplied). It does not call for “a version of an operating system [*that can execute or is executable*] on said programmable device” or “*for* controlling the processing capabilities.” Patent Owner’s arguments in support of its claim construction essentially replace the claimed terms “executing” and “controlling” respectively with “executable” and “for controlling.” See PO Resp. 15. Claim 1 does not refer to a class of versions that *will* execute on, or *are executable* on, the (PC) device. Therefore, this claim language weighs as an additional factor in support of the panel’s construction.

updating) in light of the '252 patent Specification. *See* Ex. 2001, 265:51–57 (specifying PC-DOS and MS-DOS versions 1.00, 1.10, etc. as expanding versions over earlier versions). Furthermore, sending out the latest known version of MS-DOS to all IBM PCs that may be executing PC-DOS would not necessarily constitute updating a PC-DOS, because an MS-DOS version would not necessarily constitute an update for a PC-DOS version.

Under Patent Owner's claim construction, which does not require storing information specifying a currently executing OS, the '252 patent Specification fails to provide the necessary written description support in claim 1 for disclosed embodiments that *update* a hardware device (IBM PC) with the most recent OS version (e.g., MS-DOS 1.0, PC-DOS 1.0).⁴ Based on the foregoing discussion, the updating feature constitutes part and parcel of the IBM PC embodiments that describe updating specific OS versions, as opposed to just being a central objective of the invention.

Furthermore, Patent Owner's proposed construction also fails to provide support for merely specifying the hardware of generic devices such as generic decoders and processors (with RAM or other memory), because

⁴ During the oral hearing, Patent Owner verified its claim construction meant that the methods and systems would not guarantee providing updates; rather, they may download *earlier* OS versions and overwrite *newer* versions. *See* Tr. 69:9–70:19; 75:24–76:16. Patent Owner similarly argued that the claimed method only compares hardware: “Well, those receivers aren’t making any determination other than that they are the specific type of hardware that is being designated in the control signal, then the central station is sending whatever operating system update is required for the specific types of machines.” Tr. 76:11–15. (This latter statement fails to explain how the central station knows what specific OS will update a generic IBM PC.)

absent some placeholder name for such a device tied to specific OS versions, those generic programmable hardware devices, by themselves, do not indicate a specified class of possible OS versions, let alone an OS version executing on the generic programmable device. See Ex. 1035 ¶ 13 (“[I]n many cases a version of hardware will indicate very little about what version of operating system will execute on the hardware”); Pet. Reply 6–7. As a general rule, “there is a strong presumption against a claim construction that excludes a disclosed embodiment.” *In re Katz Interactive Call Processing Patent Litig.* 639 F.3d 1303, 1324 (Fed. Cir. 2011).⁵

Our construction provides for updating (and standardization) for programmable devices, and thereby, hews most closely to written descriptive support for the generic hardware embodiments and embodiments that employ the updating feature.

4. Placeholder Name as Specifying Hardware

In addition to specifying the OS itself as explained above, the ’252 patent Specification and evidence also indicate that specifying the programmable device, if a “hardware” name must be stored, may include

⁵ The *Katz*-type presumption, the claim language, and the Specification all operate against any presumption for reading a version as one or more OS versions (e.g., MS-DOS), and show the requisite intent to point to a single OS version (e.g., Apple II OS, MS-DOS 1.0). Cf. *TiVo, Inc. v. EchoStar Commc’ns Corp.*, 516 F.3d 1290, 1303 (Fed.Cir.2008) (“As a general rule, the words ‘a’ or ‘an’ in a patent claim carry the meaning of ‘one or more.’”) In addition *TiVo*’s general rule is not clearly on point, at least in part, because the term “version of an operating system” itself implies one version—i.e., the “version of an operating system executing on said programmable device and controlling the processing capabilities of said programmable device,” as claim 1 requires.

providing a placeholder name (e.g., Apple II, URS decoder example #1) that indicates to the system (in the determining step) the specific OS version executing on that device.

The following table, Table 1, produced by the Board, summarizes various embodiments of the '252 patent Specification for discussion purposes:

Hardware	OS Versions			
IBM PC	PC-DOS	1.00	1.10	2.00
IBM PC	MS-DOS	1.00	1.20	2.00
Apple II	Apple II			
URS Dec. Ex. #1			X	
URS Dec. Ex. #3				Y
URS Decoder	Not specified			
Program. device	Not specified			

Table 1 above, shows certain “Hardware” associated with a different “OS” version executing on that “Hardware.” As explained above, the IBM PC includes OS versions 1.00, 1.10, etc., and Apple II includes an Apple II OS. *See* Ex. 1001, 265:53–57 (IBM PC), 268:12–17 (Apple II). The '252 patent Specification does not specify a name for the software or OS for the URS decoders. *See* Ex. 1001, 266:40–46 (discussed further below). To facilitate discussion, they have been named simply as OS versions X and Y. The claimed and disclosed programmable devices, including, for example,

the URS decoder, or other generic processors, do not identify a specific OS version.

As Table I summarizes, and as explained above, the '252 patent Specification implies that a specific PC-DOS or MS-DOS version specifies an IBM PC, but not vice versa—an IBM PC may indicate any number of “‘PC-DOS’ or ‘MS-DOS’ [OS versions] . . . of the . . . ‘IBM PC’” (Ex. 1001, 265:28–33), “[f]or example, versions 1.00, 1.10, 2.00, etc. . . . of PC-DOS and MS-DOS” (*id.* at 265:51–55). An Apple II OS specifies an Apple II, and vice versa. *See id.* at 268:12–17, 48–50. As noted, the '252 patent Specification describes an IBM PC and typical PCs as being “preprogrammed with an operating system that controls [its] fundamental aspects, as for example, its so-called input/output functions.” *See id.* at 265:28–30. In other words, a specific PC (Apple II, IBM PC, programmable device, etc.) includes its “preprogrammed” OS.

Turning to the URS decoder, an example relied upon by Patent Owner, the Specification describes the information that identifies preprogrammable decoders 203 as follows: “[I]nformation that identifies *not only specific preprogrammable apparatus* such as URS decoders, 203, *but also the particular version of said apparatus* (for example URS decoders, 203, of the version illustrated above in example #1 rather than example #3).” Ex. 1001, 266:43–46 (emphases added); *see* PO Resp. 6 (citing Ex. 1001, 265:62–266:4).

Similar to the discussion of the IBM PC above, this passage does not indicate what hardware and/or OS “information” identifies a “specific *preprogrammable apparatus*.” The passage implies that generic names

themselves (i.e., “URS decoders, 203,” “example #1, rather than example #3,” etc.) may “identif[y]” the “particular version of said apparatus.” *See id.*

No dispute exists over the fact that one of the main reasons, if not the main reason, for any identification in the disclosed invention, is to provide software program or OS updates. *See* Ex. 1001, 265:45–61, 266:24–26; PO Resp. 10–11, 16–17. The challenged claims broadly recite storing information specifying a version of “said programmable device,” wherein that version indicates an OS. The ’252 patent Specification fails to show how a generic programmable device indicates an OS version. This implies that the challenged claims refer to storing (and comparing, i.e., determining) either OS information and/or a placeholder name that specifies that OS version.

By way of example, the two different URS decoder examples cited (#1 and # 3) differ in hardware and in software. *See* Ex. 1001, 84:27 (“Example # 3 differs from example #1 in just two respects.”).⁶ It appears that the two examples differ with respect to an OS and hardware. *See id.*,

⁶ In example #3, a preferred embodiment, decoder 203 includes an integrated controller 39, whereas in example #1, decoder 203 employs controller 39 and SPAM controller 205C as separate units. *See* Ex. 1001, 84:42–48; 81:14–17 (in the preferred embodiment, example #3, “controller 39, of decoder, 203, and SPAM-controller, 205C, are one and the same (and are called hereinafter, controller 39)”). *See also id.* at 83:27–61 (decoder 203 preprogrammed in example #3), 69:43–45 (decoder 203 preprogrammed in example #1), 70:40–41 (CPU of controller 39, part of decoder 203), Figs. 2A, 3 (decoder 203 including controller 39). URS decoder 203 includes controller 39 with CPUs, processors, and RAM. *See* 69:69:45, 70:1–54, 81:24–65, Ex. 1001, 266:43–46, Figs. 2, 3A, 3. “‘Operating systems’ . . . generally comprise the most basic form of processor control systems.” Ex. 1001, 265:22–24.

note 6. (The outcome would not change even if all of the decoders use the same OS.) These different OS versions have been generally referenced as X or Y.

Similar to the IBM PC or Apple II examples, the decoder passages do not describe storing a specific list of hardware items for URS decoder example # 1 or example #3; rather, they simply describe a difference in hardware (and perhaps software X or Y, or perhaps both use X, etc.). See note 6. Controller 39, which is part of, or the same as decoder 203 is “pre-programmed at the RAM and/or ROM associated with control processor, 39J.” Ex. 1001, 83:30–32; see note 6. This sentence shows that a different hardware version, for example, one having only a ROM, as compared to another having a RAM and a ROM (or two, etc.), would not by itself indicate a specific or different OS version, because that OS (and other OS versions) could be stored on either hardware version and all manner of different “RAM and/or ROM” hardware for supporting the same or a different processor hardware.

Therefore, at most, the passages imply using a name such as “URS decoder, 203, example #3” as a placeholder name that indicates, by way of example, that OS version Y executes in the example # 3 programmable apparatus. Even if somehow the '252 patent system stores a list of hardware for each URS decoder 203 “example #,” the record does not show, and Patent Owner does not point to, a disclosure that shows that the system uses a specific hardware list for anything. Rather, the system somehow must know the OS to update it, as explained above.

5. Relationship between OS and Hardware

Patent Owner contends that an OS cannot specify a hardware device, because any given OS may run on different types of hardware devices.

PO Resp. 17–18. Patent Owner also contends that although the panel’s statement that

*“a generic computer device becomes a specific computer device defined by the operating system it employs” [is] correct, [it] is off-point. The specific operating system determines the specific computer device by reprogramming it, but the operating system does not **identify** the specific computer device. Knowing a computer was running the MS-DOS operating system does not tell us whether the programmable device is an IBM PC, a Commodore 64, or a Radio Shack Tandy.*

Id. at 17–18 (discussing Institution Decision) (first emphasis added).

As quoted above, Patent Owner’s argument admits that “the specific operating system *determines* the specific computer device by reprogramming it,” but Patent Owner advances a distinction between “determining” a device and “**identify**[ing] a device. PO Resp. 17–18 (first emphasis added). Claim 1 recites “specifying said specific version of said programmable device.” Patent Owner does not explain persuasively how determining the device materially differs from specifying it. Patent Owner also characterizes as “correct” the panel’s initial finding that “an algorithm defines the structure of the device,” thereby supporting the panel’s claim construction. *Id.* (quoting Dec. on Inst. 7).

Patent Owner’s arguments also support the determination that “storing information specifying said specific version of said programmable device” includes storing either the OS information, and/or a placeholder name tied to that OS information (e.g., via the column information in Table 1 *supra*). For example, specifying an MS-DOS 1.00 version determines the device as an

IBM PC, as opposed to an Apple II, according to the examples in '252 patent Specification as outlined above. *See supra* Table 1. This rationale parallels Patent Owner's rationale, which is limited to comparing two disclosed devices, an IBM PC and Apple II: "A person of ordinary skill in the art would understand that information about the particular version of programmable apparatus (*e.g.*, IBM PC, Apple II) can indeed be indicative of the version of operating system that will execute on the device. (*The IBM PC's operating system would not execute on an Apple II, and vice versa.*)" PO Resp. 15 (emphasis added). In other words, the '252 patent Specification describes an updating scenario that includes a limited and finite quantity of hardware and OS versions known to the system to be on a particular network to be updated. *See Ex. 1001, 265:36–57* (updating IBM PCs). This also suggests, in the alternative of identifying an OS, using a placeholder "hardware" name for each OS version to be updated. *See also supra* Table 1 (showing how different OS versions as described identify a certain "Hardware" versions, but not vice versa in all cases (*i.e.*, IBM PC)).

Patent Owner's argument that specifying MS-DOS may not necessarily specify an IBM PC, Commodore 64, or a Tandy, simply does not apply to the '252 patent Specification. *See* PO Resp. 18. As indicated, and as summarized above in Table 1, the '252 patent Specification discloses that only an IBM PC, as opposed to an Apple II, executes MS-DOS and PC-DOS. Again, this is because Patent Owner's system operates with a finite set of known device types operating on a given network, whereas Patent Owner's argument pertains to a hypothetical scenario in which all the device types are not known. To the extent the '252 patent contemplates other devices, for example, a network with a Tandy and an IBM PC, a placeholder

name could be “Tandy MS-DOS 1.00.” However, if both Tandys and IBM PCs use MS-DOS 1.0, the “Tandy” information would be superfluous.⁷

Patent Owner also points to the ’252 patent Specification and prosecution history as allegedly supporting Patent Owner’s construction. PO Resp. 16–17 (citing Ex. 1001, 265:62–269:31; Ex. 1002, 793–96). Patent Owner’s citation, without an explanation, to four pages of an Office Action Response filed during prosecution, fails to provide meaningful claim construction guidance. *Id.* at 17 (citing Ex. 1002 at 793–796). Regarding the ’252 patent Specification, Patent Owner characterizes an embodiment therein as follows: “If the specific version of the programmable device at the receiver station is not the same as the designated version of programmable device in the control signal, the information sent with the control signal is discarded.” *Id.* at 17 (citing Ex. 1001, 269:26–31). The embodiment cited—wherein an Apple II computer discards instructions intended for IBM PC—refers to determining if “information that identifies a specific preprogrammable apparatus version matches information that is preprogrammed at station specific EPROM 20B . . . in other words, to determine that an IBM PC is the microcomputer 205.” Ex. 1001, 269:7–12.

⁷ Stated differently, the argued example supports the determination that, at most, a disclosed placeholder name (e.g., “Z” for an IBM PC, a Commodore 64, and a Radio Shack Tandy) that identifies all three PC types, may be used to indicate that MS-DOS versions execute on semi-generic device Z. Therefore, with respect to the disclosed system and challenged claims, determining “whether the programmable device is an IBM PC, a Commodore 64, or a Radio Shack Tandy” would not matter where MS-DOS 1.00 runs on all three devices.

Similar to the examples discussed above, this example fails to describe what information specifies “*an IBM PC*” (emphasis added). This example also does not preclude storing information such as MS-DOS version 1.00, which specifies “*an IBM PC*,” or even using a name like IBM PC MS 1.00 or MS 1.00 as a stored placeholder name (which the receiver compares to a control signal having the same placeholder name). *See id.* at 268:18–269:31. In addition, Patent Owner’s cited example describes sending specific OS instructions, *id.* at 269:1–2, but the Specification does not describe clearly (and Patent Owner does not explain), how the transmitter knows to send that specific OS version to the designated IBM PC receiver based on a mere hardware name (i.e., unless it is some type of a placeholder name tied to an OS version as explained above). *See id.* 268:18–269:31.

Patent Owner also argues that Petitioner’s Declarant, Dr. Lesk, concedes that an OS does not specify a hardware version. PO Resp. 15 (citing Ex. 2006: 27:15–29:24; Ex. 2009 ¶¶ 58–59); *see* Ex. 2006, 29:1–25 (testifying basically that a skilled artisan could not tell if he or she logged onto DEC hardware or AT&T manufactured hardware “[j]ust from the fact” of knowing the OS is a UNIX System V). Citing a June 9, 2015 “Android Compatibility Definition” (Ex. 2008), Patent Owner also argues that

[i]n distributing a new version of Android, you need to know the apparatus version of the device, *e.g.*, a Samsung Note 4, a Galaxy S6, *etc.*, in order to download the new Android version. A simple designation of the Android operating system version would not be sufficient to tell you what type of device is executing that Android operating system version, *e.g.*, a Galaxy S5 phone or a Galaxy S3 phone.

PO Resp. 16.

These arguments do not address the scope of the challenged claims, or the disclosed IBM PCs, Apple II PCs, and other generic programmable devices of the '252 patent Specification, wherein a network operates with a finite set of known hardware devices (and which does not involve later-marketed Androids or other devices or OS's).⁸ At most, the arguments tend to show that the parties apparently agree, that in some scenarios, which may or may not be after the earliest effective filing date of the '252 patent Specification (1987), information about an OS may not specify hardware and vice versa. See Pet. Reply 6–7; Ex. 1035 ¶ 13; PO Resp. 16.

Petitioner similarly argues in general that “there is no definitive correspondence between hardware versions and specific OS versions.” Pet. Reply 6–7 (citing Ex. 1033, 78:24–79:11; Ex. 1035 ¶ 13). Citing the MS-DOS and PC-DOS versions of an IBM PC, Petitioner points out that “an apparatus version cannot indicate any particular OS version, and Russ admits this.” Pet. Reply 9 (citing Ex. 1033, 98:16–23; Ex. 1035 ¶ 16).⁹

Dr. Lesk persuasively supports Petitioner's contention that no clear correspondence generally existed at the time of the invention between an OS and hardware:

⁸ Testimony and argument about more recent Android systems do not inform of the knowledge of a skilled artisan at the time of the invention. See PO Resp. 15; Ex. 2009 ¶ 59 (“a common example *today* concerns the Android operating system”).

⁹ Petitioner adds that “[i]f there were a definitive correspondence, there would be no reason for the claims to recite an OS version at all,” and that the “wherein” clause would be superfluous. Pet. Reply 7. Claim 1 does not support the argument, because the wherein clause specifies that the programmable device must support an OS as opposed to any software.

First, in many cases a version of hardware will indicate very little about what version of operating system will execute on the hardware. For example, Apple computers can run many versions of Apple OS X, Microsoft Windows, Linux, and other operating systems. Even in 1987 the IBM PC could run PC-DOS, MS-DOS, and the newly-released OS/2. Second, and conversely, a version of an operating system may indicate a specific hardware version because that operating system may support only one version of hardware. For example, the initial release of Apple's iOS supported only Apple's original iPhone hardware, so that iOS version indicated what hardware version it was running on. *As these counter-examples show, there is no generally applicable correspondence one way or the other between hardware versions and operating system versions. Nor was there any such correspondence in 1980s.* For example, the PDP-11 series of minicomputers available in the 1980s supported multiple operating systems, and conversely the UNIX operating systems available in the 1980s supported multiple hardware architectures.

Ex. 1035 ¶ 13 (emphasis added).

The record shows a lack of a definitive correspondence between hardware and OS versions *in general*. The record evidence, however, does not contradict what the '252 patent Specification discloses: the finite network of known hardware devices and their corresponding OSs. *See, e.g., Table I supra.* As another example, during cross-examination, Dr. Russ's answers indicate that information that a particular apparatus is an "IBM PC" merely indicates that the device will support an OS capable of running on an IBM PC and will not support an Apple II operating system. *See Ex. 1033, 78:24–79:11.*

6. Summary

As explained above, the panel's construction agrees with Patent Owner's Motion to Amend, which adds an apparatus limitation and shows written descriptive support for the panel's construction concerning the OS,

and the district court's characterization of the claims as comparing OS versions. Mot. to Amend 4–7, Ex. 1040, 8; *see supra* Section II.A.1, 2. The record does not support Patent Owner's construction, because merely specifying hardware does not include support for merely specifying hardware for generic processors or a central embodiment of updating an IBM PC with the latest MS-DOS or PC-DOS version.

Under our construction, “storing information specifying said . . . programmable device . . . wherein said specific version indicates a version of an OS executing on said programmable device” means “storing information specifying said . . . programmable device . . . [by] . . . indicat[ing] a version of an OS executing on said programmable device.” This comports with our initial construction that “‘said specific version of said programmable device’ includes a variation of a programmable device indicated by the operating system it employs.” Dec. on Inst. 7.

In other words, the stored information identifies the currently executing OS system on a specific version of a programmable device, which specifies the programmable device. The stored OS information may be stored as a placeholder name (e.g., IBM MS DOS 1.00, or decoder example #3) or stored as OS information (e.g., MS DOS 1.00) tied to programmable device IBM PC in some fashion, including by virtue of a network having a known and finite number of hardware types each operating a given OS.

B. Obviousness

1. Nachbar's Teachings

Nachbar discloses a data and software updating system for “machines running the UNIX operating system” called “track.” Ex. 1005, 159, Abstract. In track, subscriber machines make copies of updated files from a

librarian machine. *Id.* at 161. Using the librarian as a metaphor for its human counterpart, the

electronic *librarian* has two tasks: 1) to provide a listing of its files that are available to *subscribers*, as well as some information about the currentness of the each file. Track uses the time of last modification as its measure of currentness. . . . And 2) to give out *copies* of files to authorized *subscribers* upon request. Given such a cooperative *librarian*, a *subscriber* machine need only inspect the *librarian's* listing of available files and request new *copies* of files that are more current than its own.

Id.; Dec. on Inst. 10 (quoting same).

In addition to updating data and files on different UNIX based VAX machines, Nachbar states that “[t]rack can be used for distributing bug fixes as well as new releases of software packages or even UNIX itself.” Ex. 1005, 163 (VAX machines), 166 (emphasis deleted).

2. Schmidt’s Teachings

Schmidt discloses a system for “automatically collecting and recompiling updated versions of component software objects comprising a software program for operation on a plurality of personal computers coupled together in a distributed software environment via a local area network.” Ex. 1007, 9:28–33. Schmidt discloses a modeler that updates its internal representation of the model to reflect the new version “if the object being edited is in the model.” *Id.* at 45:11–13. For example, in a “Notice Operation,” “[t]he model[]er searches its internal data structure for a reference to an earlier version of the file. If one is found, the model[]er changes the internal data structure to refer to the new version.” *Id.* at 45:57–62.

3. Combining Nachbar and Schmidt

Relying on its declarant, Dr. Lesk (Ex. 1008), Petitioner asserts that the combination of Nachbar and Schmidt renders obvious claims 1–3, 5, 9–14, 18–23, 27–32, 36–41, 45, 46, and 50–52. Pet. 17–42. Patent Owner initially directs attention to independent claims 1, 10, 28, 37, and 45, but focuses on claim 1. See PO Resp. 29–41. For purposes of discussion in this section, claims 1, 3, 10, 12, 19, 21, 28, 30, 37, and 39 materially have similar scope and are treated as a group represented by claim 1. See PO Resp. 34–41; Pet. Reply 9–19 (responding to the arguments as grouped by Patent Owner); Pet 26 (grouping features of claims 1, 3, 12, 19, 21, 30, 39, and 50).¹⁰

Petitioner generally asserts that Nachbar teaches or suggests updating the UNIX OS on different subscriber machines using a program called “*track*.” Pet. 17–19; Ex. 1008 ¶ 45. Petitioner asserts that such reprogramming of a subscriber machine constitutes reprogramming a receiver station as set forth in the preamble of claim 1. See Pet. 19, n.3 (citing Ex. 1008 ¶ 45 n.1).

Petitioner alleges that Nachbar discloses most of the claim steps of claim 1. See Pet. 22–26. According to Petitioner, “[t]he only difference

¹⁰ Claim 12 recites two receiver stations and claim 28 recites “at least one of a plurality of receiver stations,” but the claims otherwise are similar to claim 1 in material respects based on this record. To the extent more than one receiver station presents a separate issue, the analysis of claims 14, 22, 31, 32, 40, 45, 50, and 51 below addresses materially similar limitations. In any event, a further review of the Petition in light of the record evidence and arguments shows that it persuasively addresses all the challenged claim limitations.

between claim 1 of the '252 patent and the process described in Nachbar is that Nachbar discloses communicating operating system instructions if the designated version is newer than the specific version, whereas the claim requires that the versions are equal.” Pet. 20 (citing Ex. 1008 ¶ 57).

Petitioner and Dr. Lesk explain that Nachbar’s system updates an older version on a subscriber machine after comparing relative ages of the versions, whereas Schmidt teaches updating a version if the currently executing version matches a designated version. Pet. 21 (citing Ex. 1007, 43:62–44:4; Ex. 1008 ¶¶ 58–59; 1005, 162).

Further addressing OS updates, the Petition asserts that Nachbar discloses that the files to be updated may[]be operation system instructions. . . . Nachbar further discloses that only files that are compatible with the architecture will be copied. *See id.* at 163 (“For instance, almost everything in /etc can be shared between machines of the same architecture. However, some files, such as ‘/etc/fstab’ and ‘/etc/ttys’ usually have slight differences on all machines. . . . Given this design, *subscription list* entries define pruned subtrees of the filesystem.”)

Pet. 24 (quoting Ex. 1005 at 163).

The record supports Petitioner’s contention that Nachbar discloses updating OS versions on specific machines. For example, Nachbar discloses “booting a new UNIX kernel” and “the UNIX operating system” (*id.* at Abstract), and describes a “plan to port track to other flavors of UNIX (i.e., System V and Research Version 8)” (*id.* at 166). Nachbar also discloses that track distributes these new kernels to a machine named “gorp” and to “the other VAX machines.” *Id.* at 164.

Petitioner maintains that, even if the challenged claims require storing and comparing hardware information, Nachbar discloses that feature or renders it obvious. *See* Pet. 18–19, 22, 24, 35–36 (obvious to accommodate

OS and machines of different architecture), 38 (similar). Petitioner explains that in Nachbar, comparing “files on the subscriber machine with the entries from the librarian’s *statfile*, the specific hardware as well as the software version of the subscriber device . . . is necessarily known for selection of software from the librarian machine.” *Id.* (emphasis added) (citing Ex. 1008 ¶ 60). Petitioner contends that accommodating machines of different architecture is necessary to ensure OSs are compatible and would have been obvious for the purpose of providing OS updates. *See id.* at 22, 38.

The record supports Petitioner’s position. As explained more fully below, in response to the *statfile* as a control signal in *track*, Nachbar’s subscriber compares files containing a specific OS kernel (e.g., GORP/vmunix) listed in its stored *subscription list* to that same path in the *statfile*, wherein the GORP file path name relates only to a specific machine named gorp. *See* Ex. 1005, 162, 165–166. Nachbar also generally discloses “the great ease with which one can maintain software *on machines of different architecture.*” *Id.* at 164 (emphasis added).

Relevant to the comparison, Petitioner contends that Nachbar discloses that the subscriber machine stores information in a “*subscription list*,” which “describes the set of files to be considered for potential update from the *librarian.*” Pet. 23 (quoting Ex. 1005, 161) (emphases added to conform to original). Petitioner notes that an embodiment in Nachbar involves up to 20 computers all using Berkeley 4.2 UNIX connected via the Ethernet. *Id.* (citing Ex. 105, 165). Petitioner contends that “[t]he stored information specifies the specific version of the operating system,” including the currentness of the file represented by the time of last modification. Pet. 23.

Claim 1 also recites “receiving an information transmission at said receiver, said information transmission including a control signal which designates a designated version of programmable device,” and further recites “determining whether said specific version is said designated version in response to said control signal.” Addressing the determining step in claim 1, step d, which relates back to the storing information step, step a, Petitioner relates it to the just-described comparing operation in Nachbar, as follows:

Nachbar discloses determining whether the specific version (*i.e.*, the version of the operating system software that is stored on the subscriber machine) is newer or older than the designated version (*i.e.*, the version that is communicated in the *statfile* and is available from the librarian machine). *Id.* Nachbar discloses that *track* will “request from the librarian, those files that are included in the subscription list and have newer modification date” than its own version of the file.

Pet. 24.

The record, including Petitioner’s specific citations to Nachbar and Schmidt, supports Petitioner’s position. *See* Ex. 1005, 161–62; Ex. 1007, 9:28–33, 54:57–62. In summary, as Petitioner contends, the record shows that in response to a control signal (*i.e.*, Nachbar’s *statfile* entries as modified by Schmidt), each specific named subscriber uses *track* to compare stored information in each subscriber machine that identifies that specific machine and that subscriber’s current OS, to an OS system file sent by a librarian for potential updating, wherein the file to be updated also includes information identifying each specific subscriber machine and OS. *See* Ex. 1005, 161–62; Pet. 23 (“Nachbar discloses that the subscriber machine stores information in a ‘subscription list,’ which ‘describes the set of files to be considered for potential update from the librarian.’ . . . The stored information [also] specifies the specific version of the operating system

executing on the device, namely the ‘currentness’ of the file, which is represented by the time of last modification.” (quoting Ex. 1005, 161–162)). The record also supports Petitioner’s contention that “[t]he subscriber machine will then ‘see if the file is in its own *subscription list* . . . examine the modification date of its own version of the file . . . [and] request from the *librarian*, those files that are included in the *subscription list* and have newer modification dates.” Pet. 23–24 (citing Ex. 1005, 162).

4. Modified Control Signal

Further regarding the information transmission including a control signal recited in claim 1, the Petition identifies three items in Nachbar’s *statfile* as sent by the *librarian*, including a set of files to be considered for potential update, where each line in the *statfile* includes the type of object being handled, the name of the object, and “some measure of *currentness* of the object.” See Pet. 23 (quoting Ex. 1005, 162). Relying on Schmidt, Petitioner contends that the combination of Schmidt and Nachbar suggests modifying the currentness date so that one of the dates in Nachbar’s *statfile* matches the date of the OS currently executing in a subscriber machine. See Pet. 20–22, 24–25. The record, including specific citations to Nachbar, supports Petitioner’s contentions and rationale for combining the references, as discussed above and more fully below. See Ex. 1005, 161–62, 165.

Patent Owner contends that Nachbar, as modified by Schmidt, does not disclose or suggest a control signal, because the modified control signal does not include a hardware version. See PO Resp. 32–33. Patent Owner also argues that “[t]he ‘measure of currentness’ of a source code file does not indicate a version of a programmable device.” *Id.* at 33 (citing Ex. 2009 ¶ 108; Ex. 2006, 92:19–93:6 (“The currentness is not identifying a particular

hardware version. It's identifying the date on this particular piece of software.”)).

Patent Owner's arguments do not rebut Petitioner's showing, because they rely on Patent Owner's claim construction that requires an identification of a hardware version of a programmable device. *See* PO Resp. 33–35. Contrary to Patent Owner's claim construction and arguments, the '252 patent system indicates (and updates) OS or software versions executing on a programmable device, not merely hardware versions. *See supra* Section II.A (Claim Construction). To the extent the argument challenges Petitioner's showing that the prior art combination renders obvious the claimed control signal, the Petition does not rely solely on the measure of currentness as the control signal. *See* Pet. 19, 23. Rather, the Petition relies on three pieces of information in the *statfile*, including the measure of currentness. *Id.* Patent Owner agrees that the *statfile* includes three items of information. PO Resp. 34 (listing the three items).¹¹

Notwithstanding Patent Owner's reliance on Dr. Lesk's deposition testimony, Dr. Lesk clarifies that he does not rely solely on the measure of

¹¹ The librarian creates the *statfile* from the subscription list. Ex. 1005, 162. Each subscriber stores the librarian's subscription list, its own list, and the *statfile*, at least temporarily. *See* Ex. 1005, 161–163, 165. In general, to copy files, the librarian sends its subscription list and *statfile* to each subscriber. *See id.* The subscriber, using *track*, compares files in its subscription list to the *statfile* for a list for possible updates, and if a new update exists, the subscriber copies the latest file. *Id.* at 162. After any copy, the subscriber resets the date of last modification of the new local copies. *Id.* at 161–62. “Often, the subscription lists are identical on librarian and subscriber machines. However, this is not always the case. . . . [B]y editing local (subscriber) copies of the subscription list, local administrators can control updates on their machines.” Ex. 1005, 161.

currentness in his original Declaration: “In my opinion and what I wrote, Nachbar discloses the information transmission as a librarian statfile.” Ex. 2006, 93:2–4. Furthermore, specifying an update specifies the update time, the file being updated, and ultimately, the specific hardware that the uses the “appropriate” file, regardless of whether or not “the files are transmitted in source code form.” See PO Resp. 32–33 (arguing that each subscriber only obtains a file “appropriate file for its machine” and obtains source code files).

5. Patent Owner’s Additional (Hardware) Arguments

Patent Owner contends that “neither Nachbar nor Schmidt discloses or suggests a step of receiving an information transmission at the receiver which ‘includ[es] a control signal which designates a designated version of programmable device,’ as required by each independent claim (and the claims dependent thereon). PO Resp. 32. Patent Owner stresses that the prior art combination does not suggest specifying a *hardware* device, rather it only suggests communicating software updates when a subscriber version of a software differs from a librarian model. *Id.* at 3–4, 24–26.

Patent Owner also contends that the combination does not suggest modifying software when the two hardware versions are the same as required by each of the claims. *Id.* Patent Owner contends that the claimed invention “allows different types of devices **with different apparatus versions** to be updated remotely where desired. It also allows a service provider to have several different models of hardware in the field and be able to manage software downloads to correct, selected hardware models without fear of hardware models accepting incompatible software downloads.” PO Resp. 11. Patent Owner also argues that “the Nachbar

system is a pull-type of software distribution system rather than a push-type system. In Nachbar, updates are provided to the subscriber machines only upon request from the subscriber machines. . . . And, Nachbar specifically teaches not copying files that are hardware dependent.” PO Resp. 31–32.

6. Analysis of Patent Owner’s Contentions Regarding Nachbar’s Modified System

Patent Owner’s arguments are not persuasive and do not rebut Petitioner’s showing. Patent Owner’s arguments largely turn on claim construction, or point to features not required by the challenged claims. Patent Owner does not appear to dispute that the claimed combination renders obvious “storing information specifying said . . . programmable device . . . [by] . . . indicat[ing] a version of an OS executing on said programmable device,” according to the claim construction of the storing information step. *See supra*, Section II.A.7. As noted above, Petitioner relies on updating Nachbar’s UNIX OS. Pet. 24 (citing Ex. 1005, 162–63, 165–66).

As Petitioner contends, the challenged claims do not recite a “push-type” system or preclude a “pull-type” system. *See* Pet. Reply 4 (citing Ex. 1035 ¶ 7, 15, Ex. 1001, 286:50–287:10). Similarly, although Patent Owner characterizes Nachbar’s system as “file dependent” (PO Resp. 31–32), this argument fails to show unobviousness. The challenged claims do not preclude a file dependent system. According further to Patent Owner, “[t]he subscriber machine [in Nachbar] will request only those files that are appropriate for its machine.” *Id.* at 32 (citing Ex. 2009 ¶ 104). Patent Owner’s characterization implies that Nachbar’s subscriber knows to request a file containing an OS “appropriate” for itself—i.e., to its hardware system.

As explained in Section II.A (Claim Construction), the '252 patent's disclosed system includes a network with a finite and known number of hardware devices each running a specified OS. In this scenario, knowing the OS identifies the specific machine, or machine type, i.e., an Apple II OS points to an Apple II, a certain OS may identify an explicit decoder, and MS-DOS 2.0 specifies an IBM PC as opposed to an Apple II or decoder. Nachbar's system operates in the same type of environment. The VAX machines that use a UNIX OS or other files in Nachbar's system either share them if the slightly varying architectures each support them, or do not share them if the files or OSs are not compatible across requisitely different architectures: "Other complexities arise because some files are shared by machines of different architecture, while others are not." Ex. 1003, 159, 164 (distributing new VAX kernels to the other VAX machines). Nachbar also describes "the great ease with which one can maintain software *on machines of different architecture*." *Id.* at 164 (emphasis added).

According to Petitioner, and as the record shows, each subscriber in Nachbar responds to the *statfile* (i.e., control signal). *See* Ex. 1005, 162; *supra* Section II.B.5. The Petition explains and the record shows that "Nachbar discloses that *track* will 'request from the librarian, those files that are included in the subscription list and have newer modification date' than its own version of the file." Pet. 24 (citing Ex. 1005, 162); *see also id.* at 22 ("*track* compares files on the subscriber machine with the entries from the librarian's *statfile*"). In other words, based on its corresponding file name stored in the *subscription list* and *statfile* (e.g., GORP), each subscriber, using *track*, specifies that subscriber as a specific programmable device by identifying a UNIX OS version (as a timed update) currently executing on

each specific subscriber device, e.g., gorp. See Ex. 1005, 162,164; Pet. Reply 9–10.

Further discussing the control signal and any hardware information that the claims may require, Petitioner contends that

[i]n one example from Nachbar, the name “gorp” identifies a specific physical machine (*i.e.*, hardware), and the file path “/sys/conf/GORP /vmunix” refers to the version of the Unix kernel software that is compatible with that physical machine. (*Id.*, Ex. 1035 ¶ 18.) Russ agrees. (Ex. 1033 at 189:17–191:5.) This “GORP” file path is stored in the subscription list on the gorp machine. (Ex. 1005 at 165.) As disclosed earlier in Nachbar, files listed in the subscription list on a subscriber (receiver) machine are compared during the update process to files listed in a statfile from a librarian (transmitter) machine. (*Id.* at 162 (“for each file in the *statfile* . . . see if the file is included in [the subscriber’s] own subscription list”).) Thus, the “GORP” file path that specifies a specific hardware version for the subscriber is matched to a corresponding file path for that same hardware version on the librarian machine. (Ex. 1035 ¶ 18.) As a result, Nachbar discloses exactly what PMC asserts is missing—a “method for dealing with different versions of workstation hardware” by matching a specific hardware version at a subscriber machine.

Pet. Reply 10.

Patent Owner’s characterization of Nachbar follows:

The librarian machine provides a listing of files that are available to subscribers, called a statfile. *Id.* Statfiles have one line for each object. Each line contains three pieces of information: 1) a single character describing the type of object being handled, 2) the name of the object, and 3) some measure of the currentness of the object (the “measure of currentness” in *track* is always the modification date of the file expressed in seconds). *Id.* at 162. For each file in the statfile, the subscriber machine “see[s] if the file is included in its own subscription list” and “examine[s] the modification date of its own version of the file.” *Id.* at 162. The subscriber machine can “request from the librarian, those files

that are included in the subscription list and have newer modification dates.” *Id.* The subscriber machine then requests those files in the librarian’s statfile that have time of last modification dates that differ from the modification dates of its own version of the files. Russ Dec., ¶ 81 (Ex. 2009).

A subscriber machine “inspect[s] the librarian’s listing of available files and request s] new copies of files that are more current than its own.” Ex. 1005 at 162. This quoted passage makes clear that *track* simply operates off of file names and file-creation dates. It has no method for dealing with different versions of workstation hardware.

PO Resp. 23–24.

Except for the last two sentences in each summary above (characterizing Nachbar’s disclosure concerning different hardware versions), the record reveals that the parties materially agree as to how Nachbar’s system operates. The disagreement about hardware only concerns claim construction. On that issue, the record supports Petitioner for the reasons explained above. The challenged claims read on comparing OS versions on specific machines, and/or, at most, comparing a placeholder name such as “gorp” tied to the OS running on that specific machine, each to determine if a stored specific version is designated version in a control signal. As noted, Petitioner relies on Nachbar’s *statfile* as corresponding to the claimed control signal in claim 1. Pet. 19, 23; Reply Br. 10–12. The GORP/vmunix file path, stored in the subscription list, constitutes the required stored information in claim 1 that at least identifies the “appropriate” OS version for a specific hardware device (i.e., the device named “gorp”). See Pet. 19 (relying on the subscription list for the stored information as set forth in claim 1); Ex. 1005, 165 (the subscription list entry includes the file path GORP/vmunix); PO Resp. 32 (arguing that Nachbar’s subscriber obtains “appropriate” files “for its machine”).

Summarizing relative to claim 1, the record shows that Nachbar's system operates as follows: The filename or file path (e.g., GORP/vmunix) in the control signal (*statfile* as modified by Schmidt's system as explained further below), "designates a designated version of programmable device" gorp. See Ex. 1005, 161–162, 165; Ex. 1035 ¶ 18. Track causes subscriber gorp to compare GORP/vmunix or vmunix in a subscription list to GORP/vmunix in the *statfile*, and thereby "determine whether said specific version," as stored in file vmunix on gorp, is an older version of the "designated version" in response to the *statfile* control signal.¹² See Ex. 1005, 164–165 (explaining that "vmunix" has the same file name on machine "blob" in file GORP and on machine "gorp," and a comparison of versions allows subscriber gorp to "copy and install a new kernel without human intervention"); 162 ("for each file in the *statfile* . . . see if the file is included in [the subscriber's] own subscription list"); 164 ("Gorp's kernel resides in /sys/conf/GOPR/vmunix on blob and in /vmunix on gorp itself" and "track . . . compare[s] blob's file named sys/conf/GORP/vmunix with the local file named /vmunix and if the versions [i.e., times] differ, the new kernel will be copied to yet a third location"); Ex. 1035 ¶ 18.

In other words, to check for a potential update, gorp searches for its OS kernel stored in GORP/vmunix on blob, and compares it to the kernel stored in a local file vmunix on gorp, thereby specifying a programmable device gorp that runs a specific UNIX OS version. See Ex. 1105, 164 ("[T]he Unix kernel (/vmunix) is different on almost all machines. . . .

¹² The proposed modification by Schmidt involves comparing the same OS date versions, instead of checking for an older OS version, as explained above and further below.

Gorp's kernel resides in /sys/conf/GORP/vmunix on blob and in /vmunix on gorp."). In essence, only the machine named "gorp" responds to the statfile control signal that includes its file corresponding file name "GORP." The subscription list entry: blob: /sys/conf/GORP/vmunix causes gorp to compare OS versions in gorp's vmunix file, thereby rendering obvious, in light of Schmidt's suggestion, the storing and determining steps of claim 1.¹³ See Ex. 1005, 164–165; Reply Br. 10.

More generally, Nachbar discloses a software updating system for "machines running the UNIX operating system." Ex. 1005, Abstract, 165 ("At present, the system handles updates to about 20 machines running Berkeley 4.2BSD UNIX connected via an [E]thernet."). As noted, Patent Owner contends the similar subscriber machines that receive an OS all execute the same type of OS (i.e., the OS is not hardware dependent). See PO Resp. 32. Hence, according to Patent Owner's arguments, Nachbar discloses a set of VAX machines that all run newer or later versions of UNIX OS. It follows that identifying any machine in that embodiment, for example gorp, and associating it with a specific UNIX OS version via the file name GORP, specifies a specific programmable device that also indicates a class of UNIX OS versions executing on that specific hardware machine, thereby satisfying the challenged claim 1, even under Patent Owner's proposed construction. See Ex. 1005, 164–165.

¹³ Nachbar refers to the noted command on page 165 as "*subscription list entry*," and on page 162 explains that such an entry combined with the librarian's machine name "blob" also renders it a *statfile*. Compare Ex. 1005, 162, with 165; see Reply Br. 10 (discussing the command), 12 (explaining it is a *statfile*).

According to the '252 patent Specification, claim 1 embraces identifying one or more machines as an Apple II machine that runs an Apple II OS (or vice versa). *See supra* Section II.A (Claim Construction). It also embraces a system wherein all the machines on the network are Apple II machines, or IBM PCs, or both. *See id.* At the least, one of Nachbar's disclosed embodiments, which stores UNIX OS versions executing on a set of VAX machines, according to Patent Owner's arguments and the record, operates similarly to a network having only IBM PCs that execute different versions of MS-DOS or PC-DOS (i.e., 1.0, 2.0.). *See* Ex. 1003, 164–165. Accordingly, under the scenario of a finite and known set of OS and hardware machines on a given system, which Nachbar discloses, or at least suggests, identifying the UNIX OS version executing on a specific machine also implicitly indicates to the system that a compatible VAX machine executes that OS on that machine.

Even if some of the challenged claims require identifying different hardware versions (e.g., Apple II and IBM PC on the same network), as explained above and further below in connection with claims 10–12, 14, 22, 31, 32, 40, 45, 50, and 51, as noted, Nachbar discloses “the great ease” of updating “machines of different architecture.” *See* Ex. 1003, 164. Patent Owner agrees that each subscriber only receives the “appropriate” OS, and Patent Owner admits that “matching of apparatus versions” was known, albeit “not common practice.” Ex. 1003, 159 (different architectures), 164 (different VAX kernels); PO Resp. 32 (appropriate files); Mot. to Amend. 20 (admission). Accordingly, as Petitioner persuasively contends, matching hardware versions, if required, would have been obvious in order to ensure each machine obtained the appropriate OS. *See* Pet. 22, 35–36 (updating

requires ensuring compatible hardware); Ex. 1008 ¶ 77 (discussing Nachbar as rendering obvious updating “machines of different architecture” to ensure each machine has the latest update).

Patent Owner also filed Observations on Cross Examination (“Observation”) contending that Dr. Lesk testified that the name gorp “is not a class or type of machine or device,” which allegedly contradicts his Rebuttal Declaration (Ex. 1035) that “Nachbar clearly teaches a method for dealing with different versions of hardware.” See Paper 46, 3–4. Patent Owner’s Observation contentions do not rebut Petitioner’s showing but turn on Patent Owner’s claim construction. The challenged claims do not require comparing and storing manufacturer information designating a hardware class of machines. Rather, they require storing some type of information that indicates the OS executing on a specific programmable device, by storing an OS tied to that device, or by storing a placeholder name indicating the OS on that device: for example, URS decoder, example #3; Apple II OS or Apple II, PC DOS 1.0, etc. See Section I.A (Claim Construction); Ex. 1001, 266:40–46.

7. Analysis of the Parties’ Contentions Regarding Same or Different Versions

Turning to the portion of the claims requiring stored and control signal versions to be the same, Petitioner contends that the combination of Schmidt and Nachbar suggests modifying the currentness date of Nachbar so that the information in the control signal includes the same date (i.e., version) as the stored information identifying the OS version executing in a subscriber machine. See Pet. 19, 21–22, 25. The Petition explains as follows:

For example, one method, disclosed in Schmidt, could first ensure that the version of the operating system executing on the programmable device matches a designated version before providing updated instructions. *See* Ex. 1007 at 43:62–44:4. Another method could simply determine if the version of the operating system running on the receiver station is outdated and should be updated with a newer designated version. *See* Ex. 1005 at 162.

Pet. 21.

Schmidt's system generally discloses updating versions of software by identifying earlier versions to be updated. *See* Ex. 1007, 43:62–44:4, 45:57–62. Therefore, modifying Nachbar, in light of Schmidt, entails modifying Nachbar's *statfile* control signal, and determining if the currently executing OS version, in the subscription list, matches that of the *statfile*. According to Petitioner, using Schmidt's comparison step in place of Nachbar's similar comparison step to find a software version would have been an obvious mere substitution of known steps to obtain the same desired result of finding a previous version in order to update it. *See* Pet. 21–22. The version to be updated must be identified to be updated, and the version to be updated would be an older version relative to the update. *See id.*

In response, Patent Owner contends that “neither Nachbar nor Schmidt discloses or suggests a step of receiving an information transmission at the receiver which ‘includ[es] a control signal which designates a designated version of programmable device,’ as required by each independent claim (and the claims dependent thereon). PO Resp. 32. Patent Owner explains that the prior art combination does not suggest specifying a hardware device, rather it only suggests communicating software updates when a subscriber version of a software differs from a librarian model. *See id.* at 38–40. Further, Patent Owner argues it does not

suggest modifying software when the two hardware versions are the same as required by each of the claims. PO Resp. 3–4, 24–26, 40 (“Neither reference, nor the combination, discloses or suggests comparing an apparatus version at a receiver station to an apparatus version specified in a control signal. As explained above, there was no need to track hardware in these systems because they are pull-type distribution systems rather than the push-type distribution of the claims.”).

Patent Owner also argues that the relied-upon “excerpt [in Schmidt] describes *the notice operation* that is used to replace the unique identifier for a file when that file has been revised by a user. It does not relate to making a comparison of versions before a software update.” PO Resp. 27 (discussion Ex. 1007, 45:59–62), *see also id.* at 26 (discussing Ex. 1007, 18:48–59). Patent Owner explains that “Schmidt fails to disclose or suggest the claim limitations because searching for an earlier version is not the same as determining whether a specific version is the same as a designated version. An earlier version is not the same version, by definition.” *Id.* at 27. Patent Owner also argues that modifying Nachbar with Schmidt would have rendered Schmidt inoperative. *See id.* at 39–40 (citing Ex. 2006, 135:16–136:18).

According to the discussion above, Patent Owner’s arguments concerning hardware turn on claim construction and are not persuasive for the reasons discussed. Also, if claim 1 requires merely identifying when hardware versions are the same, and that hardware indicates an OS, for example, identifying an Apple II PC as using an Apple II OS, then Nachbar discloses this alleged claim requirement because Nachbar identifies gorp as using UNIX by using the file GORP. *See* discussion above and Pet. Reply

13 (citing Ex. 1005, 164–65 and stating that Nachbar matches hardware versions). In other words, gorp is the hardware identified and compared in a subscription list and *statfile* by use of the file path that includes /GORP/vmunix or /vmunix. See Ex. 1003, 164–65.

Regarding the modification of matching OS versions, Patent Owner’s argument that Schmidt’s teachings relate to the notice operation instead of updating software constitute an unpersuasive separate attack of the references. The argument that comparing to an earlier version does not suggest matching versions is not persuasive either, because the earlier version constitutes the desired version to be found. As Petitioner persuasively contends, “the difference between the comparison disclosed in Nachbar (*i.e.*, a mismatch with a new version) and the comparison in the claims (*i.e.*, a match with an old version) is a matter of mere design choice that one of ordinary skill would have readily understood.” Reply Br. 13.

Petitioner explains further with respect to Schmidt:

Schmidt also provides an example of the Notice Operation, and the example specifically discloses that the modeler searches for the earlier version of an updated file in the model, *e.g.*, the version “(Jan. 14, 1983, 14:44:09)” of the file “BtreeImpl.cedar.” (Ex. 1007 at 43:62–44:4; Ex. 1035 ¶ 29.) This earlier version of the updated file, which is designated when the Notice Procedure is called, corresponds to the “designated version” in the claims. (Ex. 1035 ¶ 29.) It is matched to a “specific version” specified in the model.

Pet. Reply 13; see also Pet. Reply 15 (citing Ex. 1007, 43:62–44:4; Ex. 1035 ¶ 30; Ex. 1001, 287:1–3).

Petitioner also addresses persuasively Patent Owner’s related argument that Schmidt does not teach looking for matches:

Schmidt looks for a match between versions at one step of the update process, and looks for a mismatch between versions at other steps. (*Id.*) [Patent Owner] inexplicably focuses only on the steps that look for a mismatch (Resp. at 35–37), while conveniently ignoring the step that looks for a match.

Id. at 14.

The record supports Petitioner’s explanations as quoted above. Patent Owner does not dispute that at least with respect to the notice operation, Schmidt teaches looking for an earlier version. *See* PO Resp. 27. Dr. Russ also does not dispute that the notice operation searches for earlier versions. Ex. 2009 ¶ 90.

Searching for an earlier version constitutes a date matching operation. *See* Ex. 1007, 43:62–44:4, 45:57–62; Ex. 1008 ¶¶ 57–58; Pet. Reply 13–15. Addressing his deposition testimony and Dr. Russ’s testimony, Dr. Lesk persuasively explains that in Schmidt’s “notice operation, the system looks for a match in versions rather than a mismatch.” Ex. 1035 ¶ 21 (citing Ex. 1007, 45:57–62).

Whether the notice operation involves software or OS updates goes to the strength of the suggestion for a modification, because the operation itself suggests a ready manner for finding files in a software system that at the least, also provides updates. In any event, contradicting Patent Owner’s related argument that the notice operation does not involve updates, Dr. Lesk also explains that “[t]he notice operation is an essential step in the distribution of updates.” *Id.* ¶ 22 (citing Ex. 1007, 9:50–59; Ex. 2009 ¶ 90). Schmidt supports Dr. Lesk’s testimony at the cited portions. *See* Ex. 1007, 9:50–59, 43:62–44:4, 45:57–62. Schmidt’s system modeler notifies the system editor of updates and uses a notice operation matching routine to find files to be updated. *See id.*; Ex. 1035 ¶¶ 21–23.

Regarding Patent Owner’s allegation of inoperability, Petitioner persuasively responds that Patent Owner relies on Dr. Lesk’s deposition testimony that addresses Nachbar alone, not a modification of Nachbar using Schmidt. See Pet. Reply 16 (citing Ex. 1035 ¶¶ 32–33). Dr. Lesk explains that Dr. Russ provides no explanation for why the system would be rendered inoperable, and points to his (Dr. Lesk’s) first Declaration, which provides a well-reasoned articulation of how to modify Nachbar—a modification he characterizes as “well within the capabilities of a person having ordinary skill in the art.” See Ex. 1035 ¶ 33 (citing Ex. 1008 ¶¶ 57–59). Relying in part on Schmidt’s method to suggest the proposed modification, Dr. Russ explains that “the *statfile* in Nachbar could also include a version number that must first be found on the subscriber’s subscription list before an update is provided.” Ex. 1008 ¶ 57, ¶¶ 57–60. In other words, modifying Nachbar’s *statfile* to identify a desired version constitutes a simple substitute for the similar method Nachbar uses to perform the same function of identifying a desired version in a predictable fashion, as Schmidt’s method suggests. See *id.* ¶¶ 57–60. “[W]hen . . . the prior art . . . is altered by the mere substitution of one element for another known in the field, the combination must do more than yield a predictable result,” *KSR Int’l. Co. v. Teleflex Inc.*, 550 U.S. 398, 416 (2007).

8. Erasing

Petitioner relies on Nachbar to reach the erasing feature recited in the challenged claims. See Pet. 24–25. Petitioner relies on Nachbar’s disclosure that “[b]ecause of the way UNIX operates, the old version of the shell must be moved aside before the new version can be moved into place.” Pet. 24–25 (quoting Ex. 1005, 164).

In the Decision to Institute, the panel reasoned as follows:

Petitioner sufficiently demonstrates that Nachbar’s disclosure of creating an updated version (e.g., booting a new kernel, updating UNIX) reasonably suggests putting the new version in the same memory location as the old version, thereby implying or suggesting erasing, in order to allow the system to access the updated version [in] the same manner it previously accessed the older version.

Dec. on Inst. 15.

Patent Owner argues that “neither Schmidt nor Nachbar, nor the combination, discloses or suggests” the erasing step of the challenged independent claims: “erasing any operating system instructions stored within an erasable portion of memory and then storing the communicated operating system instructions within the erasable portion of the memory, as required by each independent claim.” PO Resp. 31. Patent Owner contends that “moving aside” an old version as taught in Nachbar and relied upon by Petitioner “is not the same as the claimed requirement that such instructions be erased.” PO Resp. 41 (citing Ex. 1005, 164; Ex. 2006, 117:4–8; Ex. 2009 ¶ 2009).

Petitioner and Dr. Lesk disagree with Patent Owner. Petitioner explains that old instructions at the /vmunix location constitutes an erasable *portion* of memory, whence they are moved to a new location, /ovmunix, and then the new instructions are moved from a third location, /nvmunix, to the first location, /vmunix. Pet. Reply 16–17; see Pet. 24–25 (discussing moving old versions aside in Nachbar “before the new version can be moved into place”). In other words, Petitioner contends that moving files into place requires erasing of the original file from that original file location, and that the claims do not require erasing instructions from the device, only erasing

them from a portion of memory. *See* Pet. Reply 16–17; Pet. 24–25.

The record supports Petitioner. *See* Ex. 1005, 164–165; Ex. 1035 ¶ 34. Moving files aside and moving files in their place at least suggests erasing files from a memory portion, because the original files would no longer be at that original memory portion, and they would be replaced. *See* Pet. 24–25 (discussing Ex. 1005, 164); Ex. 1035 ¶ 34. Even if somehow moving original files aside and moving new files in their place, and erasing, are not “the same” in the context of the claims, Nachbar at least suggests erasing, contrary to Patent Owner’s arguments. *See* PO Resp. 41 (citing Ex. 2006, 117:4–8).

Contrary to Patent Owner’s arguments that address erasing in general, Dr. Lesk testifies that moving files from a memory portion in the context of the claims constitutes erasing from that portion. *See* Ex. 1035 ¶ 34; Ex. 2006, 116:19–117:8; PO Resp. 41 (discussing Ex. 2006, 117:4–8). Other than repeating the disputed claim limitation, Dr. Russ’s Declaration does not address erasing in the context of erasing from the claimed “portion of said memory.” *See* Ex. 2009 ¶¶ 125–127. Responding to Patent Owner’s arguments, Petitioner contends that Dr. Russ “had no opinion about whether files are erased from the first storage location in a scenario like this.” Pet. Reply 17 (citing Ex. 1033, 204:21–205:5). During his cross-examination, Dr. Russ answered that he had not “considered” a similar hypothetical to the situation in Nachbar: whether moving a file from one disk to another without retaining that file on the first disk constitutes erasing from the first disk. *See* Ex. 1033, 204:21–205:5.

The record implies that erasing instructions from memory was well-known prior to the invention. *See* 1035 ¶ 34 (testifying that Nachbar at least

suggests erasing to an artisan of ordinary skill). As a matter of common sense possessed by artisans of ordinary skill, a finite memory space that involves writing data thereto eventually requires erasing or overwriting for continued use thereof. *See* Ex. 1035 ¶ 34 (testifying it would have been obvious to erase to save space).

Based on the foregoing discussion, Nachbar discloses or suggests erasing instructions from a portion of memory as set forth in the challenged claims. Ex. 1005, 164. Even if moving files from one portion of memory to another and replacing the old file with a new file does not constitute erasing, given the feature of replacing files in the context of Nachbar, erasing would have been obvious for the purpose of saving memory space, especially where the system would no longer have use for the old files. *See* Pet Reply 17; Ex. 1035 ¶ 34 (erasing obvious to save space); Pet. 29 (arguing that the similar function of discarding saves space, which provides motivation for discarding).

Based on the foregoing discussion and the record, including the secondary evidence of nonobviousness discussed below in Section II.B.13, Petitioner demonstrates by a preponderance of evidence that claims 1, 3, 10, 12, 19, 21, 28, 30, 37, and 39 would have been obvious.

9. Claims 2, 11, 20, 29, 38, and 46

Claim 2 depends from claim 1 and recites “wherein said information transmission includes said operating system instructions.” Claims 11, 20, 29, 38, and 46 recite similar limitations. Petitioner sets forth a detailed showing that Nachbar discloses or renders obvious the limitation. *See* Pet. 26–28 (contending that providing OS instructions with Nachbar’s *statfile* would have been obvious because central administration was widespread

and it would have ensured uniformity across different stations). Patent Owner contends that a single information transmission control signal must also include the OS instructions, and it would not have been obvious to implement OS instructions in Nachbar's *statfile*. PO Resp. 42.

In the Institution Decision, as Patent Owner notes, we determined that an "'information transmission,' on this record, is not limited to a single transmission at a single time." Inst Dec. 31 (citing Ex. 1001, 8:14–17 (discussing or defining a "signal word" and a "signal unit," and stating that "[i]n all cases, signals may convey information in discrete words, transmitted at separate times or in separate locations, that receiver apparatus must assemble in order to receive one complete instruction")); PO Resp. 42.

Patent Owner contends that the cited passage relied upon in the Institution Decision does not apply to claims 2, 11, 20, 29, 38, and 46. PO Resp. 42. According to Patent Owner, "[t]he specification passage cited by the Board (Ex. 1001, 8:14–17) does not support the Board's proposition that *the claimed single information transmission* can be sent in multiple transmissions." *Id.* (emphasis added).

Contrary to this argument, however, these claims do not recite a "single" transmission. The '252 patent Specification specifically states that "in all cases, signals may convey information . . . transmitted at separate times or in separate locations, that receiver apparatus must assemble in order to receive one complete instruction." Ex. 1001, 8:14–17. Patent Owner's denial fails to address why the passage does not apply "in all cases," including those covered by claims 2, 11, 20, 29, 38, and 46, or is "contrary to the claim language." See PO Resp. 42. Generally discussing the '252 patent, Patent Owner also contends that an "information transmission

includes several messages supported by the system called ‘SPAM messages.’” PO Resp. 8 (citing Ex. 1001, 21:38–39).

Even if the claims require a single transmission, Petitioner presents persuasive rationale and evidence establishing that it would have been obvious to modify Nachbar’s *statfiles* to include operating instructions. *See* Pet. 27; Ex. 1008 ¶ 62.

Patent Owner contests Petitioner’s position and contends that the proposed modification would have been inoperable because in Nachbar, the librarian first transmits the *statfile*, which the subscriber inspects for newer dates, and then the subscriber requests updates based on that comparison. *See* PO Resp. 44. According to Patent Owner,

the librarian has no way to know which files to transmit to the subscriber until after the subscriber reviews the transmitted statfile to determine which files need to be updated on its machine. Consequently, in order to make the modification suggested by Petitioner[], the librarian would have had to transmit literally thousands of operating system files with the statfile instead of only transmitting those files that the subscriber actually needed to update. Such a modification would have completely defeated the purposes of Nachbar, namely, to make software updating efficient and driven by subscriber machines rather than the central administrator.

PO Resp. 44 (citing Ex. 2009 ¶ 132).

Dr. Russ’s Declaration repeats this argument. *See* Ex. 2009 ¶ 132. This argument assumes that the claimed method requires sending “thousands of operating system files.” Contrary to the argument, the claims do not require a set number of machines or thousands of OS files. As noted above, Nachbar describes an embodiment of “about 20 machines running UNIX . . . connected via an [E]thernet.” Ex. 1005, 165. Skilled artisans would have recognized that implementing Nachbar’s system to include OS files to be

updated in a *statfile* would have transferred OS files efficiently to a limited number of machines by using a single transmission instead of several transmissions, and also would have ensured uniformity where a limited number of machines require the same UNIX update version. See Ex. 1008 ¶ 62 (discussing central administration and uniformity).

Patent Owner's and Dr. Lesk's contention that Nachbar's system is "driven by subscriber machines rather than the central administrator," so that the modification defeats the purpose of Nachbar, oversimplifies or mischaracterizes Nachbar's system and Petitioner's proposed modification. See PO Resp. 44; Ex. 2009 ¶ 132. In Nachbar, for example, even under the proposed modification, the subscriber, using *track*, would still choose what files to update based on the subscription dates relative to the *statfile* dates, just as the subscriber does in the unmodified system of Nachbar. See Ex. 1005, 162. In both cases, the librarian provides a measure of central control, but a subscriber may choose not to run *track*, when to run *track*, and otherwise control updates by controlling what it lists in its own subscription list. See *id.*

Based on the foregoing discussion and the record, including the secondary evidence of nonobviousness discussed below in Section II.B.13, Petitioner demonstrates by a preponderance of evidence that claims 2, 11, 20, 29, 38, and 46 would have been obvious.

10. Claims 5, 23, and 41

Claim 5 recites "[t]he method of claim 1, said method further comprising the step of discarding instructions when said step of determining determines said specific version is not said designated version." Claim 23 recites "the method of claim 19, wherein when said receiver station cannot

connect said source of operation system instructions to a programmable device of said designated version, said control signal is discarded.” Claim 41 recites “the method of claim 37, further comprising the step of: discarding instructions upon determining that a programmable device of said specific version is not present at said receiver station.”

Petitioner contends that a person of ordinary skill in the art would have modified the combined system of Nachbar and Schmidt, because he or she “would have understood that data may be discarded if they are no longer useful.” Pet. at 28–29 (citing Ex. 1008 ¶ 65). Petitioner relies on the known example of discarding incorrectly addressed Ethernet packets, which generally suggests “discard[ing] unusable instructions to save space.” *Id.* at 29. Petitioner also contends that “for the same reasons, a person of ordinary skill in the art at the time of the alleged invention would have understood that the control signal may be discarded once the receiver station determines that it cannot connect to the appropriate source of operating system instructions.” *Id.*

Patent Owner contends that Nachbar and Schmidt “teach away from claims 5, 23 and 41” because “updates are only requested and provided if the user machines’ copies differ from a master. Thus, ***communicated instructions are never discarded in either system.***” PO Resp. 45. Patent Owner also contends that discarding Ethernet packets “is not comparable to the claim requirement for comparing a specific version of programmable device . . . in a control signal.” *See id.*

Addressing claims 5 and 23, Patent Owner contends that “[a]n Ethernet MAC address does not designate a version of programmable device, or even a version of operating system.” *Id.* at 46 (citing Ex. 2009

¶ 40). Without specifying a claim, Patent Owner contends that “[a]n Ethernet packet is discarded on the basis of its MAC address, not on the basis of comparing the operating system version or hardware version.” *Id.* at 46 (citing Ex. 2009 ¶¶ 138–39).

Patent Owner’s arguments are not persuasive for the following reasons. Claim 5 calls for “discarding instructions when said step of determining” does not result in a match. Claim 41 is similar. These claims do not specify what instructions to discard. *See* Pet. Reply 18 (noting that any instructions may be discarded). Claims 5 and 41 may refer to the transmitted OS instructions recited respectively in claims 1 and 37, or may refer to any other instructions. Claim 23 essentially reads on discarding a control signal when the system does not find a programmable device match.

Nachbar discloses requesting an OS file update “only if the file [in the *statfile*] is included in [the subscriber’s] own subscription list.” *See* Ex. 1005, 162. In other words, as discussed in connection with the independent claims, the combined system of Nachbar and Schmidt only provides updated OS instructions when there is a programmable device match. Also, one version of the combined system, as advanced by Petitioner and as discussed above in connection with the dependent claims 2, 11, 20, 29, 38, and 46, includes OS instructions with the *statfile*. Petitioner contends that “the claims are satisfied if any instructions are discarded for any reason, as long as the versions happen to be mismatched at the same time.” Reply Br. 19.

Based on the foregoing discussion, claims 5, 23, and 41 read on discarding any instructions in Nachbar’s *statfile* (i.e., control signal) and modified *statfile* (i.e., having OS instructions sent with the *statfile*), and claims 5 and 41 read on discarding any instructions, after a mismatch. As

Petitioner shows, in general, ordinarily skilled artisans would have discarded any unusable information, including any instructions or control signals that necessarily would not be required (i.e., would be stale) after a mismatch, in order to save memory space. *See* Pet. 28–29; Ex. 1008 ¶¶ 65–66.

Stated differently, not discarding such information, or saving everything, would have required an unlimited memory. *See* Ex. 1008 ¶ 65 (testifying that an artisan of ordinary skill “would have been motivated to discard unusable instructions to save space”). Discarding unused or stale instructions, was generally known, including via the Ethernet packet example cited by Petitioner. *See id.* That Ethernet teaching need not be bodily incorporated into the combined system of Nachbar and Schmidt to suggest generally discarding stale or unused instructions. Also, Nachbar uses Ethernet packets to transmit its instructions, thereby further suggesting the use of known packet methods in general operations. *See* Ex. 1005, 165. Based on the foregoing discussion, Nachbar and Schmidt, suggest, and do not teach away from, discarding instructions or a control signal after a mismatch.

Based on the foregoing discussion and the record, including the secondary evidence of nonobviousness discussed below in Section II.B.13, Petitioner demonstrates by a preponderance of evidence that claims 5, 23, and 41 would have been obvious.

11. Claims 10, 12–14, 31, and 32

Claim 14, which depends from claim 10, requires that an intermediate transmitter station transmit additional control signals, each additional control signal designating one of a plurality of different versions of programmable devices at a plurality of receiver stations, each additional control signal

including operating system instructions of one of the plurality of different versions, until a control signal that designates each one of the plurality of different versions is transmitted. Patent Owner contends that “[c]laim 32 is similar to claim 14 but depends from claim 28.” PO Resp. 47. Patent Owner contends that “[f]or similar reasons, the combination fails to disclose or suggest the added element of claim 31.” *Id.* n.3. Claims 12 and 13 recite similar limitations.

Petitioner contends that it would have been obvious to modify Nachbar to arrive at a system with the claimed features of intermediate transmitter stations. Pet. 37–38. According further to Petitioner, “on a given network of computers regularly operating *track*, as in Nachbar, the librarian machines will eventually transmit *statfiles* until a *statfile* is transmitted for each of the different versions of operating systems present on the network.” Pet. 37 (citing Ex. 1008 ¶ 76).

Petitioner also contends that the ’252 patent admits that intermediate transmitter stations were known and used in the television industry. Pet. 31–32 (citing Ex. 1001, 43:52–67). Petitioner provides persuasive evidence showing that transmitter relay stations were well known, including in packet switching networks like the Ethernet system of Nachbar, and that it would have been obvious to employ such intermediate stations in order to ensure the complete transmission of information. *See* Pet. 31–32 (citing Ex. 1017, 12–13; Ex. 1023; Ex. 1024, 1:11–12; Ex. 1008 ¶¶ 70–71). In addition to citing similar evidence, Dr. Lesk testifies that “[i]ntermediate relay stations were common in Morse telegraphy in the 19th century.” Ex. 1008 ¶ 70. Dr. Lesk provides several reasons for using an intermediate device in a packet-switched network, including to ensure that all packets have been received

before proceeding. *Id.* ¶ 71. Dr. Lesk also testifies that Nachbar’s Ethernet network employs 20 machines operating UNIX, and that Ethernet includes intermediate stations for retransmitting packets. *Id.* ¶ 69; Ex. 1005, 164. The record supports Petitioner’s position, according to the citations provided above, and as discussed further below.

Patent Owner relies on arguments discussed above in connection with claim 1, focusing on an alleged lack of a teaching for matching hardware versions, the lack of a push type system in Nachbar, and the alleged preferences for compiling locally and for discarding files for machines of different architecture. PO Resp. 47–49. Patent Owner’s arguments are not persuasive. Patent Owner fails to rebut Petitioner’s showing that using intermediate transmitters would have been obvious in order to ensure that all transmission packets arrive.

As discussed above in Section II.A (Claim Construction), the challenged claims do not require matching hardware versions, and even if they do, such matching was known, as Patent Owner admits. Mot. to Amend 20. Furthermore, even though Nachbar discloses discarding files due to “slight differences on all machines,” Nachbar employs this method only “sometimes,” to update “6000 files,” noting that “almost everything in /etc can be shared between machines of the same architecture.” Ex. 1005, 163. As Petitioner also persuasively points out, Nachbar discloses “different versions of the UNIX kernel targeted to multiple different physical machines. . . . In order for the librarian machine in Nachbar to provide different kernel versions to multiple physical machines, it has to send a statfile (*i.e.*, control signal) to each machine designating the kernel version for that machine.” Pet. Reply 20 (citing Ex. 1035 ¶ 18; Ex. 1005, 164–165).

Nachbar supports Petitioner. *See* Ex. 1005, 164 (noting the UNIX kernel “(/vmunix) is different on almost all machines. . . . New kernels are . . . distributed to the other VAX machines.”).

Therefore, the combined system of Nachbar and Schmidt would have suggested to a person of ordinary skill in the art, that updating OSs on different programmable devices, and accommodating hardware differences to the extent required by challenged claims 10, 12–14, 31, and 32, in order to render files compatible, especially when updating a small number of files for a small number of computers.

Based on the foregoing discussion and the record, including the secondary evidence of nonobviousness discussed below in Section II.B.13, Petitioner demonstrates by a preponderance of evidence that claims 10, 12–14, 31, and 32 would have been obvious.

12. Claims 22, 40, 45, 50, and 51

Claims 22, 40, 45, 50, and 51 recite additional limitations that are materially similar to the limitations recited in claims 10, 12–14, 31, and 32, and discussed above, including a second control signal and/or second programmable device, which the Petition persuasively addresses. *See* Pet. 38–42. Patent Owner contends that Nachbar does not disclose “different versions of programmable devices” and corresponding control signal for a second version. *See* PO Resp. 49–50. Patent Owner also generally maintains that the combination of Nachbar and Schmidt do not render obvious the limitations of claims 22, 40, and 51. PO Resp. 49–50 & n.4. Patent Owner characterizes Petitioner’s showing as contending that “*track* can maintain multiple software applications that exist on the same subscriber

machine,” which does not correspond to claim 22. *See* PO Resp. 50 (citing Pet. 39).

Patent Owner’s arguments are not persuasive. The receiver station recited in challenged claims reasonably covers at least two “different versions of subscriber machines or other devices at the same general location,” as Nachbar discloses. *See* Ex. 1005, 159 (describing “40 CPU’s of 4 different architectures running 3 different flavors of UNIX”), 165 (“the system handles updates to about 20 machines running Berkley 4.2BSD UNIX connected via the [E]thernet”)); *see* Dec. on Inst. 23 (reaching a similar determination based on the same or similar findings). Patent Owner does not address this determination and findings as outlined in the Institution Decision, which the record supports.

As also discussed above in connection with claims 10, 12–14, 31, and 32, the combined system of Nachbar and Schmidt suggests updating OS versions for different programmable devices having different kernels and architecture, using a variety of *statfile* control signals, and comparing hardware differences, when desired (and if the claims require that), in order to render files compatible, especially when updating a small number of files for a small number of computers.

Based on the foregoing discussion and the record, including the secondary evidence of nonobviousness discussed below in Section II.B.13, Petitioner demonstrates by a preponderance of evidence that claims 22, 40, 45, 50, and 51 would have been obvious.

13. Claims 9, 18, 27, 36, and 52

These claims require booting of the OS instructions. Weighing the cited evidence and arguments, including the secondary considerations of

unobviousness, we determine that Petitioner shows by a preponderance of evidence that challenged claims 9, 13, 18, and 36 would have been obvious. *See* Pet. 29–30 (citing Ex. 1008 ¶ 68; Ex. 1005, Abstract (disclosing “booting a new UNIX kernel”)). As Petitioner persuasively shows, Nachbar discloses booting OS instructions, as the claims generally require, and therefore, at least would have suggested booting. *See id.*; Ex. 1008 ¶ 68; Ex. 1005, 165. Booting in the context of challenged claims would have been obvious because the instructions must be booted to operate a machine and take advantage of the update. *See* Ex. 1005, Abstract. Patent Owner does not present separate arguments for claims 9, 18, 27, 36, and 52.

14. Secondary Considerations

i. Commercial Success: Licensing

Relying on testimony by its declarant, Mr. Holtzman, Patent Owner contends that “PMC has numerous licensees to the ’252 Patent, as well as to other patents in the portfolio.” PO Resp. 60 (citing Ex. 2007). Petitioner replies that Patent Owner fails to show “a nexus to the inventions recited in the challenged claims of the ’252 patent.” Pet. Reply 24. Petitioner’s argument is persuasive. Patent Owner’s single statement and blanket citation to an entire exhibit, Ex. 2007, without more, does not constitute a sufficient argument showing how any licenses demonstrate unobviousness.

In cases in which the proffered evidence of commercial success constitutes licenses, rather than sales of products embodying the invention, the licenses may have been taken only because they were cheaper than defending an infringement suit or for other business reasons. *See EWP Corp. v. Reliance Universal Inc.*, 755 F.2d 898, 908–09 (Fed. Cir. 1985) (“[I]t is not unusual to see astute businessmen capitalize on [a licensing

program for an otherwise invalid patent] by erecting a temporarily successful licensing program thereon. . . . They sometimes succeed because they are mutually beneficial to the licensed group or because of business judgments that it is cheaper to take licenses than to defend infringement suits, or for other reasons unrelated to the unobviousness of the licensed subject matter.”).

Similar to the characterization in *EWP*, Mr. Holtzman notes that some of the provided licenses to the “’252 Patent family,” or “the ’252 Patent, among other patents,” included a “substantial payment” in “settlement of litigation” or with knowledge of pending *inter partes* reviews. See Ex 2007 ¶¶ 16–24. Mr. Holtzman also contends that some companies made equity investments in addition to licenses, and also “many of the license amounts are orders of magnitude above the cost of defense in an infringement action.” Ex. 2007 ¶¶ 25–26.

This testimony about “substantial payments” relative to “cost of defense,” and equity investments, lacks a requisite factual support and fails to delineate anything specific to the ’252 patent family or the ’252 patent, let alone the challenged claims of the ’252 patent. It is entitled to little weight. Even affording it some weight, the testimony does not address whether or not any settlement or license has anything to do with the particular merits of the challenged claims of the ’252 patent. The testimony, at face value, at most shows that a license to the whole ’252 patent family may cost substantially more than defending any litigation with respect to one of the patents in the ’252 patent family, but it does not even mention specific litigation for the ’252 patent or relative cost to a license thereto. Also,

litigation advances with certain risk whereas a license provides a known outcome.

The tenuous testimony fails to show “affirmative evidence of nexus.” *Iron Grip Barbell Co. v. USA Sports, Inc.*, 392 F.3d 1317, 1324 (Fed. Cir. 2004). Patent Owner must demonstrate “a nexus between the *merits of the invention* and the licenses of record”; otherwise, the evidence of licenses garners little weight. *In re GPAC Inc.*, 57 F.3d at 1580 (emphasis added) (citation omitted). “[W]ithout a showing of [a] nexus, ‘the mere existence of . . . licenses is insufficient to overcome the conclusion of obviousness.’” *In re Antor Media Corp.*, 689 F.3d 1282, 1293 (Fed. Cir. 2012) (quoting *Iron Grip Barbell*, 392 F.3d at 1324).

In *In re GPAC Inc.*, 57 F.3d 1573, 1580 (Fed.Cir.1995), the court found that “in affidavits reciting the license history of the ’111 patent, GPAC did not establish which claims(s) of the patent the licensing program incorporates, GPAC has not shown that licensing of [applicant’s] invention arose out of recognition and acceptance of the subject matter claimed.” Like the applicants with licenses in *EWP Corp*, *GPAC*, and *Antor Media*, Patent Owner has not shown a nexus to the challenged “subject matter claimed”: Mr. Holtzman’s testimony lists patent family licenses and revenue, but does not discuss the merits of any of the challenged claims as they relate to any particular license for the ’252 patent in the portfolio of licenses. See PO Resp. 60 (citing Ex. 2007). Furthermore, not all of the ’252 patent claims have been challenged.

The cited testimony of Mr. Holtzman does not establish whether a specific license (or licensing clause, etc.) for the ’252 patent occurred because of the merits of the challenged claims, the merits of unchallenged

claims, for other patented inventions, or for other economic reasons related to the whole '252 patent family. The evidence at best implies that some settlements included licenses for the '252 patent family perhaps to avoid some litigation for a patent or patents in the family, or that equity investments may have occurred based on the company itself, PMC. *See* Ex. 2007 ¶¶ 8–27. Mr. Holtzman's summary coalesces with our summary: "This success is attributable not to any particular feature itself, but rather to the combination of features of PMC's inventions." Ex. 2007 ¶ 27.

Accordingly, Patent Owner's licensing program evidence is afforded little weight in showing that any one of the challenged claims in the '252 patent would not have been obvious.

ii. Industry Praise

Industry praise for an invention may provide evidence of nonobviousness where the industry praise is linked to the claimed invention. *See Geo. M. Martin Co. v. Alliance Mach. Sys. Intern. LLC*, 618 F.3d 1294, 1305 (Fed. Cir. 2010); *Asyst Tech'n, Inc.* 544 F.3d at 1316. Patent Owner asserts that "PMC has received professional acclaim and industry recognition of its inventions. Additionally, there are numerous patents and publications that cite to the '252 Patent family." PO Resp. 60 (citing Ex. 2007).

Petitioner replies that Patent Owner fails to show "a nexus to the inventions recited in the challenged claims of the '252 patent." Pet. Reply 24. Petitioner's argument is persuasive. Patent Owner's brief statements and blanket citation to an entire exhibit, Ex. 2007, without more, does not constitute a sufficient argument showing how any alleged praise

demonstrates unobviousness with respect to a challenged claim in the '252 patent. See PO Resp. 60.

Similar to its licensing argument, Patent Owner's citation to Mr. Holtzman's entire Declaration, and the Declaration itself, fails to explain how evidence of value or "numerous" citations," constitutes industry praise with nexus to a feature, let alone a novel feature, of the challenged claims in the '252 patent. See PO Resp. 60 (citing Ex. 2007); Ex. 2007 ¶¶ 28–33. Considering just the '252 patent, it includes 52 claims, 292 columns, and 199 pages of information, and includes much background information. See Ex. 1001. Patent Owner's argument fails to show how its alleged evidence of value or praise, which goes to the whole '252 patent family, distinguishes over many citations to, and great reviews of, popular text books or journal articles. See Ex. 2007 ¶¶ 28–32.

Patent Owner does not show a nexus to relevant industry praise for a product, or anything else, and also does not show that any evidence of praise is reasonably commensurate in scope with, and covered by, a challenged claim. See *Bayer Healthcare Pharms., Inc. v. Watson Pharms., Inc.*, 713 F.3d 1369, 1377 (Fed. Cir. 2013) (finding that brief discussions of Patent Owner's product in journal articles "fall well short of demonstrating true industry praise" and reasoning that "industry praise of what was clearly rendered obvious by published references is not a persuasive secondary consideration"). A challenged '252 patent claim does not cover all information disclosed in all the patents of the '252 patent family—information that allegedly, at least in part, constitutes the object of praise. Any alleged praise is not linked to the claimed invention, rather, even if it

does constitute industry praise, it is linked to the whole '243 patent family. *See Geo. M. Martin Co.*, 618 F.3d at 1305.

Patent Owner has not established a sufficient nexus between the merits of the claimed invention of the challenged claims in the '252 patent and the alleged industry praise. Accordingly, Patent Owner's evidence of secondary considerations based on industry praise is entitled to little or no weight.

iii. Summary

Based on the foregoing, we conclude that Patent Owner's evidence of indicia of nonobviousness is entitled to little or no weight.

15. Tamaru, Nachbar, and Schmidt

Patent Owner's Response and Petitioner's Reply make it clear that the central issue involved in this ground revolves around what Nachbar and Schmidt fairly teach or suggest, and the claim construction issue of hardware comparison. *See* PO Resp. 51–60; Pet. Reply 21–22. For example, Petitioner's Reply responds to Patent Owner's Response as follows:

[Patent Owner] argues that Tamaru is only directed to different versions of software and does not discuss versions of hardware. (Resp. at 51.) But the '252 patent claims do not recite hardware versions. (*Supra*, IV.A.) Even if the claims required different versions of hardware, the combination would not be missing this feature because Nachbar discloses versions of hardware. (*Supra*, VI.A.)

[Patent Owner] next argues that Tamaru, Nachbar, and Schmidt do not disclose matching a specific version of a programmable device. (Resp. at 52-54.) Yet both Nachbar and Schmidt disclose such matching. (*Supra*, Section VII.A.)

Pet. Reply 21 (Sections VI.A and VII.A refer back to Petitioner's analysis of the ground involving Nachbar and Schmidt). Patent Owner also argues that

combining Schmidt and Nachbar with Tamura would not have been obvious, but this argument would be rendered moot if Nachbar and Schmidt were deemed to be deficient. *See* PO Resp. 54.

In other words, Petitioner relies on Nachbar and Schmidt to address Patent Owner's allegations of shortcomings in the ground of Tamura, Nachbar, and Schmidt. Based on the arguments presented, if the ground involving Nachbar and Schmidt does not survive an appeal, the ground involving Tamara, Nachbar, and Schmidt likely also will not survive for the same reason. If the former ground does survive, that would constitute sufficient reason not to reach the latter ground. Therefore, the Final Decision reached here on the Nachbar/Schmidt ground renders it unnecessary to reach the Tamara/Nachbar/Schmidt ground. *Cf. In re Gleave*, 560 F.3d 1331, 1338 (Fed. Cir. 2009) (not reaching obviousness after finding anticipation).

16. Patent Owner's and Petitioner's Motions for Observations

Patent Owner and Petitioner filed Motions for Observations, and Petitioner filed an Amended Motions for Observation. Paper 46; Paper 47; Paper 51. Patent Owner and Petitioner each filed Responses to the Motions for Observations, and Patent Owner filed an Amended Response to Motions for Observation. Paper 50; Paper 49; Paper 53.

Having considered the filings, and having commented on them where material and appropriate in the Final Decision above, we determine that the filings warrant no further discussion as the filings have been afforded due consideration.

17. Conclusion

Weighing the cited evidence and arguments, including the secondary considerations of unobviousness, we determine that Petitioner shows by a preponderance of evidence that challenged claims 1–3, 5, 9–14, 18–23, 27–32, 36–41, 45, 46, and 50–52 the '252 patent would have been obvious.

C. Patentability of Proposed Substitute Claims

In an *inter partes* review, amended claims are not added to a patent as of right, but rather must be proposed as a part of a Motion to Amend. 35 U.S.C. § 316(d). As the moving party, Patent Owner bears the burden of proof in establishing that it is entitled to add proposed substitute claims 53–85. 37 C.F.R. § 42.20(c); *see also Microsoft Corp. v. Proxyconn, Inc.*, 789 F.3d 1292, 1306–08 (Fed. Cir. 2015) (patentee bears the burden of showing that its proposed substitute claims are patentable over the prior art of record); *Prolitec, Inc. v. ScentAir Techs., Inc.*, 807 F.3d 1353, 1363–64 (Fed. Cir. 2015) (same); *Idle Free Systems, Inc. v. Bergstrom, Inc.*, Case IPR2012-00027 (PTAB June 11, 2013) (Paper 26) (informative) (“For a patent owner’s motion to amend, 37 C.F.R. § 42.20(c) places the burden on the patent owner to show a patentable distinction of each proposed substitute claim over the prior art.”); *MasterImage 3D, Inc. v. RealD, Inc.*, Case IPR2015-00040 (PTAB July 15, 2015), Paper 42 (same).

As part of this showing, Patent Owner must demonstrate (1) the amendment responds to a ground of unpatentability involved in the trial; (2) the amendment does not seek to enlarge the scope of the claims of the patent or introduce new subject matter; (3) the amendment proposes a reasonable number of substitute claims; and (4) the proposed claims are supported in the original disclosure. 37 C.F.R. § 42.121.

Upon review of the Motion to Amend, Patent Owner has not met the requirements of 37 C.F.R. § 42.121.

1. Responsive to a Ground of Unpatentability and Reasonable Number of Substitute Claims

Contingent upon the determination of unpatentability of the challenged claims, Patent Owner's Motion to Amend seeks to replace all of the challenged claims, claims 1–3, 5, 9–14, 18–23, 27–32, 36–41, 45, 46, and 50–52, with proposed substitute claims substitute claims 53–85. Mot. to Amend 1. That contingency has manifested. Patent Owner's Motion to Amend satisfies the burden with respect to a reasonable number of substitute claims and responsiveness. *See* 37 C.F.R. § 42.121.

2. Written Description Support

A motion to amend claims must identify clearly the written description support for each proposed substitute claim. 37 C.F.R. § 42.121(b). The requirement that the motion to amend must set forth the support in the original disclosure of the patent is with respect to *each claim*, not for a particular feature of a proposed substitute claim. In other words, it is inadequate to show written description support for just the claim feature added by the proposed substitute claim. The motion must account for the claimed subject matter as a whole, i.e., the *entire* proposed substitute claim, when showing where there is sufficient written description support for each claim feature. *See Nichia Corp. v. Emcore*, IPR2012-0005, slip op. at 4 (PTAB June 3, 2013) (Paper 27).

In its Motion to Amend, Patent Owner does not address written description support for claims 54, 55, 57, 59, 61–63, 65–69, 71–75, 77–80, and 82–85. *See* Mot. to Amend 2–12. Accordingly, Patent Owner has not

satisfied the burden of showing written description support with respect to claims 54, 55, 57, 59, 61–63, 65–69, 71–75, 77–80, and 82–85.

3. Impermissible Broadening

A motion to amend claims “may be denied where . . . [t]he amendment seeks to enlarge the scope of the claims of the patent.” 37 C.F.R. § 42.121(a)(2)(ii). Patent Owner groups all proposed amendments together to show patentability. See Mot. to Amend 16, 13–21. The proposed substitute claims are similar in scope in material respects to each other based on the arguments presented, and we deem claim 53 representative of the proposed substitute claims.

Proposed substitute independent claim 53 follows:

53. (Substitute for claim 1, if found unpatentable)

A method of reprogramming a receiver station, said receiver station including a programmable device of a specific version having a memory, a signal detector, and a receiver operatively connected to said signal detector, said method comprising the steps of:

storing information specifying: (a) said specific version of said programmable device, wherein said specific version indicates a version of an operating system executing on said programmable device and controlling the processing capabilities of said programmable device, and (b) a specific apparatus version of said programmable device;

receiving, from a transmitter station, an information transmission at said receiver, said information transmission including a control signal which designates a designated apparatus version of programmable device;

passing said information transmission to said signal detector and detecting said control signal;

determining whether said specific apparatus version is said designated apparatus version of programmable device in response to said control signal;

communicating operating system instructions to said memory only when said step of determining determines that said specific apparatus version is said designated apparatus version, wherein said communicating comprises erasing any operating system instructions stored within an erasable portion of said memory and then storing said communicated operating system instructions within said erasable portion of said memory; and
executing said communicated operating system instructions to control operation of said programmable device.

Mot. to Amend A1.

Regarding the proposed amendments, “Patent Owner submits that the term ‘apparatus version’ can be simply defined as a hardware version.”

Mot. to Amend 13. In contrast, regarding the original claims, Patent Owner contends that “under the Board’s construction, the original claims arguably could cover operating system reprogramming based on matching information of *either* the software version *or* the hardware version between the programmable device of a targeted receiver station and the control signal.” Mot to Amend 16.

Patent Owner supports its characterization of the original claim scope as follows:

To the extent the Board’s construction of “said specific version of said programmable device” is applied, which “includes a variation of a programmable device indicated by the operating system it employs” . . . the Board has indicated that “said specific version of said programmable device” in the original claims would encompass both software (*i.e.*, operation system) version and hardware version.

Id. (quoting Dec. on Inst. 7).

Comparing the scope of the proposed amendments to the original claims, Petitioner disagrees with Patent Owner’s characterization of the scope of the original claims:

Claim 53, for example, recites “determining whether said specific apparatus version is said designated apparatus version.” (Mot. at A-1.) In the original claim, this comparison step involved “said specific version,” which by virtue of its antecedent basis had to indicate “a version of an operating system.” (Ex. 1001 at 286:66–287:3.) But in substitute claim 53, the added word “apparatus” has changed the antecedent basis, so the operating system version is no longer implicated in the comparison. *Thus, the substitute claim attempts to remove a requirement that was present in the original claim—namely the requirement that one operating system version is compared to another.*

Opp. 16–17 (emphasis added).

As the above passage shows, Petitioner contends that the original claims require comparing stored information that indicate an OS, and the substitute claims only require comparing hardware versions. Therefore, Patent Owner and Petitioner only agree regarding the scope of the proposed substitute claims—they agree that the claims require comparing only hardware versions.

Claim 53 supports the agreed-upon characterization, because although the system stores OS indicative information at step (a), nothing in claim 53 explicitly requires comparing such OS information. Claim 53 also explicitly delineates stored hardware information at step (b) from the stored OS information in step (a), and the control signal only “designates a designated apparatus version of programmable device,” which the claimed method compares to “said specific [stored information about an] apparatus version.”

In other words, comparing the apparatus version does not indicate an OS version, because claim 53 separates those items of information, and stores the latter without specifying that comparing the apparatus version constitutes indicating an OS version. *See Merck & Co. v. Teva Pharm. USA,*

Inc., 395 F.3d 1364, 1372 (Fed. Cir. 2005) (“A claim construction that gives meaning to all the terms of the claim is preferred over one that does not do so.”); *Power Mosfet Techs., L.L.C. v. Siemens AG*, 378 F.3d 1396, 1410 (Fed. Cir. 2004) (“[I]nterpretations that render some portion of the claim language superfluous are disfavored.”). Furthermore, where Patent Owner (1) carries the burden, (2) does not argue that proposed substitute claims include an OS comparison, and (3) the parties agree to that construction, we ordinarily decline to “examine the claims in greater detail than argued.” See *In re Baxter Travenol Labs*, 952 F.2d 388, 391 (Fed. Cir. 1991) (“It is not the function of this court to examine the claims in greater detail than argued by appellant”); *Asyst Technologies Inc. v. Empak, Inc.*, 268 F.3d 1364, 1369 (Fed. Cir. 2001) (“[W]e see no reason to depart from the position consistently taken on this issue by the parties and the district court.”).

Following the general guidance in *Baxter Travenol Labs* and *Ayst*, we see no reason to depart from the parties’ agreement regarding the scope of the substitute claims, as the record facially supports the agreement to the extent advanced. In addition to the claim 53 language discussed above, the scope of claim 53, and the proposed substitute claims, includes programmable devices with memory. Per the decoder and generic processor examples discussed in above in Section II.A (Claim Construction), different apparatus versions may include different types and numbers of memory chips (e.g., RAM and/or ROM chips). Without more, these hardware differences would fail to provide information about “the version of the operating system executing on said programmable device,” as claim 53 and the independent claims specify. As noted, Patent Owner’s arguments group the claims together with respect to the issue of broadening, and Patent

Owner has the burden on its Motion to Amend to point out any variances in scope upon which it relies.¹⁴

On the other hand, as noted, the parties disagree over the original scope of claim 1. According to Patent Owner, “under the Board’s construction, the original claims were broad enough to potentially cover the storage and matching of *either* a hardware version *or* a software version. *In contrast, the substitute claims recite storing “a specific apparatus version” and matching apparatus versions*, which clearly narrows the scope as compared to the Board’s construction of the original claims.” PO Reply 12 (citing Dec. on Inst. 6) (emphasis added).

Patent Owner similarly argues as follows:

Here, the inclusion of the term “apparatus version” in the substitute claims differentiates those claims from claims where “said specific version of said programmable device” can purportedly include an operating system version (under the PTAB’s construction), and limits the reprogramming criteria specifically to a match of the “apparatus version” (i.e., version information about the programmable device hardware). Therefore, the amendments add limitations to, rather than removing limitations from, the original claims.

¹⁴ See *Proxyconn, Inc.*, 789 F.3d at 1306–08 (patent owner carries the burden to show patentability of proposed amendments). “It is the applicants’ burden to precisely define their invention, not the PTO’s.” *In re Morris*, 127 F.3d 1048, 1056 (Fed. Cir. 1997). To the extent Patent Owner’s arguments are not clear during prosecution, which logically applies to presenting claim amendments, our reviewing court “interprets this as a veiled attempt to avoid the potential future effects of prosecution history estoppel. Such evasiveness we cannot condone, particularly when the public must rely on the written record to define the resulting property right.” *Id.* (“The problem in this case is that the appellants failed to make their intended meaning explicitly clear.”).

Mot. to Amend. 16–17 (emphasis added).

The record does not support Patent Owner’s argument. As explained above in Section II.A (Claim Construction), and as Petitioner argues, original claim 1 *requires* comparing stored OS information that indicates that OS version executes on a programmable device. Contrary to Patent Owner’s characterization of the Institution Decision, it also does not contemplate a claim construction that includes merely comparing hardware versions:

[A] specific PC-DOS operating system version indicates that an IBM PC uses that version. *See* Ex. 1001, 265:51–55. Stated differently, a generic computer device becomes a specific computer device defined by the specific operating system it employs—i.e., an algorithm defines the structure of the device.

Therefore, on this record, “said specific version of said programmable device” includes a variation of a programmable device indicated by the operating system it employs.

Dec. on Inst. 7.

In summary, Patent Owner neither argues nor shows that the proposed substitute claims require comparing stored information that identifies an OS version. Rather, as noted above, Patent Owner argues to the contrary.

Patent Owner has not met the burden in showing 1) the original claims do not require comparing OS versions and 2) the substitute claims require comparing OS versions. Therefore, the “amendment” impermissibly “seeks to enlarge the scope of the claims of the patent.” *See* 37 C.F.R.

§ 42.121(a)(2)(ii).

Based on the foregoing discussion, Patent Owner’s Motion to Amend is *denied*.

III. CONCLUSION

Petitioner has shown by a preponderance of the evidence that claims 1–3, 5, 9–14, 18–23, 27–32, 36–41, 45, 46, and 50–52 of the '252 patent would have been obvious under 35 U.S.C. § 103(a) as unpatentable over the combination of Nachbar and Schmidt. In addition, we conclude Patent Owner has not demonstrated, by a preponderance of the evidence, that the Motion to Amend meets the requirements set forth in 37 C.F.R. § 42.121.

IV. ORDER

For the reasons given, it is

ORDERED that, by a preponderance of the evidence, claims 1–3, 5, 9–14, 18–23, 27–32, 36– 41, 45, 46, and 50–52 the '252 patent are unpatentable;

FURTHER ORDERED that Patent Owner's Motion to Amend is *denied*; and

FURTHER ORDERED that because this is a Final Written Decision, parties to the proceeding seeking judicial review of the decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

PETITIONER:

Brenton R. Babcock
Jeremy Anapol
Mark Lezama
Joseph Cianfrani
Colin B. Heideman
Kent N. Shum
KNOBBE, MARTENS, OLSON & BEAR, LLP
2BRB@knobbe.com
2JAA@knobbe.com
2MZL@knobbe.com

IPR2014-01528
Patent 7,783,252 B1

2JSC@knobbe.com
2cbh@knobbe.com
2kns@knobbe.com
BoxAmazon1@knobbe.com

PATENT OWNER:

Stephen T. Schreiner
Ce Li
Eleanor Yost
Jennifer Albert
Phong Dinh
GOODWIN PROCTER LLP
SSchreiner@goodwinprocter.com
cli@goodwinprocter.com
EYost@goodwinprocter.com
JAlbert@goodwinprocter.com
PDinh@goodwinprocter.com

Thomas J. Scott, Jr., Vice President & General Counsel
PERSONALIZED MEDIA COMMUNICATIONS, LLC
tscott@pmcip.com