UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION

NETFUEL, INC.,	)	
	)	
Plaintiff	)	
v.	)	**No. 13 C 7895**
	)	
F5 NETWORKS, INC.,	)	**Judge Rebecca R. Pallmeyer**
	)	
Defendant.	)	

MEMORANDUM OPINION AND ORDER

As nearly anyone employed in the modern world can attest, computer networks do not always function as expected. When things go wrong, there can be significant disruptions for users of the computer network, which can result in widespread interruptions for employees and customers. Entities that maintain computer networks, therefore, try to catch and resolve problems before they lead to a disruption of the whole network. But relying on human intervention to monitor networks for potential problems can be cumbersome and time-consuming. The patent at issue uses software "agents" to anticipate problems in a network and correct those problems without human input.

Plaintiff NetFuel, Inc. ("NetFuel") is a Delaware corporation with offices in Los Gatos, California and Park Ridge, Illinois. (Compl. [1] ¶ 3.) NetFuel makes "programmable infrastructure for software-defined networking" (Compl. ¶ 6), which the court takes to mean that NetFuel makes machinery on which to run computer networks; this "infrastructure" can run its own software programs to manage the network. NetFuel also provides network engineering consulting and professional services to clients. (*Id.*) As part of its "programmable [network] infrastructure," NetFuel uses a system for managing computer networks, patented by James Harlow, one of NetFuel's founders, in U.S. Patent No. 7,747,730 ("the '730 patent"). (NetFuel's Responsive Claim Construction Brief [98] ("NetFuel Mem.") 2.) The system uses "software agents" to manage the network, reducing the need for human input. (*Cf.* '730 Patent, Joint Appendix ("JA") Ex. 1 [91-1], col. 1 ll. 11–24, col. 2 ll. 8–20, col. 3 ll. 1–10.)

NetFuel has filed this suit against Defendant F5 Networks, Inc. ("F5"), a Washington corporation with offices in Oak Brook Terrace, Illinois. (Answer [20] ¶ 4.) F5 is NetFuel's competitor; it offers products that, like NetFuel's, integrate hardware and software. (*Id.* at ¶ 9); *see BIG-IP Platform*, F5 NETWORKS, https://f5.com/products/big-ip (last visited June 19, 2017). NetFuel complains that F5 is infringing the '730 patent. The parties have proposed competing interpretations of eight terms in the '730 patent. The court construes these terms as follows.

BACKGROUND

A. The Patented Invention

Computer networks must be constantly managed to ensure that they perform securely and efficiently. ('730 Patent, JA Ex. 1, col. 1 ll. 11–12.) Specifically, networks must be managed to ensure (1) that they will "continue operating even if nodes [or points where individual machines connect] fail[,]" (2) that they continue to provide sufficient quality of service, and (3) that they can operate efficiently even when the network includes a very large number of devices. (*See id.* at col. 1 ll. 11–16.) The '730 patent pertains to a technique for managing a computer network that limits the need for human input. (*See id.* at col. 2 ll. 8–10.)

Instead, the invention supplants some human input with what are known as software agents. (Claim Construction Hr'g Tr. [103] 24:20–25:15, Dec. 4, 2015; '730 Patent, JA Ex. 1, col. 2 ll. 10–12, 22–26, 47–50.) A software agent is essentially a software version of a concept familiar in the law: an entity that performs a task, with some degree of autonomy, on behalf of someone else.[1] An agent in the physical world can perform its task without input from the principal; this is equally true when an agent is a machine, such as a robot on a factory floor,

[1] *See Software Agent*, WIKIPEDIA, https://en.wikipedia.org/wiki/Software_agent (last visited Apr. 25, 2017); *see also* Hyacinth S. Nwana, *Software Agents: An Overview*, 11 KNOWLEDGE ENGINEERING REV., no. 3, Sept. 1996, http://agents.umbc.edu/introduction/ao/, at § 4.

which can perform its repetitive task without needing constant human guidance. A software agent operates in the same way—it can perform its task without human input. For example, a software agent useful to shoppers could scan a large number of websites for a certain product, and identify the website offering the product at the lowest price;[2] without such a program, the human user would have to look at each website herself.

NetFuel's system of software agents is typically "implemented on a large network comprising a number of network devices." ('730 Patent, JA Ex. 1, col. 2 ll. 47–48.) Each device "hosts" (apparently meaning "includes as part of its programming") an agent runtime environment (ARE), in which software agents can operate. (*See id.* at col. 2 ll. 12–13, 48–50.) A runtime environment generally provides a software program with the tools that the program needs to execute its task.[3] For example, most software programs, including software agents, likely need to access the memory of the device that they are operating on. Though the patent does not provide details about the characteristics of these particular AREs, the tools within the ARE allow agents to communicate with their devices and with each other. (*Id.* at col. 2 ll. 13–17.)

Agents perform these routine tasks quickly, effectively automating them, to accomplish their goals. These tasks and goals together are referred to as "policies." At their most basic level, policies are a series of "if . . ., then . . ." tests; in many cases these tests are compounded, which increases complexity. (*Id.* at col. 22 ll. 58–65.) In the shopping software agent example, the "policy" of such an agent would consist of both the task (scanning websites and comparing

[2] *Id.*

[3] *Cf. Runtime*, DICTIONARY OF COMPUTER SCIENCE, ENGINEERING & TECHNOLOGY 431 (Philip A. Laplante ed., 2001) (defining "runtime" as "adjective describing the support libraries needed to execute a program); *Runtime System*, WIKIPEDIA, https://en.wikipedia.org/wiki/Runtime_system (last visited June 14, 2017); WALTER BRENNER, RÜDIGER ZARNEKOW & HARTMUT WITTIG, INTELLIGENT SOFTWARE AGENTS: FOUNDATIONS AND APPLICATIONS 126 (Anthony S. Rudd trans., 1998).

prices) and the goal (finding the lowest price). (*Cf.* ‘730 Patent, JA Ex. 1, col. 2 ll. 50–56, col. 22 ll. 50–52; Claim Construction Hr’g Tr. 5:15–6:9.)

The patented system is effectively a hierarchy. The upper layers of the system are called “modelers.” (*Id.* at col. 2 ll. 29–32, col. 3 ll. 30–34.) The modelers, appropriately, “model network behavior,” that is, they apparently use numerical models to test what the network will do in certain situations. (*Id.* at col. 3 ll. 1–10.) By modeling network behavior, the patented system can observe the effects of new policy (that is, a different set of tasks, goals, and priorities for managing network traffic) on the network without having to actually implement the new policy; the patented system can effectively test the new policy via the modeler first.[4] (*See id.*) This modeled policy is called “test policy.” (Claim Construction Hr’g Tr. 78:11–18.) Through this modeling, the modelers can determine the “optimal policy” for the desired network behavior—the policy for the network to operate most efficiently, depending on the circumstances. (‘730 Patent, JA Ex. 1, col. 3 ll. 1–10.) Together, the modelers are an “agent control mechanism;” they send the optimal policy to the agents, who are the lower tiers of the system and actually implement the policy.[5] (*See id.* at col. 2 ll. 29–31, col. 3 ll. 1–10, 30–34; *see also id.* at FIG. 1, col. 2 ll. 49–50, col. 3 ll. 11–27.)

But communication is not one-way between the modelers and the agents. A modeler gets the data that it uses for its modeling from the agents themselves. (*Id.* at col. 2 l. 65–col. 3 l. 1.) Thus, the agents monitor the system and send data to the modeler, which uses the information to constantly update the optimal policy, which is in turn sent back to the agents. (*Id.* at col. 3 ll. 1–10.) As a result, the network policy, which ensures the most efficient operation of the network, is updated dynamically without human input. (*Id.*)

[4] By modeling different scenarios this way, the modeler can also predict what circumstances could cause a part of the network to “fail.” (Claim Construction Hr’g Tr. 5:22–6:9.)

[5] The agent control mechanism is able to communicate with the agents through the agent runtime environment. (‘730 Patent, JA Ex. 1, col. 2 ll. 13–17.)

When an agent detects a problem with the network—that is, where a device within the network has "deviated outside its normal operational parameters" (*id.* at col. 3 ll. 39–40)—the system works from bottom to top to distribute new policy to the agents to correct the problem. First, the agent detects the abnormality. (*See id.* at col. 3 ll. 38–42.) Second, the agent determines whether it already has a policy to correct the deviation; if it does, it will apply that policy to the device. (*Id.* at col. 3 ll. 40–45.) If the agent does not have the necessary policy, it will request the necessary policy first from its ARE,[6] and then from other agents and AREs within the patented system. (*Id.* at col. 3 ll. 45–50.) If the agent still does not have an appropriate policy to correct the deviation, it goes to the modelers, which as described above, create new policy by modeling the network parameters. (*Id.* at col. 3 ll. 50–64, col. 21 ll. 40–51.)

The specification explicitly defines an agent: "As used herein the term 'agent' denotes a program that performs some type of operation, which may be information gathering or some processing task, in the background." (*Id.* at col. 7 ll. 21–24.) The specification also describes software agents:

> In particular, a software agent is a virtual entity which: (a) is capable of acting in a runtime environment; (b) can communicate directly with other agents (messaging); (c) is driven by a set of tendencies (expressed in the form of individual objectives or of a satisfaction/survival function which it tries to optimize or policy); (d) possesses resources of its own (logic and algorithms); (e) is capable of perceiving its environment (state); (f) has only a partial representation of its environment; and (g) is able to reproduce/clone itself[.]

(*Id.* at col. 7 ll. 27–37.) Finally, the specification also lists "general characteristics" of agents: "As will be appreciated various implementations of the agents are possible, however each of the agents will have the following general characteristics[.]" (*Id.* at col. 10 l. 32–col. 11 l. 11.) What follows is a long list of additional characteristics, including, for example:

> (a) agents assume a secured environment already exists wherever they run (there is only one Security Manager per ARE);
> . . .

[6] Though the patent does not address this, it appears that AREs themselves can also have policies that are available to agents.

> (f) each agent maintains information regarding its management status and the management domain under which it is being administered;
> . . .
> (h) each agent is assigned a named thread group upon creation;
> . . .
> (l) agents have the ability to be passivated and/or re-activated by the ARE.

(*Id.* at col. 10 l. 32–col. 11 l. 11.)

Two subsets of software agents are "intelligent" agents and "autonomous" agents. Intelligent and autonomous agents have many characteristics in common, which are identified in the specification. (*Id.* at col. 7 ll. 50-57, col. 8 ll. 26–36.) Yet there are also distinctions between intelligent and autonomous agents, which pertain to the agents' respective abilities to solve problems with or without other input.[7] (*Id.* at col. 7 ll. 50–67.)

B. The Disputed Claim Terms

The '730 patent sets forth 36 claims. (*Id.* at col. 34 l. 54–col. 38 l. 11.) The disputed terms are: "network component," "failure," "predict/predicting," "software agent," "autonomous agent," "embodied in hardware," "clone," "computer network," and "kill;" there is also one lengthy term: "Modeler embodied in hardware to create test policy and to model a behavior of the computer network based on the test policy thereby to determine an optimal policy for the computer network said modeler comprising a predictive algorithm to predict a failure of a network component; wherein the modeler determines appropriate policy based on the prediction" (hereinafter "modeler . . ."). In an Appendix to this opinion, the court has provided relevant material from the '730 patent to place these terms in the context of the claims; the disputed terms are found in independent claims 7 and 30, and dependent claims 10, 11, 16–19, 21, and 26. (*Id.*)

[7] These distinctions and similarities are described below in the court's discussion of the term "autonomous agent."

C. Prosecution History

Harlow filed his patent application on June 28, 2002, but the patent was not issued until June 29, 2010. (*Id.* at [22], [45].) In the meantime, the Patent Office issued U.S. Patent Number 6,839,850 to Campbell ("the Campbell patent"). (Campbell Patent, Ex. 3 to NetFuel Mem. [98-4] at [45].) The subject matter of the Campbell patent, issued in 2005, is distinct from that of the '730 patent: the Campbell patent covers a system for detecting unauthorized access to computer networks. (Campbell Patent, col. 1 ll. 7–11.) Yet in reviewing the application for the '730 patent at issue in this case, the examiner rejected earlier versions as unpatentable over the Campbell patent. (JA Ex. 2 at 483.) The examiner determined that the Campbell patent involved use of software agents; specifically, the examiner identified a process in the Campbell patent, known as the "MANAGER" process,[8] as a software agent. (*See id.*) The examiner found that Harlow's claims would have been obvious to a person skilled in the art of computer programming, who would recognize the possibility of combining the functionality of the MANAGER process with other prior art. (*Id.*) Harlow responded that the MANAGER process differed from his software agents; while the MANAGER process interacted with "audit agents"[9] and could clone itself, "Campbell does not disclose that the audit agents *themselves* are able to clone themselves." (*Id.* at 503 (emphasis added).) Harlow added that "Campbell does not disclose that the audit agents are capable of perceiving their own state." (*Id.*)

Although the examiner was not initially persuaded by Harlow's response (*id.* at 546), Harlow amended the claims, adding a limitation that described the agents' ability to request further policy from other agents or the modeler (as described above). (JA Ex. 2 [91-4] at 567,

[8] The MANAGER process involves "a forking server daemon that clones a copy of itself for each Audit Agent." (Campbell Patent, col.11 l. 65–col. 12 l. 9.) A daemon, in turn, is "a background process capable of initializing other processes with little or no input from the user." (*Id.*) The court thus recognizes at least some similarity between the MANAGER process and software agents.

[9] It is unclear whether "audit agents" are a subset of software agents.

609.) After considering Harlow's argument that this amendment resolved any problem involving patentability, the examiner rescinded his rejection on this basis. (*Id.* at 622.)

But the examiner then rejected several claims for another reason: the software agent, agent support mechanism, and modeler could be "implemented as software," and therefore "the claims can cover software per se and software per se does not fit into a statutory category."[10] (*Id.* at 623.) NetFuel overcame this defect by reciting that the agents, agent support mechanism, and modeler "are embodied by hardware." (*Id.* at 636.) The Patent and Trademark Office issued the '730 patent on June 29, 2010. ('730 Patent, JA Ex. 1 at [45].) NetFuel filed suit against F5 on November 4, 2013, alleging that certain products in F5's BIG-IP Product Suite, which are systems of application delivery services on hardware or software platforms, infringe the '730 patent.[11] (Compl. ¶¶ 7, 9, 11.)

<u>DISCUSSION</u>

A. Legal Standards Governing Claim Construction

"[T]he claims of a patent define the invention to which the patentee is entitled the right to exclude." *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). Claim construction "giv[es] proper meaning to the claim language[,]" *Abtox, Inc. v.*

[10] Though it is not well-settled that software is indeed an abstract idea (hence unpatentable), *cf. Alice Corp. Pty. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2358 (2014); *Intellectual Ventures I LLC v. Symantec Corp.*, 838 F.3d 1307, 1314–16 (Fed. Cir. 2016), it seems commonplace for patent examiners to evaluate claims as though this is the case. *See CMG Fin. Servs., Inc. v. Pac. Trust Bank, F.S.B.*, 50 F. Supp. 3d 1306, 1318 (C.D. Cal. 2014), *aff'd sub nom. CMG Fin. Servs., Inc. v. Pac. Trust Bank*, 616 F. App'x 420 (Fed. Cir. 2015) ("The Board also explained that computer instructions, or 'software per se' were necessarily only an abstract idea where they were not explicitly tied to any particular machine."); Ebby Abraham, *Bilski v. Kappos: Sideline Analysis from the First Inning of Play*, 26 BERKELEY TECH. L.J. 15, 45 (2011) ("[T]he BPAI has yet to find a software claim patent eligible when the claim does not limit the code on a computer readable medium[.]").

[11] NetFuel also alleged that F5 infringes another of NetFuel's patents (Compl. ¶ 12); those allegations are not before the court for claim construction, and neither party has mentioned it in its brief. The court is uncertain whether NetFuel is continuing to pursue that infringement claim.

Exitron Corp., 122 F.3d 1019, 1023 (Fed. Cir. 1997), and thus defines the scope of the protected invention. Claim construction is "exclusively within the province of the court[.]" *Teva Pharm. USA, Inc. v. Sandoz, Inc.*, 135 S. Ct. 831, 837 (2015) (quoting *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 372 (1996).

Terms in a claim "are generally given their ordinary and customary meaning[]" to a person skilled in the art. *Phillips*, 415 F.3d at 1312 (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)). When this ordinary meaning is "readily apparent," claim construction "involves little more than the application of the widely accepted meaning of commonly understood words." *Id.* at 1314. "In such circumstances, general purpose dictionaries may be helpful." *Id.* But a person of ordinary skill in the art ("POSITA") "is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." *Id.* at 1313. Importantly, the claims "are part of 'a fully integrated written instrument,' consisting principally of a specification that concludes with the claims." *Id.* at 1315 (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 978 (Fed. Cir. 1995) (en banc), *aff'd,* 517 U.S. 370 (1996)). The specification "is the single best guide to the meaning of a disputed term." *Id.* (quoting *Vitronics*, 90 F.3d at 1582).

Like the claim language and the specification, the prosecution history is also intrinsic evidence of the claim term's meaning. *Id.* at 1317. "[T]he prosecution history provides evidence of how the PTO and the inventor understood the patent." *Id.* The prosecution history, however, "often lacks the clarity of the specification and thus is less useful for claim construction purposes." *Id.*

In addition to the intrinsic evidence, the court may consider extrinsic evidence, which consists of "evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises." *Id.* (quoting *Markman*, 52 F.3d at 980). Extrinsic evidence "is 'less significant than the intrinsic record in determining the legally operative meaning of claim language.'" *Id.* (quoting *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 862 (Fed. Cir. 2004)). Thus, when the intrinsic evidence resolves an ambiguity in a disputed claim term, "it is improper to rely on extrinsic evidence." *Vitronics*, 90 F.3d at 1583. Here, the parties have presented some extrinsic evidence to construe the disputed claim terms, but much of the evidence is intrinsic.

B. Network Component

Claim Term	**NetFuel's Proposed Construction**	**F5's Proposed Construction**
Network Component	Hardware or software that is part of a configuration of data processing devices and software connected for information exchange	A device that is part of a computer network

The first term that the parties dispute is "network component," in claims 7, 16, and 30. At the heart of the parties' disagreement is whether this term includes software; F5's construction limits a network component to a device, which would mean exclusively hardware.

A "component" in general is "a constituent part." *Component*, MERRIAM-WEBSTER COLLEGIATE DICTIONARY 255 (11th ed. 2003). NetFuel effectively contends that the analysis ends here, but the court must still decide whether software is a constituent part of a network—that is, a "network component." The term "network component" does not appear in the specification, so the parties focus their arguments on the language of the claims and the patent as a whole.

First, F5 points out that the only independent claims in which "network component" appears discuss it in the same context: "a predictive algorithm to predict a failure of a network component" (claim 7) and "predicting a failure of a network component based on a predictive algorithm" (claim 30).[12] In the specification, the functionality of the invention to predict a failure is described, in one embodiment, as "predict[ing] the failure of a network device." ('730 Patent,

[12] F5 also contends that the patent does not enable the prediction of the failure of a network component (Def. F5's Reply Claim Construction Br. [100] ("F5 Reply") 1; Claim Construction Hr'g Tr. 29:10–14), but that argument is more appropriate for summary judgment.

JA Ex. 1, col. 23 ll. 57–60.) This means, according to F5, that a component must be a device. F5 notes, further, that nothing in the patent predicts the failure of software. (Def. F5's Opening Claim Construction Br. [96] ("F5 Mem.") 11.) But this argument assumes the conclusion: if a "network component" includes software, then the claims could involve predicting the failure of software. F5's construction attempts to import a limitation from one embodiment described in the specification. The Federal Circuit has cautioned against this practice. *See Phillips*, 415 F.3d at 1323 ("[W]e have expressly rejected the contention that if a patent describes only a single embodiment, the claims of the patent must be construed as being limited to that embodiment.").

The court will not read limitations from an embodiment "absent a clear indication in the intrinsic record that the patentee intended the claims to be so limited." *Epos Techs. Ltd. v. Pegasus Techs. Ltd.*, 766 F.3d 1338, 1341 (Fed. Cir. 2014) (quoting *Liebel-Flarsheim v. Medrad, Inc.*, 358 F.3d 898, 913 (Fed. Cir. 2004); *see Liebel-Flarsheim*, 358 F.3d at 906 ("Even when the specification describes only a single embodiment, the claims of the patent will not be read restrictively unless the patentee has demonstrated a clear intention to limit the claim scope using 'words or expressions of manifest exclusion or restriction.'") (quoting *Teleflex Inc. v. Ficosa N. Am. Corp.*, 299 F.3d at 1327). In *Liebel-Flarsheim*, the patents at issue concerned devices for injecting fluids; none of the claims expressly referred to a pressure jacket surrounding the syringe, but all embodiments did. 358 F.3d at 900–01. The defendant argued that "the claims of those patents must be construed as limited to devices that use pressure jackets[,]" *id.* at 905–06, but the Federal Circuit rejected this argument. *Id.* at 908.

Here, similarly, the fact that a single embodiment describes predicting a failure of a network device is not sufficient to limit the claims reciting "network component" to a device. In *Liebel-Flarsheim*, language in the specification "focus[ed] on the use of the invention in conjunction with pressure jackets"—indeed, every embodiment had them—but that was not enough to disclaim uses of the invention without a pressure jacket. *Id.* This case is even stronger; the specification does not “focus on” predicting a failure of a network device; F5 has identified just one instance where such functionality is mentioned, with respect to one embodiment. In fact, it is not even clear to the court that “predicting a failure of a network device” would never involve software; a network device could “fail” because of a problem with software. That one embodiment describes predicting the failure of a network device will not limit network components to network devices.

F5 also points to one place where the patent purportedly uses “network component” and “network device” interchangeably: Claim 16 recites “wherein the agent support mechanism comprises an agent runtime environment configured for a particular network component.” F5 claims that this can only refer to a network device, citing the third sentence of the specification’s Detailed Description: “The software agents operate within an agent runtime environment . . . which is hosted on a particular network (host) device.” (‘730 Patent, JA Ex. 1, col. 2 ll. 12–14.) These two references to an ARE, one as “configured for a particular network component” and another as “hosted on a particular network device,” demonstrate, according to F5, that network and device are synonymous.

“Different terms or phrases in separate claims may be construed to cover the same subject matter where the written description and prosecution history indicate that such a reading of the terms or phrases is proper.” *Nystrom v. TREX Co.*, 424 F.3d 1136, 1143 (Fed. Cir. 2005). The court is not convinced that these passages support such a reading, however. It is not clear to the court that “configured for” and “hosted on” mean the same thing. If they do not, then the court can glean nothing from the fact that AREs are both configured for a particular network component and hosted on a particular network device. Without such evidence or explanation, the court cannot assume that “configured for a particular network component” and “hosted on a particular network device" are truly interchangeable, and limit the meaning of the term "network component."[13]

Arguing for a broader understanding of the term "network component," NetFuel points to language of the specification that it claims distinguishes components from devices: "The system includes discovery agents which are used to examine and determine the capability of a network device (host) and the equipment or *components* of the *device*." ('730 Patent, JA Ex. 1, col 12 ll. 18-21 (emphasis added)). "There is an inference, however, that two different terms used in a patent have different meanings." *Comaper Corp. v. Antec, Inc.*, 596 F.3d 1343, 1348 (Fed. Cir. 2010). Yet *Comaper* noted that "this inference is not conclusive[.]" 596 F.3d at 1348. And it is not obviously conclusive here; after all, even if "network component" is construed to mean a "device," the device itself could have components, and it would make sense to refer to them as such, rather than to say "the equipment or devices of the device." That the patent distinguishes between a device and its own components does not defeat F5's construction.

More helpful to NetFuel are the parts of the patent that explicitly describe software as "components" of other things. For example, Figure 2 of the '730 patent depicts software agents running within an ARE as "components" of the ARE. (*See* '730 Patent, JA Ex. 1, FIG. 2, col. 4 ll. 24–27.) NetFuel also notes that the patent lists software modules as "system components." (NetFuel Mem. 7) (citing '730 Patent, JA Ex. 1, col. 30 l. 18–col. 34 l. 55). Thus, NetFuel contends, the patent itself implies that device components and system components can be software, so a POSITA would understand that a network component could also be software. F5

[13] F5 also points to other places in the specification that purportedly require "network component" to mean "network device" when read in conjunction with claim 16. (*See* '730 Patent, JA Ex. 1, col. 2 ll. 23–25 ("The system **10** is vertically layered comprising an ARE **12** which is loaded on a host platform **14** defined on a network device."); *id.* at col. 2 ll. 47–49 ("Typically, the system **10** is implemented on a large network comprising a number of network devices, each hosting an ARE.")). These passages provide no more guidance than the one described in the text: there is no support for the contention that "defined on" and "implemented on a large network comprising" means the same as "configured for."

does not explain why a skilled artisan would understand that system components and device components could consist of software, but a network component could not.

Extrinsic evidence, cited by F5, also does not answer this question. F5 cites the 2002 Microsoft Computer Dictionary, which defines "network" as "[a] group of computers and associated devices that are connected by communications facilities." (*Network*, MICROSOFT COMPUTER DICTIONARY 362 (5th ed. 2002), Ex. A to F5 Mem. [96-1].) According to F5, a "component" of a network must be one of these devices. In assessing this construction, the court notes, first, that as described above, the intrinsic evidence does not support it, and the court cannot prefer a construction supported only by extrinsic evidence that contradicts the intrinsic evidence. *See Vitronics*, 90 F.3d at 1583. Second, the definition F5 relies on requires that the devices be "connected by communications facilities." The court presumes that "connection" in this context means more than simply putting cables between devices; devices connected by cables do not truly become a network unless they also become connected through software that enables these devices to communicate with each other. A group of devices would not be "connected" without software to provide instructions for those devices. A dictionary that NetFuel cites (albeit not for this purpose) bolsters this conclusion by describing an attribute of a network as "running a network operating system." *Network*, WEBSTER'S NEW WORLD DICTIONARY OF COMPUTER TERMS 352 (6th ed. 1997), Ex. 2 to NetFuel Mem. [98-3].) The court concludes that software can be a constituent part, or component, of a network.[14]

[14] NetFuel relies on other extrinsic evidence, as well, including a definition of "network architecture": "the complete set of hardware, software and cabling standards for a local area network (LAN) design." (NetFuel Mem. 6); *Network Architecture*, WEBSTER'S NEW WORLD DICTIONARY OF COMPUTER TERMS 352 (6th ed. 1997), Ex. 2 to NetFuel Mem [98-3]. The court finds this reference less useful. "Architecture" describes "the manner in which the components of a computer or computer system are organized and integrated[.]" *Architecture*, MERRIAM-WEBSTER COLLEGIATE DICTIONARY 65 (11th ed. 2003). In the network context, as with ordinary architecture, the set of parts that make up a coherent structure are distinct from the instructions or plans for that structure. That software could describe the manner in which network components are organized does not require the conclusion that software can be a network component. NetFuel's construction finds much better support in the intrinsic evidence.

The parties' proposed constructions of "network component" demonstrate that they also disagree about the definition of "network." Because "computer network" is itself a disputed term, the court addresses it separately below. "Network component" is construed as "hardware or software that is a constituent part of a computer network."

C. Failure

Claim Term	NetFuel's Proposed Construction	F5's Proposed Construction
Failure	The inability of a network component to operate reliably or to operate at all	The inability of a network component to operate reliably or to operate at all (not including suboptimal performance)

The parties dispute the construction of the term "failure," which appears in claims 7 and 30. "Failure" appears only once in the specification: "[i]n another embodiment, the refinery, [sic] uses predictive algorithms to predict the failure of a network device and to determine appropriate corrective policy ahead of the failure." ('730 Patent, JA Ex. 1, col. 23 ll. 57–60.) Though both parties initially proposed different constructions, they now agree that the construction should include "inability of a network component to operate reliably or to operate at all." (F5 Reply 2.) In its reply brief and at oral argument, however, F5 argued for the caveat that failure does not include "suboptimal performance." (Claim Construction Hr'g Tr. 38:20–22.)

F5 urges that there is a distinction between something that continues to perform its function, albeit suboptimally, and something that functions only intermittently. (F5 Reply 2–3; Claim Construction Hr'g Tr. 38:22–39:1.) As an example of suboptimal performance that is not "failure," F5 cites a car whose engine had not been tuned up, or a computer that runs slowly. (Claim Construction Hr'g Tr. 39:2–14.) F5 also cites to the language of the patent: "failure" is used in the claims in the context of "predict a failure"—an expression that, in F5's view, must mean something more cataclysmic than a mere problem with the system.[15] (*Id.* at 41:14–19.)

[15] NetFuel points out that this is not entirely correct; the reference to "failure" in the specification continues into the following sentence: "In another embodiment, the refinery uses

Thus, F5 urges that the court should reject a definition of failure that would include mere "suboptimal performance."

The problem with F5's proposal, however, is that the expression "suboptimal performance" is arguably broader than "failure." "Sub-" denotes performance that is "less than completely, perfectly, or normally." *Sub-*, MERRIAM-WEBSTER COLLEGIATE DICTIONARY 1241 (11th ed. 2003). "Suboptimal," thus can mean anything less than the best or perfect performance. "Suboptimal" performance could refer to a car with an engine that needs a tune-up, but also to a car that starts only half the time. But both parties seem to agree that a car that fails to start, even only some of the time, has "failed." (*See* Claim Construction Hr'g Tr. 39:2–4.) "Suboptimal," accordingly, may be understood to encompass failure.

To support the argument that "failure" does not include what it deems the less-serious category of "suboptimal performance," F5 refers to dictionary definitions. First, the Microsoft Computer Dictionary notes that "[a] common cause of system failure is loss of power." *Failure*, MICROSOFT COMPUTER DICTIONARY 205 (5th ed. 2002), Ex. A to F5 Mem. As F5 reads this language, it means that failure must be something akin to a power loss. Second, F5 points to the IBM Dictionary of Computing, which tracks F5's original proposed construction: "the termination of the ability . . . to perform its required function." *Failure*, IBM DICTIONARY OF COMPUTING 262 (10th ed. 1993), Ex. B to F5 Mem. [96-2]; (F5 Mem. 11–12.) These dictionaries do support the contention that a power loss and the termination of the ability to perform are failures. But that is not a matter of dispute; both sides agree that failure includes "the inability of a network component to operate at all." Both sides also agree that the inability to operate *reliably* is a failure, as well. The dictionaries do nothing to distinguish between "inability to operate reliably," which F5 concedes is part of the construction, and "suboptimal performance," which F5 wants excluded from the construction.

predictive algorithms to predict the failure of a network device and to determine appropriate corrective policy ahead of the failure. At **292** a determination is made as to whether the *problem or abnormality* has been remedied." ('730 Patent, JA Ex. 1, col. 23 ll. 57–61 (emphasis added).)

To carve out “suboptimal performance” from the definition of “failure” would exclude events that the parties agree are failures. Because “suboptimal” encompasses a broad category of performance, that word injects unnecessary ambiguity into the definition. *Cf. O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1360 (Fed. Cir. 2008) (“The purpose of claim construction is to ‘determin[e] the meaning and scope of the patent claims asserted to be infringed.’”) (alteration in original) (quoting *Markman*, 52 F.3d at 976). Adding this vague caveat would not facilitate understanding of the term’s meaning. The court construes “failure” as “the inability of a network component to operate reliably or to operate at all.”

D. Predict/Predicting

The parties have agreed to construe “predict/predicting” as “determine/determining that something will or might happen in the future.” In particular, F5 agreed in its reply brief to NetFuel’s construction, but changed “determine” to “determine/determining.” (NetFuel Mem. 8; F5 Reply 3.) “Predict/predicting” is therefore construed as “determine/determining that something will or might happen in the future.”

E. Software Agent

Claim Term	**NetFuel’s Proposed Construction**	**F5’s Proposed Construction**
Software Agent	A virtual entity which: (a) is capable of acting a [sic] runtime environment; (b) can communicate directly with other agents (messaging); (c) is driven by a set of tendencies (expressed in the form of individual objectives or of a satisfaction/survival function which it tries to optimize or policy); (d) possesses resources of its own (logic and algorithms); (e) is capable of perceiving its environment (state); (f) has only a partial	A virtual entity which: (a) is capable of acting in a runtime environment; (b) can communicate directly with other agents (messaging); (c) is driven by a set of tendencies (expressed in the form of individual objectives or of a satisfaction/survival function which it tries to optimize or policy); (d) possesses resources of its own (logic and algorithms); (e) is capable of perceiving its environment (state); (f) has only a partial

	representation of its environment; (g) is able to reproduce/clone itself	representation of its environment; (g) is able to reproduce/clone itself, and **(h) has the general characteristics described at col 10, line 31 to col. 11, line 11 of the ‘730 patent**

“Software agent” appears in claims 7, 10, 17, and 30. Parts (a) through (g) of both parties’ proposed construction are verbatim quotations from the specification. (‘730 Patent, JA Ex. 1, col. 7 ll. 20–36 (emphasis added) (“Column 7 Characteristics”).) This is perfectly consistent with the principle that “the specification may reveal a special definition given to a claim term by the patentee that differs from the meaning it would otherwise possess.” *Phillips*, 415 F.3d at 1316. The patentee acts as his own lexicographer when the patentee “‘clearly set[s] forth a definition of the disputed claim term,’ and ‘clearly express[es] an intent to define the term.’” *GE Lighting Sols., LLC v. AgiLight, Inc.*, 750 F.3d 1304, 1309 (Fed. Cir. 2014) (quoting *Thorner v. Sony Comput. Entm't Am. LLC*, 669 F.3d 1362, 1365 (Fed. Cir. 2012)). The parties agree that the Column 7 Characteristics form an explicit definition of “software agent” in the patent.

The parties disagree, however, on whether the definition incorporates additional “general characteristics” of software agents that appear elsewhere in the specification: “As will be appreciated various implementations of the agents are possible, however*, each of the agents will have the following general characteristics*[.]” (‘730 Patent, JA Ex. 1, col. 10 l. 31–col. 11 l. 11 (emphasis added) (“Column 10 Characteristics”).) F5 argues that these “general characteristics” are part of the explicit definition of “software agent” in the patent and must be part of the construction, while NetFuel claims that they are not part of the definition and should not be included in the construction.

The reference to “various implementations” identifies some embodiments of the invention. The dispute concerns whether “each of the agents” in the subsequent clause refers to each of the agents in those specific embodiments, or to each of the agents in the invention as a whole. If it is the former, these characteristics should not be part of the construction, because the court "do[es] not read limitations from the embodiments in the specification into the claims." *Hill-Rom Servs., Inc. v. Stryker Corp.*, 755 F.3d 1367, 1371 (Fed. Cir. 2014). If it is the latter, then the description of characteristics of each agent is part of the patentee's explicit definition of the term, and should be part of the construction. *See SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.*, 242 F.3d 1337, 1341 (Fed. Cir. 2001) ("Where the specification makes clear that the invention does not include a particular feature, that feature is deemed to be outside the reach of the claims of the patent, even though the language of the claims, read without reference to the specification, might be considered broad enough to encompass the feature in question."). In such a case, the patentee would have used the specification to further limit the term "software agent" by including the Column 10 Characteristics in an explicit definition.

Here, the latter interpretation is correct; "each of the agents" does not refer to the embodiments described in the preceding clause. The word "however" signals that the specification now refers to the invention as a whole, as opposed to the "various implementations" of the invention. The specification articulates that *each* software agent in the invention shall have the Column 10 Characteristics listed. The word "each" provides the definitional element; it declares that there is *no* software agent within the patented invention that does not have the listed characteristics.[16] If the patentee intended that the general characteristics refer only to the agents in the embodiments, it could have used different

[16] NetFuel seems to argue, for the first time at the hearing, that "each" referred to each type of agent, such as autonomous, intelligent, or mobile. (Claim Construction Hr'g Tr. 53:15–21.) Yet the text specifies "each agent" shall have the characteristics, not "each type of agent." The discussion of the distinction between different types of agents (autonomous or intelligent) appears several paragraphs earlier in the specification. ('730 Patent, JA Ex. 1, col. 7 l. 41–col. 8 l. 35.) In any event, even if "each agent" did mean "each type of agent," the result would still apply these characteristics to every agent, regardless of its classification as mobile, intelligent, or autonomous.

language, such as by removing the “however,” which signals a change in direction from the previous clause, or by specifying that the characteristics apply to “each of the agents in the embodiments” as opposed to “each of the agents.”

NetFuel effectively argues that “general characteristics” means “characteristics that apply generally, but not always” to software agents. (*See* Decl. of James D. Harlow, attached to NetFuel’s Mem. [98-9] ¶ 6.) These phrases, however, are not synonymous. The phrase “general characteristics” refers to characteristics that are not completely specific; for example, Column 10 Characteristic (b) is that “each agent assumes certain services . . . are available to it from the ARE” (‘730 Patent, JA Ex. 1, col. 10 ll. 38–42), but does not specify what service each agent has available. In this context, “general characteristics” are not characteristics that “apply generally, but not always;” to the contrary, the patent states that *each* software agent has the Column 10 Characteristics. If the Column 10 Characteristics applied generally, but not always, there would be no need to specify that *each* agent had those characteristics; such language expresses clear intent that the Column 10 Characteristics apply to all software agents. *See Aventis Pharma S.A. v. Hospira, Inc.*, 675 F.3d 1324, 1330 (Fed. Cir. 2012) (explaining that the clear intent to define a claim term or disavow its ordinary scope “may be inferred from clear limiting descriptions of the invention in the specification or prosecution history.”) Accordingly, those characteristics should be part of the construction.

This distinction is confirmed by the analogy that both parties made at oral argument. NetFuel argues that an automobile has the “general characteristics” of having side windows, four wheels, a steering wheel, and a crankshaft, but some electric cars, while automobiles, do not have a crankshaft. (Claim Construction Hr’g Tr. 47:18–48:6.) According to NetFuel, these “general” characteristics do not apply to all automobiles, just as the “general characteristics” in the patent do not apply to all software agents. As F5 points out, however, the structure of the patent is first to define the particular characteristics: in the case of the car, the patent would first indicate that it was a sports car and name particular characteristics, such as a small size, a convertible roof, and an advanced steering wheel. (*Id.* at 55:1–17.) After such particular characteristics are listed, a reference to general characteristics of each sports car takes on a different meaning: while each automobile under the sun might not have a crankshaft, every sports car covered by the patent *would* have such general characteristics. Each software agent covered by the patent in this case, similarly has the identified "general characteristics."

NetFuel cites to testimony from James Harlow, the inventor, that there are software agents that do not have one of the listed characteristics. In particular, NetFuel identifies mobile agents, a type of software agent that can migrate from machine to machine—yet Column 10 Characteristic (h) dictates that "each agent is assigned a named thread group upon creation" ('730 Patent, JA Ex. 1, col. 10 l. 66–67), a characteristic that, according to NetFuel, excludes migration.[17] (NetFuel Mem. 11.) As F5 points out, however, the software claimed by the patent need not be a mobile agent, and neither party has identified a claim that requires the software to be a mobile agent. (F5 Reply 5.) NetFuel has no response to this. True, "'[a] claim construction that excludes a preferred embodiment . . . is rarely, if ever, correct[,]'" *Pfizer, Inc. v. Teva Pharm., USA, Inc.*, 429 F.3d 1364, 1374 (Fed. Cir. 2005) (omission in original) (quoting *SanDisk Corp. v, Memorex Prods., Inc.*, 415 F.3d 1278, 1285 (Fed. Cir. 2005)), but NetFuel does not suggest that mobile agents are part of the invention's *preferred* embodiment.[18]

[17] F5 appears to contest the argument that mobile agents cannot possess this characteristic for the first time at oral argument. (Claim Construction Hr'g Tr. 44:24–45:16.) NetFuel disagrees. (*Id.* at 51:9–23). As explained further, the court need not resolve whether mobile agents can possess this characteristic.

[18] F5, ironically, asserts that Harlow does claim that "that one of the general characteristics is inconsistent with one of the preferred embodiments." (F5 Reply 5.) But the court can find no explicit reference to a preferred embodiment anywhere in the patent or in Harlow's testimony; the court finds only references to illustrative embodiments, but no indication that these embodiments are preferred. NetFuel made no reference to any "preferred" embodiments in its brief, nor did either party mention "preferred embodiments" at oral argument. One of the embodiments described may indeed be preferred, but the court declines to draw such a conclusion on its own. *See Honeywell Int'l, Inc. v. ITT Indus., Inc.*, 452 F.3d 1312, 1318 (Fed. Cir. 2006) (declining to characterize a description of the invention as a preferred embodiment when "the written description does not indicate that a fuel filter is merely a preferred embodiment of the claimed invention"); *cf.* Tom Brody, *Preferred Embodiments in*

Indeed, "[i]t is not necessary that each claim read on every embodiment." *Baran v. Med. Device Techs., Inc.*, 616 F.3d 1309, 1316 (Fed. Cir. 2010); *see Sinorgchem Co., Shandong v. Int'l Trade Comm'n*, 511 F.3d 1132, 1138 (Fed. Cir. 2007) ("Where . . . multiple embodiments are disclosed, we have previously interpreted claims to exclude embodiments where those embodiments are inconsistent with unambiguous language in the patent's specification or prosecution history."). That mobile agents would appear to not fall within the explicit definition set forth in the patent does not doom the construction consistent with that definition. Moreover, Harlow's testimony "is entitled to no deference." *Markman*, 52 F.3d at 983; *see Bell & Howell Document Mgmt. Prod. Co. v. Altek Sys.*, 132 F.3d 701, 706 (Fed. Cir. 1997) ("The testimony of an inventor often is a self-serving, after-the-fact attempt to state what should have been part of his or her patent application[.]") (citing *Markman*, 52 F.3d at 983). This evidence does not persuade the court to abandon the patent's explicit definition.

NetFuel also refers to the prosecution history, where the examiner rejected several claims as unpatentable over U.S. Patent Number 6,839,850 to Campbell. (JA Ex. 2 at 483.) The Campbell patent states that "the MANAGER process . . . clones a copy of itself for each Audit Agent," processes data with little input from the user, and forwards the data on. (Campbell Patent, col. 11 l. 65–col. 12 l. 9.) In particular, the examiner stated that the Campbell patent disclosed a software agent that had many of the Column 7 Characteristics, but the examiner made no mention of the Column 10 Characteristics. NetFuel also points out that when Harlow responded to the patent examiner's rejection, distinguishing the MANAGER process in Campbell from a software agent, he also did not mention the Column 10 Characteristics. (*See* JA Ex. 2 at 503.) NetFuel urges that because the Column 10 Characteristics were absent from Harlow's and the examiner's discussion of Campbell, the

Patents, 9 JOHN MARSHALL REV. INTELL. PROP. L. 398, 427, 436 (2010) (observing that the strong preference against a construction that excludes a preferred embodiment comes into effect when a patentee "label[s] an embodiment as preferred").

examiner must have concluded (as would a POSITA) that only the Column 7 Characteristics were part of the patent's definition of "software agent."[19] (NetFuel Mem. 10–11.)

This conclusion does not follow. NetFuel is correct that neither Harlow nor the examiner mentioned the Column 10 Characteristics with respect to the Campbell patent, but this does not alter the court's determination. NetFuel appears to assume that only those characteristics of software agents mentioned by the patent examiner can be part of the definition, but this cannot be true. After all, the examiner did not mention all of the Column 7 Characteristics, but NetFuel acknowledges that all those characteristics are part of the definition. In short, what the examiner highlighted as part of the definition of "software agent" does not necessarily exclude the Column 10 Characteristics.

Accordingly, the court adopts F5's construction of a software agent: "a virtual entity which: (a) is capable of acting in a runtime environment; (b) can communicate directly with other agents (messaging); (c) is driven by a set of tendencies (expressed in the form of individual objectives or of a satisfaction/survival function which it tries to optimize or policy); (d) possesses resources of its own (logic and algorithms); (e) is capable of perceiving its environment (state); (f) has only a partial representation of its environment; (g) is able to reproduce/clone itself, and (h) has the general characteristics described at col. 10, line 31 to col. 11, line 11 of the '730 patent."

F. Autonomous Agent

Claim Term	**NetFuel's Proposed Construction**	**F5's Proposed Construction**
Autonomous Agent	An agent with the ability to independently solve complex problems up to a certain level without human interaction.	An agent (a) with the capacity to anticipate future events and to prepare for them, (b) that is able to use a capacity for algorithmically induced reasoning based

[19] As noted above, the examiner was not initially persuaded by Harlow's argument distinguishing the MANAGER process from software agents, but this issue became moot when Harlow amended the claims to include an additional limitation.

		on representation of the environment to memorize situational parameters (data points), analyze them, and request additional algorithmic components (policy from the agent control mechanism), (c) that has a degree of sophistication which allows it to algorithmically interpret the state of its environment and to perform tasks, independently without human interaction, (d) that possesses representations of its environment and inference mechanisms that allow it to function independently of other agents, and (e) with the ability to independently solve complex problems up to a certain level without human interaction.[20]

As with “software agent,” the dispute over the construction of “autonomous agent” centers on whether to include certain characteristics listed in the specification in the construction. NetFuel claims that “autonomous agent” does not need construing, and that parts (a) through (d) of F5’s definition refer to limitations from exemplary embodiments, which should not be used to limit the term’s construction. F5 responds that the patent gives an express definition of “autonomous agent” and lists parts (a) through (d) of its construction as characteristics of all autonomous agents, not only some embodiments. The characteristics of the proposed construction appear in the patent as follows:

> According to embodiments of the invention, agents may be autonomous and/or intelligent. Both intelligent and autonomous agents **have a degree of sophistication which allows them to algorithmically interpret the state of**

[20] F5 modified its proposed construction to add (e) in its reply brief. (F5 Reply 6.)

> **their environment and to perform tasks independently without human interaction**. They **possess representation of their environment and inference mechanisms which allow them to function independently of other agents**. The difference between intelligent and autonomous agents lies in the scope of the representation of the environment that they possess, an intelligent agent having a greater representation of its environment than its autonomous agent. Accordingly an intelligent agent is able to solve complex problems without reference to outside entities whereas as [sic] an autonomous agent **is able to independently solve complex problems up to a certain level beyond which the autonomous agent will have to request further policy or data from an outside entity**
>
> . . . .
>
> A further attribute of intelligent and autonomous agents is that they **have the capacity to anticipate future events and to prepare for them**. These agents **are able to use a capacity for algorithmically induced reasoning based on representations of the environment to memorize situational parameters (data points), analyze them, and request additional algorithmic components (policy from the agent control mechanism).** In the event that there is a conflict in goals between these agents, these agents are able to negotiate among themselves to determine which goals/policy are more relevant to satisfy needs.

('730 Patent, JA Ex. 1, col 7 ll. 50–67, col. 8 ll. 26–35 (emphasis added).) The question here is whether the bolded characteristics are found in all autonomous agents, or alternatively whether certain embodiments have autonomous agents with these characteristics, but other autonomous agents lack these characteristics.

The court is not persuaded that the listed characteristics of autonomous agents are limited to autonomous agents in certain embodiments, as opposed to autonomous agents in the invention as a whole. The language of the specification itself is instructive: the specification first recites that some embodiments may have intelligent and/or autonomous agents. The specification then goes on to explain the implications of an embodiment having such an agent, rather than one of the other types of agents described elsewhere in the specification (such as simple or mobile agents). (*Id.* at col. 7 ll. 41–48). If only a subset of autonomous agents have these characteristics, there would be no need to characterize them as autonomous agents; instead they would simply be agents with these characteristics. If these characteristics do not apply to autonomous (and intelligent) agents generally, then the sentence before the list of characteristics ("[a]ccording to embodiments of the invention, agents may be autonomous and/or intelligent") would be unnecessary; there would be no need to label the agents as intelligent and/or autonomous if the characteristics were limited to the agents in those embodiments. Indeed, the source of the attributes identified in clauses (a) and (b) of F5's proposal is a paragraph that does not even mention the word "embodiment." Defining characteristics or attributes, such as those in the patent, constitute a definition. See *Define*, MERRIAM-WEBSTER COLLEGIATE DICTIONARY 327 (11th ed. 2003) (To "define" is "to determine or identify the essential qualities or meaning of.") That is the case here.

Having determined that this list of characteristics defines an autonomous agent, the court is also not persuaded by NetFuel's argument that only one such characteristic—the ability to independently solve complex problems up to a certain level without human interaction—is properly part of the construction. That single characteristic does distinguish an autonomous agent from an intelligent agent, but the specification makes clear that autonomous agents have other characteristics as well. That such characteristics are not unique to autonomous agents does not mean that they do not define autonomous agents. NetFuel urges that "the specification . . . teaches what intelligent and autonomous agents are" (NetFuel Mem. 13), and that is exactly the point: the specification teaches that these kinds of agents have these certain characteristics. NetFuel offers no reason that the specification limits autonomous agents to a single characteristic. A patentee is entitled to choose a broad term and employ its ordinary meaning, but that rule does not apply when the patentee explicitly defines the term. Because the list of attributes of an autonomous agent functions as a definition of "autonomous agent," and is not limited to one embodiment, the court adopts F5's construction.

The court recognizes that this is a contrary result to the one expressed at the claim construction hearing. The court expressed some frustration about the complicated nature of the parties' proposed claim construction and suggested the term did not need construing. (Claim Construction Hr'g Tr. 58:12–59:5.) The court acknowledged, however, that "defendant is correct that the language cannot be ignored, and to the extent that [the listed attributes of autonomous agents are] in the patent and the F5 product or products [do not] have those characteristics, then they're not infringing." (*Id.* at 62:6–10.) On the issue of infringement, F5 may argue that its products lack one of these characteristics. Had the court not construed the term, the question would turn on whether these characteristics are part of an embodiment or a definition. The court therefore determines that construction is appropriate and resolves it here.

G. Embodied In Hardware

Claim Term	**NetFuel's Proposed Construction**	**F5's Proposed Construction**
Embodied In Hardware	Fulfilled, performed, or carried out in hardware circuitry	Implemented in hardware circuitry

The next term, "embodied in hardware" appears several times in claim 7, as "agent embodied in hardware," "agent support mechanism embodied in hardware," and "modeler embodied in hardware." F5 argues that this term should be construed as "implemented in hardware circuitry," while NetFuel urges that the term need not be construed, but if it does, it means "fulfilled, performed, or carried out in hardware circuitry."

F5 contends that the specification supports its construction; in particular, F5 notes that the specification refers to two categories of embodiments:

> It will be apparent from this description [that] the aspects of the present invention **may be embodied, at least partly, in software**. In **other embodiments, hardware circuitry may be used in combination with software instructions** to implement the present invention. Thus, the techniques are not limited to any specific combination of hardware circuitry and software.

('730 Patent, JA Ex. 1, col. 29 ll. 39-44 (emphasis added).) As F5 reads this language, it "draws a distinction between hardware and software, acknowledging that some aspects of the invention may be implemented by hardware circuitry, rather than embodied in software." (F5 Mem. 17.)

In the court's view, these two embodiments do not shed any light on what it means to be "embodied in hardware." One potential embodiment is "at least partly" embodied in software, but that does not necessarily mean that the embodiment is *exclusively* embodied in software. The second embodiment contemplates hardware that "may" be used in conjunction with software. It is not clear from the language whether these embodiments can occur exclusively in software, or in some combination of both hardware and software, but nothing about the language suggests that embodiments are limited to hardware alone. In short, this language gives no direction as to what "embodied in hardware" means.

F5 points to another part of the specification, as well, which F5 claims also distinguishes between software and hardware: "computer executable instructions may be written in a computer programming language or may be embodied in firmware logic." ('730 Patent, JA Ex. 1, col. 29 ll. 61–63.) According to F5, this language shows that when the patentee wanted to specify that something was "embodied" in *software*, it did so explicitly, and "embodied in hardware" must therefore not involve software.

Again, the court does not find this language particularly illuminating. F5's construction still leaves uncertainty about what "embodied" means, and offers no real guidance on whether "implemented" or "fulfilled, performed, or carried out" is an appropriate definition. Moreover, the context that F5 emphasizes uses "written" in programming language to mean something like "implement using programming language," but even if "embodied in software (or firmware)"[21] is understood by a POSITA, it is not clear that the POSITA would also understand how something can be embodied in *hardware*. That the patent can make use of software and firmware in this way does not explain what "embodied in hardware" means.

NetFuel relies on the prosecution history to support its proposed construction. The examiner rejected an earlier version of claim 7 on the grounds that the software agent, agent support mechanism, and modeler could be "implemented as software," and therefore "the

[21] Firmware is also a set of instructions, *see Firmware*, McGraw-Hill Dictionary of Scientific and Technical Terms 806 (6th ed., 2003) (Firmware is "[a] computer program or instruction, such as a microprogram, used so often it is stored in a read-only memory instead of being included in software."), and to "implement" instructions is something that can be understood by a person skilled in the art; "hardware" on the other hand, is not a set of instructions, so "implementation" or "embodiment" of hardware may mean something different.

claims can cover software per se and software per se does not fit into a statutory category."[22] (JA Ex. 2 at 623.) NetFuel overcame this defect by simply reciting that the agent, agent support mechanism, and modeler "are embodied by hardware."[23] (*Id.* at 636.) NetFuel urges that this change merely shows that "embodied in hardware" does not mean "software per se." (NetFuel Mem. 15.) More particularly, NetFuel urges, "the patent office and patentee merely meant for [the] agent, agent support mechanism, and modeler to not be software ***only***, but software running on a processor or hardware environment[.]" (*Id.* (emphasis in original).)

The court recognizes that this term must have some meaning sufficient to overcome the rejection during prosecution history: "Claims may not be construed one way in order to obtain their allowance and in a different way against accused infringers." *Southwall Techs., Inc. v. Cardinal IG Co.*, 54 F.3d 1570, 1576 (Fed. Cir. 1995). But, lacking programming experience, the court is uncertain what guidance, if any, the phrase provides to a POSITA. It appears that the patentee added "embodied in hardware" simply to explain that the software must run on some machine and to avoid the invention's being classified as "software per se," considered by the PTO to be an unpatentable abstract idea. Adding such language to avoid unpatentability appears to be common practice. *See CMG Fin. Servs., Inc. v. Pac. Trust Bank, F.S.B.*, 50 F. Supp. 3d 1306, 1318 (C.D. Cal. 2014), *aff'd sub nom. CMG Fin. Servs., Inc. v. Pac. Trust Bank*, 616 F. App'x 420 (Fed. Cir. 2015) ("The Board also explained that computer instructions, or "software per se" were necessarily only an abstract idea where they were not explicitly tied to any particular machine."); *Ex Parte George Henry Forman & Henri Jacques Suermondt*, No. 2007-1546, 2007 WL 4480714, at *3 (B.P.A.I. Dec. 21, 2007) (explaining "how one may avoid having claims interpreted as directed to software per se" by requiring interaction between "'the

[22] NetFuel cites *DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1257 (Fed. Cir. 2014) to show that the claimed invention is not an abstract idea and is patent-eligible, but such arguments are not properly addressed at claim construction.

[23] The patent uses the phrase "embodied in hardware." When Harlow amended the claims, however, he told the patent examiner that the amendment reflected that the invention was "embodied *by* hardware."

computer program and the rest of the computer'") (quoting *Manual of Patent Examining Procedure (MPEP)* § 2106.01(I), p. 2100-18 (8th Ed., Rev. 6, Sept. 2007)); Ebby Abraham, *Bilski v. Kappos: Sideline Analysis from the First Inning of Play*, 26 BERKELEY TECH. L.J. 15, 45 (2011) ("[T]he BPAI has yet to find a software claim patent eligible when the claim does not limit the code on a computer readable medium[.]").

Whatever the merit of this practice, if "embodied in hardware" was added to distinguish the invention from software per se, this does not satisfy the court that it should adopt NetFuel's construction. That a software program cannot exist in a vacuum is something that is well understood and does not require claim construction to confirm. More importantly, NetFuel acknowledges that the verbs it uses in its proposed construction—fulfill, perform, and carry out—themselves define F5's proposed verb, implement. (NetFuel Mem. 16.) The parties' meanings, therefore, are effectively identical. Implement, fulfill, perform, and carry out add no more clarity than "embody." Claim construction "is not an obligatory exercise in redundancy." *U.S. Surgical Corp. v. Ethicon, Inc.*, 103 F.3d 1554, 1568 (Fed. Cir. 1997). Because these are synonyms, the court would accomplish nothing by preferring one over the other, and concludes that this term does not need construction.

H. Clone

Claim Term	**NetFuel's Proposed Construction**	**F5's Proposed Construction**
Clone	Reproduce	Create a replica of

The disputed term "clone" appears in claims 7 and 30 of the '730 patent, both of which refer to a software agent that is "able to clone itself." F5 urges a construction of this term as "create a replica of," while NetFuel would have the court construe the term as "reproduce." F5 urges that its construction is commensurate with the term's ordinary meaning. NetFuel argues that the "replica" construction is improperly limiting, and that F5 has not shown that the patentee clearly intended to define the term that way, or to disavow the term's ordinary full scope.

The requirement that a party offer evidence of such clear intent applies, however, only when the party wants to interpret a claim more narrowly than its ordinary meaning to a skilled artisan. *See Aventis Pharma S.A.*, 675 F.3d at 1330. In this case, the court concludes it is F5's construction that adheres to the ordinary meaning of "clone." As the court noted at the claim construction hearing, "clone" means to make an exact copy or a replica of something; the word appears to have come into common usage precisely to distinguish cloning from ordinary reproduction. (Claim Construction Hr'g Tr. 71:24–72:8.) NetFuel's construction inappropriately broadens the term's ordinary meaning.

NetFuel insists that "reproduce" is the ordinary meaning of "clone," and notes, in support, that F5's own dictionary defines clone as "to produce a copy of; imitate" as opposed to "produce a replica." (NetFuel Mem. 16–17); *Clone*, THE AMERICAN HERITAGE COLLEGE DICTIONARY 263 (3d ed. 1997), Ex. C to F5 Mem. [96-3]. But of course, "replica" is a type of copy, specifically, a precise copy. *Replica*, MERRIAM-WEBSTER COLLEGIATE DICTIONARY 1056 (11th ed. 2003) (defining "replica" as "a copy exact in all details"). "Reproduce," on the other hand, connotes a copy, but not necessarily an exact copy. *Reproduce*, MERRIAM-WEBSTER COLLEGIATE DICTIONARY 1057 (11th ed. 2003) (defining "reproduce" as "to imitate *closely*") (emphasis added). NetFuel's proposed construction (and indeed, F5's dictionary definition) leaves uncertainty about whether a close imitation is sufficient for an agent to have cloned itself. The court construes this term as "to create an exact copy of," consistent with its ordinary meaning. This more precisely captures the meaning of "clone" than "copy" or "reproduce." To the extent that dictionary definitions are inconsistent with the court's construction, the court notes that it is inappropriate to "elevat[e] the dictionary to such prominence is that it focuses the inquiry on the abstract meaning of words rather than on the meaning of claim terms within the context of the patent." *Phillips*, 415 F.3d at 1321.

In the context of this patent, "clone" does not mean "reproduce" or simply "copy." In arguing against this conclusion, NetFuel cites testimony of James Harlow that the specification's use of “reproduce/clone” tells a POSITA what he or she needs to know. (NetFuel Mem. 17.) Respectfully, this assertion does not constitute evidence that “reproduce” and “clone” are synonymous terms. The specification itself uses two different words, and the court assumes that different words have different meanings. *Cf. Nystrom*, 424 F.3d at 1143. And the inventor’s testimony is itself extrinsic evidence, which is afforded no deference in claims construction issues. Nor will the court honor extrinsic evidence that contradicts the plain and ordinary meaning, or the intrinsic evidence in the patent itself. *See Phillips*, 415 F.3d at 1318 (characterizing extrinsic evidence as “less reliable”); *see also Indus. Tech. Research Inst. v. Int'l Trade Comm'n*, 567 F. App'x 914, 917–18 (Fed. Cir. 2014) (“If, and only if, the intrinsic evidence does not establish the meaning of a claim, we can turn to the extrinsic evidence[.]”). The court construes “clone” as “to create an exact copy of.”

I. Modeler Embodied in Hardware to Create Test Policy and to Model a Behavior of the Computer Network Based on the Test Policy Thereby to Determine an Optimal Policy for the Computer Network Said Modeler Comprising a Predictive Algorithm to Predict a Failure of a Network Component; Wherein the Modeler Determines Appropriate Policy Based on the Prediction (“Modeler . . .”)

Claim Term	**NetFuel’s Proposed Construction**	**F5’s Proposed Construction**
Modeler . . .	An entity that obtains information regarding the state of each agent or other modeler.	112(6) Means-plus-function Term The function(s) of this element are as follows: (1) creating test policy and modeling a behavior of the computer network based on the test policy thereby to determine an optimal policy for the computer network, (2) predicting a failure of a network component, (3) determining appropriate policy based on the prediction. The structure of this term is indefinite.

F5 contends that “modeler . . .” is a means-plus-function term under 35 U.S.C. § 112(f). Under that section, a claim may be expressed “as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof.” 35 U.S.C. § 112. Unlike other claim terms, means-plus-function terms are “construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.” *Id.* The threshold question is whether the term “modeler . . .” is a means-plus-function term and subject to § 112(f). If it is, construction of the term is a two-step process. *Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1351 (Fed. Cir. 2015). First, the court must “identify the claimed function.” *Id.* Second, the court determines “what structure, if any, disclosed in the specification corresponds to the claimed function.” *Id.* “Where there are multiple claimed functions . . . the patentee must disclose adequate corresponding structure to perform all of the claimed functions.” *Id.* at 1351–52. “If the patentee fails to disclose adequate corresponding structure, the claim is indefinite.” *Id.* at 1352.

To answer the threshold question and determine whether “modeler . . .” is a means-plus-function term, “the essential inquiry is not merely the presence or absence of the word ‘means.’” *Id.* at 1348. When a claim term lacks the word “means,” as in this case, the court will exercise a presumption that it is not a means-plus-function term, but “the presumption can be overcome and [§ 112(f)] will apply if the challenger demonstrates that the claim term fails to ‘recite[] sufficiently definite structure’ or else recites ‘function without reciting sufficient structure for performing that function.’” *Id.* at 1349 (alteration in original) (quoting *Watts v. XL Sys., Inc.*, 232 F.3d 877, 880 (Fed. Cir. 2000)).

The court notes that the question of whether the presumption is overcome and step two of the means-plus-function inquiry appear somewhat similar. So far as the court can discern, the difference lies in whether the court looks for a corresponding structure in the patent claim, or in the entire specification. In evaluating whether the presumption is overcome, the court looks only to the claim term itself, without looking to the specification, to determine if it is a means-plus-function term. *Cf. Williamson*, 792 F.3d at 1349–1351. If the structure in the claim is a nonce word similar to "means" (for example, "mechanism" or "device"), then the claim term itself is a means-plus-function term (by contrast, if the structure is definite, like "hammer," then the term is not a means-plus-function term). *See id.* at 1350–51. Once the presumption is overcome, only then does the court look to the entire specification to find the corresponding structure—the details about the structure that accomplishes the relevant function.[24] *Cf. id.* at 1349–52.

The "modeler . . ." term lacks the word "means," but F5 urges that the presumption should be overcome, contending that "modeler" is analogous to "means." In further support of this argument, F5 contends that "modeler" does not recite sufficiently definite structure because it does not have a reasonably well understood meaning in the art. Courts may can inquire "into whether the 'term, as the name for structure, has a reasonably well understood meaning in the art'" to determine whether it recites a sufficiently definite structure. *Watts*, 232 F.3d at 880–81 (quoting *Greenberg v Ethicon Endo-Surgery, Inc.*, 91 F.3d 1580, 1583 (1996)). F5's support for its contention that "modeler . . ." is a means-plus-function term is that the inventor of the patent testified at his deposition that no "such modelers" existed in the prior art. (F5 Mem. 20–21.)

NetFuel responds that "modeler" does recite sufficient structure because a POSITA would understand the structure recited by "modeler." (NetFuel Mem. 20.) In support, NetFuel cites to the declaration of Nancy Miracle, a person with "expertise in the areas of system design

[24] The court is uncertain whether F5 itself conflates these two questions. It complains that "there is no algorithm provided in the claim language to impart structure in the modeler" (F5 Reply 11), but it is unclear whether this portion of its argument addresses the threshold question (whether the presumption is overcome) or step two of the means-plus-function inquiry. Yet courts have applied the requirement that the patent recite an algorithm at step two of the analysis. *See Williamson,* 792 F.3d at 1352 (proceeding to step two after concluding that the claim term was a means-plus-function term and identifying the function: "[w]e require that the *specification* disclose an algorithm for performing the claimed function") (emphasis added); *Net MoneyIN, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1367 (Fed. Cir. 2008) (addressing the algorithm requirement in step two: "a means-plus-function claim element for which the only disclosed structure is a general purpose computer is invalid if the *specification* fails to disclose an algorithm for performing the claimed function") (emphasis added). Because this is a step two question, as explained below, the court need not reach this argument.

and intellectual property analysis with a focus on software and hardware product design," who asserts that a POSITA "would have understood what was described by the term 'modeler.'" (Decl. of Nancy Miracle, attached to NetFuel Mem. [98-8] ("Miracle Decl.") ¶¶ 2, 5.) Miracle observed that only the claimed modeler was novel, not that modelers in general are novel. (*Cf. id.* at ¶¶ 6–7.)

"[T]he fact that a particular mechanism . . . is defined in functional terms is not sufficient to convert a claim element containing that term into a 'means for performing a specified function[.]'" *Greenberg*, 91 F.3d at 1583. "Many devices take their names from the functions they perform . . . . such as 'filter,' 'brake,' 'clamp,' 'screwdriver,' or 'lock.'" *Id.* Here, the fact that a "modeler" could also be expressed as a "means for modeling" does not resolve the question; a "lock" could also be described as "means for locking," but a claim reciting a lock that performs a certain function would not be a means-plus-function element.

That the purported means for performing the functions listed is a "modeler" perhaps does recite less structure than a "lock," but it certainly suggests more structure than "means" or "mechanism." "Modeler" signals that, at its core, the structure involves creating models. A similar term was at issue in *Personalized Media Communications, LLC v. InternationalTrade Commission*, 161 F.3d 696, 704 (Fed. Cir. 1998), where the Federal Circuit concluded that "'[d]etector' is not a generic structural term such as 'means,' 'element,' or 'device'; nor is it a coined term lacking a clear meaning, such as 'widget' or 'ram-a-fram.'" To put it another way, if the term recited "a means to create test policy," that would communicate significantly less information and recite significantly less structure than "a modeler embodied in hardware to create test policy." A POSITA would know that it creates models, and, like the words "detector" or "lock," "modeler" expresses a function, but is more definite than a means-plus-function term. While "modeler" may not be so definite as to communicate exactly what or how it models, this does not render the structure indefinite: by its term alone, "lock" does not necessary communicate what or how it locks, but again, "lock" would not recite a means-plus-function element. The court concludes that "modeler" has a reasonably well understood meaning to a person with skill in the art.[25]

The court further notes that the fact that the "modeler" element performs certain objectives does not automatically convert that element into a means-plus-function element. Notably, F5 does not contend that the element that appears in claim 7 immediately before this element—"an agent support mechanism embodied in hardware to provide support to the agent"—is a means-plus-function element. Nor should it; a POSITA knows what "agent support mechanism" means. The point, however, illustrates that the mere fact that the element is defined in functional terms does not render it a means-plus-function element.[26]

Nor is the court's conclusion undermined by the testimony of the inventor, Harlow, that "such modelers" did not exist in the prior art. In the cited testimony, Harlow testified that "a modeler embodied in hardware to create a test policy and to model a behavior of the computer network based on the test policy to determine an optimal policy for the computer network" was not in the prior art. (Deposition of James D. Harlow, Ex. F to F5 Mem. ("Harlow Dep.") [96-6] 114:23–115:7.) Yet the entire line of questioning makes clear that while the modelers "as characterized [in the patent]" were new (Harlow Dep. 115:17–19), models in general had existed in the prior art:

[25] NetFuel contends that Miracle's declaration establishes that a POSITA would have understood that "modeler" recited sufficiently definite structure. Miracle declares that "models" that can improve device functionality have been around for decades. (Miracle Decl. ¶ 6.) This argument is not particularly persuasive on its own; that models have been around for decades does not clarify whether a POSITA would have understood the concept of a "modeler"—the thing that creates those models. Yet as Miracle observes, the specification describes the modelers' decision-making process and hierarchy. ('730 Patent, JA Ex. 1, col. 20 l. 23–col. 21 l. 59.) Miracle's opinion thus does reinforce the court's conclusion that a POSITA would be able to implement these steps and understand considerably more structure from "modeler" than from "means."

[26] The court recognizes that the specification uses "module" in place of "modeler" in some places, but this, again, does not require the conclusion that "modeler" is a means-plus-function term. *See Greenberg*, 91 F.3d at 1583 ("[W]e do not agree with the district court that the term "detent mechanism" in the '501 patent should be treated as synonymous with the term 'detent means' simply because the patent uses the term 'detent means' in place of 'detent mechanism' on two occasions in the 'summary of the invention' portion of the specification.").

> Q. . . . .
> [R]ight now you'd agree that there were, for example, econometric models in computer software in the prior art?
>
> . . . .
>
> A. Yes.

(Harlow Dep. 118:10–19.) The claimed modelers have some uniqueness over the prior art, but this is not surprising; indeed, it is expected given the Patent Act's requirements of novelty and nonobviousness. *See* 35 U.S.C. §§ 102, 103. Harlow's testimony confirms that the particular claimed modelers are not in the prior art, but that does not mean that a POSITA would be unable to implement the modeler after reading the specification, which describes the modelers' decision-making process and hierarchy. ('730 Patent, JA Ex. 1, col. 20 l. 23–col. 21 l. 59.) That such modelers were not in the prior art does not, by itself, indicate indefinite structure.

This question of precisely how definite a structure must be in order for the claim not to be a means-plus-function element has plagued district courts. This was true even before the Federal Circuit adopted the now-abandoned "strong presumption" in *Lighting World, Inc. v. Birchwood Lighting, Inc.*, 382 F.3d 1354, 1358 (Fed. Cir. 2004) *overruled by Williamson*, 792 F.3d at 1349. *Compare Aguayo v. Universal Instruments Corp.*, Civil Action No. H-02-1747, 2003 WL 25787593, at *4 (S.D. Tex. June 9, 2003) (finding that "indicator" recited sufficiently definite structure) *with Nilssen v. Motorola, Inc.*, 80 F. Supp. 2d 921, 933 (N.D. Ill.), *opinion modified and supplemented*, 130 F. Supp. 2d 976 (N.D. Ill. 2000) (concluding that "input" did not recite sufficiently definite structure). On close questions, the court takes guidance from the presumption. But F5 has not shown that the element does not recite sufficiently definite structure: "modeler" is, like "detector," more definite than "means" or "mechanism." Although it does not recite quite as much structure as "lock," this does not overcome the presumption. The court concludes that "modeler . . ." is not a means-plus-function element.

Though NetFuel proposes a construction, it apparently offers it only in the event that the court construes the term; it primarily argues that "[t]he term has a plain and ordinary meaning." (NetFuel Mem. 19.) Because the term is not a means-plus-function element, the court sees no reason to convert a string of over fifty words into a single term—the words employ their individual plain and ordinary meanings.

J. Computer Network

Claim Term	NetFuel's Proposed Construction	F5's Proposed Construction
Computer Network	No construction is necessary, but if the court construes the term, NetFuel's proposed construction is: A group of one or more connected computers	A group of two or more connected computers (The preamble of claim 7 is limiting)

In its proposed construction, F5 asks the court both to construe the term "computer network" as two or more computers and to hold that the preamble to claim 7 is limiting. The term "computer network" appears in both the preamble and the body; the court assumes such a limitation means that the "computer network" in the body of the claim would be the same "computer network" referred to in the preamble. NetFuel disputes both F5's proposed construction—instead, NetFuel claims that a computer network may have only one computer—and disputes F5's contention that the preamble is limiting.[27]

The preamble to claim 7 simply recites "a computer network, comprising[,]" and the body of the claim follows. "[A] preamble limits the invention if it recites essential structure or steps, or if it is 'necessary to give life, meaning, and vitality' to the claim." *Catalina Mktg. Int'l, Inc. v. Coolsavings.com, Inc.*, 289 F.3d 801, 808 (Fed. Cir. 2002) (quoting *Pitney Bowes Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1305 (Fed. Cir. 1999)). A preamble that "states a necessary and defining aspect of the invention" limits the claim, *see On Demand Mach. Corp. v.*

[27] NetFuel contends that "computer network" appears only in the preamble to the claims. (NetFuel Mem. 23.) This is not correct. "Computer network" appears in the body of claim 7: "[W]herein the software agent . . . is able to communicate with other software agents in the computer network." ('730 Patent, JA Ex. 1, col. 35 ll. 25–29.) NetFuel's contention that the claim appears only in the preamble cites to claim 1, which is not an asserted claim here, and more importantly, *does* include "computer network" in the body of the claim. (*Id.* at col. 34 l. 55–col. 35 l. 8.) As F5 points out, "computer network" appears in the body of *each* of the asserted independent claims. (F5 Reply 14; '730 Patent, JA Ex. 1, col. 35 l. 22–col. 36 l. 64.)

Ingram Indus., Inc., 442 F.3d 1331, 1343 (Fed. Cir. 2006), but a preamble is not a limitation if it "is simply an introduction to the general field of the claim," *see id.*, or "only [states] a purpose or intended use for the invention," *Rowe v. Dror*, 112 F.3d 473, 478 (Fed. Cir. 1997). *Catalina Marketing* sets forth several "guideposts" for determining whether a preamble limits the claim scope. 289 F.3d at 808. A preamble may limit a claim: (1) under *Ex parte Jepson*,1917 Dec. Comm'r Pat. 62,[28] when the preamble defines the claimed invention, (2) where other parts of the patent depend on the preamble for an antecedent basis, (3) when "the preamble is essential to understand limitations or terms in the claim body," (4) when the preamble "recit[es] additional structure or steps underscored as important by the specification," and (5) where the inventor relies "on the preamble during prosecution to distinguish the claimed invention from the prior art[.]" *Id.*

NetFuel characterizes this list of guideposts as a bright-line test, and concludes that a preamble is only limiting if all of the guideposts are present (though NetFuel only identifies three guideposts). Because the preamble does not recite additional steps and the inventor did not rely on it to distinguish prior art, NetFuel concludes, the preamble fails this test and is not limiting. But the court does not understand the guideposts as a bright-line test. For one thing, the *Catalina Marketing* court explicitly stated, immediately before reciting the guideposts, that "[n]o litmus test defines when a preamble limits claim scope." *Id.* For another, the guideposts themselves are loaded with conditional language: "*Jepson* claiming *generally* indicates intent . . . " and "dependence on a particular disputed preamble phrase for antecedent basis *may* limit claim scope . . . ." *Id.* (emphasis added).

Furthermore, the Federal Circuit has not treated the guideposts as a test where all factors must be present. For example, in *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 953 (Fed. Cir. 2006), the Federal Circuit concluded that the preamble was limiting after observing that

[28] *Jepson* claiming is a type of claim that "allows a patentee to use the preamble to recite 'elements or steps of the claimed invention which are conventional or known.'" *Rowe*, 112 F.3d at 479 (quoting 37 C.F.R. § 1.75(e) (1996)).

other references to the disputed term derived their antecedent basis from the preamble, and that the preamble recited additional structure, but without mentioning other guideposts. Similarly, in *Pacing Techs., LLC v. Garmin International, Inc.*, 778 F.3d 1021, 1024 (Fed. Cir. 2015), the court found a preamble to be limiting after observing that the preamble provided an antecedent for the term in another claim, with no discussion of any other guideposts. Thus, a court need not conclude that all the guideposts are present to find a preamble to be limiting.

NetFuel has no other argument that the preamble is not limiting, which is unsurprising, because it is clear that "computer network" in the preamble of claim 7 does provide the antecedent basis for several references to "computer network" in the asserted claims. Though the preamble of claim 7 refers to "a computer network," the body of the claim refers to "the computer network." The use of a definite article, rather than the indefinite article of the preamble, shows that the body of the claim refers to a specific computer network, and the only logical computer network it could signify is the network referred to in the preamble. The computer network in the body of the claim, therefore, derives its antecedent basis from the preamble. Further, Claims 10, 11, 16, 21, and 26 all refer to "the computer network of claim 7;" claims 17 and 19, in turn, refer to the "computer network of claim 16," while claim 18 refers to "the computer network of claim 17." ('730 Patent, JA Ex. 1, col. 35 l. 51–col. 36 l. 30.) The "computer network" in the preamble of claim 7 forms the antecedent basis for the computer network in claims 7, 10, 11, 16, 17, 18, 19, 21, and 26. This preamble clearly "helps to define the claimed invention," *see In re Cruciferous Sprout Litig.*, 301 F.3d 1343, 1347 (Fed. Cir. 2002), and the repeated references in other claims show that the preamble "states a necessary and defining aspect of the invention." *See On Demand Mach.*, 442 F.3d at 1343. In short, the preamble is limiting.

The question then is whether a single computer may constitute a network. F5 points to several computer dictionaries that define "network" as multiple computers or reference computers and devices in plural. (F5 Mem. 22.) F5 also points to parts of the specification that describe a network as encompassing *devices* (in plural) or refer to *a device* of a network (implying that there is more than one such device). Furthermore, the court refers back to the definition of "network" in the 2002 Microsoft Computer Dictionary (which F5 relied on earlier for its proposed construction of "network component"), which defines "network" as "[a] group of computers and associated devices that are connected by communications facilities." *Network*, MICROSOFT COMPUTER DICTIONARY 362 (5th ed. 2002), EX. A to F5 Mem. Not only does this definition refer to computers and devices in the plural, it also requires that such devices be connected. The court cannot envision how a single computer, unconnected to any other device, can form a network.

NetFuel's only response, first articulated at oral argument, is that multiple chips or cores may be networked together to form a computer network within a physical box, unit, or structure. (Claim Construction Hr'g Tr. 88:9–16; 90:8–12.) This argument seems premised on the characterization of a chip or core (or CPU or processor) as a single computer. NetFuel provides no support for this characterization; as F5 pointed out at argument, a laptop often contains multiple chips and processors, but is generally referred to as a single "computer." (*Id.* at 92:21–93:2.)

NetFuel further argues that multiple computers could be housed in a single structure and connected. (*Id.* at 90:8–12.) The court has no basis to conclude that such a structure would *not* simply be referred to as multiple computers within one structure. True, this understanding essentially depends on the definition of "computer"—multiple chips may be housed in one laptop as a single computer, but perhaps a single sufficiently large structure could house multiple computers—but NetFuel has presented no evidence concerning this distinction, or as support for its construction at all. F5's dictionaries refer to a network as "computers" in plural, and NetFuel provides no counter evidence that a network is instead a group of connected chips, cores, or processors, nor has NetFuel asked the court to construe "computer." The court adopts F5's construction.

K. Kill

The parties have agreed to construe the term “kill” as “to terminate a process.” (F5 Mem. 23, NetFuel Mem. 24).

CONCLUSION

The claim terms in the ‘730 patent are construed as follows:

Claim term	**Construction**
“network component”	hardware or software that is a constituent part of a computer network
“failure”	the inability of a network component to operate reliably or to operate at all
“predict/predicting”	determine/determining that something will or might happen in the future
“software agent”	a virtual entity which: (a) is capable of acting in a runtime environment; (b) can communicate directly with other agents (messaging); (c) is driven by a set of tendencies (expressed in the form of individual objectives or of a satisfaction/survival function which it tries to optimize or policy); (d) possesses resources of its own (logic and algorithms); (e) is capable of perceiving its environment (state); (f) has only a partial representation of its environment; (g) is able to reproduce/clone itself, and (h) has the general characteristics described at col. 10, line 31 to col. 11, line 11 of the ‘730 patent
“autonomous agent”	an agent (a) with the capacity to anticipate future events and to prepare for them, (b) that is able to use a capacity for algorithmically induced reasoning based on representation of the environment to memorize situational parameters (data points), analyze them, and request additional algorithmic components (policy from the agent control mechanism), (c) that has a degree of sophistication which allows it to algorithmically interpret the state of its environment and to perform tasks, independently without human interaction, (d) that possesses representations of its environment and inference mechanisms that allow it to function independently of other agents, and (e) with the ability to independently solve complex problems up to a certain level without human interaction
“embodied in hardware”	plain and ordinary meaning
“clone”	to create an exact copy of
“modeler embodied in hardware to create test policy and to model a behavior of the computer network based on the test policy thereby to determine an optimal policy for the computer network said modeler	plain and ordinary meaning

comprising a predictive algorithm to predict a failure of a network component; wherein the modeler determines appropriate policy based on the prediction”	
“computer network”	a group of two or more connected computers (The preamble of claim 7 is limiting)
“kill”	to terminate a process

ENTER:

Dated: June 29, 2017

REBECCA R. PALLMEYER
United States District Judge

Appendix

The disputed terms are emphasized in the claims below.

7. A **computer network**, comprising:

a **software agent** having an assigned goal which is a programmatic expression of a predefined task for the **software agent embodied in hardware**; wherein the **software agent** has its own runtime environment; is able to communicate with other **software agents** in the **computer network**; is capable of perceiving its own state; and is able to **clone** itself;

an agent support mechanism **embodied in hardware** to provide support to the agent;

a modeler ***embodied in hardware*** *to create test policy and to model a behavior of the* ***computer network*** *based on the test policy thereby to determine an optimal policy for the* ***computer network*** *said modeler comprising a predictive algorithm to* ***predict*** *a* ***failure*** *of a* ***network component****; wherein the modeler determines appropriate policy based on the prediction*;[29] and

a network control mechanism to dynamically modify the assigned goal of the **software agent** by replacing the assigned goal based on the optimal policy; wherein the **software agent** comprises an **autonomous agent** operable to request further policy when it lacks an ability to perform the predefined task.

10. The **computer network** of claim 7, wherein the **software agent** comprises a monitoring agent having the assigned goal to monitor an operation characteristic of the network.

11. The **computer network** of claim 7 . . . .

16. The **computer network** of claim 7, wherein the agent support mechanism comprises an agent runtime environment configured for a particular **network component**.

17. The **computer network** of claim 16, wherein the agent runtime environment controls and operation of the **software agent** . . . .

18. The **computer network** of claim 17, wherein the operation is selected form the group comprising of spawn, **kill** and suspend.

19. The **computer network** of claim 16 . . . .

21. The **computer network** of claim 7 . . . .

26. The **computer network** of claim 7 . . . .

[29] Italicization indicates the alleged means-plus-function term, which incorporates several other disputed terms.

30. A machine-readable storage medium that provides instructions which when executed by a processor causes the processor to perform a method comprising:

assigning a goal to a **software agent**; wherein the **software agent** has its own runtime environment; is able to communicate with other **software agents** in the **computer network**; is capable of perceiving its own state; and is able to **clone** itself; and comprises an **autonomous agent** operable to request further policy when it lacks and ability to perform the predefined task, and wherein the goal is a programmatic expression of a redefined task for the **software agent**; creating test policy and modeling a behavior of the **computer network** based on the test policy to determine an optimal policy for the **computer network**, including **predicting** a **failure** of a **network component** based on a predictive algorithm; wherein said modeling comprises determining appropriate policy based on the prediction; and

dynamically modifying the assigned goal of the **software agent** according to a desired operational characteristic of the **computer network** by replacing the assigned goal based on the optimal policy.

(*Id.* at col. 34 l. 54–col. 38 l. 11.)