

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
TYLER DIVISION**

**BEDROCK COMPUTER,
TECHNOLOGIES, LLC**

v.

**SOFTLAYER TECHNOLOGIES,
INC., ET AL.**

§
§
§
§
§
§
§

No. 6:09cv269 LED-JDL

JURY DEMANDED

MEMORANDUM OPINION AND ORDER

This claim construction opinion construes the disputed terms in U.S. Patent No. 5,893,120 (“the ‘120 patent”). Plaintiff Bedrock Computer Technologies, LLC (“Bedrock”) alleges Softlayer Technologies, Inc., Citiware Technology Solutions, LLC, Google, Inc., Yahoo! Inc., MySpace Inc., Amazon.com Inc., Match.com, Inc., and AOL Inc.¹ (collectively, “Defendants”) infringe the ‘120 patent. The parties have presented their claim construction positions (Doc. Nos. 275, 284, & 300). On October 7, 2010, the Court held a claim construction hearing and heard further argument. The Court issued a provisional claim construction order on October 29, 2010. (Doc. No. 326). For the reasons stated herein, the Court adopts the constructions set forth below.

OVERVIEW OF THE PATENT

The ‘120 patent is related to an “on-the-fly” garbage collection procedure designed to remove expired data from information storage and retrieval systems while “other types of access to the storage space are taking place.” *See* ‘120 patent at 1:1-5; 1:21-23; 2:56-57. In particular, the system stores information using a hashing technique in the form of external chaining. *See* ‘120 patent at

¹ Defendants CME Group Inc. and PayPal, Inc. settled earlier in the litigation. *See* (Doc. Nos. 168 and 202).

2:60.

The goal of the '120 patent is to efficiently remove expired records so as to reclaim storage and speedily access data. '120 patent at 2:19-21. In order to do so, the disclosed method accesses the linked list of data and removes expired records during normal search procedures, as illustrated in Figure 3. '120 patent at 2:57-63 and Fig. 3.

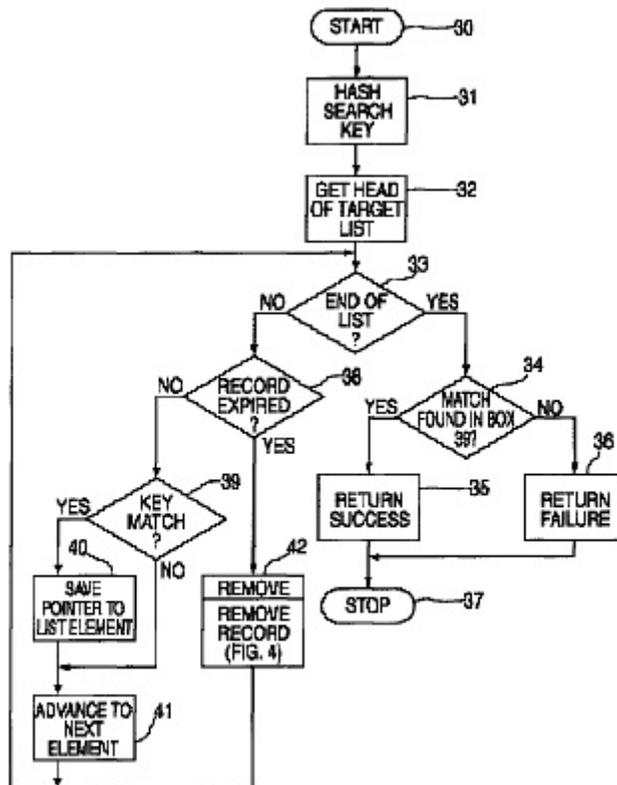


FIG. 3

Claim 1 of the '120 patent is set forth below as a representative claim with disputed claim terms² set forth in bold:

1. An information storage and retrieval system, the system comprising:

² The disputed means-plus-function terms were not highlighted in order to emphasize the individually disputed claim terms. The means-plus-function elements in dispute are discussed further below.

a linked list to store and provide access to records stored in a memory of the system, at least some of the records **automatically expiring**;
a record search means utilizing a search key to access the linked list;
the record search means including a means for identifying and **removing** at least some of the expired ones of the records **from the linked list³ when the linked list is accessed**, and
means, utilizing the records search means, for accessing the linked list and, at the same time, removing at least some of the **expired** ones of the records in the linked list.

‘ 120 patent at 13:24-39 (claim 1).

CLAIM CONSTRUCTION PRINCIPLES

“It is a ‘bedrock principle’ of patent law that ‘the claims of a patent define the invention to which the patentee is entitled the right to exclude.’” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). The Court examines a patent’s intrinsic evidence to define the patented invention’s scope. *Id.* at 1313-1314; *Bell Atl. Network Servs., Inc. v. Covad Commc’ns Group, Inc.*, 262 F.3d 1258, 1267 (Fed. Cir. 2001). Intrinsic evidence includes the claims, the rest of the specification and the prosecution history. *Phillips*, 415 F.3d at 1312-13; *Bell Atl. Network Servs.*, 262 F.3d at 1267. The Court gives claim terms their ordinary and customary meaning as understood by one of ordinary skill in the art at the time of the invention. *Phillips*, 415 F.3d at 1312-13; *Alloc, Inc. v. Int’l Trade Comm’n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

Claim language guides the Court’s construction of claim terms. *Phillips*, 415 F.3d at 1314.

³ The parties dispute “identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed.” However, the Court chooses to define certain terms within the disputed phrase. *See* (Doc. No. 326) at 3, n.4.

“[T]he context in which a term is used in the asserted claim can be highly instructive.” *Id.* Other claims, asserted and unasserted, can provide additional instruction because “terms are normally used consistently throughout the patent.” *Id.* Differences among claims, such as additional limitations in dependent claims, can provide further guidance. *Id.*

“[C]laims ‘must be read in view of the specification, of which they are a part.’” *Id.* (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995)). “[T]he specification ‘is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.’” *Id.* (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002). In the specification, a patentee may define his own terms, give a claim term a different meaning that it would otherwise possess, or disclaim or disavow some claim scope. *Phillips*, 415 F.3d at 1316. Although the Court generally presumes terms possess their ordinary meaning, this presumption can be overcome by statements of clear disclaimer. *See SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.*, 242 F.3d 1337, 1343-44 (Fed. Cir. 2001). This presumption does not arise when the patentee acts as his own lexicographer. *See Irdeto Access, Inc. v. EchoStar Satellite Corp.*, 383 F.3d 1295, 1301 (Fed. Cir. 2004).

The specification may also resolve ambiguous claim terms “where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone.” *Teleflex, Inc.*, 299 F.3d at 1325. For example, “[a] claim interpretation that excludes a preferred embodiment from the scope of the claim ‘is rarely, if ever, correct.’” *Globetrotter Software, Inc. v. Elam Computer Group Inc.*, 362 F.3d 1367, 1381 (Fed. Cir. 2004) (quoting *Vitronics Corp.*, 90 F.3d at 1583). But, “[a]lthough the specification may aid

the court in interpreting the meaning of disputed language in the claims, particular embodiments and examples appearing in the specification will not generally be read into the claims.” *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988); *see also Phillips*, 415 F.3d at 1323.

The prosecution history is another tool to supply the proper context for claim construction because a patentee may define a term during prosecution of the patent. *Home Diagnostics Inc. v. LifeScan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir. 2004) (“As in the case of the specification, a patent applicant may define a term in prosecuting a patent”). The well established doctrine of prosecution disclaimer “preclud[es] patentees from recapturing through claim interpretation specific meanings disclaimed during prosecution.” *Omega Eng’g Inc. v. Raytek Corp.*, 334 F.3d 1314, 1323 (Fed. Cir. 2003). The prosecution history must show that the patentee clearly and unambiguously disclaimed or disavowed the proposed interpretation during prosecution to obtain claim allowance. *Middleton Inc. v. 3M Co.*, 311 F.3d 1384, 1388 (Fed. Cir. 2002). “Indeed, by distinguishing the claimed invention over the prior art, an applicant is indicating what the claims do not cover.” *Spectrum Int’l v. Sterilite Corp.*, 164 F.3d 1372, 1378-79 (Fed. Cir. 1988) (quotation omitted). “As a basic principle of claim interpretation, prosecution disclaimer promotes the public notice function of the intrinsic evidence and protects the public’s reliance on definitive statements made during prosecution.” *Omega Eng’g, Inc.*, 334 F.3d at 1324.

Although, “less significant than the intrinsic record in determining the legally operative meaning of claim language,” the Court may rely on extrinsic evidence to “shed useful light on the relevant art.” *Phillips*, 415 F.3d at 1317 (quotation omitted). Technical dictionaries and treatises may help the Court understand the underlying technology and the manner in which one skilled in the

art might use claim terms, but such sources may also provide overly broad definitions or may not be indicative of how terms are used in the patent. *Id.* at 1318. Similarly, expert testimony may aid the Court in determining the particular meaning of a term in the pertinent field, but “conclusory, unsupported assertions by experts as to the definition of a claim term are not useful.” *Id.* Generally, extrinsic evidence is “less reliable than the patent and its prosecution history in determining how to read claim terms.” *Id.*

The patent in suit may contain means-plus-function limitations that require construction. Where a claim limitation is expressed in means-plus-function language and does not recite definite structure in support of its function, the limitation is subject to 35 U.S.C. § 112 ¶ 6. *Braun Med., Inc. v. Abbott Labs.*, 124 F.3d 1419, 1424 (Fed. Cir. 1997). In relevant part, § 112 mandates that “such a claim limitation be construed to cover the corresponding structure . . . described in the specification and equivalents thereof.” *Id.* (citing 35 U.S.C. § 112 ¶ 6.). Accordingly, when faced with means-plus-function limitations, courts “must turn to the written description of the patent to find the structure that corresponds to the means recited in the [limitations].” *Id.*

Construing a means-plus-function limitation involves two inquiries. The first step requires “a determination of the function of the means-plus-function limitation.” *Medtronic, Inc. v. Advanced Cardiovascular Sys., Inc.*, 248 F.3d 1303, 1311 (Fed. Cir. 2001). Once a court has determined the limitation’s function, “the next step is to determine the corresponding structure disclosed in the specification and equivalents thereof.” *Medtronic*, 248 F.3d at 1311. A structure is corresponding “only if the specification or prosecution history clearly links or associates that structure to the function recited in the claim.” *Id.* Moreover, the focus of the corresponding structure inquiry is not merely whether a structure is capable of performing the recited function, but rather whether the

corresponding structure is “clearly linked or associated with the [recited] function.” *Id.*

DISCUSSION

The terms in dispute, and their corresponding constructions, are set forth below.

I. “a linked list to store and provide access to records”⁴

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|---|--|
| A list in which each record contains a pointer to the next record or information indicating that there is no next record. | Two or more records in which each record contains a pointer to the next record in the list or information indicating that there is no next record. |

The parties agreed to the construction of this term at the October 7, 2010 claim construction hearing and the Court adopts the agreed construction, “a list, capable of containing two or more records, in which each record contains a pointer to the next record or information indicating there is no next record.”

II. “automatically expiring”/ “expired”⁵

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|---|---|
| After a limited period of time or after the occurrence of some event, becoming obsolete and therefore no longer needed or desired in the storage system / obsolete and therefore no longer needed or desired in the storage system | Becoming obsolete and no longer needed or desired in the storage system by comparing some portion of the contents of the record to some external condition / obsolete and no longer needed or desired in the storage system because of some external condition |

The parties dispute what triggers a record to automatically expire. Defendants contend that an automatically expiring record does so because of some external condition. DFTS’ RESPONSE AT

⁴ The term “a linked list to store and provide access to records” is contained in claims 1 and 3. Claims 5 and 7 discuss a “linked list of records.” The construction applies to claims 1, 3, 5, and 7.

⁵ This term is contained in claims 1, 3, 5, and 7.

7. Plaintiffs argue that a record expires automatically after a limited time or the occurrence of some event. PLTFF'S BRIEF AT 6.

In support of its construction, Bedrock points to the specification, which states:

Some forms of information are such that individual data items, after a limited period of time, become obsolete, and their presence in the storage system is no longer needed or desired. Scheduling activities, for example, involve data that become obsolete once the scheduled event has occurred.

PLTFF'S BRIEF AT 6 (citing '120 patent at 2:7-11) (emphasis in original). Bedrock further argues that "records can become obsolete merely by the passage of time or by the occurrence of some event." *See id.* (citing '120 patent at 5:38-41). Defendants, on the other hand, argue that the determination of whether a record has expired results "by comparing some portion of the contents of the record to some external condition." DFTS' RESPONSE AT 8 (citing '120 patent at 6:5-13). Defendants further assert that Bedrock's construction of the term merely sets forth examples of external conditions. *Id.*

The specification supports both Bedrock and Defendants' constructions. Specifically, Defendants' construction encompasses Bedrock's; a period of time or an event are examples of external conditions. But an expiration of a record may be triggered by something other than an external condition. The system may be programmed to compare record timestamps to an internal clock or timer that triggers expiration. *See* '120 patent at 6:9-11 ("A timestamp in the record, for example, could be compared with the current time-of-day value maintained by all computers.").

Therefore, the Court construes "automatically expiring" as, "becoming obsolete and therefore no longer needed or desired in the storage system because of some condition, event, or period of time." The Court finds that internal and external events and conditions are encompassed by the construction.

In addition, the Court construes “expired” as, “obsolete and therefore no longer needed or desired in the storage system because of some condition, event, or period of time.” Again, internal and external events and conditions are encompassed by the construction.

III. “removing . . . from the linked list”⁶

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|--|---|
| <p>Bedrock believes that this term language is properly construed pursuant to 35 U.S.C. § 112(6)⁷</p> <p>No construction necessary; however, should the Court construe this term; “removing at least some of the automatically expiring records from the linked list when the linked list is accessed for a purpose other than garbage collection, using the same linked list traversal performed for the purpose other than garbage collection.”⁸</p> | <p>While traversing the linked list, both adjusting the pointers in the linked list to bypass the previously identified expired records and deallocating the memory occupied by these records</p> |

The issue at dispute concerns whether the removal step requires the memory occupied by a record to be deallocated.⁹ See DFTS’ RESPONSE AT 14. Therefore, the Court finds that only “removing . . . from the linked list” requires construction.

Defendants assert that to operate in accord with the purpose of the invention, the removal step requires deallocating memory. *Id.* Defendants reason that the memory associated with a

⁶ This term is contained in claims 1, 3, 5, and 7.

⁷ Bedrock’s proposed construction applies to claims 1 and 5.

⁸ This proposed construction applies to claims 3 and 7.

⁹ With respect to Claims 1 and 5, the parties dispute the phrase “identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed.” As for Claims 3 and 7, the parties dispute the phrase “removing at least some of the automatically expired records from the linked list when the linked list is accessed.”

removed record must be deallocated in order “to reclaim the storage and maintain fast access to the data.” *Id.* (citing ‘120 patent at 2:19-21). In support of their argument, Defendants cite to the Appendix Code illustrated below. *Id.* at 15.

Remove Procedure

```

procedure remove (var elem_to_del: list_element_pointer;
                  previous_elem: list_element_pointer;
                  index: 0 . . . table_size - 1);
/* Delete elem_to_del↑ from list, advancing elem_to_del to next element. previous_elem points to
   elem_to_del's predecessor, or nil if elem_to_del↑ is 1st element in list.*/
var p: list_element_pointer;                               /* Save pointer to elem_to_del for disposal. */
begin
  p := elem_to_del;                                       /* Save so we can dispose when finished adjusting pointers. */
  elem_to_del := elem_to_del↑.next;
  if previous_elem = nil                                  /* Deleting 1st element requires changing */
  then table[index] := elem_to_del                        /* head pointer, as opposed to */
  else previous_elem↑.next := elem_to_del;               /* predecessor's next pointer. */
  dispose (p)                                            /* Dynamically de-allocate node. */
end                                                       /* remove*/

```

Defendants contend that the “APPENDIX contains PASCAL-like pseudocode listings for all of the programmed components **necessary to implement** an information storage and retrieval system operating **in accordance with the present invention.**” *Id.* (citing ‘120 patent at 8:60-64) (emphasis in original). Therefore, according to Defendants, the removal procedure outlined above includes deallocating the memory associated with the removed record after adjustment of the pointers in the linked list. *Id.*

In addition, Defendants point to Figures 3 through 7 of the ‘120 patent, emphasizing the remove procedure outlined in Figure 4 below. *Id.* at 16.

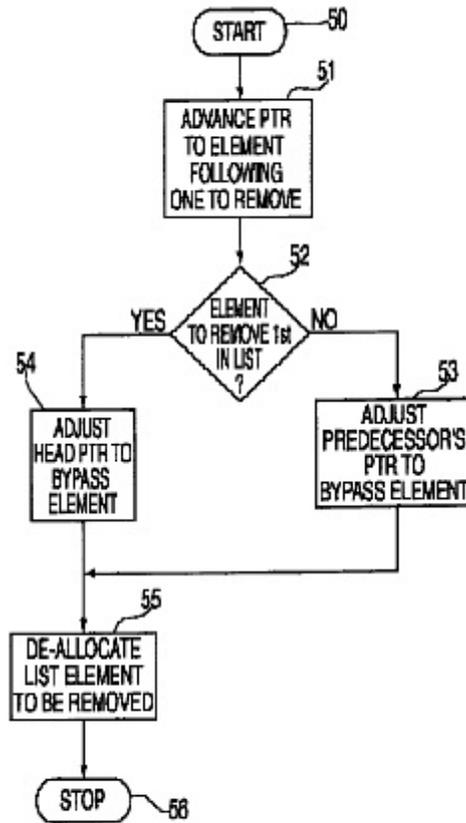


FIG. 4

Defendants argue that Figure 4, like the APPENDIX, as well as all other references to the removal procedure, require (1) adjusting the pointers in the linked list and (2) deallocating memory. *Id.* at 15 (citing ‘120 patent at 5:25-33; 6:35-45; 7:43-51, 7:56-64).

In response, Bedrock argues that the deallocation step is separate from the removal step. *See* PLTFF’S REPLY AT 7. In particular, Bedrock cites, “The remove procedure *causes actual removal* of the designated element by *adjusting the predecessor pointer* so that it bypasses the element to be removed Following the pointer adjustments, the storage occupied by the *removed* element is returned to the system storage pool for future allocation.” *Id.* at 5 (citing the ‘120 patent at 7:43-50)

(emphasis in original). In addition, Bedrock replies that the patent need not claim the goal of the invention. *Id.* at 4; *see also* PLTFF’S SLIDES AT 36 (citing *i4i Ltd. Partnership v. Microsoft Corp.*, 598 F.3d 831, 843 (Fed. Cir. 2010)). In other words, the objective of the technology need not be legally claimed. *Id.* To address Defendants’ point regarding the APPENDIX, Bedrock states that the code is not limiting, but rather shows a preferred embodiment in accordance with the present invention. *Id.* at 4-5.

Defendants are correct that the removal procedure includes adjusting the pointers in the linked list. Claim 1 of the ‘120 patent, which is exemplary for the purpose of resolving this dispute, recites in relevant part:

the record search means including a means for identifying
and removing at least some of the expired ones of the
records from the linked list when the linked list is
accessed, and

‘120 patent at 13:31-34. Specifically, the expired record(s) is removed *from the linked list*, but not memory storage. The specification goes on to state, “The remove procedure causes actual removal of the designated element by adjusting the predecessor pointer so that it bypasses the element to be removed.” *Id.* at 7:43-50. Therefore, removal consists of taking the expired record(s) out of the linked list and adjusting the pointer from the previously identified expired record to the successor element in the linked list or NIL if the removed record was the last record in the linked list. *See* ‘120 patent at 7:33-37.

Moreover, the specification supports the idea that removal is a step separate from deallocation. The removal procedure involves adjusting the pointers to bypass the removed record:

The remove procedure causes actual removal of the designated element by adjusting

the predecessor pointer so that it bypasses the element to be removed Following the pointer adjustments, the storage occupied by the *removed* element is returned to the system storage pool for future allocation.

‘120 patent at 7:43-50 (emphasis added)). Only after the pointers are adjusted is the “removed element,” returned to system storage, which is the actual step of deallocation. Accordingly, the Court finds that the removal procedure is distinct from deallocating memory, and the proper construction of the term “removing . . . from the linked list” is “adjusting the pointer in the linked list to bypass the previously identified expired records.”

IV. “dynamically determining”¹⁰

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|---|---|
| <p>Bedrock believes that this term language is properly construed pursuant to 35 U.S.C. § 112(6)¹¹</p> <p>Determining, during the program execution, maximum number of expired ones of the records to remove when the linked list is accessed¹²</p> | <p>Immediately before the linked list is traversed, determining a single number that serves as an upper limit on the number of records to remove while the linked list is traversed</p> |

The parties debate whether “dynamically determining” means “immediately before the linked list is accessed” or “during program execution.” *See* PLTFF’S BRIEF AT 10. Consequently, the Court finds that only “dynamically determining” needs construction.¹³

Bedrock argues that one of ordinary skill in the art at the time of the invention would have

¹⁰ This term is contained in claims 2, 4, 6, and 8.

¹¹ Bedrock’s proposed construction applies to claims 2 and 6.

¹² This proposed construction applies to claims 4 and 8.

¹³ The parties dispute the phrase “dynamically determining a maximum number of expired ones of the records to remove when the linked list is accessed.”

understood the term “dynamically” to mean “during program execution.” *Id.* Bedrock offers external evidence in the form of the *New IEEE Standard Dictionary of Electrical and Electronic Terms* (5th ed. 1990) to show that “dynamic” means “[p]ertaining to an event or process that occurs during computer program execution.” *Id.* Defendants, on the other hand, argue that the dynamic determination occurs after the linked list is accessed. DFTS’ RESPONSE AT 20 (citing ‘120 patent at 3:11-14). Defendants contend the dynamic determination occurs immediately after accessing the linked list because Claims 4 and 8 state the determination occurs “when the linked list is accessed.” *Id.* at 21. Defendants further assert that Bedrock’s construction of “during program execution” is vague because the dynamic determination could occur at any time, days before or after the linked list is accessed, as long as the program is in execution. *Id.*

The construction Defendants propose is incorrect. There is no reference to “immediate” or “before” in the portion of the specification to which Defendants cite: “the method provides for dynamically determining maximum number of expired ones of the records to be removed when the linked list is accessed.” ‘120 patent at 3:11-14.

Additionally, “when the linked list is accessed” does not modify “dynamically determining,” as Defendants contend. Rather, the claims state that removal occurs “when the linked list is accessed,” not that a dynamic decision is made “when the linked list is accessed.” Looking at Claim 1, it states in relevant part, “identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed.” ‘120 patent at 13:31-34. Thus, it seems Claim 1 declares that “identifying and removing” occurs when the linked list is accessed. Claims 4 and 8 seem to indicate the same: “dynamically determining maximum number of expired ones of the records to remove when the linked list is accessed.” ‘120 patent at 13:55-57; 14:57-59 (emphasis

added). Thus, “dynamically determining” modifies “expired ones of the records to remove,” not “when the linked list is accessed.” Therefore, the claims do not specifically say when the dynamic decision occurs and Defendants’ proposed construction is incorrect.

Instead of discussing when the dynamic decision takes place, the specification discusses dynamic decisions in the context of changing factors under which the decision may be made dynamically. The specification describes an alternative method of on-the-fly garbage removal whereby “the procedure can be readily adapted to remove some but not all of the expired records, thereby shortening the linked list traversal time and speeding up the search at the expense of perhaps leaving some expired records in the list.” ‘120 patent at 6:59-63. The specification goes on to describe the alternate method:

The implementor even has the prerogative of choosing among these strategies dynamically at the time search table is invoked by the caller, thus sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them. *Such dynamic runtime decision might be based on factors such as*, for example, how much memory is available in the system storage pool, general system load, time of day, the number of records currently residing in the information system, and other *factors both internal and external to the information storage and retrieval system itself*.

‘120 patent at 6:67-7:1-10 (emphasis added). The specification does not require a temporal limitation on the dynamic decision-making process, but it does state that the dynamic determination is based on changing factors under which the caller may make certain decisions, such as available memory, time of day, number of records, etc. Therefore, the dynamic determination depends on “factors both internal and external to the information storage and retrieval system itself.”

Accordingly, the Court finds the term “dynamically determining” means “making a decision based on factors internal or external to the information storage and retrieval system.” This

construction describes what the decision is and how it is made, not when it is made.

V. “maximum number”¹⁴

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|--|--|
| Bedrock believes that this term language is properly construed pursuant to 35 U.S.C. § 112(6) ¹⁵ Determining, during the program execution, maximum number of expired ones of the records to remove when the linked list is accessed ¹⁶ | Immediately before the linked list is traversed, determining a single number that serves as an upper limit on the number of records to remove while the linked list is traversed |

Like the term “dynamically determining” above, the Court finds that only “maximum number” requires construction.¹⁷

The parties disagree whether the term “maximum number” should even be construed. DFTS’ RESPONSE AT 21. Defendants contend the term means “a single number that serves as an upper limit of the number of records to remove while the linked list is traversed.” *Id.* In support, Defendants cite a portion of the prosecution history, taking almost verbatim a definition provided by the ‘120 patent’s inventor. *Id.* at 22 (citing EX. 5, ‘120 PATENT FILE HISOTRY, AUG. 10, 1998 RESPONSE AT 4).

Bedrock, on the other hand, states no construction is needed. Bedrock reasons that once the terms “dynamically determined” and “expired” are construed, the jury will understand the term

¹⁴ This term is contained in claims 2, 4, 6, and 8.

¹⁵ Bedrock’s proposed construction applies to claims 2 and 6.

¹⁶ This proposed construction applies to claims 4 and 8.

¹⁷ The parties dispute the phrase “dynamically determining a maximum number of expired ones of the records to remove when the linked list is accessed.”

“maximum number.” PLTFF’S BRIEF AT 11. In response to Defendants’ prosecution history arguments, Bedrock argues that the inventor was merely distinguishing a piece of prior art, which allowed for a system to select between organizing records in a linear probing technique or the external chaining technique. *Id.* at 12. According to Bedrock, the prior art has an upper and lower threshold that determined whether the system would reorganize the records according to linear probing or external chaining. *Id.* Thus, the statement made in the prosecution history merely distinguished between the methods of using thresholds to determine the use of particular techniques (linear vs. external) and a maximum number of records to remove. *Id.* Therefore, Bedrock argues, the inventor was not acting as his own lexicographer and did not define the term “maximum number.” *Id.* In its Reply, Bedrock further argues that Defendants’ proposed construction for the term reads out embodiments of the specification, particularly the alternative method previously discussed above. PLTFF’S REPLY AT 6 (citing ‘120 patent at 6:66-7:4).

Defendants are incorrect in requiring a “maximum number” to be a single number. In light of the discussion above regarding “dynamically determining,” the alternative method of on-the-fly garbage removal allows for the removal of all, none, or some of the expired records:

The implementor even has the prerogative of choosing among these strategies dynamically at the time search table is invoked by the caller, thus sometimes *removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them. . . .* A person skilled in the art will appreciate that the technique of removing all expired records while searching the linked list can be expanded to include techniques whereby not necessarily all expired records are removed, and that the decision regarding if and how many records to delete can be a dynamic one.

‘120 patent at 6:66-7:15 (emphasis added). Because the decision to remove records may be a dynamic one, the number of records to be removed may vary. At times, the maximum number of

records to be removed could be all the expired records in the list. At other times, the maximum number of removed records could be the expired records in the list before a match is found. Therefore, the maximum number of removed records does not necessarily need to be a single, constant number. The maximum number need only be an upper limit as to the records to be removed.

Therefore, in light of the terms “dynamically determining” and “expired” construed above, the Court finds that no construction is necessary for the term “maximum number.” The Court, however, does find that the term “maximum number” is not limited to a single number.

VI. “external chaining”

The parties have agreed that the term “external chaining” means “a technique for resolving hash collisions using a linked list(s).”

VII. “when the linked list is accessed”¹⁸ and Ordering of Method Steps¹⁹

| Claim Term or Phrase | Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|---|--|---|
| when the linked list is accessed | Bedrock believes that this term language is properly construed pursuant to 35 U.S.C. § 112(6) ²⁰ No construction necessary ²¹ | Both identification and removal of the automatically expired record(s) occurs during the same traversal of the linked list. |

¹⁸ This term is contained in claims 1, 3, 4, 5, 7, and 8.

¹⁹ The Ordering of Method Steps applies to claims 3 and 7.

²⁰ Bedrock’s proposed construction applies to claims 1 and 5.

²¹ This proposed construction applies to claims 3 and 7.

| | | |
|------------------------------|--|---|
| Order of Method Steps | <p>No construction necessary.</p> <p>If the Court is inclined to address this issue, then it should hold that the steps of claim 3 may be performed in a consecutive manner, in a repeating manner, in an overlapping manner, or a combination of the three.</p> <p>If the Court is inclined to address this issue, then it should hold that the steps of claim 7 may be performed in a consecutive manner, in a repeating manner, in an overlapping manner, or a combination of the three, except that the ultimate step of claim 7 must follow or at least partially follow the penultimate step of claim 7.</p> | <p>The elements of claim 3 and 7 must be executed in order.</p> <p>Moreover, “when the linked list is accessed” in the removing step refers to the accessing step, and the identifying and removing steps must occur during the same traversal of the linked list of records.</p> |
|------------------------------|--|---|

A. “when the linked list is accessed”

The main dispute between the parties is whether identifying and removing occurs “when the linked list is accessed.” DFTS’ RESPONSE AT 9. In other words, the parties disagree as to whether identifying and removing occur in the same access. Therefore, the Court finds that only “when the linked list is accessed” needs construction.²²

Defendants assert that the language of Claims 1 and 5 make clear that the “identification” and “removing” occur in the same traversal of the linked list, citing the claim language of “identifying and removing at least some of the expired ones of the records from the linked list when the linked

²² The parties actually dispute the phrase “identifying and removing at least some of the expired records from the linked list when the linked list is accessed.”

list is accessed.” *Id.* at 9-10. Further, Defendants assert that the specification repeatedly describes identifying and removing as occurring during the same traversal of the linked list. *Id.* at 11-12 (citing Abstract, 2:54-63; 3:7-11; 5:57-6:27; 6:35-35; 6:46-53; 6:56-59; Fig. 3; Appendix). Defendants additionally note that the ‘120 patent Appendix describes the concept of the identification and removal elements performed together as the “HEART OF THE TECHNIQUE.” *Id.*(citing Appendix, cols. 11-12). According to Defendants, the goal of the invention would not be achieved if identification of expired records occurred on one traversal and removal occurred on another. *Id.* at 13. To further support their construction, Defendants cite a Response filed during prosecution in which the applicant stated “what is recited in claims 1, 3, 5, and 7, for example, means and methods for identifying and removing ‘at least some expired ones of the records’ from the linked list ‘when the linked list is accessed.’” *Id.* at 10 (citing August 10, 1998 Response at 6). Defendants note that the Response did not distinguish between the means claims (claims 1 and 5) and the method claims (claims 3 and 7).

In response, Bedrock states that the portions of the specification to which Defendants cite actually repeat the claim language “when the linked list is accessed,” rather than the “during the same traversal” language Defendants propose. PLTFF’S REPLY AT 4. In addition, Bedrock asserts that Defendants’ “traversal” construction comes from the preferred embodiments described within the specification, and therefore, Defendants’ construction should be rejected. *Id.*

First, the parties agree that the identifying step must be initiated before the removing step can begin. That said, the Court finds that the identifying and removing steps occur during the same access. The “access” referred to in “removing at least some of the automatically expired records

from the linked list when the linked list is *accessed*²³ is the same “access” of “accessing the linked list of records.” *See* ‘120 patent at 13:48-53.

Furthermore, the preferred embodiment describes the identifying and removing steps as occurring in the same access. In discussing Figure 3, the specification recites in relevant part:

If decision box 38 determines that the record under question has expired, box 42 is entered to perform the on-the-fly removal of the expired record from the linked list and the return of the storage it occupies to the system storage pool. . . . It can be seen that the search table procedure of FIG. 3 operates to examine the *entire linked list of records* of which the searched-for record is a part, *and to remove expired records*

‘120 patent at 6:21-26; 6:34-38 (emphasis added). Moreover, the specification states that identification and removal occur “while other types of access to the storage space are taking place,” indicating that identification and removal occur during the same access. *See* ‘120 patent at 2:56-63.

This is not a case of merely reading into a claim a preferred embodiment. The language of Claims 1 and 5 dictate such a finding. Furthermore, Defendants note that the specification repeatedly describes the importance and advantage of the identifying and removing elements occurring during the same access. And, during prosecution, the means and method claims were distinguished from prior art on this basis without differentiation between the means claims (claims 1 and 5) and the method claims (claims 3 and 7).

Therefore, the Court finds the term “when the linked list is accessed” to mean “both identification and removal of the automatically expired record(s) occurs during the same access²⁴ of

²³ Claims 3 and 7, emphasis added.

²⁴ Note that the Court has adopted the majority of Defendants’ proposed construction. However, Defendants proposed the word “traversal” where the Court has used “access.” Looking to Defendants’ arguments regarding “removing . . . from the linked list,” Defendants cite the ‘120 patent at 5:25-33; 6:35-45; 7:43-51, 7:56-64. None of the cited portions of the specification use “traversal.” To be consistent with the claim, the Court substitutes the word “access” for “traversal.”

the linked list.”

B. Ordering of Method Steps

Although the Court has determined that identification and removal of expired records occurs during the same access, issues concerning the order of the method steps in Claims 3 and 7 remain. Specifically, the parties dispute whether the method steps of Claims 3 and 7 must be performed in order as listed. Claim 3, which is exemplary for the purpose of resolving this dispute, is below.

3. A method for storing and retrieving information records using a linked list to store and provide access to the records, at least some of the records automatically expiring, the method comprising the steps of:
accessing the linked list of records,
identifying at least some of the automatically expired ones of the records, and
removing at least some of the automatically expired records from the linked list when the linked list is accessed.

Defendants assert that logic dictates that the method steps of Claims 3 and 7 must be performed in order. DFTS’ RESPONSE AT 30. According to Defendants, the linked list of records must first be accessed before identifying which of those records has expired. *Id.* Similarly, the expired records cannot be removed until they have been identified. *Id.* Therefore, Defendants claim, the method claims must be performed in order. *Id.*

Bedrock argues the method steps need not be performed in the order they are listed because Federal Circuit precedent does not require such a reading. PLTFF’S BRIEF AT 27 (citing *Altiris v. Symantec Corp.*, 318 F.3d 1363, 1369 (Fed. Cir. 2003)). Although Bedrock agrees that the identifying step must begin before the removing step can be completed, Bedrock asserts that the jury would understand this, and thus, no construction is necessary. *Id.* However, Bedrock maintains that the method steps can be performed in a “consecutive, repeating, and/or overlapping manner without

offending the logic or grammar of the claim.” *Id.* Moreover, Bedrock contends that in the same access, multiple identifying steps could be executed before any removing steps begin. PLTFF’S REPLY AT 10.

Method steps that do not actually recite an order do not usually require one. *Altiris*, 318 F.3d at 1369 (citing *Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1342-43 (Fed. Cir. 2001)). To decide whether method steps require an order, the court must look at the claim language to determine whether logic or grammar dictate that the steps be performed as written. *Id.* If not, the court must look to the specification to determine whether the specification “directly or implicitly requires such a narrow construction.” *Id.* (citing *Interactive*, 256 F.3d at 1343). If the specification does not support an explicit order, then the order in which the steps are written is not limiting. *Id.*

Examining the language in Claim 7, it is clear that the ultimate step of Claim 7 must follow, or at least partially follow, the penultimate step of Claim 7. Claim 7 states in relevant part:

removing at least some of the automatically expired records from the linked list when the linked list is accessed, and
inserting, retrieving or deleting one of the records from the system following the step of removing.

‘120 patent at 14:51-55. Because Claim 7 explicitly states that the “inserting, retrieving or deleting” step must follow “the step of removing,” the last step described in Claim 7 must follow, at least in part, the removing step.

As for the remaining method steps, as stated above, logic dictates, and the parties agree, that the “identifying” step must start before “removing” can begin. However, the identifying step need not complete before the removing step can begin, thus allowing execution of the steps to overlap.

For example, removal could begin before the system has identified all the expired records in the linked list to be removed: “The search table procedure shown in FIG. 3 . . . and described above traverses the entire linked list removing all expired records as it searches for a key match.” ‘120 patent at 3:56-59. The “as it” language indicates the possibility of contemporaneous performance of the identifying and removing steps. Thus, nothing in the specification “directly or implicitly requires such a narrow construction.” *See Altiris*, 318 F.3d at 1369 (citing *Interactive*, 256 F.3d at 1343).

Therefore, neither logic nor grammar dictate that the method steps of the ‘120 patent must be executed in the order they are written, with the exception of (1) the steps in Claim 7 described above and (2) the requirement that identifying must begin before removing can begin.

VIII. “a record search means utilizing a search key to access the linked list”²⁵

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|-----------------------------------|-----------------------------------|
|-----------------------------------|-----------------------------------|

²⁵ This term is contained in claims 1 and 5.

The precise language of claims 1 and 5 differ slightly. The disputed portion of Claim 1 is recited above. Claim 5 states, in relevant part, “a record search means utilizing a search key to access a linked list of records having the same hash address.” Although claims 1 and 5 differ slightly, the parties briefed these claim terms together. *See* PLTF’s BRIEF AT 13. Therefore, for ease of reference, the Court focuses its discussion on Claim 1. However, the discussion applies equally to the term contained in Claim 5 because the issues raised by the parties are similar.

| | |
|---|-------------------|
| <p>Function: The <u>recited function</u> is record searching utilizing a search key to access the linked list</p> <p>Structure: The <u>corresponding structure</u> is: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56; and (2) Executable software instructions as illustrated in Boxes 31-36 and Boxes 39-41 of FIG. 3, or as portions of the pseudo-code of Search Table Procedure (cols. 11 and 12) or Alternate Version of Search Table Procedure (cols. 11, 12, 13, and 14), and described in col. 5, line 57-col. 6 line 4 and col. 6 lines 15-20, or the equivalents thereof.</p> | <p>Indefinite</p> |
|---|-------------------|

The primary dispute between the parties is whether the term “a record search means utilizing a search key to access the linked list” is indefinite. *See* DFTS’ RESPONSE AT 23; DFTS’ SUMMARY JUDGMENT MOTION (Doc. No. 283). Defendants argue hashing is the only means for “using a search key to access the linked list,” but the specification does not provide a hashing algorithm. (Doc. No. 283) at 8. Therefore, according to Defendants, the term is indefinite because it lacks corresponding structure. *Id.*

Bedrock counters Defendants’ indefiniteness argument, stating that the function of the means-plus-function element is not to execute a hashing function, which would require a hashing algorithm, but rather to perform “record searching utilizing a search key to access the linked list.” *See* PLTFF’S SJ RESPONSE AT 6 (Doc. No. 299); JOINT CLAIM CONSTRUCTION CHART (Doc. No. 309-1). Bedrock states that the corresponding structure for the function may be found in Figures 1, 2, and 3, or portions of the psuedo-code of Search Table Procedure (cols. 11 and 12) or Alternate

Version of Search Table Procedure (cols. 11, 12, 13, and 14), and equivalents thereof. PLTFF'S RESPONSE AT 14-17.

As an initial matter, the Court must determine whether the claim should be construed pursuant to 35 U.S.C. § 112 ¶ 6. A claim is presumed to be a means-plus-function limitation governed by § 112 ¶ 6 if it “contains the word ‘means’ and recites a function.” *Envirco Corp. v. Clestra Cleanroom, Inc.*, 209 F.3d 1360, 1364 (Fed. Cir. 2000). However, the presumption fails if the claim reveals sufficient structure to perform the function. *Id.* Looking at the claim language, the Court finds that § 112 ¶ 6 applies, and the Court will construe “a record search means utilizing a search key to access the linked list” as a means-plus-function term.

Next, the Court must determine the function claimed by the means-plus-function limitation. “The statute [35 U.S.C. § 112 ¶ 6] does not permit limitation of a means-plus-function claim by adopting a function different from that explicitly recited in the claim.” *Micro Chemical, Inc. v. Great Plains Chemical Co., Inc.*, 194 F.3d 1250, 1258 (Fed. Cir. 1999). Therefore, the function of the term “a record search means utilizing a search key to access the linked list” is “utilizing a search key to access the linked list.”

Bedrock and Defendants seem to agree that the function cannot be performed without a hashing method. Plaintiff proposes that the corresponding structure includes Box 31 of Figure 3, a Hash Search Key, and Defendants argue that the recited function cannot be performed without a hashing algorithm.

The corresponding structure of a means-plus-function element must be clearly linked to the claimed function, *Medtronic, Inc.*, 248 F.3d at 1311, and must actually perform the recited function. *Asyst Tech., Inc. v. Empak, Inc.*, 268 F.3d 1364, 1370 (Fed. Cir. 2001). In this case, the function

is “utilizing a search key to access the linked list,” as opposed to “executing a hashing function” or a “means for hashing.” *See* PLTFF’S SLIDES AT 14. A Hash Search Key (Box 31) allows the system to access the linked list. The hashing step in Box 31 provides the search key, which is used to access/search the hash table, or externally chained linked list. ‘120 patent at 5:58-60.

However, further structure is needed to execute the search key to access the linked list. Therefore, Plaintiff’s proposed construction is essentially correct. The information storage and retrieval can be executed by “application software packages 23-35, or used by other parts of the system, such as user access software 20 or operating system 21 software.” ‘120 patent at 4:45-48. The application software, user access software, or operating system software²⁶ stored on RAM 11 is accessed by CPU 10. *See* ‘120 patent at 3:56-58. Therefore, the corresponding structure requires CPU 10, RAM 11, portions of application software, user access software or operating system software.²⁷

In addition, the software referred to above must be programmed with instructions. Figure 3 shows “a flowchart of the search table procedure for searching the hash table.” ‘120 patent at 5:53-54. The specification states, in relevant part:

[T]he search key of the records being searched for is hashed in box 31 to provide the subscript of an array element. In box 32, the hash table array location indicated by the subscript generated in box 31 is accessed to provide the pointer to the target linked list. Decision box 33 examines the pointer value to determine whether the end of the linked list has been reached. If the end has been reached, decision box 34 is entered to determine if a key match was previously found in decision box 39 If so, the search is successful and returns success in box 35, followed by the

²⁶ Looking to Figure 2, once a user has acquired access to the computer system, the user is launched into the operating system, which coordinates the hardware components and application software of the computer system. ‘120 patent at 4:22-44; FIG. 2.

²⁷ These elements are all part of the corresponding structures for the remaining means-plus-function claim terms. *See* Sections IX-XIII, *infra*.

procedure's termination in terminal box 37. If not, box 36 is entered where failure is returned and the procedure again terminates in box 37. In any case, if the record has not expired, decision box 39 is entered to determine if the key in this record matches the search key. If it does, the address of the record is saved in box 40 and box 41 is entered. If the record does not match the search key, the procedure bypasses box 40 and proceeds directly to box 41. In box 41, the procedure advances forward to the next record in the linked list and the procedure returns to box 33.

'120 patent at 5:58-6:4; 6:13-20. The search procedure described above explains the steps of "utilizing a search key to access the linked list." As stated above, Box 31 provides the search key that is used to access the linked list. Decision boxes 34 and 39 then compare the search key to the keys associated with the records in the linked list to determine whether a match has been found. If the record does not match the search key, the search procedure bypasses that record and moves on to the next element, box 41, unless the searched record is the last of the list, which was determined at decision box 33. Alternatively, the software could be programmed with instructions illustrated in the psuedo-code of Search Table Procedure (cols. 11 and 12) or Alternate Version of Search Table Procedure (cols. 11, 12, 13, & 14), which provide instructions similar to those described above. Thus, the specification provides sufficient structure to perform the search function claimed in the means-plus-function term.

Therefore, the corresponding structure is: CPU 10 and RAM 11 of FIG. 1 and col. 3 lines 52-56 and portions of the application software, user access software or operating system software, as described at col. 4 lines 22-48, programmed with software instructions as described in Boxes 31-36 and Boxes 39-41 of FIG. 3 and in col. 5 line 53-col. 6 line 4 and col. 6 lines 14-20, and/or programmed with software instructions as described in the pseudo-code of Search Table Procedure (cols. 11 and 12) or Alternate Version of Search Table Procedure (cols. 11, 12, 13, and 14), and equivalents thereof.

IX. “the record search means including a means for identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed”²⁸

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|---|--|
| <p>Function: record searching including identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed.</p> <p>Structure: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56. (2) Executable software instructions as illustrated in Boxes 33-42 of FIG. 3, or as pseudo-code of Search Table Procedure (cols. 11 and 12) or Alternate Version of Search Table Procedure (cols. 11-14), starting at the line “while . . . /*HEART OF THE TECHNIQUE . . .” and ending at the end of each procedure, and/or as described in col. 5, line 63 - col. 6, line 34, or the equivalents thereof.</p> | <p>Function: identifying and removing at least some [of the] expired ones of the records from the linked list [of records] when the linked list is accessed.</p> <p>For the construction of this function, see proposed constructions above, as further described in Joint Claim Construction Statement [Dkt. 251].</p> <p>Structure: Boxes 10 and 11 of Fig. 1, Boxes 38 and 42 of Fig. 3, Fig 4, pseudocode in the Search Procedure (cols. 11-14) and Remove Procedure (cols. 13-14), and corresponding portions of the specification.</p> |

The parties agree that the term should be construed as a means-plus-function term, and they essentially agree that the function of the term contains the claim language. However, Defendants propose additional limitations on the function, indicating that both identification and removal occur during the same traversal. DFTS’ RESPONSE AT 25. Defendants further propose that the “utilizing a search key to access the linked list” function is performed “when the linked list is accessed.” *Id.* In addition, Defendants contend removal requires deallocating the memory associated with the

²⁸ This term is contained in claims 1 and 5.

removed record.²⁹ *Id.* Bedrock contends Defendants improperly read limitations into the term that are not explicitly found within the recited claim. PLTFF’S REPLY AT 7. As stated above, the function of a means-plus-function term should explicitly recite the claim language. Therefore, the function is “identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed.”

The main dispute between the parties is whether the corresponding structure includes programming to deallocate elements to be removed. *See* DFTS’ RESPONSE AT 25. Defendants argue that because Bedrock concedes that Box 42 of Figure 3 is part of the corresponding structure, it necessarily follows that the corresponding structure includes the deallocation algorithm in Figure 4, depicted to the right. *Id.* According to Defendants, Box 42 explicitly states that the removal algorithm is disclosed in Figure 4, but Box 42 itself does not disclose any structure. *Id.* Therefore, Defendants contend the removal algorithm includes adjusting the pointers as well as deallocation. *Id.* at 26.

Bedrock replies that Defendants’ construction improperly injects deallocating memory into the function, which erroneously imports deallocating structure into the proposed corresponding structure. PLTFF’S REPLY AT 7. Bedrock emphasizes that the rearrangement of pointers, not deallocation, causes

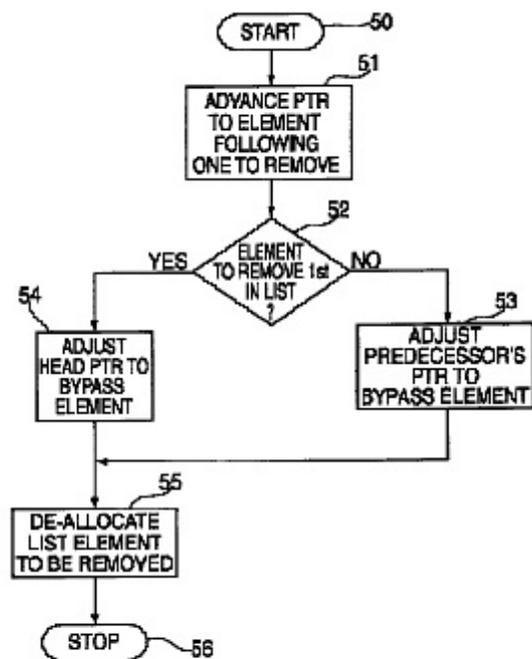


FIG. 4

²⁹ The Court has already addressed these issues in Sections III, VII.A and VIII.

actual removal:

The remove procedure causes actual removal of the designated element by adjusting the predecessor pointer so that it bypasses the element to be removed. . . . Following pointer adjustments, the storage occupied by the removed element is returned to the system storage pool for future allocation.

Id. (citing ‘120 patent at 7:43-50). Thus, Bedrock contends, the corresponding structure is the portion of Figure 4 and/or the pseudo-code of the Remove Procedure that relate to pointer adjustment. *See id.* at 8.

The corresponding structure should be limited to “only that which is necessary to perform the recited function.” *NOMOS Corp. v. Brainlab USA, Inc.*, 357 F.3d 1364, 1368 (Fed. Cir. 2004) (citing *Asyst Tech*, 268 F.3d at 1371). As stated above, the function is “identifying and removing at least some of the expired ones of the records from the linked list when the linked list is accessed.” Because the Court finds the removal step is separate from the deallocation step, deallocation is not part of the recited function or the corresponding structure. *See* discussion *supra* in Section III.

Furthermore, the Court has construed “removing . . . from the linked list” to mean “adjusting the pointer in the linked list to bypass the previously identified expired records.” Therefore, the corresponding structure for the term should include elements related to adjusting pointers.

The corresponding structure requires CPU 10, RAM 11, application software, user access software, and/or operating system software to execute the information storage and retrieval process. *See* discussion *supra* Section VIII. The software must be programmed with instructions similar to those described in Section VIII because both functions involve accessing the linked list to search for records. *See* discussion *supra* Section VIII; ‘120 patent at 5:58-6:4; 6:13-20. In addition, the instructions must include a procedure for identifying expired records and removing said records,

which should include directions for adjusting pointers to bypass the removed element.

The specification supports such corresponding structure. In addition to the portions of the specification that describe the search procedure illustrated with Boxes 31-36 and Boxes 39-41, *see* ‘120 patent at 5:58-6:4; 6:13-20, the specification goes on to provide instructions for identifying and removing expired records when the linked list is accessed:

If the end of the list has not been reached as determined by decision box 33, decision box 38 is entered to determine if the record pointed to has expired. . . . If decision box 38 determines that the record under question has expired, box 42 is entered to perform the on-the-fly removal of the expired record from the linked list and the return of the storage it occupies to the system storage pool, as will be described in connection with FIG. 4. In general, the remove procedure of box 42 (FIG. 4) operates to remove an element from the linked list by adjusting its predecessor’s pointer to bypass that element. (However, if the element to be removed is the first element of the list, then there is no predecessor and the hash table array entry is adjusted instead.) On completion of procedure remove [*sic*] invoked from box 42, the search table procedure returns to box 33.

‘120 patent at 6:5-8; 6:21-34. Moreover, the pseudo-code of the Search Table Procedure and Alternate Version of Search Table Procedure provide software instructions to perform the recited function. ‘120 patent at cols. 11-14.

Therefore, the corresponding structure is: CPU 10 and RAM 11 of FIG. 1 and col. 3 lines 52-56 and portions of the application software, user access software or operating system software, as described at col. 4 lines 22-48, programmed with software instructions as described in Boxes 33-42 of FIG. 3 and in col. 5 line 53-col. 6 line 34, and/or programmed with software instructions as described in the pseudo-code of Search Table Procedure (cols. 11 and 12) or Alternate Version of Search Table Procedure (cols. 11, 12, 13, and 14), and equivalents thereof.

X. “means, utilizing the record search means, for accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list”³⁰

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|--|--|
| <p>Function: utilizing the record search means, accessing the linked list, and at the same time, removing at least some of the expired ones of the records in the linked list.</p> <p>Structure: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56. (2) Executable software which provides the insert, retrieve, or delete record capability illustrated in the flowchart of FIG. 5, FIG. 6, or FIG. 7, respectively, and/or as pseudo-code of Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9, 10, 11, and 12), or Delete Procedure (cols. 11 and 12), respectively, and/or described in col. 7, line 65 - col.8, line 32, col. 8, lines 33-44, or the equivalents thereof.</p> | <p>Function: utilizing the record search means, [accessing the linked list / inserting, retrieving, and deleting from the system] and, at the same time, removing at least some of the expired ones of the records in the linked list.</p> <p>"At the same time" means during the same traversal of the linked list as [accessing the linked list / inserting, retrieving, and deleting records from the system].</p> <p>For the construction of “removing ...”, see proposed construction above, as further described in Joint Claim Construction Statement [Dkt. 251].</p> <p>Structure: Boxes 10 and 11 of Fig. 1; Figs. 4-7, pseudocode in the Search Procedure (cols. 11-14), Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9 and 10), Delete Procedure (cols. 11-12), and Remove Procedure (cols. 13-14), and corresponding portions of the specification.</p> <p>Inserting, retrieving, and deleting are all required.</p> |

As an initial matter, the Court finds the term falls under § 112 ¶ 6 and will construe the term as a means-plus-function term.

The primary dispute between the parties is whether the “means . . . for accessing the linked list” requires the steps of inserting, retrieving *and* deleting, or merely inserting, retrieving *or*

³⁰ This term is contained in claim 1.

deleting. DFTS' RESPONSE AT 28. Defendants assert that the specification consistently describes "accessing" to include the inserting, retrieving *and* deleting steps. DFTS' RESPONSE AT 28. Defendants further contend that accessing the linked list and the removal of records occurs during the same traversal of the linked list. *Id.* (citing '120 patent at 11:25-40; 12:25-40). Bedrock, on the other hand, contends Defendants improperly import limitations into the function, rather than letting the claim language control. *See* PLTFF'S REPLY AT 9. Bedrock further argues that nothing in the specification clearly disavows that an access could not be performed by any one of the insert, retrieve, or delete steps. *See id.*

The function of the term is merely the claimed function, and therefore additional limitations should not be imported into the function. Therefore, the function is "utilizing the record search means, accessing the linked list and, at the same time, removing at least some of the expired ones of the records in the linked list."

In addition, the limitation in Claim 5, which claims a means for "inserting, retrieving, *and* deleting," indicates that the means-plus-function in Claim 1 does not require the additional limitation Defendants propose:

means, utilizing the record search means, for inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.

'120 patent at 14:32-36 (emphasis added). The use of "accessing" in Claim 1 and the use of "inserting, retrieving, and deleting" in Claim 5 seem to indicate different meanings. Therefore, the function of "accessing" does not require inserting, retrieving, *and* deleting.

As for the corresponding structure, the parties reiterate their arguments with regard to

whether removal requires instructions for deallocation. *See* DFTS' RESPONSE AT 28. Bedrock again argues that "any one of the insert, retrieve, or delete structures can serve as the corresponding structure." PLTFF'S BRIEF AT 23.

Because the Court finds that the function does not require the "access" to be performed by the insert, delete, *and* retrieve structures, the corresponding structure for the means-plus-function limitation requires only that the software be programmed with instructions to insert, retrieve *or* delete records. Furthermore, the discussion in Section III states that the removal step does not include deallocation. Therefore, the structure for removing does not require code for deallocating memory.

The specification provides algorithms for the insert, delete, and retrieve processes. Specifically, Figure 5 is a flowchart of the insert procedure, '120 patent at 7:65-8:32; Figure 6 shows a flowchart for a retrieve procedure, '120 patent at 8:33-44; and Figure 7 depicts a delete procedure, '120 patent at 8:45-59. In addition, the specification provides pseudo-code for each procedure. *See* Insert Procedure, '120 patent at cols. 9-10; Retrieve Procedure, '120 patent at cols. 9, 10, 11 & 12; Delete Procedure, '120 patent at cols. 11-10.

Therefore, the corresponding structure is: CPU 10 and RAM 11 of FIG. 1 and col. 3 lines 52-56 and portions of the application software, user access software or operating system software, as described at col. 4, lines 22-48, programmed with software instructions that provide the insert, retrieve, or delete record capability as described in the flowchart of FIG. 5 and col. 7 line 65 – col. 8 line 32, FIG. 6 and col. 8 lines 33-44, or FIG. 7 and col. 8 lines 45-59, respectively, and/or programmed with software instructions that provide the insert, retrieve or delete record capability as described in the pseudo-code of Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9, 10,

11, and 12), or Delete Procedure (cols. 11 and 12), respectively, and equivalents thereof.

XI. “a hashing means to provide access . . .”³¹

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|--|-----------------------------------|
| <p>Function: The <u>recited function</u> is using hashing to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address, at least some of the records automatically expiring.</p> <p>Structure: The <u>corresponding structure</u> is: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56; and (2) Executable software instructions corresponding to pseudo-code “var table: array [0 . . . table_size - 1] of list_element_pointer /* Hash table.*” which point to records of type “list_element” in cols. 9-10 that allocates in memory an external chaining hash table, and/or as described in col. 5, lines 16-41, or the equivalents thereof.</p> | <p>Indefinite</p> |

Defendants argue that this term is indefinite. DFTS’ RESPONSE AT 23. Defendants contend the specification fails to disclose a hashing algorithm to perform the function of hashing. DFTS’ SUMMARY JUDGMENT MOTION AT 7. Bedrock, on the other hand, argues that a person of ordinary skill in the art would understand the scope of this term, and proposes that the function is not “executing a hashing function,” but rather to “provide access to records stored in a memory of the system.” PLTFF’S SJ RESPONSE AT 5. According to Bedrock, the linked lists provide access to

³¹ This term is contained in claim 5.

records stored in memory and therefore the corresponding structure includes code, such as the pseudo-code of Definitions depicted below, that defines a hash table and the externally chained linked lists. *Id.* at 4-5.

Definitions

The following formal definitions are required for specifying the insertion, retrieval, and deletion procedures. They are global to all procedures and functions shown below.

```

1. const table_size                                /* Size of hash table. */
2. type list_element_pointer = ↑ list_element      /* Pointer to elements of linked list. */
3. type list_element =                             /* Each element of linked list. */
    record
        record_contents: record_type;
        next: list_element_pointer                /* Singly-linked list. */
    end
4. var table: array [0 . . . table_size - 1] of list_element_pointer /* Hash table. */
    Initial state of table: table[i] = nil ∀ i 0 ≤ i < table_size /* Initially empty. */

```

‘120 patent at cols. 9-10.

The Court finds § 112 ¶ 6 applies, and will construe the term “a hashing means to provide access . . .” as a means-plus-function term. The claim language explicitly states the function is “to provide access to records stored in a memory of the system and using an external chaining technique to store the records with same hash address at least some of the records automatically expiring.”

As indicated by the explicit functional language of the claim, the structure corresponding to this particular means-plus-function limitation does not require structure executing a hashing function. Rather, Claim 5 describes the use of a hash table to provide access to records. Because the corresponding structure must perform the function stated in the claim language, structure for forming a hash table is not needed.³² However, software instructions are needed to provide a hash

³² However, the specification does provide hashing execution algorithms. In addition to stating known methods of hashing, i.e. “truncation, folding, transposition, modulo arithmetic, and combinations of these operations,” ‘120 patent at 5:5-7, the specification provides a discussion of hashing techniques. ‘120 patent at 4:53-5:52.

table with an attached linked list. In other words, software must be programmed with code to provide a hash table having a pointer to the head of a linked list of externally chained records.

Bedrock is correct to the extent that external chaining to a hash table provides access to stored records. The specification states in relevant part:

A common collision resolution strategy, with which the present invention is concerned, is called external chaining. *Under external chaining, each hash table entry stores all of the records that collided at that location by storing not the records themselves, but instead a pointer to the head of a linked list of those same records.* Such linked lists are formed by storing the records individually in dynamically allocated storage and maintaining with each record a pointer to the location of the next record in the chain of collided records. *When a search key is hashed to a hash table entry, the pointer found there is used to locate the first record.*

‘120 patent at 5:16-26 (emphasis added). The pointer from the hash table entry to the linked list allows the system to acquire the first record. In other words, the hash table with a pointer to an external chain of linked records provides “access to records stored in a memory of the system and us[es] an external chaining technique to store the records with same hash address at least some of the records automatically expiring.” In addition, the specification provides pseudo-code to provide a hash table with a pointer to the head of the linked list. *See* ‘120 patent at cols. 9-10 (Definitions section of Appendix). Therefore, the specification provides adequate structure to perform the function recited in the means-plus-function limitation.

Thus, the corresponding structure³³ is: CPU 10, and RAM 11 of FIG. 1 and col. 3 lines 52-56 and portions of the application software, user access software or operating system software, as described at col. 4, lines 22-48, programmed with software instructions to provide a hash table

³³ At the October 7, 2010 hearing, Defendants argued that claiming CPU and RAM merely claims a general purpose computer. Bedrock asserted that the ‘120 patent described a specially programmed computer. The Court does not disagree, but does not need to find structure based on a specially programmed computer. Instead, the corresponding structure refers to hashing execution algorithms. *See supra* note 32.

having a pointer to the head of a linked list of externally chained records as described in col. 5 lines 16-26 and/or programmed with software instructions as described in the pseudo-code of Definitions, definition number 4, and equivalents thereof.

XII. “means for dynamically determining maximum number”³⁴

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|---|-----------------------------------|
| <p>Function: The recited function is dynamically determining maximum number of records for the record search means to remove in the accessed linked list of records.</p> <p>Structure: The corresponding structure is: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56; and (2) Executable software, as described in col. 6, line 56 - col. 7, line 15, that dynamically chooses among removal strategies (e.g., chooses whether to execute Search Table Procedure [cols. 11-12] or Alternate Version of Search Table Procedure [cols. 11-14]) “at the time the record search means is invoked by the caller, thus sometimes removing all expired records, at other times removing some but not all of them, and yet at other times choosing to remove none of them. Such a dynamic decision can be based on factors such as, for example, how much memory is available in the system storage pool, general system load, time of day, the number of records currently residing in the information system, and other factors both internal and external to the information storage and retrieval system itself” (col. 7, lines 1-10), or the equivalent thereof.</p> | <p>Indefinite</p> |

³⁴ This term is contained in claims 2 and 6.

The Court finds the term falls under § 112 ¶ 6 and will construe the term as a means-plus-function term.

It seems that the parties agree that the function of the term is “dynamically determining maximum number of records for the record search means to remove in the accessed linked list of records.” *See* DFTS’ SUMMARY JUDGMENT MOTION AT 9. However, Defendants contend that this term is indefinite because the specification does not provide a corresponding algorithm “that determines the maximum number of records to remove.” *Id.*; *see* DFTS’ RESPONSE AT 23-24. Bedrock suggests otherwise, pointing to the Search Table Procedure and the Alternate Version of Search Table Procedure to show that such an algorithm is disclosed. PLTFP’S BRIEF AT 26 (citing ‘120 patent at 6:56-7:15). Bedrock further argues that the choice between using the Search Table Procedure or the Alternate Version of Search Table Procedure is the dynamic determination of the maximum number of records. PLTFP’S SJ RESPONSE AT 7-8.

The function recited in the term “means for dynamically determining maximum number” is “dynamically determining maximum number for the record search means to remove in the accessed linked list of records.” Therefore, the corresponding structure must dynamically determine the maximum number of records for the record search means to remove.

Bedrock is correct that the dynamic decision is the decision to run the Search Table Procedure or the Alternate Search Table Procedure. The specification describes methods of on-the-fly garbage removal that removes (1) all expired records; (2) some, but not all, expired records; or (3) none at all. *See* ‘120 patent at 6:66-7:4. First, the specification discusses the process by which all records are removed, as described in Figure 3 and in the pseudo-code of the Search Table

Procedure. See '120 patent at 6:56-59, cols. 11-12, and Figure 3. Next, the specification states that "PASCAL-like pseudocode for [an] alternate version of search table appears in the APPENDIX," which adapts the procedure described in Figure 3 to remove some, but not all, of the expired records, or even no records at all. '120 patent at 6:59-7:4; *see also* '120 patent at cols. 11, 12, 13 & 14. Then the specification states that "[s]uch a dynamic runtime decision might be based on" various factors. '120 patent at 7:4-15. The language of the specification indicates that the decision to run the Search Table Procedure or the Alternate Search Table Procedure is the dynamic runtime decision referred to in the term "means for dynamically determining maximum number." Therefore, the portions of the specification describing the Search Table Procedure, the Alternate Search Table Procedure, and the corresponding pseudo-code are the structures corresponding to the recited function.

Thus, the corresponding structure is: CPU 10, and RAM 11 of FIG. 1 and col. 3 lines 52-56 and portions of the application software, user access software or operating system software, as described at col. 4, lines 22-48, programmed with software instructions to dynamically determine a maximum number of records to remove by choosing a search strategy of removing all expired records from a linked list or removing some but not all of the expired records as described in col. 6 line 56 – col. 7 line 15 and/or programmed with software instructions to dynamically determine a maximum number of records to remove by choosing between the pseudo-code of the Search Table Procedure (cols. 11 and 12) or Alternative Version of Search Table Procedure (cols. 11, 12, 13, and 14), and equivalents thereof.

XIII. “mea[n]s, utilizing the record search means, for inserting, retrieving, and deleting from the system and, at the same time, removing at least some of the expired ones of the records in the accessed linked list of records”³⁵

| Plaintiff’s Proposed Construction | Defendants’ Proposed Construction |
|---|---|
| <p>Function: utilizing the record search means, inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.</p> <p>Structure: (1) Portions of the application software, user access software or operating system software, as described at col. 4, lines 30-48 and illustrated in FIG. 2, of a computer system that includes at least a CPU 10 and RAM 11, see FIG. 1 and col. 3 lines 52-56. (2) Executable software which provides the insert, retrieve, or delete record capability illustrated in the flowchart of FIG. 5, FIG. 6, or FIG. 7, respectively, and/or as pseudo code of Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9, 10, 11, and 12), or Delete Procedure (cols. 11 and 12), respectively, and/or described in col. 7, line 65 - col. 8, line 32, col. 8, lines 33-44, or col. 8 lines 45-59, or the equivalents thereof.</p> | <p>Function: utilizing the record search means, [accessing the linked list / inserting, retrieving, and deleting from the system] and, at the same time, removing at least some of the expired ones of the records in the linked list.</p> <p>"At the same time" means during the same traversal of the linked list as [accessing the linked list / inserting, retrieving, and deleting records from the system].</p> <p>For the construction of “removing ...”, see proposed construction above, as further described in Joint Claim Construction Statement [Dkt. 251].</p> <p>Structure: Boxes 10 and 11 of Fig. 1; Figs. 4-7, pseudocode in the Search Procedure (cols. 11-14), Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9 and 10), Delete Procedure</p> |

As an initial matter, the claim term is governed by § 112 ¶ 6 and will be construed as a means-plus-function term.

The parties essentially reiterate the arguments set forth in Section X above. However, instead of arguing whether the function requires inserting, retrieving *and* deleting, the parties agree that Claim 5 requires all three. DFTS’ RESPONSE AT 23, n.12. Therefore, the Court finds the claimed function of the term “mea[n]s utilizing the record search means, for inserting, retrieving and

³⁵ This term is contained in claim 5.

deleting” is “utilizing the record search means, inserting, retrieving, and deleting records from the system and, at the same time, removing at least some expired ones of the records in the accessed linked list of records.” Additionally, the corresponding structure is: CPU 10 and RAM 11 of FIG. 1 and col. 3 lines 52-56 and portions of the application software, user access software or operating system software, as described at col. 4, lines 22-48, programmed with software instructions that provide the insert, retrieve, and delete record capability as described in the flowchart of FIG. 5 and col. 7 line 65 – col. 8 line 32, FIG. 6 and col. 8 lines 33-44, or FIG. 7 and col. 8 lines 45-59, respectively, and/or programmed with software instructions that provide the insert, retrieve and delete record capability as described in the pseudo-code of Insert Procedure (cols. 9 and 10), Retrieve Procedure (cols. 9, 10, 11, and 12), and Delete Procedure (cols. 11 and 12), respectively, and equivalents thereof. *See discussion supra* Section X.

CONCLUSION

For the foregoing reasons, the court adopts the constructions set forth above.

So ORDERED and SIGNED this 10th day of January, 2011.



JOHN D. LOVE
UNITED STATES MAGISTRATE JUDGE