

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

SOFTWARE TREE, LLC

§

v.

§

NO. 6:09-CV-097

§

RED HAT, INC., et al.

§

§

MEMORANDUM OPINION & ORDER

This claim construction opinion construes the disputed terms in U.S. Patent No. 6,1366,776 (“the ‘776 patent”). Plaintiff Software Tree, LLC (“Plaintiff”) accuses Defendant Red Hat, Inc. (“Defendant”)¹ of patent infringement. The parties have submitted a number of claim terms for construction and filed claim construction briefs (Doc. Nos. 114, 131, 149, 157). A claim construction hearing was held on February 2, 2010. On March 29, 2010, the Court issued a provisional claim construction order (Doc. No. 177). For the reasons stated herein, the Court adopts the constructions set forth below.

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*

¹ Defendants Genuitec, LLC, Dell, Inc., and Hewlett-Packard Company were dismissed from this case after the initial claim construction briefing (Doc. Nos. 145, 187, 188).

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. "[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 than 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. Elan 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. 1998) (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.*

DISCUSSION

A. Overview of the Patent

The ‘776 Patent discloses a “system and method for exchanging data and commands between an object oriented system and a relational system.” ‘776 Patent 1:15-16. The patent relates to object-relational mapping (“ORM”), which provides an interface between an object-oriented user program and a relational database. DEF.’S BR. at 2. ORM enables a user to “obtain information from, or add information to, a database without having to interact with the database directly. Instead the user can interact with a user-friendly application that presents data in a visually appealing and intuitive manner.” *Id.* The patented system includes “an [ORM] Grammar, an ORM specification, Object Class Definitions, a relational database, an operating system, a Database Exchange Unit including an OR mapping unit, a schema generator, a schema reverse engineering unit and applications.” ‘776 Patent 2:57-62.

B. Agreed Terms

The Court adopts the parties' agreed constructions for the following terms:

Term	Parties' Agreed Construction
ORM Specification <i>Claims 1, 2, 4, 6, 9, 11, 28, 43, 44, 46, 48, 51, 57, 60-62, 65</i>	"a textual specification that is based on an ORM grammar and includes information for defining a mapping between an object-oriented system and a relational system"
ORM Data Structure(s) / object relational mapping data structure(s) <i>Claims 1, 2, 4, 5, 6, 10-12, 14, 28, 43, 44, 46, 47, 48, 57, 58, 59, 65</i>	"in-memory data structure(s) that are based, in part, on either an ORM specification or an ORM Template Specification, and contain information about an object-relational mapping"
beginning a new transaction / committing the transaction to the database <i>Claim 40</i>	"starting one or more operations against a database" / "ending the transaction, including saving to the database any changes made as part of the transaction"
query context <i>Claims 41, 42</i>	"set of data describing the state of the current object streaming process, including at least a query cursor"
unit <i>Claim 1, 2, 3, 4, 6, 9-14, 28, 40, 42-44, 46, 48, 49, 51, 57, 60-62, 65</i>	"one or more programs, routines, modules or tools"

C. Disputed Terms

The Court construes the following disputed terms.

Term	Plaintiff's Proposed Construction	Defendant's Proposed Construction
ORM grammar <i>Claims 1, 2, 4, 43, 46, 57</i>	"the extensible set of rules, including syntax, for textually describing a mapping between an object-oriented system and a relational	"the extensible set of rules that define syntax and vocabulary for generating and interpreting an ORM specification"

	system in a declarative way”	
--	------------------------------	--

The Court begins by noting the specification states ORM grammar “define[s] the mapping between an object-oriented system and a relational system” by providing “the rules for textually describing an [ORM] in a declarative way.” ‘776 Patent 5:42-48. The parties build on this initial definition to further clarify the term. Although they agree ORM grammar is an “extensible set of rules,” they disagree as to whether it: 1) includes both syntax and vocabulary; 2) may be used to “textually describe[] a mapping . . . in a declarative way;” and 3) whether it must be used to “generate” an ORM specification.

Plaintiff argues the term “vocabulary” does not appear in the specification and therefore has no place in the Court’s construction. PL.’S BR. at 7-8. Defendant cites a technical dictionary to describe a grammar as having a “set of rules” and an “alphabet” of symbols, which they analogize to syntax and vocabulary, respectively. DEF.’S BR. at 7. Defendant further cites the inventor’s deposition testimony, wherein when asked “And by rules, you mean the set of vocabulary and syntax necessary to describe how you map an object to a relational database, right?” he responded, “Right, vocabulary, key words, syntax, semantics.” *Id.* Plaintiff argues this testimony is unpersuasive because the inventor had previously described ORM Grammar as “defin[ing] the rules for textually specifying the [ORM] specification” and only used the term vocabulary in response to a leading question. PL.’S REPLY. at 2-3.

Defendant does not cite, and the Court does not find, anything in the intrinsic evidence supporting the inclusion of vocabulary in the construction of ORM grammar. Furthermore, the Court is not persuaded by Defendant’s extrinsic evidence. *See Phillips*, 415 F.3d at 1317 (cautioning against reliance on extrinsic evidence); *see also Voice Techs. Group, Inc. v. VMC Sys., Inc.*, 164 F.3d

605, 615 (Fed. Cir. 1999) (holding an “inventor can not by later testimony change the invention and the claims from their meaning at the time the patent was drafted and granted”). Accordingly, the Court declines to include vocabulary in its construction of the term.

Likewise, the Court is not persuaded by Defendant’s argument that the phrase “The ORM Grammar 200 [sic] the rules for textually describing an [ORM] in a declarative way” describes the ORM specification, rather than the ORM grammar. DEF.’S BR. at 9-10 (quoting ‘776 Patent 5:46-48). Relying solely on attorney argument, Defendant states the terms “textually” and “declarative” describe the ORM specification. Despite the typographical error, the specification’s meaning is clear. ORM grammar is the set of rules used to textually describe an ORM in a declarative way. ‘776 Patent 5:42-48. The Court finds it appropriate to include these terms in its construction.

Finally, the Court finds Defendant’s proposal that the ORM grammar is “for generating and interpreting an ORM specification” overly limiting. Plaintiff argues this phrasing implies ORM grammar must be used to automatically create an ORM specification, thereby excluding disclosed embodiments where users can manually create an ORM specification. PL.’S BR. at 7 (citing ‘776 Patent 6:17-22). Defendant argues the specification expressly refers to using the ORM grammar to generate and interpret the ORM specification, DEF.’S BR. at 8 (citing ‘776 Patent 5:55-58), and contends manual creation is simply a form of “generation.” *Id.* at 9 n.16. Defendant is correct that the specification teaches use of the ORM grammar to interpret and generate the ORM specification, *see* ‘776 Patent 5:55-58 (stating “the ORM Grammar 200 may in alternate embodiments be used to generate and interpret an ORM Specification 202”), and the Court agrees that an ORM grammar must be used to interpret or generate an ORM specification. *Id.* 21:10-11 (reciting the “ORM Specification [is] based on an ORM Grammar”). However, the Court agrees with Plaintiff that the

word “generation” can mislead jurors by implying automation, to the exclusion of the manual creation embodiment. *See* ‘776 Patent 6:17-22 (stating “[a]ny one of a variety of text editor or graphical user interfaces can be used to create, modify and review ORM Specifications 202”).

For the foregoing reasons, the Court adopts Plaintiff’s proposal. In the Court’s provisional claim construction order, the Court originally appended the phrase “and that may be used to generate and interpret an ORM specification” (Doc. No. 177 at 2). Having further considered the matter, the Court finds the phrase unnecessary. The Court originally suggested the phrase during the *Markman* hearing in an effort to find agreement between the parties’ positions. The purpose of the language was to communicate that “generation,” as understood to mean automated or computer-implemented, was not required in every case. Plaintiff’s original proposal, however, does not use this potentially confusing word. The ORM grammar is a set of rules for textually describing an ORM specification, *i.e.*, a textual description of a mapping between an object-oriented system and a relational system, in a declarative way. Plaintiff’s original proposal accurately conveyed this meaning and this definition corresponds to the parties’ agreed definition of ORM specification. Accordingly, the Court revises its provisional construction and adopts Plaintiff’s proposal as its final construction.

For the foregoing reasons, the Court finds the proper construction of this term is “the extensible set of rules, including syntax, for textually describing a mapping between an object-oriented system and a relational system in a declarative way.”

Term	Plaintiff’s Proposed Construction	Defendant’s Proposed Construction
input(s) and output(s) <i>Claims 2, 5, 13, 43, 58, 62</i>	Plaintiff contends that this term does not require construction. In the alternative only, Plaintiff	“information received and sent”

	<p>contends that this term should accorded the meaning that would have been understood by one of ordinary skill in the art at the time of the invention, which is:</p> <p>“the input interface(s) and output interface(s)”</p>	
--	--	--

The parties disagree whether the terms “inputs” and “outputs” describe structures or information. *Compare* Pl.’s Br. at 21 *with* DEF.’s BR. at 15. Plaintiff argues the terms are used in apparatus claims and refer to structure, such as the input and output interfaces on a software module. PL.’s BR. at 21. Defendant contends the specification uses the terms to refer to information, such as the data received or sent by a software module. DEF.’s BR. at 15. Defendant further contends the specification uses the terms in conjunction with the term “device” when reference to an interface is intended. *Id.* at 17. Defendant argues if the applicant had intended to claim a structural interface, then the applicant could have so indicated by using the term “input device.” *Id.*

“Input” and “output” are ordinary words that may be used in various ways having different meanings and even embodying different parts of English grammar (e.g., they can be verbs or nouns). Thus, it is not surprising that strict analysis of the specification will find such varying uses that may be characterized as inconsistency. Defendant accurately observes the intrinsic evidence at times refers to inputs and outputs as data or information. *See* ‘776 Patent 7:13-16 (stating “[t]his Schema Generator 212 takes the name of the file (e.g., abcjdx) containing the [ORM] information as input . . .”); DEF.’s BR. at EX. 6, June 19, 2006 Office Action Response at 23 (explaining in a prior art reference “an SMDL parser takes SMDL as input and produces SMIR as output”). In other instances, however, the intrinsic record uses these terms to refer to interfaces to a software module.

See ‘776 Patent 13:21-23 (stating “[t]he output of the Relational Schema Statements Generation Unit 602 is coupled to the input of the Relational Schema Statements Application Unit 606”); 12:53-56 (explaining the Unit 602 and Unit 606 “are coupled by Bus 114”). The Court also recognizes in certain instances the specification refers to input and output devices. *See* ‘776 Patent 4:51-54 (stating “the output device 108 is preferably a video monitor; and the input device 106 is preferably a keyboard and mouse-type controller”). Thus, the specification at times uses the terms to refer to information that is sent or received, and at other times to refer to interfaces between components of the invention.

The Court finds, within the context of the claims, the terms input and output refer to interfaces. Although claims must be viewed in light of the specification, *Vitronics*, 90 F.3d at 1582, the claim language and the context in which terms are used guides the Court’s construction. *Phillips*, 415 F.3d at 1314. In the claim language, the terms are most clearly used to describe interfaces. *See* ‘776 Patent 21:30-32 (reciting “a database interface unit having inputs and outputs . . . the inputs and outputs coupled to the relational system and the mapping unit”); 21:57-60 (reciting “the input of the schema generator coupled to the object class definition and the object relational data structure, the output of the schema generator coupled to the relational system”). For example, in some claims, the terms are referring to input and output interfaces through which data or information travels. *See id.* at 21:17-18 (reciting “an object call processing unit having inputs and outputs *for receiving object calls*” (emphasis added)), 21:29-30 (reciting “a database interface unit having inputs and outputs, *for retrieving and storing data*” (emphasis added)). Relying on the written description’s use of the terms, Defendants argue it is possible to read the claims in a way where inputs and outputs are information. If Defendant’s argument were applied in the context of the claim phrase “the output

of the schema generator coupled to the relational system,” Defendants would presumably argue that the word “coupled” simply indicates the schema generator provides output information to the relational system. While tenable, the Court finds Defendant’s interpretation forced and awkward. The Court declines to exclude the contextually ordinary and commonsense interpretation of these words in favor of Defendant’s proposal.

Finally, as to Defendant’s argument that the term “device” is determinative of when input or output are referring to an interface, the Court rejects it for two reasons. First, in portions of the written description, the terms refer to interfaces without the use of the word “device.” *See, e.g., id.* at 13:21-23 (describing output as coupling two units together). Second, in the context of the claim language the terms are used more clearly and consistently to refer to interfaces and not data. *See id.* at 21:29-30 (distinguishing “inputs and outputs” from data). Indeed, as used in claim 2, inputs and outputs must be distinct from data because they are used “for retrieving and storing data.” *Id.* Thus, in the context of the claims, input and output are interfaces.

Having resolved the parties’ claim scope dispute, the Court finds the terms do not require construction because their meanings are clear in the context of the claim and will be readily understandable to the jury. *O2 Micro Int’l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1362 (Fed. Cir. 2008); *Fenner Inv. Ltd. v. Microsoft Corp.*, No. 6:07-cv-8, 2008 WL 3981838 at *3 (E.D. Tex. Aug. 22, 2008) (finding that a court need not construe a disputed term as long as it has resolved the claim scope dispute between the parties). Although the Court does not construe this term, the parties may not interpret this term in a manner that is inconsistent with this opinion.

Term	Plaintiff’s Proposed Construction	Defendant’s Proposed Construction
-------------	--	--

<p>(a) a database interface unit having inputs and outputs, for retrieving and storing data in the relational system <i>Claim 2</i></p> <p>(b) a database interface unit for retrieving and storing data and metadata in the relational system <i>Claim 57</i></p> <p>(c) a database interface unit for applying the statements on the relational system <i>Claims 6, 44, 48</i></p>	<p>Plaintiff contends these phrases do not require construction. In the alternative only, Plaintiff contends these phrases should be accorded the meanings that would have been understood by one of ordinary skill in the art at the time of the invention, which are:</p> <p>(a) “Unit having the input interface(s) and output interface(s), for retrieving and storing data in the relational system.”</p> <p>(b) “Unit used to retrieve and store data, including metadata, in the relational system.”</p> <p>(c) “Unit used to retrieve and store data in a relational database system, and to apply statements to the database.”</p> <p>Plaintiff further contends these limitations are not governed by 35 U.S.C. § 112(6). In the alternative only, the following constructions are proposed:</p> <p>(a) <u>Function</u>: retrieve and store data in a relational database system.</p> <p><u>Structure</u>: the Database Interface Unit with input interface(s) and output interface(s) as illustrated in</p>	<p>Defendant contends these limitations are means-plus-function limitation under 35 U.S.C. § 112, paragraph 6, are propose the following constructions:</p> <p>(a) <u>Function</u>: “retrieving and storing data in the relational system” <u>Structure</u>: Not disclosed.</p> <p>(b) <u>Function</u>: “retrieving and storing data and metadata in the relational system” <u>Structure</u>: Not disclosed.</p> <p>(c) <u>Function</u>: “applying the statements on the relational system” <u>Structure</u>: Not disclosed.</p> <p>Alternatively, if these are not considered means-plus-function limitations:</p> <p>(a) A unit having inputs and outputs that interface with a database by retrieving and storing data in the relational system.</p> <p>(b) A unit that interfaces with a database by retrieving and storing data and metadata in the relational system.</p> <p>(c) A unit having inputs and outputs that interfaces with a database by sending the</p>
---	--	--

	<p>Figs. 4, 5, 6, 7, 8, and 9 of various embodiments.</p> <p>(b) <u>Function</u>: retrieve and store data, including metadata, in a relational database system</p> <p><u>Structure</u>: the Database Interface Unit with input interface(s) and output interface(s) as illustrated in Figs. 4, 5, 6, 7, 8, and 9 of the various embodiments disclosed in the patent, and their equivalents.</p> <p>(c) <u>Function</u>: retrieve and store data in a relational database system, and to apply statements to the database.</p> <p><u>Structure</u>: the Database Interface Unit with input interface(s) and output interface(s) as illustrated in Figs. 4, 5, 6, 7, 8, and 9 of various embodiments.</p>	<p>statements to the relational database to be acted on by the relational database.</p>
--	---	---

“[A] claim term that does not use ‘means’ will trigger the rebuttable presumption that § 112 ¶ 6 does not apply.” *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1369 (Fed. Cir. 2002). “This presumption can be rebutted ‘by showing that the claim element recite[s] a function without reciting sufficient structure for performing that function.’” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 469 F.3d 1005, 1023 (Fed. Cir. 2006) (quoting *Watts v. XL Sys., Inc.*, 232 F.3d 877, 880-81 (Fed. Cir. 2000)). “[T]he presumption flowing from the absence of the term ‘means’ is a

strong one that is not readily overcome.” *Lighting World, Inc. v. Birchwood Lighting, Inc.*, 382 F.3d 1354, 1358 (Fed. Cir. 2004).

Defendant attempts to rebut the law’s strong presumption by arguing “[t]he phrase ‘database interface unit . . . for’ is a mean-plus-function limitation under § 112 ¶ 6 because it recites function without ‘sufficiently definite’ structure.” DEF.’S RESP. at 19. Defendant further argues “unit” is a nonce word and that “database interface unit” does not have a generally understood, non-generic meaning to a skilled artisan. DEF.’S RESP. at 19-20; DEF.’S REPLY at 7-8. Plaintiff responds, arguing the “database interface unit” terms connote sufficient structure such that the presumption that § 112 ¶ 6 does not apply cannot be overcome. PL.’S BR. at 27-28.

Although the “database interface unit” terms connote some function, they more prominently connote structure and are stated as structure. Indeed, a skilled artisan would understand their implementation as a well-known and definite structure within a software architecture. The written description of the invention describes a software architecture for accomplishing the claimed invention. *See generally* ‘776 Patent 4:39-20:67. Particular figures and their accompanying text detail the software components that makeup this overall architecture. *See, e.g., id.* at 9:66-10:27 (describing the functions performed by and the connections to Database Exchange Unit 210). These components consist of smaller modules or routines which operate in a specifically claimed concert with other modules or routines to perform the described functions. *See, e.g., id.* at 10:28-44 (describing the modular makeup of Database Exchange Unit 210). The “database interface unit” terms describe a particular module utilized by the Database Exchange Unit, the Schema Generator, and the Schema Reverse Engineering Unit. *See id.* at Figures 5-9, 10:28-14:34. Throughout the written description and the claim language, the terms are used to signify a software structure or

module, connected to other software structures and possessing a specific place in the overall software architecture. *See, e.g., id.* at 21:29-32 (claiming “a database interface unit having inputs and outputs . . . coupled to the relational system and the mapping unit”). In each instance, a “database interface unit” is simply the module or component that interacts with the relational database. *E.g., id.* Accordingly, the terms recite sufficient structure and the presumption that § 112 ¶ 6 does not apply is not rebutted. One of ordinary skill in the art would understand the database interface unit connotes structure and would further understand the nature of that structure in the overall claimed architecture. *See Alcatel USA Res., Inc. v. Microsoft Corp.*, No. 6:06-cv-500, 2008 WL 2625852, at *17-18 (E.D. Tex. June 27, 2008).

Therefore, the Court finds these terms are not means-plus-function claims subject to interpretation under § 112 ¶ 6. Having resolved the parties’ dispute, the Court finds the terms do not require construction because their meanings are clear in the context of the claim and will be readily understandable to the jury. *O2 Micro Int’l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1362 (Fed. Cir. 2008); *Fenner Inv. Ltd. v. Microsoft Corp.*, No. 6:07-cv-8, 2008 WL 3981838 at *3 (E.D. Tex. Aug. 22, 2008) (finding that a court need not construe a disputed term as long as it has resolved the claim scope dispute between the parties). The Court recognizes the terms, standing alone, would be difficult for a lay jury to understand. Read in context, however, the claim language explains what a database interface unit is and what it does.² Although the Court does not construe these terms, the parties may not interpret them in a manner that is inconsistent with this opinion.

² The Court’s conclusion is further supported by the parties’ alternative proposals, which essentially restate the claim language. The Court finds a restatement of the claim language unnecessary.

Term	Plaintiff’s Proposed Construction	Defendant’s Proposed Construction
<p>persistently unique sequence numbers <i>Claims 13, 62</i></p>	<p>Plaintiff contends that this term does not require construction. In the alternative only, Plaintiff contends that this term should be accorded the meaning that would have been understood by one of ordinary skill in the art at the time of the invention, which is:</p> <p>“Unique sequence numbers used to ensure that all objects using the particular named sequence generator (e.g., whether newly created or persisted in a database) will have different identification numbers.”</p>	<p>“unique sequence numbers used to ensure that all objects (i.e., whether newly created or already persisted in the database) will have different identification numbers”</p>

Plaintiff argues the construction of “persistently unique sequence numbers” must allow for multiple named sequence generators. PL.’S BR. at 14-15. Defendant argues the word “unique” and portions of the specification support a conclusion that the sequence numbers are universally unique. DEF.’S RESP. at 29-31. Essentially, Plaintiff’s construction would only require that every sequence number generated by a given sequence generator is unique amongst the numbers generated by that sequence generator, whereas Defendant’s construction would require every sequence number to be unique amongst all numbers generated by all sequence generators.

The specification describes the behavior of named sequence generators. ‘776 Patent 9:9-65. “Named Sequence Generators 226 are routines or modules that generate persistently unique sequence numbers.” *Id.* at 9:10-12. “These sequence numbers . . . can be used to assign unique ids to different

objects.” *Id.* at 9:26-27. The specification describes sequence numbers as unique only within a particular domain. *Id.* at 9:34-39 (describing “the way sequences are named by the Named Sequence Generators 226 is done in a manner convenient for application programmers because it uses meaningful names (e.g., imageIdSequences) for the sequence generators, and allows the creation of multiple sequence number domains (e.g., partIdSequences, customerIdSequences, etc.)”). Although portions of the specification describe reserving blocks of numbers for use with a particular application, *see id.* at 9:55-65, this utility operates within a given domain and does not require universal uniqueness. *See id.* at 20:9-19 (describing a reserved block of numbers as “a unique set of numbers with respect to the given sequenceName”). Thus, Defendant’s proposal is unsupported by the specification and improperly excludes a preferred embodiment. *See Vitronics Corp.*, 90 F.3d at 1583 (noting an interpretation that excludes a preferred embodiment is “rarely, if ever, correct”).

Accordingly, the Court finds the proper construction of this term is “unique sequence numbers used to ensure that all objects using the particular named sequence generator (i.e., whether newly created or already persisted in a database) will have different identification numbers.”

Term	Parties’ Partially Agreed Construction
<p><i>The order of the steps in claim 40.</i></p> <p>40. A method for object streaming comprising the steps of: [a] beginning a new transaction; [b] generating a query call to a database exchange unit for a plurality of objects; [c] returning the predetermined number (X) of objects; [d] processing the</p>	<p>Plaintiff does not believe Claim 40 should be construed to include an order of steps. If, however, the Court finds it should be construed to include an order of steps, then Plaintiff agrees:</p> <ol style="list-style-type: none"> 1) Step (a) must start before step (c); 2) Step (a) must occur before step (h); 3) Step (b) must occur at least once before step (c) begins; and 4) Step (e) must occur at least once before step (f) begins. <p>Defendant concedes the remainder of the steps, other than steps (c) and (d), either are not required to be performed in any order or may be performed simultaneously.</p>

<p>returned objects; [e] determining whether more objects are to be retrieved through the current stream of objects; [f] retrieving and processing an additional number (m) of objects if more objects are to be retrieved through the current stream of objects; [g] generating a query close to the database exchange unit; and [h] committing the transaction to the database.</p>	<p>The parties disagree whether step (c) must be completed before step (d) begins.</p>
---	--

“Unless the steps of a method actually recite an order, the steps are not ordinarily construed to require one. However, such a result can ensue when the method steps implicitly require that they be performed in the order written.” *Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323-1342 (Fed. Cir. 2001). To determine whether the steps of a method claim must be performed in a given order, the Court first looks “to the claim language to determine if, as a matter of logic or grammar, they must be performed in the order written.” *Altiris, Inc. v. Symantec Corp.*, 318 F.3d 1363, 1369 (Fed. Cir. 2003). If the claim language itself does not require such a construction, the Court may find the rest of the specification requires it either implicitly or explicitly. *Id.* at 1370.

This is a case where the claim language requires, as a matter of logic, that some of the steps be performed in a certain order. For example, the step of “beginning a new transaction” certainly must occur before the step of “committing the transaction to the database,” *i.e.*, ending the transaction. Accordingly, the Court construes claim 40 to include an order of steps. The Court adopts the parties’ agreement that: 1) step (a) must start before step (c); 2) step (a) must occur before step (h); 3) step (b) must occur at least once before step (c) begins; 4) step (e) must occur at least

once before step (f) begins; and 5) the remainder of the steps, other than steps (c) and (d), either are not required to be performed in any order or may be performed simultaneously.

The parties disagree whether step (c) must be completed before step (d) begins. The claim language logically requires, and the rest of specification supports, *see* '776 Patent Figure 15, 18:8-53, the conclusion that an object must be returned before a "returned object" can be processed. Defendant argues step (c) must be wholly completed before step (d) begins; that is, all of the objects must be returned before any of the objects can be processed. Plaintiff concedes that step (c) must be completed in part - that a given object must be returned before it can be processed - but argues that step (d) may begin with respect to a returned object before all objects are returned. Having reviewed the claim language and the rest of the specification, the Court is persuaded Plaintiff's position is correct. Although a given object must be returned before it is processed, neither the claim nor the specification requires that all objects must be returned before any are processed. The Court notes that the literal words of step (d) ("processing the returned objects") allows for either; step (c) to entirely complete prior step (d) beginning (as in "processing returned objects" after the last predetermined number is returned); or step (d) to begin before step (c) completes (as in "processing the returned objects" as they are returned). Accordingly, the Court finds step (c) must start before step (d), and for any particular object step (c) must be completed before step (d) starts.

CONCLUSION

For the foregoing reasons, the Court interprets the claim language in this case in the manner set forth above. For ease of reference, the Court's claim interpretations are set forth in the table attached to this order as Appendix A.

So ORDERED and SIGNED this 28th day of May, 2010.

APPENDIX A

Term	Court's Construction
ORM grammar	“the extensible set of rules, including syntax, for textually describing a mapping between an object-oriented system and a relational system in a declarative way”
ORM specification	“a textual specification that is based on an ORM grammar and includes information for defining a mapping between an object-oriented system and a relational system”
ORM Data Structure(s) / object relational mapping data structure(s)	“in-memory data structure(s) that are based, in part, on either an ORM specification or an ORM Template Specification, and contain information about an object-relational mapping”
input(s) and output(s)	No Construction
<p>(a) a database interface unit having inputs and outputs, for retrieving and storing data in the relational system</p> <p>(b) a database interface unit for retrieving and storing data and metadata in the relational system</p> <p>(c) a database interface unit for applying the statements on the relational system</p>	No Construction
beginning a new transaction / committing the transaction to the database	“starting one or more operations against a database” / “ending the transaction, including saving to the database any changes made as part of the transaction”

<p>query context</p>	<p>“set of data describing the state of the current object streaming process, including at least a query cursor”</p>
<p>persistently unique sequence numbers,</p>	<p>“unique sequence numbers used to ensure that all objects using the particular named sequence generator (i.e., whether newly created or already persisted in a database) will have different identification numbers.”</p>
<p>unit</p>	<p>“one or more programs, routines, modules or tools”</p>
<p><i>The order of the steps in claim 40.</i></p> <p>40. A method for object streaming comprising the steps of: [a] beginning a new transaction; [b] generating a query call to a database exchange unit for a plurality of objects; [c] returning the predetermined number (X) of objects; [d] processing the returned objects; [e] determining whether more objects are to be retrieved through the current stream of objects; [f] retrieving and processing an additional number (m) of objects if more objects are to be retrieved through the current stream of objects; [g] generating a query close to the database exchange unit; and [h] committing the transaction to the database.</p>	<ol style="list-style-type: none"> 1) step (a) must start before step (c); 2) step (a) must occur before step (h); 3) step (b) must occur at least once before step (c) begins; 4) step (e) must occur at least once before step (f) begins; 5) the remainder of the steps, other than steps (c) and (d), either are not required to be performed in any order or may be performed simultaneously; and 6) step (c) must start before step (d), and for any particular object step (c) must be completed before step (d) starts.