

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SERVICENOW, INCORPORATED,
Petitioner,

v.

INQUISIENT INCORPORATED,
Patent Owner.

IPR2023-01072
Patent 8,219,585 B2

Before BRIAN J. McNAMARA, GREGG I. ANDERSON, and
CHARLES J. BOUDREAU, *Administrative Patent Judges*.

McNAMARA, *Administrative Patent Judge*.

DECISION
Denying Institution of *Inter partes* Review
35 U.S.C. § 314

I. INTRODUCTION

ServiceNow, Inc. (“Petitioner”) filed a petition, Paper 2 (“Petition” or “Pet.”), to institute an *inter partes* review (“IPR”) of claims 1–47 (the “challenged claims”) of U.S. Patent No. 8,219,585 B2 (Ex. 1001, “the ’585 patent”) under 35 U.S.C. § 311. InQuisient, Inc. (“Patent Owner”) filed a Preliminary Response, Paper 6 (“Prelim. Resp.”), contending that the Petition should be denied as to all challenged claims. We authorized Petitioner’s filing of a Reply to Patent Owner’s Preliminary Response (Paper 7, “Pet. Prelim. Reply”) and Patent Owner’s filing of a Preliminary Sur-reply (Paper 8, “PO Prelim. Sur-reply”).

We have jurisdiction under 35 U.S.C. § 6 (2018). This Decision on Institution is issued pursuant to 35 U.S.C. § 314, which provides that an *inter partes* review may not be instituted unless the information presented in the Petition “shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.”

A decision to institute under § 314 may not institute on fewer than all claims challenged in the petition. *SAS Inst., Inc. v. Iancu*, 138 S. Ct. 1348, 1359–60 (2018). In addition, per Board practice, if the Board institutes trial, it will institute “on all of the challenged claims and on all grounds of unpatentability asserted for each claim.” *See* 37 C.F.R. § 42.108(a) (2023).

Having considered all the arguments and the associated evidence of record, for the reasons described below, we decline to institute *inter partes* review.

II. REAL PARTIES IN INTEREST

The Petition identifies itself (ServiceNow, Inc.) as its real party-in-interest. Pet. 69.

Patent Owner identifies itself (InQuisient, Inc.) as its real party-in-interest. Paper 5, 1.

III. RELATED MATTERS

The parties state that the '585 patent is asserted in the following litigation: *InQuisient Inc. v. ServiceNow, Inc.*, 1-22-cv-00900 (D. Del.). Pet. 69; Paper 5, 1.

Petitioner also identifies U.S. Patent No. 7,979,468 B2 (“the '468 Patent”) that Petitioner challenges in *ServiceNow v. InQuisient, Inc.*, IPR2023-01071, as a patent related to the '585 patent.

IV. THE '585 PATENT

The '585 patent relates to systems and methods for storing, managing, indexing, interrelating and/or retrieving data sets. Ex. 1001, 1:15–18. The '585 patent states that a conventional system implements its own data model and rules according to a user interface, requiring developers to write dedicated code and user interface code that displays or manages data. *Id.* at 1:22–28. The '585 patent seeks to provide a system and method to store, manage, index, interrelate, and/or retrieve data sets in a manner that is independent of the data model. *Id.* at 1:52–54. The system includes a processor and data repository. *Id.* at 1:61–65. An exemplary entity-relationship diagram of an elements metadata repository (EMR) shown in Figure 1 includes eight modules recited in claim 1. *Id.* at 3:3–6. A database creator enters database information in the modules rather than writing dedicated code and user interface code. *Id.* at 8:55–65. In this elements-based approach, each of the elements conforms to rules of generalized things in the universe, e.g., through entries in an element relation table. *Id.* at 9:24–29.

The challenged claims recite a data repository that includes an element module configured to store and uniquely identify a plurality of elements each identified by a unique element identification. *Id.* at 2:1–3; 6:13–34 (describing elements as uniquely identified by an ID, name, and class identifier that points to an indicator of the type or classification of the element). The EMR places “things” into an element table—an element stored in an element table implies that the user can find similarity between any “thing” the user would like to discuss, report, store, or manage, without regard to their exact nature. *Id.* at 6:3–10.

The data repository also includes an element relation module configured to store one or more relationships between the elements in the element module. *Id.* at 2:4–6; 6:35–45 (describing an element relation table that stores relationships between elements, e.g., each entry in the element relation table contains a pointer to at least two elements in the elements table and a type definition describing the type of relationship).

A class module is configured to define at least one class of the elements and store the class. *Id.* at 1:66–2:7. The EMR also includes an attribute module configured to define and store the attributes and a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class, and a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements. *Id.* at 2:6–11. An attribute table 110 holds the definition of each attribute, e.g., the physical name and type of data that corresponds to the database column name and column type where the actual values are stored. *Id.* at 5:4–9. A class attribute table may associate a given attribute with one or more classes in a class table. *Id.* at 5:9–10.

A type definition module is configured to define and store one or more types of the class, the attributes related to the class and the relationship between the elements. *Id.* at 2:11–14.

The EMR can utilize entries in a named_text table to name structured query language (SQL) queries and filter query entries into one or more dynamic folders, such that the result of a query may vary over time. *Id.* at 10:64–11:39.

V. ILLUSTRATIVE CLAIMS

Claim 1, reproduced below with paragraph designations used in the Petition, is illustrative of the subject matter of the '585 patent.

- 1[pre]. A computerized system for manipulating data sets, the computerized system comprising:
- [a] a hardware-based processor; and
 - [b] a data repository adapted to process, retrieve, and store: data contained in the data sets; and one or more layers of metadata included in the data sets, wherein the data repository comprises:
 - [c] an element module configured to store and identify a plurality of elements, wherein the plurality of elements are identified by unique element identification;
 - [d] an element relation module configured to store one or more relationships among the plurality of elements stored in the element module;
 - [e] a class module configured to define at least one class of the plurality of elements and store the at least one class;
 - [f] an attribute module configured to define one or more attributes and store the one or more attributes;
 - [g] a class attribute module configured to define and store one or more class-attribute associations between at least one of the one or more attributes and the at least one class;
 - [h] a type definition module configured to define and store one or more types of the at least one class, a plurality of attributes related to the at least one class and the one or more relationships among the plurality of elements;

- [i] a named text module configured to store textual representations of query, control, and display mechanisms that relate to the plurality of elements, wherein the named text module defines one or more folders and one or more filters; and
- [j] a tuple module configured to generate and store one or more tuples, wherein the one or more tuples link the one or more folders and the one or more filters to generate a hierarchy of one or more data entries.

Ex. 1001, 16:19–54. Independent claim 17 recites a method with limitations generally corresponding to those of claim 1. *Id.* at 17:56–18:20.

Independent claim 33 recites a non-transitory computer readable medium with limitations generally corresponding to those of claim 1. *Id.* at 19:15–47.

VI. ASSERTED GROUNDS

Petitioner asserts that claims 1–47 would have been unpatentable on the following grounds:

Claim(s) Challenged	35 U.S.C. § ¹	Reference(s)
1, 4–11, 14–17, 20–27, 30–33, 36–43, 46, 47	103(a)	Althoff, ² Moore ³
2, 3, 18, 19, 34, 35	103(a)	Althoff, Moore, Nye ⁴
12, 13, 28, 29, 44, 45	103(a)	Althoff, Moore, Surjanto ⁵

Pet. 1.

¹ The Leahy-Smith America Invents Act (“AIA”) included revisions to 35 U.S.C. § 103 that became effective on March 16, 2013. Because the ’585 patent issued from an application filed before March 16, 2013, we apply the pre-AIA version of § 103.

² U.S. Patent No. 6,374,252 B1, Apr. 16, 2002 (Ex. 1006).

³ U.S. Patent No. 7,627,552 B2, Dec. 1, 2009 (Ex. 1007)

⁴ U.S. Patent No. 6,341,279 B1, Jan. 22, 2002 (Ex. 1005)

⁵ *XML Content Management Based on Object-Relational Database Technology*, Surjanto *et al.*, January 2000 (Ex. 1008).

VII. LEVEL OF ORDINARY SKILL IN THE ART

Petitioner describes a person of ordinary skill (“ordinarily skilled artisan”) as one having “a Bachelor’s degree in Computer Science and three to five years of industry experience in application development, database development, enterprise application management; or a Ph.D. specializing in the field.” Pet. 3 (citing Ex. 1003, Declaration of Dr. Jules White, (“White Decl.”) ¶ 23). Petitioner further cites Dr. White as stating that more education could compensate for less experience and vice versa. *Id.* The Patent Owner Preliminary Response does not directly address the level of ordinary skill.

The level of ordinary skill in the art usually is evidenced by the references themselves. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001); *In re GPAC Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995); *In re Oelrich*, 579 F.2d 86, 91 (CCPA 1978). As Petitioner’s unchallenged description of the level of ordinary skill appears appropriate to the subject matter of the ’585 patent and the references, we apply it for purposes of this Decision.

VIII. CLAIM CONSTRUCTION

We interpret claim terms using “the same claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. 282(b).” 37 C.F.R. § 42.100(b). In this context, claim terms “are generally given their ordinary and customary meaning” as understood by a person of ordinary skill in the art in question at the time of the invention. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–13 (Fed. Cir. 2005) (citations omitted) (en banc). “In determining the meaning of the disputed claim limitation, we look principally to the intrinsic evidence of record, examining the claim language itself, the written description, and the prosecution history, if in

evidence.” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 469 F.3d 1005, 1014 (Fed. Cir. 2006) (citing *Phillips*, 415 F.3d at 1312–17).

Extrinsic evidence is “less significant than the intrinsic record in determining ‘the legally operative meaning of claim language.’” *Phillips*, 415 F.3d at 1317 (citations omitted).

Any special definition for a claim term must be set forth in the specification with reasonable clarity, deliberateness, and precision. *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

We construe only those claim terms that require analysis to determine whether to institute *inter partes* review. *See Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1375 (Fed. Cir. 2019) (“The Board is required to construe ‘only those terms . . . that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

A. Module

Noting that the ’585 patent Specification describes a data repository as including modules, Petitioner contends that a person of ordinary skill would have understood a “‘module’ to be a portion of a data repository.” Pet. 4. Petitioner notes that in parallel litigation, Patent Owner conceded “an example of a ‘module’ is ‘a table or a portion of a table’ (*i.e.*, a data repository).” *Id.* (citing Ex. 1013, 78). Petitioner then states that it “applies Patent Owner’s own definition of a ‘module’ throughout this Petition.” *Id.*

Patent Owner argues that, in the context of the ’585 patent, the term “module” would be understood to be a table-based structure, *i.e.*, “a module comprises one or more columns within one or more tables.” Prelim. Resp. 19. According to Patent Owner, to the extent the panel issues a formal construction, “module” should be construed to mean “a table or, one or more

columns of/within one or more tables.” *Id.* Patent Owner does not cite to any definition of “module” in the ’585 patent Specification or any disclosure that would limit a “module” to a table or a column of a table.

Patent Owner further argues that we should deny institution and “not reach the substance of the dispute” because the Petition does not apply what Petitioner contends to be the proper construction of the term “module.”

Prelim. Resp. 19–20. According to Patent Owner, because Petitioner does not apply what it believes is the correct construction, but instead applies a construction Petitioner alleges Patent Owner proposed in parallel litigation, Petitioner has failed to comply with the requirements of 37 C.F.R.

§ 42.104(b)(3). *Id.* at 20–21 (citing *Toyota Motor Corp. v. Blitzsafe Tex., LLC*, IPR2016-00422, Paper 12, 26 (PTAB July 6, 2016)).

In its Preliminary Reply, Petitioner notes that in response to a question during a hearing in the District Court on a Motion to Dismiss under 35 U.S.C. § 101, Patent Owner argued the claim terms recite structure because the claimed module is a table or a portion of a table. Pet. Prelim. Reply 1–2 (citing Ex. 1013, 78). Petitioner also notes that in a Joint Claim Construction Chart submitted to the District Court, Patent Owner proposed “module” be given its plain and ordinary meaning, without defining such plain and ordinary meaning, and that Petitioner proposed that “module” be construed to mean “[a] table or, one or more columns of/within a single table.” *Id.* at 2 (citing Ex. 1011, 2).

The premise of Patent Owner’s argument is flawed. Petitioner asserts that, in the context of the ’585 patent, a person of ordinary skill would understand “module” to be a portion of a data repository. Pet. 4. Petitioner also states its understanding that in parallel litigation, Patent Owner described a table or a portion of a tables as an *example* of a module, i.e., an

example of a portion of a data repository. *Id.*; see Ex. 1013, *InQuisient, Inc. v. ServiceNow, Inc.* Case 22-900 (D. Del), Transcript of Motion to Dismiss, 78:13–19 (“This claim has a very specific data dictionary. It’s got particular structures. The modules that are referred to in the claim, an *example* that’s used in the specification is a table or a portion of a table. A table is a structure that people know how to build.”) (emphasis added); see also PO Prelim. Sur-reply 1–2 (Patent Owner stating that at the hearing Patent Owner referred to “*an example that’s used in the specification* is a table or portion of a table”).

Petitioner’s acknowledgement of Patent Owner’s parallel litigation argument does not change Petitioner’s recognition that the ’585 patent describes a data repository as including modules, such that an ordinarily skilled person would have understood a “module” to be a portion of a data repository. The ’585 patent includes an extensive discussion of tables and relationships between tables, but does not define the term “module.” In the ’585 patent, the term “module” appears in the claims; “module” also appears in the “SUMMARY,” in a manner that parrots the language of the claims. Ex. 1001, 2:1–14.

The term “module” is a classic nonce word that does not connote definite structure. See, e.g., *Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1350 (Fed. Cir. 2015) (“‘Module’ is a well-known nonce word that can operate as a substitute for ‘means’ in the context of [pre-AIA 35 U.S.C.] § 112, para. 6” and “is simply a generic description for software or hardware that performs a specified function” that “does not provide any indication of structure.”). Nonetheless, claim 48, which is not challenged in the Petition, does not use the term “module,” instead reciting limitations similar to those of claim 1 with express use of “means”-plus-function language, thereby

limiting the scope of claim 48 to structures described in the Specification and equivalents thereof. Ex. 1001, 20:47–22:3; *see* 35 U.S.C. § 112, para. 6 (2012). Under the doctrine of claim differentiation, we ascribe a broader meaning to the term “module” in claims 1–47.

Petitioner merely recognizes that in the parallel litigation Patent Owner attempted to avoid a finding that claims 1–47 recite unpatentable subject matter or are indefinite by ascribing structure to the term “module.” Patent Owner does not, indeed cannot, contend that the structure it identifies as a module, i.e., a table or a portion of a table, is not an example of a portion of a data repository. Petitioner’s efforts to address the structure Patent Owner ascribes to the term “module” are not inconsistent with Petitioner’s proposed construction.

To the extent that claims 1–47 can be construed as definite, we construe “module” to mean a portion of a data repository, and note that an example of a structure of a module disclosed in the ’585 patent, i.e., a portion of a data repository, is a table or portion of a table.

IX. ANALYSIS

A. *Legal Principles*

“In an [*inter partes* review], the petitioner has the burden from the onset to show with particularity why the patent it challenges is unpatentable.” *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016) (citing 35 U.S.C. § 312(a)(3) (requiring *inter partes* review petitions to identify “with particularity . . . the evidence that supports the grounds for the challenge to each claim”)). This burden of persuasion never shifts to Patent Owner. *See Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015) (discussing the burden of proof in *inter partes* review).

The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) objective evidence of obviousness or nonobviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). Additionally, the obviousness inquiry typically requires an analysis of “whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007) (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006) (requiring “articulated reasoning with some rational underpinning to support the legal conclusion of obviousness”)); see *In re Warsaw Orthopedic, Inc.*, 832 F.3d 1327, 1333 (Fed. Cir. 2016) (citing *DyStar Textilfarben GmbH & Co. Deutschland KG v. C. H. Patrick Co.*, 464 F.3d 1356, 1360 (Fed. Cir. 2006)).

An obviousness analysis “need not seek out precise teachings directed to the specific subject matter of the challenged claim, for a court can take account of the inferences and creative steps that a person of ordinary skill in the art would employ.” *KSR*, 550 U.S. at 418; accord *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1259 (Fed. Cir. 2007). Petitioner cannot satisfy its burden of proving obviousness by employing “mere conclusory statements.” *In re Magnum Oil Tools Int’l, Ltd.*, 829 F.3d 1364, 1380 (Fed. Cir. 2016). Instead, Petitioner must articulate a reason why a person of ordinary skill in the art would have combined the prior art references. *In re NuVasive, Inc.*, 842 F.3d 1376, 1382 (Fed. Cir. 2016).

A reason to combine or modify the prior art may be found explicitly or implicitly in market forces; design incentives; the “interrelated teachings of multiple patents”; “any need or problem known in the field of endeavor

at the time of invention and addressed by the patent””; and the background knowledge, creativity, and common sense of the person of ordinary skill. *Perfect Web Techs., Inc. v. InfoUSA, Inc.*, 587 F.3d 1324, 1328–29 (Fed. Cir. 2009) (quoting *KSR*, 550 U.S. at 418–21).

In determining whether a claim is obvious in light of the prior art, we consider any relevant objective evidence, when in evidence, of obviousness or non-obviousness. *See Graham*, 383 U.S. at 17. At this stage of the proceeding, the parties do not present evidence of such objective considerations.

We analyze the asserted grounds of unpatentability in accordance with these principles to determine whether Petitioner has met its burden to establish a reasonable likelihood of success at trial.

B. Petitioner’s Contention That Claims 1, 4–11, 14–17, 20–27, 30–33, 36–43, 46, and 47 Are Obvious Over Althoff and Moore

1. Althoff (Ex. 1006)

Althoff relates to modeling of object-oriented database structures, translation to relational database structures, and dynamic searches of the same. Ex. 1006, code (57), 1:16–18, 3:12–14. By way of background, Althoff discloses that it is known in software systems to associate data items and data types to allow elements of a system to rely on the characteristics of the data type and relationships between one data type and other data types. Ex. 1006, 1:20–28. Althoff further discloses that users can define a class, e.g., telephone number, to allow elements of the system to store, manipulate, or retrieve telephone numbers as though they were fundamental pieces of information. *Id.* at 1:30–35. Althoff explains that it has been found that the technique of defining classes of software objects and manipulating those objects also has been useful for database applications. *Id.* at 1:41–43. In

particular, according to Althoff, in object-oriented database applications, a user can define classes of software objects, as well as properties of those classes and relationships between them, and can populate a database with data items that are instances of those objects. *Id.* at 1:43–47. For example, Althoff discloses, the class “telephone number” might be used to store telephone numbers for person, companies, and computer networks. *Id.* at 1:47–50. Althoff explains that it has been found that object-oriented database management provides advantages such as rapid application and database development and relative software reliability. *Id.* at 1:50–53. Notwithstanding such advantages, however, Althoff further discloses that a problem has arisen in the art insofar as “many database systems have been designed to operate with another, different database technique, known as ‘relational database’ management.” *Id.* at 1:54–57.

In a relational database, Althoff explains,

the database comprises a set of data tables which represent the relationships between data items. Each table comprises a set of records, one record for each data item; each record comprises a set of columns or fields, one column for each data value associated with the data item. (For example, each record in a table of persons might have a column defined for that person’s telephone number; similarly, each record in a table of companies might have a column defined for that company’s telephone number.)

Ex. 1006, 1:58–66. According to Althoff, relational databases have been found to be successful and efficient in managing large databases, and accordingly, there has been substantial investment in relational database management systems and the applications and computer programs that support them. *Id.* at 1:66–2:4. Therefore, Althoff discloses, it would be advantageous to obtain the advantages of object-oriented database

management while retaining the efficiency of and installed investment in relational database management systems, but tools for creating, implementing, and manipulating object-oriented databases are generally inconsistent with the tools for creating, implementing, and manipulating relational databases. *Id.* at 2:5–8. Althoff thus seeks to provide a method of compiling and translating between object-oriented and relational database structures that would obtain the advantages of the former while retaining the ability to use the then-existing relational database management systems. *Id.* at 3:4–9.

To achieve the stated goal, Althoff provides a method and system that allow a user to create, edit, and manipulate an object database, objects for that object database, and queries to be applied to the object database, where the method and system dynamically translates the object database, the objects for the object database, and queries to be applied to the object database into a set of relational database structures, data for those relational database structures, and queries to be applied to those relational database structures, respectively. Ex. 1006, 3:12–25. According to Althoff, the method and system “include[] a meta-model of the user’s object database, which is itself an object database, and which has itself been translated into a set of relational database structures for manipulation by a relational database engine.” *Id.* at 3:26–30. The meta-model comprises a set of classes, objects, and relationships between classes that model the classes and relationships between classes of the system. *Id.* at 3:30–32. “Each of these classes may comprise a set of searchable properties, and each of these relationships may comprise an inheritance relationship (between a base class and a derived class) or a data-model relationship (such as a one-to-one, one-to-many, or many-to-many relationship).” *Id.* at 3:31–37. Further, “[t]he data model of

the user's object database is modeled by actual objects in the meta-model, and editing or manipulating the user's object database is modeled by creating, modifying, or deleting objects in the meta-model." *Id.* at 3:37–41. "The meta-model also models itself, in the same manner as it models the user's object database, and may be manipulated in the same manner as the user's object database." *Id.* at 3:41–44.

In a preferred embodiment, each class in the user's object database is modeled by a relational database table, each searchable property of that class is modeled by a column in that table, and each object of that class is modeled by a row in that table and corresponding rows in tables representing base classes for that object. Each such table comprises an "object-ID" column generated by the system, comprising a unique ID for each object in the system. Each relationship between two objects is modeled by providing a pointer from the first object to the second object; the pointer comprises a column (generated by the system to model the relationship) in the first object's table with the object-ID of the second object. These relational database structures are created, modified, or deleted dynamically or incrementally in response to user commands to create, edit, and manipulate the user's object database.

Id. at 3:44–60. Thus, each class may comprise a set of searchable properties and the relationship may be an inheritance relationship (between a base class and a derived class) or a data-model (e.g., a one-to-one, many-to-one, or many-to-many relationship). *Id.*; *see also id.* at 7:21–25 (describing the meta-model as "comprises an object database having as its objects the classes 101 in the user's object database 100, searchable properties 102 of those classes, and relationships 103 between those classes").

Metamodel 200 comprises classes shown in Table 3-1. Ex. 1006, 10:41–11:10. Each class of the user's object database 100 is modeled by an object of class Class 301. Ex. 1006, 11:13–14. Each object of class Class

301 includes an object ID that is unique for the object, a class label (a name to be displayed for the class), a class DB name (to be used by the system for generating SQL commands and accessing relational structures corresponding to this class 101), properties 102, such as Class Group, Class Type, and Is System Specific that maintain information about the class 101 in system 200, as well as information about other properties. *Id.* at 11:12–12:23.

2. *Moore (Ex. 1007)*

Moore discloses a virtual folder system that exposes regular files and folders (also known as directories) to users in different views based on their metadata instead of the actual physical underlying file system on the disk. Pet. 5; Ex. 1007, 5:51–54; Ex. 1003, White Decl. ¶¶ 47–48. Moore discloses virtual folder system 200 includes folder processor 210, relational database 230 and virtual folder descriptions database 232. Ex. 1007, 8:5–9. A “query builder component 220 [that] builds SQL queries” from “data from the virtual folder descriptions database 232, including a list of types to display in the folder, the initial filter, and the physical location to show results from (the scopes).” *Id.* at 9:19–25. The “query builder component 220 sends the SQL queries to the relational database 230.” *Id.* at 9:25–27.

3. *Petitioner’s Contentions Concerning Reasons to Combine Althoff and Moore*

Petitioner contends that a person of ordinary skill would have had reason to combine the teachings of Althoff and Moore to improve end-user experience by avoiding the need for complicated SQL queries or for navigating complicated tables to access objects stored in Althoff’s database. Pet. 10. Petitioner contends that SQL queries to access databases are often beyond the capabilities of ordinary users, but such users are already familiar with working with folders. *Id.* at 11–12. Petitioner contends that, because

Moore provides a system that can take a property stored in a database and represent it as a container that is like a folder, a person of ordinary skill would have had reason to combine Moore’s virtual folders with Althoff’s database to simplify use of such systems. *Id.* at 11–13 (citing Ex. 1003, White Decl. ¶¶ 62–69; Ex. 1007, 3:36–41).

4. *Claim 1*

a) *Preamble*

Petitioner identifies as the preamble of claim 1 the recitation “[a] computerized system for manipulating data sets.” Pet. 13. Neither party contends that the preamble of claim 1 is limiting. Nevertheless, to the extent that the preamble is considered limiting, Petitioner cites Althoff’s general purpose computer as teaching a device that allows the user to create, edit, and manipulate the user’s object database. *Id.* at 13–14 (citing Ex. 1006, 3:15–18, 5:22–31). Petitioner further states that an ordinarily skilled artisan would understand that Althoff’s object database includes data sets. *Id.* at 14 (citing Ex. 1003, White Decl. ¶¶ 70–71). Patent Owner does not respond to Petitioner’s contentions concerning the preamble of claim 1. On the present record, we are persuaded that Petitioner has sufficiently demonstrated that Althoff teaches the preamble of claim 1.

b) *Limitation [1a]*

Petitioner identifies as limitation 1[a] the recitation “a **hardware-based processor**.” Pet. 14. Noting that Althoff discloses using a general-purpose computer, or set of general-purpose computers, Petitioner asserts that an ordinarily skilled artisan would understand that such devices include a hardware-based central processing unit, i.e., a hardware-based processor as claimed. *Id.* (citing Ex. 1003, White Decl. ¶ 73; Ex. 1006, 5:21–31). Patent Owner does not respond to Petitioner’s

contentions concerning limitation 1[a]. On the present record, we are persuaded that Petitioner has sufficiently demonstrated that Althoff teaches limitation 1[a].

c) Limitation 1[b]

Petitioner identifies as limitation 1[b] the recitation “a data repository adapted to process, retrieve, and store: data contained in the data sets; and one or more layers of metadata included in the data sets, wherein the data repository comprises.” Pet. 14. Petitioner asserts that a person of ordinary skill would have understood Althoff’s relational database to be a specific implementation of the data repository as claimed. *Id.* (citing Ex. 1006, 6:42–55; Ex. 1003, White Decl. ¶¶ 74–75).

In support of its contention that Althoff discloses a data repository that is adapted to store data contained in the data sets and one or more layers of metadata included in the data sets, Petitioner cites Althoff’s teaching of the “*user’s object database*.” *Id.* (citing Ex. 1006, 3:25–29). According to Petitioner, an ordinarily skilled artisan would understand that an object database contains data sets in the form of objects and the data associated with those objects. *Id.* (citing Ex. 1003, White Decl. ¶¶ 76–77). Petitioner further cites Althoff as disclosing a meta-model of the user’s object database. *Id.* Petitioner notes that Althoff teaches that the meta-model “is itself an object database [that] comprises a set of classes, objects and relationships between classes which model the classes and relationships between classes of the system.” *Id.* at 14–15 (citing Ex. 1006, 3:25–32). According to Petitioner, a person of ordinary skill would understand that Althoff’s meta-model of the user’s object database would contain one or more layers of metadata sets because the meta-model includes data about the data sets. *Id.* (citing Ex. 1003, White Decl. ¶¶ 78–80).

Petitioner further contends that Althoff's data repository is adapted to process, retrieve and store data contained in the data sets, as claimed, because Althoff teaches that it “allows a user to *create, edit and manipulate a user's object database* (which [Althoff's] method and system dynamically translates into a set of *relational database structures*).” Pet. 15 (quoting Ex. 1006, 3:15–21) (emphasis by Petitioner). Petitioner further notes that Althoff's relational database stores data in the data sets by translating the user's object database into a set of relational database structures. *Id.* (citing Ex. 1003, White Decl. ¶ 81).

Petitioner also contends that Althoff is adapted to process, retrieve and store one or more layers of metadata included in the data sets, as claimed. *Id.* Noting that Althoff teaches a meta-model of the user's object database that includes one or more layers of metadata of the data in the data sets, Petitioner argues that Althoff further teaches the meta-model may be manipulated in the same manner as the user's object database. *Id.* (citing Ex. 1006, 3:41–44; Ex. 1003, White Decl. ¶¶ 81–82)).

Patent Owner does not respond explicitly to Petitioner's contentions concerning claim limitation 1[b]. On the present record, we are persuaded that Petitioner has sufficiently demonstrated that Althoff teaches limitation 1[b].

d) Limitation 1[c]

Petitioner identifies as limitation 1[c] the recitation “an element module configured to store and identify a plurality of elements, wherein the plurality of elements are identified by unique element identification.” Pet. 16. According to Petitioner, Althoff teaches representing classes and objects from an object-oriented database within a relational database by explaining that each class in the user's object database is modeled by a relational

database table and that each object of that class is modeled by a row in that table. *Id.* (citing Ex. 1006, 3:45–46; Ex. 1003, White Decl. ¶ 83). Petitioner notes that Althoff describes a system of classes defined in a “class Class” table, citing Althoff’s disclosures that for each of these classes “the system 200 creates a table in the corresponding relational database 203 to correspond to the selected class 101.” *Id.* (citing Ex. 1006, 23:10–12). Petitioner also states that each table is linked to their particular class in the “class Class” table with a column containing the “class identifier” for each object. *Id.* (citing Ex. 1006, 23:33–34).

Petitioner identifies “elements” in the ’585 patent as corresponding to “objects” in Althoff. *Id.* (“Althoff further teaches that these individual class tables **store objects** (*i.e.*, **elements**) . . .”). Petitioner asserts that Althoff’s individual class tables are the claimed “element modules” because each row in the table stores objects (*i.e.*, elements). *Id.* at 16–17 (citing Ex. 1006, 23:19–25 (“When an object is populated into the corresponding relational database 203, the system 200 will create a row in the table corresponding to the object’s class 101 [and thus] **each object will correspond to one row in each table** corresponding to each class 101.”); Ex. 1003, White Decl. ¶¶ 84–85). Petitioner further contends that Althoff discloses an element module is “configured to store and identify a plurality of elements, wherein the plurality of elements are identified by unique element identification” as claimed because Althoff teaches that for each of the individual class tables the first column of the table is the unique identifier (UID) for each object; thus, in each table in which a row is created for the object, the first column corresponds to the object’s UID. *Id.* at 17 (citing Ex. 1006, 23:14–15, 23:26–29; Ex. 1003, White Decl. ¶ 86).

Patent Owner does not respond explicitly to Petitioner's contentions concerning claim limitation 1[c]. On the present record, we are persuaded that Petitioner has sufficiently demonstrated that Althoff teaches limitation 1[c].

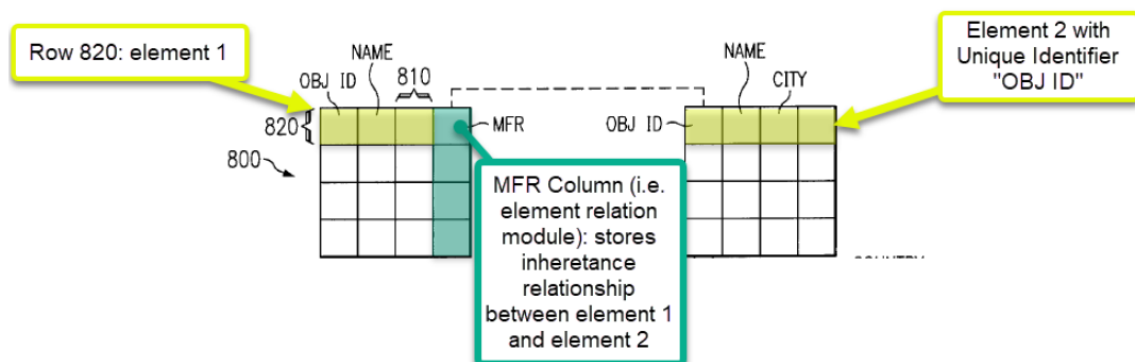
e) Limitation 1[d]

Petitioner identifies as limitation 1[d] the recitation “an element relation module configured to store one or more relationships among the plurality of elements stored in the element module.” Pet. 17. According to Petitioner, Althoff discloses the claimed “***element relation module***” in the form of columns in a class table that stores the relationship between that class and a related class. *Id.* (citing Ex. 1003, White Decl. ¶¶ 87–91). Petitioner notes that Althoff's system creates a table for each class that is populated with instanced objects of that class corresponding to each row of the table. *Id.* (citing Ex. 1006, 23:66–24:2 (“As described with regard to FIG. 7, the user relational database 230 corresponding to the user's ***object database 100 comprises a set of relational tables 800***, each corresponding to a class 101.”) (emphasis by Petitioner); Ex. 1003, White Decl. ¶ 87). Referencing Figure 8 of Althoff, Petitioner also notes that inheritance relationships, i.e., relationships between a base class and a derived class, are modeled by providing a column 810 (“Object ID”) in each class table 800. *Id.* at 18 (citing Ex. 1006, 23:51–53 (“[T]he system 200 creates a column in the table for each ***relationship*** 103 between the class 101 and a related class 101.”), 23:53–55 (“In the column for this relationship 103, the system 200 places the UID of the object in the related class 101 which is related to the object in the class 101 for this table.”), 30:10–12, Fig. 8); *see also* Ex. 1006, 3:61–4:4, 5:38–6:36 (providing examples of base classes and derived

classes). Thus, Petitioner identifies column 810 as the claimed “*element relationship module*.” Pet. 18 (citing Ex. 1003, White Decl. ¶ 88).

Limitation 1[d] also recites that the claimed element relationship module (identified by Petition as corresponding to column 801 of class table 800) is configured to store one or more relationships among the plurality of elements (identified by Petitioner as objects stored in rows 820 of table 800) stored in the element module (identified by Petitioner as individual class tables). See Section IX.B.4.d herein. Referencing Figure 8, Petitioner notes that each class table includes a UID column (OBJ ID) that uniquely identifies each element and an MFR (manufacturer) table that stores UIDs for elements in other class tables. Pet. 18 (citing Ex. 1006, 24:9–31, 23:53–55; Ex. 1003, White Decl. ¶ 88). Petitioner further cites Althoff’s disclosure that “data-model relationships 103 between source and target classes 101 are modeled by providing a column 810 in the source table 800 pointing to an object in the target 800, i.e., having the object UID of a row 820 in the target table 800.” *Id.* at 18–19 (citing Ex. 1006, 30:37–41).

Petitioner’s analysis is illustrated in an example derived from two relational tables of Althoff Figure 8 in the figure reproduced below.



Pet. 19. Petitioner cites a class Manufacturer table on the right and a class Component table on the left with the leftmost column of each table being for

OBJ ID and the elements in rows 820. *Id.* (citing Ex. 1003, White Decl. ¶ 89). According to Petitioner, the dotted line connecting the MFR of an object (i.e., element) of the class Manufacturer table with the OBJ ID of an element of the class Component table establishes the relationship 103 between objects of the class 101 component and objects of the class 101 manufacturer. *Id.* (citing Ex. 1006, 24:26–31; Ex. 1003, White Decl. ¶ 89).

Petitioner also contends that Althoff’s class Class Link 302 teaches the claimed element relation module. Pet. 19 (citing Ex. 1003, White Dec. ¶ 90). Referencing Althoff’s meta-model 220, Petitioner notes that each object of class Class 301 may have zero or more parent and child objects of class Class 301 and that these relationships 103 are represented using cross-link class Class Link 302. *Id.* at 19–20 (citing Ex. 1006, 12:28–31, 12:34–36 (“Each object in the class Class Link 302 links a child class 101 to a parent class 101. A child class 101 may have more than one parent class 101 due to multiple inheritance.”)). Referring to the Figure 8-derived example discussed above concerning the relationship between a Component object and a Manufacturer object, Petitioner states that “[t]his relationship between objects (*i.e. elements*) reflects a class relationship between the Component class and the Manufacturer class, which is a relationship stored in the ‘class Class Link 302.’” *Id.* at 20 (citing Ex. 1006, 6:12–16). For this reason, according to Petitioner, class Class Link 302 stores one or more relationships among the plurality of elements stored in tables, including the element module defined in Althoff as a specific implementation of a table. *Id.* (citing Ex. 1003, White Decl. ¶ 91).

Patent Owner does not respond explicitly to Petitioner’s arguments concerning limitation 1[d]. On the present record, we are persuaded that Petitioner has sufficiently demonstrated that Althoff teaches limitation 1[d].

f) Limitation 1[e]

Petitioner identifies as limitation 1[e] the recitation “a class module configured to define at least one class of the plurality of elements and store the at least one class.” Pet. 21. Petitioner first repeats its contention that in Althoff classes are classes of the plurality of elements, and instanced objects of a class correspond to the plurality of elements stored in the elements module. *Id.* (citing Ex. 1006, 3:45–46 (stating that each class is modeled by a relational database table and each object of the class is modeled by a row in that table); Ex. 1003, White Decl. ¶ 92).

Citing Althoff as disclosing that in meta-model table 220 each class of the user’s object database is modeled by an object of class Class 301, Petitioner argues that Althoff teaches that classes are defined and stored in a class module, i.e., class Class 301 table. Pet. 21–22 (citing Ex. 1003, White Decl. ¶ 94). Noting that in Althoff classes of elements are defined by the user, Petitioner cites Althoff as providing user 201 the capability to select values for the properties of the new object of class Class 310, which has just been created. *Id.* (citing Ex. 1006, 1:31–32, 28:28–52, 11:12–16 (each object of class Class 301 comprises information in table 3–2), 11:18–48 (such information includes the name to be displayed for the class and a description of the class); Ex. 1003, White Decl. ¶ 94).

Patent Owner argues that, under Petitioner’s construction of “module,” Petitioner fails to identify the claimed class module. Prelim. Resp. 24–37. Noting that at page 21, the Petition maps the claimed class module to class Class 301 table in Althoff, Patent Owner argues that Petitioner misquotes Althoff, and fails to recognize Althoff does not disclose such a table. *Id.* at 25, 33–35. As Patent Owner points out, the passages Petitioner cites from Althoff do not refer to a class Class 301 table and do

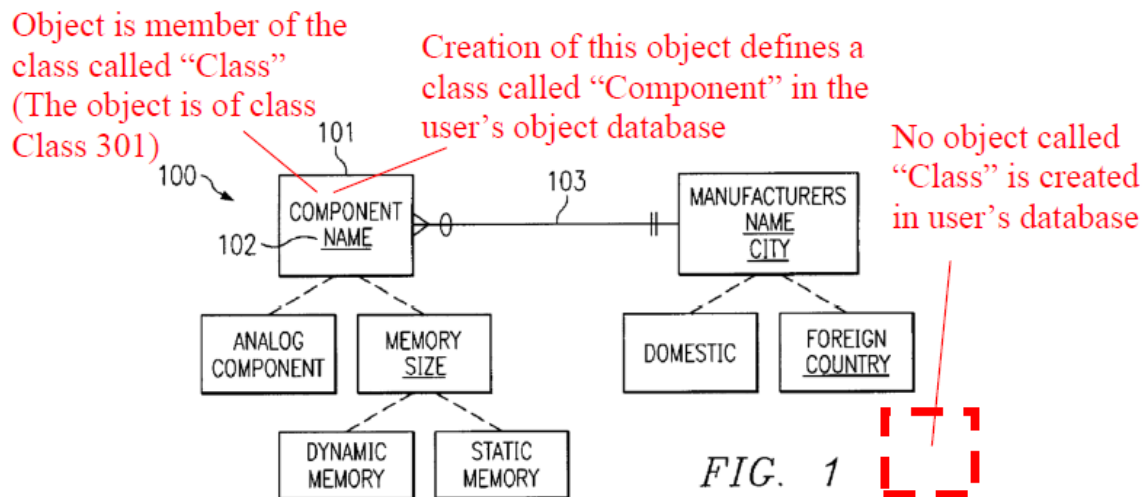
not refer to rows of a table. *Id.* at 32–33. Patent Owner argues that class Class 301 in Althoff is not an object in the user’s object database, but instead is a class (i.e., a data type) defined by the meta-model that can be used by the user’s object database as a data type when the user creates objects. *Id.* at 25–26, 28, 35. Patent Owner further states that the Petition’s reference to Table 3-2 of Althoff as describing information stored in a class Class 301 table is incorrect because Table 3-2 does not concern a row of class table; instead, Table 3-2 of Althoff simply provides a list of information that an object would have in an easily interpreted format. *Id.* at 36 (citing Pet. 21).

Class Class 301 is shown in Figure 3 of Althoff. Ex. 1006, Fig. 3. Figure 3 of Althoff is a description of meta-model 220. *Id.* at 7:21–26. Althoff’s meta-model is an object database. *Id.* at 3:65–67, 7:21–22. The objects of the meta-model object database are classes 101 in the user’s database 100. *Id.* at 7:22–23, 3:37–38 (“[t]he data model of the user’s object data base is modeled by actual objects in the meta-model”). “[E]diting or manipulating the user’s object database is modeled by creating, modifying, or deleting objects in the meta-model.” *Id.* at 3:38–39. The meta-model database also includes searchable properties 102 and relationships between those classes. *Id.* at 7:23–25.

Patent Owner emphasizes that Althoff seeks to translate between two distinct, incompatible types of database structures, i.e., an object-oriented database and a relational database. Prelim. Resp. 23–24 (citing Ex. 1006, 1:14–18, 3:4–7). Patent Owner notes that in Althoff, object database 100 is stored in user model 230 and translated into relational database 231. *Id.* at 23 (citing Ex. 1006, 8:18–28, 23:62–24:31, Figs. 1, 2, 8). Patent Owner contends that, although Althoff’s database translator translates each of the user’s class objects in the relational database table, class Class 301 is not

translated because there is no class called “Class” in the user’s object database. *Id.* at 29–31. In particular, Patent Owner contends that Althoff’s translator cannot select the non-existent class Class for translation into relational data base table. *Id.* at 31. Patent Owner also argues that because Althoff generates a separate table for each class in the user’s object database the class information is preserved and there is no reason to generate a class Class 301 table, as Petitioner states. *Id.*

Figure 1 of Althoff, as annotated by Patent Owner, is reproduced below.



Prelim. Resp. 29. Figure 1 of Althoff “shows a data model diagram of an example user’s object database.” Ex. 1006, 4:59–60. According to Patent Owner, “[i]n Althoff, a database designer defines classes by creating objects that are identified as having the type ‘Class.’” Prelim. Resp. 28. Patent Owner argues that each of the classes in Figure 1 of Althoff is defined in Althoff’s object database by creating objects of the class Class, i.e., objects that are members of the class called Class. Prelim. Resp. 28 (citing Ex. 1006, 10:41–46 (disclosing that meta-model 220 includes classes), Fig. 1). Thus, according to Patent Owner, in Althoff, there is no object

called “Class” in the user’s object database. *Id.* at 29. Instead, Althoff’s class Class 301 is a class, i.e., a data type defined in Althoff’s meta model; objects assigned to that class are identified as classes in the user’s object database. *Id.* at 28–29. A designer defines classes, e.g., “Component” and “Manufacturer,” by building objects having the type “Class,” where each object is a member of the class Class 301. *Id.*

Patent Owner illustrates this principle using the annotated version of Figure 8 of Althoff reproduced below.

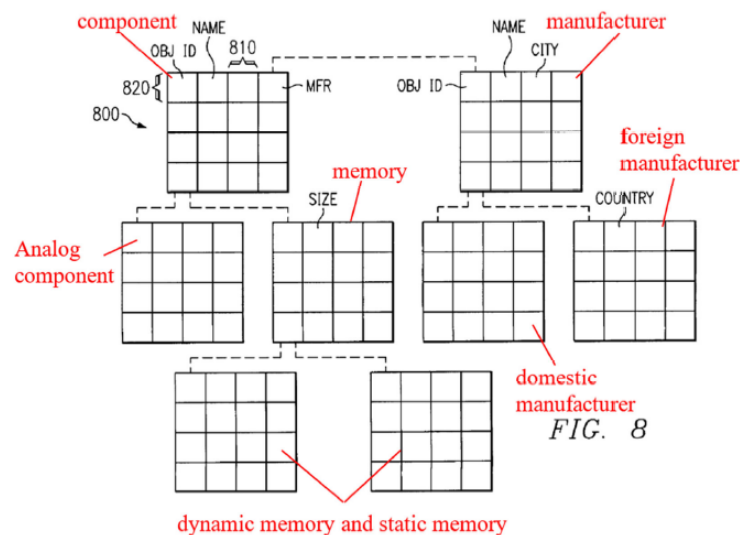


Figure 8 of Althoff as annotated by Patent Owner

Prelim. Resp. 31. Figure 8 of Althoff shows a data model of a user relational database corresponding to the sample user’s object database of Figure 1. Ex. 1006, 5:8–10. According to Patent Owner, Class 301 in Althoff’s meta-model is a data type, used by the object database to identify the classes in the user’s database. *Id.* at 32. Referencing annotated Figure 8, Patent Owner states that, in translating an object database to a relational database, Althoff’s system selects one of the eight classes in the user’s object database and creates a table in the corresponding relational database 203 corresponding to that selected class. Prelim. Resp. 29–30 (citing Ex.

1006, 5:23–6:29, 23:6–7, 23:11–13). Patent Owner notes that Althoff’s translator generates eight relational database tables, one for each class (i.e., (1) Component, (2) Analog Component, (3) Memory, (4) Dynamic Memory, (5) Static Memory, (6) Manufacturer, (7) Domestic [manufacturer], and (8) Foreign [manufacturer]), but Althoff’s system does not generate a relational database table called Class because there is no object called Class. *Id.* at 29–32 (citing Ex. 1006, 4:58–49, 5:8–10, Figs. 1, 8). We agree with Patent Owner that, contrary to Petitioner’s assertion, Althoff does not teach the claimed class module, i.e., a module (or table) configured to define at least one class of the plurality of elements (Althoff’s objects 101) and store that class.

Patent Owner’s contention that there would be no reason to generate a class Class 301 table as Petitioner asserts is also persuasive. *See* Prelim. Resp. 31–32. As Patent Owner notes, by creating separate tables for each class, e.g., a Component table and a Manufacturer table, Althoff preserves the class information. *Id.* Patent Owner also argues that Petitioner’s assertions concerning a class Class 301 table fail because “if objects were already stored as table in the object database, there would be no need for Althoff to build a translator to translate objects into table.” *Id.* at 33.

Petitioner relies on Moore’s disclosure of a virtual folders description database to teach the named text module recited in limitation 1[i]. Pet. 5, 28. Petitioner does not identify any disclosure in Moore that teaches the subject matter not disclosed by Althoff. Accordingly, we determine that Petitioner has not sufficiently shown that limitation 1[e] is taught or suggested by the combination of Althoff and Moore.

g) Conclusion Regarding Claim 1

Having reviewed the evidence and arguments of record, we determine that Petitioner has not demonstrated that a reasonable likelihood it would succeed in its challenge to claim 1.

5. Independent Claims 17 and 33

Claim 17 is drawn to a method. Claim 33 is drawn to a non-transitory computer medium. For claims 17 and 33 Petitioner relies on its arguments with respect to claim 1, merely citing for the limitations of claims 17 and 33 to corresponding limitations of claim 1 and arguing that claims 17 and 33 are unpatentable for the same reasons as alleged for claim 1. Pet. 48–49. For the reasons discussed above with respect to claim 1, we determine that Petitioner also has not demonstrated a reasonable likelihood it would succeed in its challenge to claims 17 and 33.

6. Claims 4–11, 14–16, 20–27, 30–32, 36–43, 46, and 47

Claims 4–11 and 14–16 depend from claim 1. Claims 20–27 and 30–32 depend from independent claim 17. Claims 36–43, 46, and 47 depend from independent claim 33. As discussed above, Petitioner has not identified disclosure in Althoff and Moore that teaches the subject matter of the independent claims. Therefore, we determine that Petitioner has not demonstrated a reasonable likelihood it would succeed in its challenge to claims 4–11, 14–16, 20–27, 30–32, 36–43, 46, and 47.

C. Petitioner’s Contentions That Claims 2, 3, 18, 19, 34, and 35 Are Obvious over Althoff, Moore, and Nye

Petitioner relies on Nye as disclosing an event model that allows a developer to add complex dependency logic to an existing database without having to modify the underlying structure of the database. Pet. 5, 51–52.

According to Petitioner, a person of ordinary skill would have understood that the integration of state machines in the object-oriented model taught by Althoff would provide benefits, e.g., by allowing a developer to describe different modes or operation of the system, and to make the system more robust. *Id.*

Claims 2 and 3 depend from independent claim 1. Claims 18 and 19 depend from independent claim 17. Claims 34 and 35 depend from independent claim 33. Petitioner has not identified disclosure in Nye that teaches the subject matter of the independent claims lacking in Althoff and Moore. Therefore, we determine that Petitioner has not demonstrated a reasonable likelihood it would succeed in its challenge to claims 2, 3, 18, 19, 34, and 35 as unpatentable over Althoff, Moore, and Nye.

D. Petitioner's Contentions That Claims 12, 13, 28, 29, 44, and 45 Are Obvious over Althoff, Moore, and Surjanto

Petitioner relies on Surjanto as disclosing XML document management extensions specifically designed to extend object-oriented databases like Althoff's to provide the known benefits of XML document support. Pet. 6, 62.

Claims 12 and 13 depend from independent claim 1. Claims 28 and 29 depend from independent claim 17. Claims 44 and 45 depend from independent claim 33. Petitioner has not identified disclosure in Surjanto that teaches the subject matter of the independent claims lacking in Althoff and Moore. Therefore, we determine that Petitioner has not demonstrated a reasonable likelihood it would succeed in its challenge to claims 12, 13, 28, 29, 44, and 45 as unpatentable over Althoff, Moore, and Surjanto.

X. CONCLUSION

For the reasons discussed above, we determine that Petitioner has not demonstrated a reasonable likelihood that it would succeed on any of its challenges to patentability.

XI. ORDER

In consideration of the foregoing, it is hereby:
ORDERED that the Petition is DENIED.

IPR2023-01072
Patent 8,219,585 B2

FOR PETITIONER:

James Glass
John McKee
Quincy Lu
Joshua Parzivand
QUINN EMANUEL URQUHART
& SULLIVAN, LLP
jimglass@quinnemanuel.com
johnmckee@quinnemanuel.com
quincylu@quinnemanuel.com
joshuaparzivand@quinnemanuel.com

FOR PATENT OWNER:

Steven Katz
Andrew Pearson
Jason Wolff
FISH & RICHARDSON P.C.
katz@fr.com
pearson@fr.com
wolff@fr.com