

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

ORACLE CORPORATION,  
Petitioner,

v.

THOUGHT, INC.,  
Patent Owner.

---

Case IPR2014-00117  
Patent 6,985,912 B2

---

Before KARL D. EASTHOM, BARBARA A. BENOIT, and  
STACEY G. WHITE, *Administrative Patent Judges*.

WHITE, *Administrative Patent Judge*.

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

## I. INTRODUCTION

Oracle Corporation (“Petitioner”) filed an amended Petition requesting *inter partes* review of claims 1–16 (“challenged claims”) of U.S. Patent No. 6,985,912 B2 (Ex. 1001, “the ’912 Patent”) pursuant to 35 U.S.C. §§ 311–319. Paper 6 (“Pet.”). Thought, Inc. (“Patent Owner”) did not file a Preliminary Response. On April 25, 2014, we instituted *inter partes* review of claims 1–3, 6, and 7<sup>1</sup> on three grounds of unpatentability alleged in the Petition (Paper 16, “Dec. to Inst.”). After institution of trial, Patent Owner filed a corrected Patent Owner Response (Paper 31, “PO Resp.”), to which Petitioner filed a Reply (Paper 34, “Reply”).

Patent Owner filed a Motion for Observations (Paper 37, “Obs.”) on certain cross-examination testimony of Petitioner’s declarant, to which Petitioner filed a Reply (Paper 41, “Obs. Reply”).

An oral hearing was held on November 18, 2014, a transcript of which has been entered into the record. Paper 43 (“Tr.”).

The Board has jurisdiction under 35 U.S.C. § 6(c). In this Final Written Decision, issued pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73, we determine Petitioner has shown by a preponderance of the evidence that challenged claims 1–3, 6, and 7 are unpatentable.

### A. Related Proceedings

The ’912 Patent is the subject of *Thought, Inc. v. Oracle Corp.*, No. 3:12-05601 (N.D. Cal.). Pet. 1; Mandatory Notice (Paper 10).

---

<sup>1</sup> Prior to institution, Petitioner filed an unopposed motion to withdraw its alleged grounds of unpatentability against all claims except 1–3, 6, and 7. Paper 15. We granted Petitioner’s motion and, therefore, we considered only challenges leveled against claims 1–3, 6, and 7. Dec. to Inst. 3.

### *B. The '912 Patent*

The '912 Patent is directed to “enhanced database access and performance when correlating or translating one database to another database or to an object programming application.” Ex. 1001, Abstract. It “provide[s] a simple wrapper or facade interface and API to be utilized to coordinate relational data systems and object data systems and to control multiple software modules that are utilized to load, insert, update[,] and delete data from data systems and objects or object models in object programming systems.” *Id.* at 3:64–4:2.

One of the objects of the invention is to “provid[e] an improved caching system for multiple users.” *Id.* at 4:15. Data or objects populate the cache when the data or objects first are read from a data source, or the data may be preloaded as directed by the user. *Id.* at 4:24–27. Once a unit of data or object is updated, the cached versions of that unit of data or object are invalid. *Id.* at 16:50–51. “[A] cache notification system[] termed a UNS (Unit Notification System)” provides “notification that a unit of data or object in a cache is invalid.” *Id.* at 5:26–31. Upon receiving a notification of invalid data, the user can set the cache to delete and reload the invalidated data or wait until the user attempts to read that unit of data before updating the cache. *Id.* at 16:52–55.

### *C. Illustrative Claim*

Claim 1 of the '912 Patent is the only independent claim challenged, and it recites:

1. A computer system or network comprising a local or distributed data or object caching system wherein:

(a) data or objects are (i) placed in a user cache in the caching system when the data or object is first read from a data source by a user as a lazy cache, or (ii) preloaded as directed by a user,

(b) each unit of data or object that is placed in a user cache is registered with a unit notification system (UNS) as being present in the cache and the UNS is connected to a Java Messaging Service (JMS) or to any other reliable messaging service available in the object programming language system environment,

(c) when any unit of data or object is updated in a user caches and persisted (saved) to a primary data source the UNS sends a message to all other user caches that may exist and contain the updated unit of data or object to register in the user cache a message that the unit was updated by another user and is noted as being invalid in the cache, and

(d) other user caches that have been notified as described in (c) will read from the data source the updated unit of data or object and update the cache only when a user attempts to access the invalidated unit or data or object in a user cache, may automatically update the user cache as soon as notified as has been pre-set in the caching system by the user of the user cache.

*D. Alleged Grounds of Unpatentability Instituted for Trial*

The following table summarizes the three challenges to patentability that were instituted for *inter partes* review:

| Reference(s)          | Basis | Claim(s) Challenged |
|-----------------------|-------|---------------------|
| Franklin <sup>2</sup> | § 102 | 1–3                 |

---

<sup>2</sup> Michael J. Franklin et al., *Transactional Client-Server Cache Consistency: Alternatives and Performance*, ACM, 1–44 (1997) (“Franklin,” Ex. 1005).

| Reference(s)                    | Basis | Claim(s) Challenged |
|---------------------------------|-------|---------------------|
| Franklin and JVM <sup>3</sup>   | § 103 | 2                   |
| Franklin and Chang <sup>4</sup> | § 103 | 6 and 7             |

## II. ANALYSIS

### A. Claim Interpretation

In an *inter partes* review, claim terms in an unexpired patent are given their broadest reasonable construction in light of the specification of the patent in which they appear. 37 C.F.R. § 42.100(b); *see* Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,766 (Aug. 14, 2012). Pursuant to that standard, the claim language should be read in light of the specification, as it would be interpreted by one of ordinary skill in the art. *In re Suitco Surface, Inc.*, 603 F.3d 1255, 1260 (Fed. Cir. 2010). Thus, absent any special definitions, we give claim terms their ordinary and customary meaning, as would be understood by one of ordinary skill in the art at the time of the invention. *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). Any special definitions for claim terms must be set forth with reasonable clarity, deliberateness, and precision. *See In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

For purposes of the Decision to Institute, we construed the terms (1) “when a user attempts to access the invalidated unit or data or object in a user cache, may automatically update the user cache,” (2) “lazy cache,” (3) “unit notification system (UNS),” (4) “reliable messaging service,” and

---

<sup>3</sup> Tim Lindholm et al., THE JAVA™ VIRTUAL MACHINE SPECIFICATION 1–247 (Lisa Friendly ed., Addison-Wesley 1997) (“JVM,” Ex. 1009).

<sup>4</sup> U.S. Patent No. 5,627,979 (“Chang,” Ex. 1010), issued May 6, 1997.

(5) “provides transparent persistence.” Dec. to Inst. 7–12. “Patent Owner agrees with the Board’s construction of the terms construed by the Board [in the Decision to Institute] and respectfully submits that further interpretation of that construction is needed in two instances,” the construction of “reliable messaging service” and “transparent persistence.” PO Resp. 9–10; *see id.* at 10–17. With the exception of “reliable messaging service” and “provides transparent persistence,” which we will address below, we discern no reason to modify or further discuss in this Final Written Decision our constructions for the remaining claim phrases. For convenience, those claim constructions are reproduced in the table below.

| Claim Phrase                                                                                                                                                                                                                                       | Claim Construction                                                                                                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| “update the cache only when a user attempts to access the invalidated unit or data or object in a user cache, may automatically update the user cache as soon as notified as has been pre-set in the caching system by the user of the user cache” | “update the cache only when a user attempts to access the invalidated unit of data or object in a user cache, or may automatically update the user cache as soon as notified as has been pre-set in the caching system by the user of the user cache” |
| “lazy cache”                                                                                                                                                                                                                                       | “a cache in which data or an object is first placed in the cache when it is requested.”                                                                                                                                                               |
| “Unit Notification System (UNS)”                                                                                                                                                                                                                   | “a software component that manages identifying information of data or objects in caches and can send messages to the caches in a computer system.”                                                                                                    |

1. *“reliable messaging service”*

In the Decision to Institute, we construed “reliable messaging service” to be “a messaging service that sends messages that will be received

according to guarantees made by the service.” Dec. to Inst. 11. Patent Owner asserts that in order to account fully for the context of the term as recited in claim 1, element b, we need to construe the phrase “any other reliable messaging service available in the object programming language system environment.” PO Resp. 10–11 (emphasis omitted). According to Patent Owner, a reliable messaging service should be construed as follows:

a service that:

i) provides communication between software components or applications as a loosely coupled peer-to-peer facility in which the sender and receiver do not have to be available at the same time in order to communicate;

ii) sends messages that will be received according to guarantees made by the service;

iii) has reliability capabilities similar to the JMS messaging service, namely store and forward capabilities; and

iv) has an interface for object oriented programs to use the messaging service.

*Id.* at 12–13 (citing Ex. 2001 ¶ 84).

Patent Owner asserts that a person of ordinary skill in the art would have understood the recited phrase “any other” to mean that the reliable messaging service of claim 1 is a reliable messaging service with capabilities similar to those of the Java Messaging Service (“JMS”). *Id.* at 11–12 (citing Ex. 2001 ¶ 83). In support of its position, Patent Owner relies upon the testimony of Sam Malek, Ph.D. *See* Ex. 2001 ¶¶ 80–85. Dr. Malek provides a detailed description of JMS (*id.* ¶¶ 39–47), however, he does not provide a sufficient factual basis for his assertion that “any other” would be limited to services similar to JMS. During his deposition, Dr. Malek testified that JMS was an example of a messaging service and that there were other messaging services beyond JMS. Ex. 1016, 48:19–24. He, however, asserts that

because the '912 Patent mentions JMS, that in itself is an indication that the claims are directed to a particular form of messaging performed by JMS. *Id.* at 50:19–51:7. We note, however, that the '912 Patent does not limit the messaging service to JMS; instead it lists JMS as an example of a messaging service that could be used. Ex. 1001, 5:57–60 (“caching system that utilizes a messaging system such as the Java Messaging Service (JMS)”), 16:28–30 (“the Java Messaging Service (JMS) or to any other reliable messaging service”), 16:46–49 (“JMS . . . or other reliable messaging service), 41:22–25 (“JMS (or similar) notification caching system”). At most, there is one passing reference in the specification to the messaging system being “similar” to JMS, but the specification does not describe in what way the messaging system should be similar to JMS, nor does it state how similar the messaging system should be to JMS. Thus, we find that the '912 Patent does not require JMS or any specific features of JMS. Therefore, we determine that the broadest reasonable interpretation of “any other” as recited in claim 1 of the '912 Patent does not limit the claimed messaging service to those similar to JMS.

As to the phrase “available in the object programming language system environment,” Patent Owner argues that this should mean that the recited messaging service has an interface for object oriented programs to use the messaging service. PO Resp. 12 (citing Ex. 2001 ¶ 83). In support of its position, Patent Owner cites Dr. Malek’s testimony. Dr. Malek, however, provides no analysis or factual support for that assertion outside of a discussion of JMS’s program language system environment. Ex. 2001 ¶¶ 82–83. We determine that this term is not limited to the environment of the JMS. Therefore, we determine that the broadest reasonable interpretation of

“available in the object programming language system environment” is not limited to having an interface for object oriented programs to use the messaging service.

We are not persuaded that express construction is necessary for the terms “any other” and “available in the object programming language system environment.” In addition, after reviewing the evidence of record, we determine that our preliminary construction of “reliable messaging service” to be “a messaging service that sends messages that will be received according to guarantees made by the service” is the broadest reasonable construction of the term and, as such, we do not modify our preliminary construction.

*2. “provides transparent persistence”*

In our Decision to Institute, we construed the phrase “provides transparent persistence” to mean “allows for the storage of data objects that are not inherently database aware.” Dec. to Inst. 12. The parties do not dispute the construction itself; instead, the parties dispute the viewpoint used to evaluate whether transparent persistence has been achieved. PO Resp. 13–17; Reply 8. Patent Owner asserts that transparent persistence should be evaluated at runtime. PO Resp. 14–17. Petitioner argues that it should be evaluated from the perspective of the developer of the source code. Reply 8.

Under the Petitioner’s view, transparent persistence may be achieved if the person writing the source code may create an object without including persistence code (e.g., SQL statements or JDBC calls), and this remains true even if a code generator later generates persistence code that may be embedded in the object. Reply 8. Patent Owner asserts that Petitioner’s proposal does not go far enough. Patent Owner argues that transparent

persistence requires that the object not have any code in it for implementing the storage of data in the underlying database. PO Resp. 14. According to Patent Owner's view, the existence of persistence code in the object makes the object "brittle" such that the code can be broken by even minor changes to the underlying data store. *Id.* at 4. "Transparent persistence solves these problems by completely separating the application logic of the object from its persistence logic. Transparent persistence of objects means that the objects are not database aware. In other words, the objects do not have any code that deals with their persistence." *Id.* (citing Ex. 2001 ¶ 56).

This is a question of the proper scope of the claim term. Transparent persistence is not defined expressly in the specification. Therefore, we must give the term the scope that one of ordinary skill in the art would accord to it given its ordinary and customary usage. *See Paulsen*, 30 F.3d at 1480. Petitioner's expert, Dr. Antony Hosking, testifies that "transparent persistence means that an object is not database-aware from the perspective of the object-oriented application developer that designs the objects." Ex. 1022 ¶ 5.5.1.6. Dr. Hosking and Petitioner cite extrinsic references to support the assertion that one of skill in the art would have understood transparent persistence to be seen from the viewpoint of the developer. *Id.* ¶¶ 5.5.1.5–5.5.1.6 (citing Ex. 1019, 1; Ex. 1020, 5 ("By transparent persistence, we mean that programmers should not need to be aware of . . . stable storage representations of persistent object.")); Reply 9 (citing Ex. 1018, 2–3; Ex. 1019, 1). On behalf of Patent Owner, Dr. Malek testifies that one benefit of transparent persistence is improved programmer productivity, but that the term, when properly construed, also includes other benefits such as decoupling the object and its storage code such that they could be

modified independently at runtime without requiring the recompilation of the source code. Ex. 2001 ¶ 89.

We determine that the '912 Patent does not limit transparent persistence to only the situation where persistence code is not present in the object at all points in time. Petitioner and Petitioner's expert have provided persuasive evidence that one of skill in the art would be familiar with transparent persistence as a tool to ease the development of an object by not requiring the programmer to be aware of the details of the underlying database. We are persuaded that, under the broadest reasonable interpretation of "provides transparent persistence," the '912 Patent does not require all potential benefits of transparent persistence to be achieved in order to meet the limitations of the claim. Thus, we determine that the claim limitation may be met if an object is created without persistence code, and the claim limitation still will be met if a code generator later generates persistence code and embeds that code in the object.

#### *B. Anticipation by Franklin*

Petitioner alleges that claims 1–3 of the '912 Patent are unpatentable under 35 U.S.C. § 102 over Franklin. Pet. 11–19. Petitioner relies on claim charts showing how this reference discloses the claimed subject matter. *Id.* Petitioner further relies on a Declaration from Dr. Hosking. Ex. 1014.

For the reasons discussed below, we conclude that Petitioner has proved by a preponderance of the evidence that claims 1–3 are anticipated by Franklin.

*1. Overview of Franklin (Ex. 1005)*

Franklin is a paper titled “Transactional Client-Server Cache Consistency: Alternatives and Performance.” Ex. 1005. Franklin states that “[c]lient data caching is a dynamic form of data replication. As with any form of data replication, care must be taken to ensure that the presence of multiple copies in the distributed system does not jeopardize the correctness of programs.” *Id.* at 2.

Franklin describes several algorithms used to maintain cache consistency. *Id.* at 10. One family of algorithms described in Franklin is the Optimistic Two-Phase Locking (“O2PL”) family. *Id.* at 23. In O2PL, each client has a local lock manager from which a lock must be obtained before a transaction can access a client’s data. *Id.* After a page is updated in the client’s cache, the update is sent from the client to the server. *Id.* The server then locks the pages to be updated and sends a message to each client with a cached copy of the page to be updated. *Id.* The clients then lock the local cached copy of the page. *Id.* O2PL-I is a variant of the O2PL family that always invalidates cached copies of the updated page. Under O2PL-I, the client deletes the cached copy and releases the lock. *Id.* O2PL-P is a variant that always propagates updated page copies to clients that are caching the updated pages. *Id.* Under O2PL-P, the client locks the cached page and sends a message to the server indicating it is prepared to commit the update. *Id.* at 23–24. Then, the server sends copies of the updated pages to the clients that have notified the server that they are prepared to commit the update. *Id.* at 24. The clients then update the cached page and release the lock. *Id.*

Figure 4 of Franklin is reproduced below.

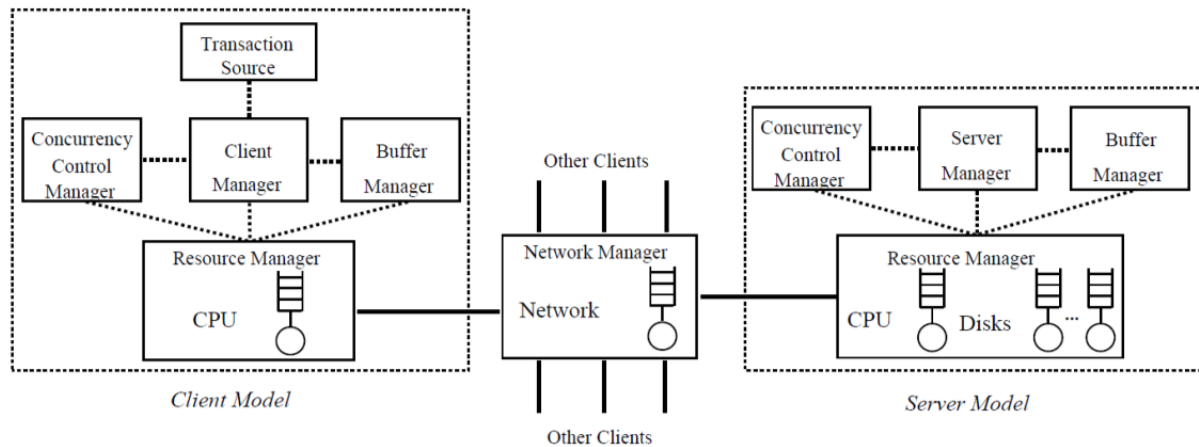


Figure 4 illustrates a performance model of a client-server database management system. *Id.* at 26. The clients and server in Figure 4 each have a Concurrency Control Manager. *Id.* at Fig. 4. The server's Concurrency Control Manager stores information about the location of page copies in the system and manages locks. *Id.* at 26.

## 2. Analysis

Petitioner contends that Franklin anticipates claims 1–3. Pet. 11–19. We have reviewed Petitioner's detailed explanation identifying where each limitation allegedly appears in Franklin, along with the testimony of Petitioner's declarant, Dr. Hosking. *See id.*; Ex. 1014 ¶¶ 9.2.1–9.2.1.1.7.2; Ex. 1022 ¶¶ 5.1–5.2.2.2. We also have reviewed Patent Owner's assertions and evidence as to why Petitioner's explanations and evidence are deficient. PO Resp. 18–34; Ex. 2001 ¶¶ 107–132. On this record, we are persuaded that Petitioner has shown by a preponderance of the evidence that Franklin discloses each limitation of claims 1–3.

Claim 1 requires placing data or objects in a cache. Petitioner argues that Franklin discloses the claimed cache. Pet. 12–13; Ex. 1014 ¶ 9.2.1.1.2.2. Element (b) of claim 1 further requires that the data be

registered with a unit notification system (“UNS”) and that the UNS be connected to JMS “or to any other reliable messaging service available in the object programming language system environment.” Petitioner asserts that Franklin’s Concurrency Control Manager is the claimed UNS. Pet. 13; Ex. 1014 ¶ 9.2.1.1.3.2. Petitioner argues that a reliable messaging service is disclosed inherently as part of the O2PL algorithms described in Franklin. Ex. 1014 ¶ 9.2.1.1.3.4. Element (c) of claim 1 requires that when any data or object is updated, the UNS sends a message to all other caches that may contain the updated data alerting them that the data has been updated and is invalid. Petitioner argues that in the O2PL algorithms discussed in Franklin, once a page has been updated, the server sends a message to each client that has cached copies of any of the updated pages to inform those clients that the data has been invalidated. Ex. 1005, 23. Finally, claim 1 requires the cache to either (1) update the data when a user attempts to access the invalidated data or (2) update the data automatically. Petitioner argues that both options are disclosed in Franklin via its discussion of the O2PL-I and O2PL-P variants of the O2PL algorithm family. Specially, Petitioner argues that Franklin discloses the O2PL-I algorithm in which a client purges a page from its cache upon receipt of an invalidation message from the server. Pet. 16. Dr. Hosking opines that this discloses the claimed updating of the cache when a user attempts to access the invalidated data because “the next request for the now purged page will be a miss, and the client will request a new copy of the page and add the new copy to the cache.” Ex. 1014 ¶ 9.2.1.1.5.3. Petitioner notes that the OP2L-P algorithm automatically updates the client cache in response to an invalidation message from the server. *Id.* ¶ 9.2.1.1.5.4; Pet. 15; Ex. 1005, 23–24.

Claim 2 depends from claim 1 and further requires the applications or users to access the data in a virtual machine environment. Petitioner directs us to Franklin’s disclosure of examples of object-oriented database management systems (“OODBMS”) including Gemstone. Pet. 17 (citing Ex. 1005, 2). Dr. Hosking explains<sup>5</sup> that “the example OODBMS described in Franklin include OODBMS, such as Gemstone, that specifically run in object-oriented virtual machine environments, including both Java and Smalltalk virtual machine environments.” Ex. 1014 ¶ 9.2.1.1.6.1. Claim 3 depends from claim 2 and further requires a distributed system with a central data source. Petitioner argues that Franklin’s Figure 4 depicts a distributed client-server database management system that discloses the limitations of claim 3. Pet. 17 (citing Ex. 1005, 25, Fig. 4).

In response, Patent Owner contends that Franklin does not disclose elements (b) and (c) of claim 1. We address each of Patent Owner’s arguments in turn.

- i. “any other reliable messaging service available in the object programming language system environment”

Much of Patent Owner’s argument on this element centers on its proposed construction of reliable messaging service. As noted in § II.A.1 of this Decision, we do not agree with Patent Owner’s proposal and we determine that the broadest reasonable construction of this term is “a

---

<sup>5</sup> Extrinsic evidence, such as Dr. Hosking’s Declaration, may be used to explain, but not expand on, what one of ordinary skill in the art would have understood from reading the purportedly anticipatory reference. *See In re Baxter Travenol Labs*, 952 F.2d 388, 390–91 (Fed. Cir. 1991) (concluding that the Board properly considered an expert declaration that explained the makeup of a prior art product).

messaging service that sends messages that will be received according to guarantees made by the service.” With that in mind, Patent Owner’s remaining arguments are that Petitioner has not provided sufficient evidence to show that Franklin inherently discloses a reliable messaging service. PO Resp. 20. In addition, Patent Owner contends that the two-step commit disclosed in Franklin can be performed using an unreliable messaging service. *Id.* at 22.

As to the inherent disclosure of a reliable messaging service, Petitioner’s expert Dr. Hosking opines that

it is inherent in the disclosure of Franklin, that the invalidation messages are sent using a reliable messaging service. . . . In the context of Franklin, messages are sent to caches to invalidate copies of data stored in those caches as part of a cache consistency algorithm. Invalidation messages sent through any other messaging service other than a reliable messaging service would result in missed invalidation messages and inconsistent caches, rendering the system describe[d] by Franklin ineffective.

Ex. 1014 ¶ 9.2.1.1.3.4. We find Dr. Hosking’s testimony to be well reasoned. “[A]nticipation by inherent disclosure is appropriate only when the reference discloses prior art that must necessarily include the unstated limitation.” *Rexnord Indus., LLC v. Kappos*, 705 F.3d 1347, 1355 (Fed. Cir. 2013) (quoting *In re Omeprazole Patent Litigation*, 483 F.3d 1364, 1378 (Fed. Cir. 2007)). According to our construction of reliable messaging service, such a service makes guarantees regarding the receipt of messages. Therefore, we find that one of ordinary skill in the art would understand an unreliable messaging service to be one that makes no such guarantees. It follows that if an unreliable messaging service is used, then there is no guarantee that invalidation messages would be received. The stated purpose

of the algorithms described in Franklin is “to ensure that such caching does not result in the violation of transaction semantics.” Ex. 1005, 1. Franklin goes on to note that “care must be taken to ensure that the presence of multiple copies [of data] in the distributed system does not jeopardize the correctness of programs.” *Id.* at 2. The absences of guarantees regarding the receipt of invalidation messages would render the disclosed algorithms ineffective for their stated purpose because one cannot ensure that messages are received if an unreliable messaging system is used. Thus, we find that one of ordinary skill in the art would read Franklin’s disclosure to require a reliable messaging service.

We also find Patent Owner’s argument regarding the two-phase commit to be unpersuasive. Patent Owner asserts that “the evidence shows that messages sent in a two-phase commit process are not required to be sent with a reliable messaging service for the commit process to function.” PO Resp. 22. According to Patent Owner, “one of the purposes of a two-phase commit process is [to] perform database transactions over a distributed network where there is no guarantee that a message will be received at its destination.” *Id.* at 23 (citing Ex. 2001 ¶ 112). Dr. Malek supports Patent Owner’s assertion by testifying that “a two-phase commit process has a mechanism for handling failures in the protocol, namely time-out mechanisms.” Ex. 2001 ¶ 112 (citing Ex. 2006, 356 (noting that timeout mechanisms are used in cases where a process may be waiting indefinitely for a message to be received)). Petitioner argues that only one of the disclosed O2PL algorithms (O2PL-P) could be performed with a two-phase commit. Tr. 10:21–25. As described in Franklin, the O2PL-I algorithm requires “the client [to] simply purges its copies of the updated pages,

releases its local locks on those pages, and then sends an acknowledgement message to the server, *a two-phase commit protocol is not necessary in this case.*” Ex. 1005, 23 (emphasis added). In addition, Petitioner argues that by including a time out mechanism, the messaging system becomes a reliable messaging system, i.e., one that makes guarantees. Reply 3; Tr. 11:5–17, 12:11–13:2. Petitioner argues that one of skill in the art would have understood the timeout mechanism to be a part of the messaging service “and specifically the part of that service that provides reliability to the overall messaging service.” Tr. 12:24–13:2. On cross examination, Patent Owner’s expert testified that “a two-phase commit process is generally a technique that it can use to achieve reliability . . . because of the fact that it has [a] time out mechanism in it.” Ex. 1016, 114:21–24. Thus, we determine that one of ordinary skill in the art would have understood that messaging performed via a two-phase commit with a time out mechanism to be a reliable messaging system. We are persuaded that O2PL-P could not be performed with an unreliable messaging system. Therefore, we determine that despite the possibility of using a two-phase commit protocol, Franklin’s disclosed O2PL algorithms necessarily require a reliable messaging system.

- ii. “when any unit of data or object is updated . . . and persisted (saved) to a primary data source the UNS sends a message . . . that the unit was updated . . . and is noted as being invalid in the cache”

Patent Owner argues Franklin does not disclose the sequence recited in element (c) of claim 1. PO Resp. 25–33. We note, however, that “as a general rule the claim is not limited to performance of the steps in the order recited, unless the claim explicitly or implicitly requires a specific order.” *Baldwin Graphic Sys., Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1345 (Fed. Cir.

2008) (citing *Interactive Gift Express, Inc. v. CompuServe Inc.*, 256 F.3d 1323, 1342–43 (Fed. Cir. 2001)). This is particularly true here because claim 1 recites a computer system and not a method.

According to Patent Owner, the inclusion of the term “when” means that the invalidation message cannot be sent to the clients before the update is persisted in the data source. PO Resp. 29. Patent Owner urges that the term “when” has an ordinary and customary meaning of “at or during which time.” *Id.* at 29 n.5 (quoting Ex. 2003, 1424). We do not agree with Patent Owner’s narrow interpretation. Franklin discloses the sending of invalidation messages and the commitment to the database to be contemporaneous activities. Ex. 1005, 23–24. Franklin describes that in the O2PL algorithms a “transaction enters its commit phase” then it sends invalidation messages to the server and the client regarding the data to be updated, and then the data is updated in accordance with the algorithm. *Id.* Thus, one of ordinary skill in the art would have understood that the sending of invalidation messages to be part of the “commit phase.” As such, the sending of invalidation messages would occur “when” the update is being committed. Therefore, we determine that this disclosure meets the limitations of element (c) of claim 1.

Patent Owner also asserts that Franklin does not disclose a message that notes that the data in the cache is invalid. PO Resp. 33–34. Specifically, Patent Owner argues that the O2PL-P merely propagates the pages without a message that the page is invalid, and the O2PL-I algorithm merely deletes the page without a message that the page is invalid. *Id.* As disclosed in Franklin, “the server sends a message to each client that has cached copies of any of the updated pages.” Ex. 1005, 23. The clients then

obtain locks for those pages. *Id.* It is only after the locks are obtained that any page is purged or updated. *Id.* Thus, the message sent to the client is a message that alerts the client that the page in question has been invalidated, and this message is received prior to any action being done to the page. As such, we determine that the message described by Franklin would have been understood by one of ordinary skill in the art to be a message that invalidates the data stored in the caches containing copies of the data that has been updated. Therefore, we find Patent Owner's argument to be unpersuasive.

After considering the arguments and evidence presented by Petitioner and Patent Owner, we determine that Petitioner has shown by a preponderance of the evidence that Franklin anticipates claims 1–3.

### *C. Obviousness over Franklin and JVM*

Petitioner alleges that claim 2 of the '912 Patent is unpatentable under 35 U.S.C. § 103 over Franklin and JVM. Pet. 25–26. For the reasons discussed below, we conclude that Petitioner has proved by a preponderance of the evidence that claim 2 would have been obvious over Franklin and JVM.

Claim 2 depends from claim 1 and further recites that in the system of claim 1, multiple applications or users access data in an object language virtual machine environment. JVM is the specification for the Java Virtual Machine. Ex. 1009. According to Petitioner, JVM was published approximately five years before the priority date of the '912 Patent. Pet. 25; Ex. 1014 ¶ 8.5.2.2. The reference describes the requirements and capabilities of the Java Virtual Machine. Pet. 25 (citing Ex. 1009, 2). Petitioner asserts that to the extent that we find Franklin to not disclose the

“object language virtual machine environment,” it would have been obvious for one of ordinary skill in the art to implement Franklin’s caching system in the virtual machine environment disclosed by JVM. *Id.* at 25. As discussed above, we find that Franklin discloses the claimed “object language virtual machine environment.” We note, however, that “obviousness under § 103 can be based on a reference which happens to anticipate the claimed subject matter.” *In re Meyer*, 599 F.2d 1026, 1031 (CCPA 1979); *see also Cohesive Techs, Inc. v. Waters Corp.*, 543 F.3d 1351, 1364 (Fed. Cir. 2008) (“it is commonly understood that prior art references that anticipate a claim will usually render that claim obvious”). Thus, we analyze the disclosures of JVM to ascertain whether it provides additional teachings that also would have rendered claim 2 obvious in light of the disclosures of Franklin.

JVM discloses that “[t]he Java Virtual Machine is an abstract computing machine. Like a real computing machine, it has an instruction set and uses various memory areas. It is reasonably common to implement a programming language using a virtual machine . . . .” *Id.* at 26 (quoting Ex. 1009, 2). Dr. Hosking opines that “[i]t would have been obvious to one of ordinary skill in the art to combine the disclosure of JVM with the disclosure[] of Franklin . . . to implement the caching systems described therein for an object oriented virtual machine environment.” Ex. 1014 ¶ 8.5.2.3. Petitioner asserts that “environments, such as JVM, were widely known and used in distributed computer systems including caching systems.” Pet. 25–26 (citing Ex. 1014 ¶ 8.5.2.3).

Patent Owner asserts that Petitioner improperly relied upon hindsight reasoning to combine JVM and Franklin. PO Resp. 38. We are not persuaded by Patent Owner’s assertion. We, instead, are persuaded that the

Java Virtual Machine was well-known, and that the implementation of a caching system, such as the one described by Franklin, in the Java Virtual Machine environment would have been within the knowledge of one of ordinary skill, and that such modification would have been motivated by the knowledge of one of ordinary skill. *See In re McLaughlin*, 443 F.2d 1392, 1395 (CCPA 1971) (“Any judgment on obviousness is . . . necessarily a reconstruction based upon hindsight reasoning, but so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made and does not include knowledge gleaned only from applicant’s disclosure, such a reconstruction is proper.”). Therefore, we are not persuaded of improper hindsight reasoning.

After considering the arguments and evidence presented by Petitioner and Patent Owner, we determine that Petitioner has shown by a preponderance of the evidence that Franklin and JVM would have rendered obvious claim 2.

#### *D. Obviousness over Franklin and Chang*

Petitioner alleges that claims 6 and 7 of the ’912 Patent are unpatentable under 35 U.S.C. § 103 over Franklin and Chang. Pet. 26–28. For the reasons discussed below, we conclude that Petitioner has proved by a preponderance of the evidence that claims 6 and 7 would have been obvious over Franklin and Chang.

##### *1. Analysis of Prior Art Challenge*

Chang is a patent directed to a system and method for providing a graphical user interface for mapping and accessing objects in a data store. Ex. 1010, [54]. Chang discloses that “[a] user may define a mapping

between object schema and data store schema by use of a high level language, Schema Mapping Definition Language (SMDL), which is data store independent, object oriented language independent, and extensible.” *Id.* at Abstract. Chang discusses the differences in structure between objected oriented and relational databases and describes a method and apparatus for allowing a user to access a relational database from an object oriented application. *Id.* at 1:48–4:43. In Chang’s disclosure, “a definition of a mapping between object schema and conventional data store schema is represented by a Schema Mapping Internal Representation [(SMIR)].” *Id.* at 5:30–32. As described in Chang, “data store independence is provided by the use of the data store independent [SMDL], the data store independent [SMIR], and the single, uniform, data store independent interface to a Data Store Manager.” *Id.* at 5:66–6:4.

Claim 6 depends from claim 1 and further recites that the system of claim 1 “also provides transparent persistence.” Petitioner asserts that the mapping and accessing of data discussed in Chang teaches transparent persistence. Pet. 26–27 (citing Ex. 1014 ¶ 9.3.4.1.1.3). Claim 7 depends from claim 1 and further requires that the system comprise at least one Object to Relational Mapping Repository. Petitioner argues that Chang’s Data Store Manager contains a SMIR that defines the mapping between and the object oriented and relational databases. Ex. 1010, 5:30–57; see Pet. 28; Ex. 1014 ¶¶ 9.3.4.1.4.2–9.3.4.1.4.3.

As to claim 6, Patent Owner argues that Chang does not teach storing objects that are not inherently database aware. PO Resp. 39. Patent Owner’s argument is premised on its proposal regarding how and when to evaluate whether an object provides transparent persistence. As discussed in

§ II.A.2, we do not agree with Patent Owner’s proposal. In that discussion, we determined that the claim limitation may be met if an object is created without persistence code. Patent Owner directs us to Chang’s disclosure of code generators 410 and 420 that later insert data access code within objects enabling those objects to manipulate data in the underlying data store. PO Resp. 43 (citing Ex. 1010, 37:9–57). As discussed above, however, we determined that one aspect of transparent persistence may be achieved if an object may be created in a manner such that the programmer does not need to be aware of (and therefore, does not write source code to access) the underlying data store. Thus, under the broadest reasonable interpretation of the term, we find that one of ordinary skill in the art would have understood Chang’s disclosure of objects that have persistence code added at a later point in time by a code generator to be a system that provides transparent persistence. *See* Ex. 1022 ¶ 5.5.1.5.

Patent Owner also contends that Petitioner’s rationale for combining Franklin and Chang is conclusory and rooted in hindsight. PO Resp. 50. Petitioner asserts that transparent persistence was a well-known concept at the time of the invention of the ’912 Patent and that it would have been obvious to add transparent persistence to the disclosures of Franklin. Reply 13. Petitioner asserts that adding transparent persistence, as taught by Chang, to the caching system of Franklin only requires “applying a known technique (e.g., transparent persistence) to a known system as disclosed in Franklin and Chang (e.g., Franklin’s system as modified by Chang and the knowledge of a skilled artisan) to yield predictable results (providing transparent persistence in the system).” *Id.* (citing Ex. 1022 ¶ 5.5.3.4); *see* Ex. 1014 ¶ 8.6.2.2. We are persuaded by Petitioner’s argument that

transparent persistence was well-known, that the addition of such to a system such as the one described by Franklin would have been within the knowledge of one of ordinary skill, and that such modification would have been motivated by the knowledge of one of ordinary skill. *McLaughlin*, 443 F.2d at 1395.

## 2. *Analysis of Secondary Considerations*

Patent Owner asserts that evidence of secondary considerations outweighs Petitioner's prior art evidence of obviousness of claim 6. PO Resp. 52–54. Specifically, Patent Owner directs us to its CocoBase product, which it alleges to embody the features of claim 6. *Id.* at 52. Patent Owner asserts that CocoBase received widespread industry recognition and awards, such as the 2003 WebSphere Advisor Editors' Choice Award in the field of Data Integration. *Id.* at 53 (citing Ex. 2004, 4). In addition, in January 2003, About.com named CocoBase the "Best of the Net." *Id.* at 54 (citing Ex. 2005, 1).

We are not persuaded that Patent Owner's evidence of industry acclaim is sufficient evidence of indicia of nonobviousness. As to the About.com article, there is only a brief mention of CocoBase and very little discussion as to why this application is praiseworthy. The entire discussion of CocoBase is as follows: "CocoBase is the leading object to relational database mapping tool for the Java platforms. It requires no EJB, JDBC, or SQL coding." Ex. 2005, 1. The WebSphere article also is very short, and it provides little detail as to why CocoBase is praiseworthy. Ex. 2004, 4. Patent Owner provides no evidence by which we may evaluate whether these web pages should be considered to be industry praise. There is no evidence of record as to how these awards were given. Nor is there evidence

as to whether these awards were well-known or respected in the industry. We are not persuaded that these two mentions on web pages constitute industry praise. *See Bayer Healthcare Pharms., Inc. v. Watson Pharms., Inc.*, 713 F.3d 1369, 1377 (Fed. Cir. 2013) (finding that brief discussions of Patent Owner’s product in journal articles “fall well short of demonstrating true industry praise”). In addition, Patent Owner only has made a conclusory statement that CocoBase embodies the limitations of claim 6 and has not presented evidence of a nexus between CocoBase and the limitations of claim 6. Therefore, we are not persuaded that Patent Owner’s evidence of secondary considerations outweighs Petitioner’s evidence of obviousness.

We determine that Petitioner has demonstrated by a preponderance of the evidence that claims 6 and 7 are unpatentable over Franklin and Chang.

### III. CONCLUSION

We conclude Petitioner has demonstrated by a preponderance of the evidence that

1. Claims 1–3 are unpatentable under 35 U.S.C. § 102 as anticipated by Franklin;
2. Claim 2 is unpatentable under 35 U.S.C. § 103 as obvious over Franklin and JVM; and
3. Claim 6 and 7 are unpatentable under 35 U.S.C. § 103 as obvious over Franklin and Chang.

### IV. ORDER

In consideration of the foregoing, it is hereby

ORDERED that, based on a preponderance of the evidence, claims 1–3, 6, and 7 of the ’912 Patent are unpatentable;

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

IPR2014-00117  
Patent 6,985,912 B2

PETITIONER:

Vaibhav P. Kadaba  
Steven D. Moore  
Kilpatrick Townsend & Stockton LLP  
[wkadaba@kilpatricktownsend.com](mailto:wkadaba@kilpatricktownsend.com)  
[smoore@kilpatricktownsend.com](mailto:smoore@kilpatricktownsend.com)

PATENT OWNER:

J. Warren Lytle, Jr.  
Brian K. Shelton  
SUGHRUE MION PLLC  
[jlytle@sughrue.com](mailto:jlytle@sughrue.com)  
[bshelton@sughrue.com](mailto:bshelton@sughrue.com)