Trials@uspto.gov
571-272-7822

Paper 10
Date: March 3, 2022

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

FACEBOOK, INC.,
Petitioner,

v.

EXPRESS MOBILE INC.,
Patent Owner.

IPR2021-01456
Patent 9,471,287 B2

Before JEFFREY S. SMITH, AMBER L. HAGY, and
AARON W. MOORE, *Administrative Patent Judges.*

HAGY, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

A. Background and Summary

Facebook, Inc. ("Petitioner") filed a petition requesting *inter partes* review (Paper 1, "Pet.") of claims 1, 2, 5–7, 11, and 12 ("the challenged claims") of U.S. Patent No. 9,471,287 B2 (Ex. 1001, the "'287 patent"). *See* 35 U.S.C. § 311. Express Mobile Inc. ("Patent Owner") timely filed a Preliminary Response. Paper 6, "Prelim. Resp." With authorization from the Board, Petitioner and Patent Owner filed, respectively, a Reply (Paper 7) and a Sur-reply (Paper 9) addressing discretionary denial issues.

Under 37 C.F.R. § 42.4(a), we have authority to determine whether to institute review. The standard for instituting an *inter partes* review is set forth in 35 U.S.C. § 314(a), which provides that an *inter partes* review may not be instituted unless the information presented in the Petition and the Preliminary Response shows "there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition."

Applying those standards, we determine that Petitioner has demonstrated a reasonable likelihood that it would prevail in showing unpatentability of at least one challenged claim of the '287 patent. We institute review on all challenged claims and on all grounds raised in the Petition.

Our findings of fact and conclusions discussed below are based on the evidentiary record developed thus far. This Decision to institute trial is not a final decision as to the patentability of any challenged claim. Any final decision will be based on the full record developed during trial.

B. Real Parties in Interest

The parties identify themselves as the real parties in interest. Pet. 1; Paper 4, 1.

C. Related Matters

As required by 37 C.F.R. § 42.8(b)(2), the parties identify various related matters, including numerous district court proceedings involving the ’287 patent and/or its parent patent, U.S. Patent No. 9,063,755 (“the ’755 patent”). Pet. 1–4; Paper 4, 1–5. Among the identified related matters are several *inter partes* matters involving the ’287 patent: IPR2021-00710 (“the ’710 IPR”), IPR2021-01145 (“the ’1145 IPR”), IPR2021-01227 (“the ’1227 IPR”), and IPR2022-00248 (“the ’248 IPR”). Paper 4, 2; Prelim. Resp. 27.[1]

D. The ’287 Patent (Ex. 1001)

The ’287 patent relates to a platform for generating and distributing programming to mobile devices over a network. Ex. 1001, Abs. The system includes a database of web services that can be obtained over a network, and an authoring tool “configured to define an object for presentation on the display, select a component of a web service included in said database, associate said object with said selected component, and produce code, that when executed on the platform, provides said selected component on the display of the platform.” *Id.* at 1:36–42. The Specification describes device-independent programs (“Applications”) and device- or

[1] Petitioner misidentifies several IPR matters as involving the ’287 patent, which in fact involve different patents owned by Patent Owner. *See* Pet. 2 (citing IPR2021-00700 (involving U.S. Patent No. 6,546,397); IPR2021-01144 (involving the ’755 patent); and IPR2021-01228 (involving the ’755 patent)). We view this misidentification as inadvertent error.

platform-specific instructions ("Players") that are provided to user devices. *Id.* at 5:8–14. A Player transforms device-independent instructions of an Application into device-specific instructions that are executable by a user device to, for example, generate one or more pages on the display of the device. *Id.* at 5:60–63, 6:4–8. The disclosed system allows for Applications to provide web service interaction and invocation to a device. *Id.* at 7:47–56.

Figure 2A illustrates the interaction between system components, and is reproduced below.

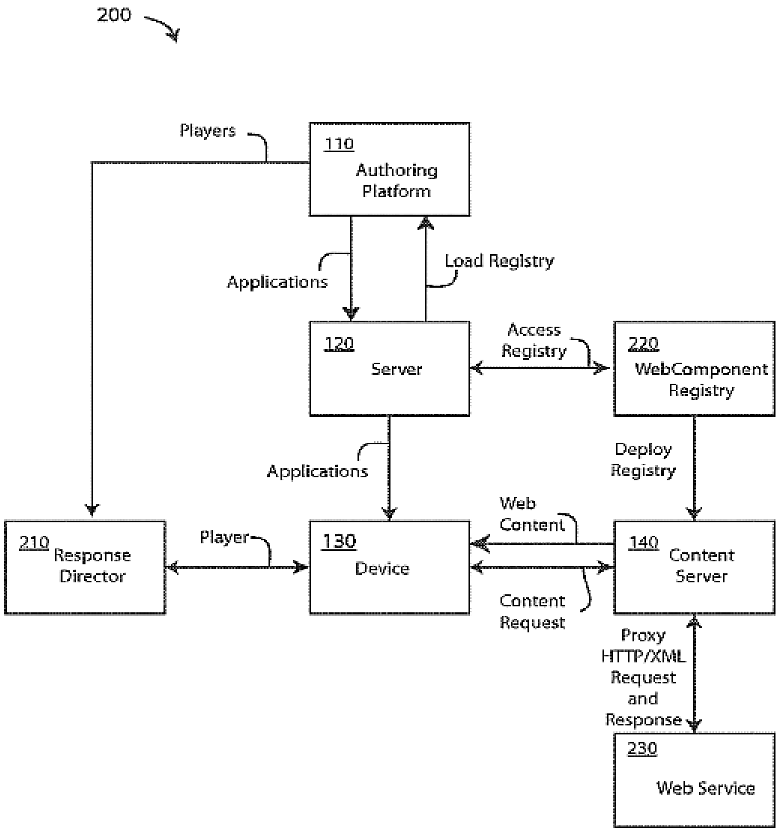


FIG. 2A

Figure 2A is a schematic illustrating the communications between different system components. Ex. 1001, 2:21–23.

As illustrated in Figure 2A, authoring platform 110 may generate one or more Players, which are provided to response director 210. *Id.* at 8:15–17. Device 130 requests a Player from response director 210, and receives and installs the Player. *Id.* at 8:18–20. Web service 230 includes a plurality of services obtainable over the Internet. *Id.* at 8:26–27. Each web service is identified in an entry in web component registry 220. *Id.* at 8:27–30. Web component registry 220 is provided through server 120 to authoring platform 110 so that a user of the authoring platform may bind web services 230 to elements to be displayed on device 130. *Id.* at 8:30–34. A web component registry 220 for each registered web service 230 is loaded into authoring platform 110. *Id.* at 8:62–64. The user of the authoring platform can then assign components of any web service 230 to an Application without any need to write code. *Id.* at 8:64–66.

E. Illustrative Claim

Of the challenged claims, claim 1 is independent. Challenged claims 2, 5–7, 11, and 12 depend from claim 1. Claim 1, reproduced below, illustrates the claimed subject matter:

> 1. A system for generating code to provide content on a display of a device, said system comprising:
>
> computer memory storing a registry of:
>
> a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where the symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service, where each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI)

objects that support the data format type of the symbolic name, and has a preferred UI object, and

b) an address of the web service;

an authoring tool configured to:

define a (UI) object for presentation on the display, where said defined UI object corresponds to a web component included in said registry selected from a group consisting of an input of the web service and an output of the web service, where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool,

access said computer memory to select the symbolic name corresponding to the web component of the defined UI object,

associate the selected symbolic name with the defined UI object, where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name, and

produce an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code; and

a Player, where said Player is a device-dependent code, wherein, when the Application and Player are provided to the device and executed on the device, and when the user of the device provides one or more input values associated with an input symbolic name to an input of the defined UI object,

1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service,

2) the web service utilizes the input symbolic name and the user provided one or more input values for generating

> one or more output values having an associated output symbolic name,
>
> 3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object.

Ex. 1001, 37:48–38:37.

F. Prior Art and Asserted Grounds

Petitioner asserts that claims 1, 2, 5–7, 11, and 12 are unpatentable on the following grounds:

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
1, 2, 5–7, 12	103(a)	Anderson,[2] Bowers,[3] Jacobs,[4] Ambrose-Haynes,[5] and Geary[6]
11	103(a)	Anderson, Bowers, Jacobs, Ambrose-Haynes, Geary, and NFS Administration[7]

[2] G. Anderson & P. Anderson, *JAVA Studio Creator Field Guide*, 2d ed. (Prentice Hall 2006) (Ex. 1003, "Anderson").

[3] B. Bowers & S. Lane, *Advanced FileMaker Pro 6 Web Development* (Wordware Pub. 2003) (Ex. 1004, "Bowers").

[4] S. Jacobs, *Foundation XML for Flash* (Springer-Verlag 2006) (Ex. 1005, "Jacobs").

[5] N. Ambrose-Haynes et al., *Professional ColdFusion 5.0* (Wrox Press. Ltd. 2001) (Ex. 1006, "Ambrose-Haynes").

[6] D. Geary & C. Horstmann, *Core JavaServer Faces* (Sun Microsystems Press, Prentice Hall 2004) (Ex. 1011, "Geary").

[7] *Unix System V NFS Administration* (D. Herman, ed., Prentice Hall 1993) (Ex. 1007, "NFS Administration").

As further support, Petitioner offers the Declaration of Vijay K. Madisetti, Ph.D. Ex. 1002.

Patent Owner submits the Declaration of Kevin C. Almeroth, Ph.D. Ex. 2001.

II. ANALYSIS

A. Discretionary Denial

Patent Owner argues that the Petition "is one of ***five*** staggered petitions for *inter partes* review for the '287 patent," citing the '710 IPR, the '1145 IPR, and the '1227 IPR, and the '248 IPR as the other challenges to the '287 patent. Prelim. Resp. 27. Patent Owner contends that we should deny institution based on the factors set forth in the Board's precedential decision in *General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 (PTAB Sept. 6, 2017) (designated precedential in relevant part) "and/or § 325(d)." Prelim. Resp. 28–35.

General Plastic sets forth a series of factors to be considered by the Board in evaluating whether to exercise discretion under 35 U.S.C. § 314(a) to deny a petition that challenges a patent that was previously challenged before the Board. *General Plastic*, Paper 19 at 15–16. These factors are as follows:

> 1. whether the same petitioner previously filed a petition directed to the same claims of the same patent;
> 2. whether at the time of filing of the first petition the petitioner knew of the prior art asserted in the second petition or should have known of it;
> 3. whether at the time of filing of the second petition the petitioner already received the patent owner's preliminary response to the first petition or received the Board's decision on whether to institute review in the first petition;

4. the length of time that elapsed between the time the petitioner learned of the prior art asserted in the second petition and the filing of the second petition;
5. whether the petitioner provides adequate explanation for the time elapsed between the filings of multiple petitions directed to the same claims of the same patent;
6. the finite resources of the Board; and
7. the requirement under 35 U.S.C. § 316(a)(11) to issue a final determination not later than 1 year after the date on which the Director notices institution of review.

Id. at 16. These factors are "a non-exhaustive list," and "additional factors may arise in other cases for consideration, where appropriate." *Id.* at 16, 18.

General Plastic addressed the situation where the same petitioner filed "follow-on petitions" against the same patents, after a first set of petitions was denied on the merits. *General Plastic*, Paper 19 at 2–3. The Petition here is not a "follow-on petition," as that term is used in *General Plastic*; there was no prior petition challenging the '287 patent filed by Petitioner, Facebook. The '710 IPR petition was filed by Google LLC ("Google"), the '1145 IPR petition was filed by SAP America, Inc. ("SAP"), the '1227 IPR was filed by Adobe, Inc. ("Adobe"), and the '248 IPR petition was filed by Booking Holdings Inc. ("Booking"). Prelim. Resp. 27. We recognize, however, that *General Plastic* has not been limited to instances where multiple petitions are filed by the same petitioner. *Valve Corp. v. Elec. Scripting Prods., Inc.*, IPR2019-00062, Paper 11 (PTAB Apr. 2, 2019) (precedential). Rather, "when different petitioners challenge the same patent, we consider any relationship between those petitioners when weighing the *General Plastic* factors." *Valve*, Paper 11 at 9.

Patent Owner does not allege any specific relationship between Petitioner and any of SAP, Adobe, or Booking; rather, Patent Owner's arguments focus on an alleged relationship between Petitioner and Google. *See* Prelim. Resp. 27–30; Sur-reply 1–2. On this record, we determine that the *General Plastic* factors do not favor denial based on the '1145 IPR, the '1227 IPR, or the '248 IPR because there is no evidence of any relationship between Petitioner and any of SAP, Adobe, or Booking that would implicate *General Plastic* under *Valve*.

Patent Owner argues that Petitioner's alleged relationship with Google in district court litigations favors denial under *General Plastic* and *Valve*. Prelim. Resp. 27–30; Sur-reply 2–3. In particular, Patent Owner argues that it sued Petitioner and Google "in the same court on the same day," and "[s]ince then, Google and Petitioner worked as a team to form theories against the infringement allegations of the '287 patent." Prelim. Resp. 28–29. According to Patent Owner, Google and Petitioner "were working closely together in secret" and have "jointly participated in meet-and-confers in the underlying cases and submitted joint papers to the court expressing the two parties' aligned positions." *Id.* at 29 (citing Exs. 2005–2007). Patent Owner asserts, therefore, that "[t]here is a significant relationship between Petitioner and Google with respect to Patent Owner's assertion of the '287 patent, as they have been working together under the veil of common interest privilege to develop their theories and positions with respect to that patent." *Id.*; Sur-reply 2.

Petitioner counters that its relationship with Google with respect to the '287 patent is simply "a garden variety, arms-length co-defendant relationship." Reply 2. Petitioner further notes that such a "common interest/joint defense relationship" is the "type of cooperation [that] is actively encouraged by district courts and often required in multi-defendant patent cases to conserve resources." *Id.* (citing Exs. 2005, 2006). Petitioner asserts that it played "no role" in preparing the IPR petition filed by Google. *Id.* In particular, Petitioner states that it "had no involvement in Google's selection of prior art, [or in] Google's preparation of the IPR petition and supporting documents," and also asserts that it "had no access to (let alone reviewed) the Google IPR petition and supporting materials prior to their filing," and also "had no input into the timing of Google's IPR filing." *Id.*

We are not persuaded that Petitioner's and Google's status as defendants in co-pending litigation involving the '287 patent rises to the type of "significant relationship" described in *Valve*. In *Valve*, the Board noted that "Valve and HTC were co-defendants in the District Court litigation and were accused of infringing the [challenged] patent based on HTC's VIVE devices that incorporate technology licensed from Valve." *Valve*, Paper 11 at 10. The Board determined that there was "a significant relationship between Valve and HTC with respect to Patent Owner's assertion of the [challenged] patent." *Id.* Nothing in the present record indicates any such relationship between Petitioner and Google. Rather, they were sued by Patent Owner on the same day—as part of a myriad of similar filings against dozens of other defendants. *See* Prelim. Resp. 28; *see also* Paper 4, 1–5 (Patent Owner's Mandatory Notices listing 49 lawsuits involving the '287 patent filed by Patent Owner). The record shows some coordination among numerous defendants in various cases on the issue of claim construction. *See* Exs. 2005 (Defendants' List of Proposed Claim Terms and Phrases for Construction), 2006 (email chain discussing claim construction issues in district court). But, as Petitioner argues without refutation by Patent Owner, this is the "type of cooperation [that] is actively encouraged by district courts." *See* Reply 2.

Patent Owner argues that "the Board has even denied under § 314(a) institution of a serial petition filed by a petitioner having ***no*** alleged relationship with an earlier one." Sur-reply 2 (citing *Samsung Elecs. Am., Inc. v. Uniloc 2017 LLC*, IPR2018-01756, Paper 7 at 24–30 (PTAB Mar. 11, 2019)). The *Samsung* case, which is not precedential, is readily distinguishable. The Board denied institution in that case "because the same or substantially the same prior art or arguments were previously presented to the Board." *Samsung*, Paper 7 at 25–26 (citing 35 U.S.C. § 325(d)). In the present case, by contrast, Petitioner represents that "prior art references cited in this Petition were not cited during prosecution or the other co-pending IPR petitions." Pet. 8. Patent Owner does not dispute this assertion.

Based on the foregoing, and under the circumstances of this case, we are not persuaded that there is a "significant relationship" between Petitioner and Google that would justify discretionary denial. In view of this determination, we also are not persuaded that the remaining *General Plastic* factors favor exercising discretion to deny review. For example, the third *General Plastic* factor is "whether at the time of filing of the second petition the petitioner already received the patent owner's preliminary response to the first petition or received the Board's decision on whether to institute review in the first petition." *General Plastic*, Paper 19 at 16. This factor "is directed to Petitioner's potential benefit from receiving and having the opportunity to study Patent Owner's Preliminary Response, as well as our institution decisions on the first-filed petitions, prior to its filing of follow-on petitions." *Id.* at 17. Here, Petitioner filed the Petition before the Board issued its decision in the '710 IPR, and, therefore, Petitioner received limited benefit from having been able to review Patent Owner's preliminary Response in that matter. *See* Reply 4.

The second, fourth, and fifth *General Plastic* factors are "to assess and weigh whether a petitioner should have or could have raised the new challenges earlier." *General Plastic*, Paper 19 at 18. Here, Petitioner has filed only one petition against the '287 patent, and, as discussed above, the evidence does not show a relationship between Petitioner and previous petitioners that would justify denial. Therefore, we are not persuaded that Petitioner needed to raise the challenges in this Petition earlier.

Finally, we do not view the sixth *General Plastic* factor ("the finite resources of the Board") or the seventh *General Plastic* factor ("the requirement under 35 U.S.C. § 316(a)(11) to issue a final determination not later than 1 year after the date on which the Director notices institution of review") as favoring exercise of our discretion to deny institution.

Therefore, weighing all of relevant factors, we decline to exercise discretion under 35 U.S.C. § 314 to deny the Petition.

Patent Owner also asserts that its arguments warrant our exercise of discretion under 35 U.S.C. § 325(d) to deny review. Prelim. Resp. 33–35. Patent Owner does not, however, explain how the prior art or the arguments asserted in the Petition are the same or substantially the same as art or arguments previously presented to the Office. *See id.*; *see also* 35 U.S.C. § 325(d) ("In determining whether to institute or order a proceeding under this chapter, chapter 30, or chapter 31, the Director may take into account whether, and reject the petition or request because, the same or substantially the same prior art or arguments previously were presented to the Office.”). Indeed, Patent Owner acknowledges the lack of overlap. Sur-reply 2 (noting that “there is no overlap in art between the Petition and [the ’710 IPR petition],” but asserting that does “not tip the scale”). Patent Owner also asserts that “Petitioner had the opportunity to view Patent Owner’s arguments and the Board’s reasons for denial” before it filed the Petition. *Id.* at 4. Patent Owner is incorrect with regard to the timing of the denial of the ’710 petition, which was denied on October 4, 2021—over a month *after* the Petition was filed in this matter. The Board’s reasons for denial of the ’710 petition could not have factored into Petitioner’s choice of prior art in the present Petition.

We also decline to exercise discretion under 35 U.S.C. § 325(d) to deny review.

B. Principles of Law

A patent claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, “would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). 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) when in evidence, objective evidence of nonobviousness.[8] *Graham v. John Deere Co.*, 383 U.S. 1,

[8] Neither party presents arguments or evidence of secondary considerations

17–18 (1966).

"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*, 815 F.3d at 1363 (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).

C. Level of Ordinary Skill in the Art

The level of skill in the art is a factual determination that provides a primary guarantee of objectivity in an obviousness analysis. *Al-Site Corp. v. VSI Int'l Inc.*, 174 F.3d 1308, 1324 (Fed. Cir. 1999) (citing *Graham*, 383 U.S. at 17–18; *Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 718 (Fed. Cir. 1991)).

Relying on the declaration testimony of Dr. Madisetti, Petitioner asserts:

> A person of ordinary skill as of April 2008 would have had at least a bachelor's degree in software engineering, computer science, computer engineering, or electrical engineering with at least two years of experience in web-based software application development, including experience in developing software and systems for storing, retrieving, and transmitting information (such as text and images) over a computer network such as the Internet (or equivalent degree or experience).

Pet. 8 (citing Ex. 1002 ¶ 16).

at this stage. Therefore, secondary considerations do not constitute part of our analysis herein.

At this stage, Patent Owner does not propose an alternative assessment. We note that Patent Owner's declarant, Dr. Almeroth, describes the level of ordinary skill in the art similarly to the assessment offered by Petitioner. *See* Ex. 2001 ¶ 31.

To the extent necessary, and for purposes of this Decision, we accept the assessment offered by Petitioner as it is consistent with the '287 patent and the asserted prior art.

D. Claim Construction

In interpreting the claims of the '287 patent, we "us[e] the same claim construction standard that would be used to construe the claim[s] in a civil action under 35 U.S.C. [§] 282(b)." *See* 37 C.F.R. § 42.100(b) (2021). The claim construction standard includes construing claims in accordance with the ordinary and customary meaning of such claims as would have been understood by one of ordinary skill in the art in light of the written description and the prosecution history pertaining to the patent. *See id.*; *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312–14 (Fed. Cir. 2005) (en banc).

Neither party proposes any terms for construction at this stage. Petitioner notes some constructions of certain terms that were proposed by the parties in underlying litigation involving the '287 patent,[9] but states that "Petitioner does not believe express claim construction is necessary at this

[9] The parties include Patent Owner and several defendants (including Petitioner as well as Expedia, eBay, Google, and Atlassian). *See* Ex. 2009. The underlying district court litigation was originally filed in the Western District of Texas, and that is where the parties submitted their claim construction briefing. *See id.* Before the court entered a final claim construction order, however, the case was transferred to the Northern District of California. *See* Prelim. Resp. 15, n.1; Paper 4, 2.

time." Pet. 10–11. Patent Owner also does not propose any constructions, but asserts that Petitioner's arguments are inconsistent with Petitioner's construction of "authoring tool configured to . . . [produce] a player" as proposed in district court. Prelim. Resp. 15–16 (alterations omitted).

Patent Owner's assertion highlights a potential claim construction dispute. In particular, as Patent Owner points out, the parties in the underlying litigation disputed whether the recited "Player" in claim 1 is "something that is produced by the 'authoring tool' (like the preceding 'Application' limitation), or whether the 'Player' is completely separate from and unrelated to the claimed 'authoring tool.'" Ex. 2009, 36. The defendants in that litigation (including Petitioner) jointly proposed that claim 1 be construed to require that the "authoring tool" that is configured to carry out certain recited actions must also "produce" the claimed Player. *Id.* Patent Owner asserted that no construction is necessary. *Id.*

Although Patent Owner highlights this dispute and characterizes Petitioner's position here as inconsistent with its position in the district court, Patent Owner does not propose any construction. Prelim. Resp. 15–17. And although Petitioner notes the parties' proposed constructions of several terms from an underlying litigation, as noted above, Petitioner does not mention this particular dispute. *See* Pet. 10–11. Thus, we are left without guidance from the parties on this issue.

Notably, in raising this issue, Patent Owner does not contend that Petitioner has failed to meet its obligations before the Board of indicating how challenged claims are to be construed. *Cf.* 37 C.F.R. § 42.104(b)(3) (stating that the petition must identify "[h]ow the challenged claim is to be construed"). Nor does Patent Owner contend that Petitioner has taken a position that contradicts a construction that has been adopted by another court. In that regard, we find unavailing Patent Owner's reliance on the Board's decision in *Orthopediatrics Corp. v. K2M, Inc.*, IPR2018-01546, Paper 10 at 10–12 (PTAB Feb. 14, 2014) (Decision on Institution) ("*K2M*"). *See* Prelim. Resp. 16. *K2M* is nonprecedential and, moreover, was decided on markedly different facts than this case. In *K2M*, the disputed claim term at issue (i.e., "grasping members") implicated mean-plus-function claiming, which, under our Rules, requires the petitioner to identify the specific portions of the specification that describe corresponding structure. *See K2M* at 10–12; *see also* 37 C.F.R. § 42.104(b)(3). Also, in *K2M*, the district court actually adopted the proposed construction that had been advocated by petitioner. *K2M* at 11. Here, in contrast, none of the claim terms at issue implicate means-plus-function claiming. And although the original district court in the underlying litigation did release, via email to the parties, "Preliminary Construction[s]" prior to the claim construction hearing (Ex. 2011), the litigation was transferred to a different venue shortly thereafter, before a final ruling was issued. *See* Paper 4, 2.[10]

[10] The current status of district court litigation between the parties is unclear. Although Patent Owner represents that the underlying litigation was transferred from the Western District of Texas to the Northern District of California (*see* Paper 4, 2), and we have confirmed a docket entry from the Western District of Texas to that effect, a review of the docket from the Northern District of California does not reveal an active litigation between the parties. We invite the parties to clarify this point during trial. We also remind the parties that because the Board interprets claim terms in accordance with the standard used in federal district court in a civil action involving the validity or infringement of a patent, any further claim construction determinations concerning a claim term at issue in parallel district court litigation should be submitted to the Board, as discussed in the

Ultimately, Patent Owner posits that Petitioner must be held to the same claim construction position that it has advocated in a different forum, even though no district court has formally adopted that construction. Our rules do not require such a result. Rather, a petitioner complies with the applicable provisions of our rules by sufficiently identifying the claim construction it proposes as the basis for requesting review of the challenged claims. *See Western Digital Corp. v. Spex Techs, Inc.*, IPR2018-00084, Paper 14 at 10–12 (PTAB April 25, 2018). *Western Digital* states that "37 C.F.R. § 104(b)(3) does not require [p]etitioner to express its subjective agreement regarding correctness of its proffered claim constructions or to take ownership of those constructions." *Id.* at 11. Thus, petitioners may advocate in different fora for seemingly inconsistent positions (although perhaps at some peril to their credibility).

Here, Petitioner states that no express construction of any terms is needed, and thus does not advocate for construing "a Player" as produced by the authoring tool. In that regard, it is not clear that Petitioner is, in fact, advocating a position before us that is *inconsistent* with its previous position in district court. In particular, the Petition does not advocate one way or the other regarding whether the claimed "Player" is produced by an authoring tool. It is simply silent on that point. Moreover, as Patent Owner acknowledges, Petitioner's designated expert, Dr. Madisetti, opines that the prior art does teach or suggest that the authoring tool is configured to produce a Player. Ex. 1002 ¶¶ 145–147. And although Patent Owner does

Trial Practice Guide. *See* Patent Trial and Appeal Board Consolidated Trial Practice Guide, 46–47 (Nov. 2019) (citing 37 C.F.R. § 42.100(b) (2019)).

not explicitly take a position either way in its Preliminary Response, Patent Owner cites to testimony of its designated expert, Dr. Almeroth, who contends that the prior art does not teach or suggest a player produced by an authoring tool. Prelim. Resp. 17 (citing Ex. 2001 ¶¶ 53–58).

Although we lack guidance from the parties on this issue, we nevertheless consider whether claim 1 requires that the Player be produced by the authoring tool, or whether the Player is simply part of the recited system for generating code (which, of course, would not exclude its being produced by the authoring tool). On its face, the recitation of "a Player" as part of a system claim does not suggest the need for indicating the origin of the Player. Therefore, construing "a Player" as something that is specifically *produced by the authoring tool* would require us effectively *to add language* to claim 1 (inserting "produce" before "a Player," to tie back to "an authoring tool configured to"). This does not strike us as a matter of claim construction but rather as a matter of correction to the claim.

In that regard, we note that district courts may correct obvious errors in patent claims by interpretation of the patent. *See, e.g.*, *CBT Flint Partners, LLC v. Return Path, Inc.*, 654 F.3d 1353, 1358 (Fed. Cir. 2011) (holding that where "there is an obvious and correctable error in the claim, the construction of which is not subject to reasonable debate," correction is proper); *Novo Indus., L.P. v. Micro Molds Corp.*, 350 F.3d 1348, 1354 (Fed. Cir. 2003). Under *Novo*'s two-part test, a district court may

> correct an error in a patent by interpretation of the patent where no certificate of correction has been issued . . . only if (1) the correction is not subject to reasonable debate based on consideration of the claim language and the specification and

> (2) the prosecution history does not suggest a different interpretation of the claims.

Novo, 350 F.3d at 1354. Board decisions have adopted *Novo*'s two-part test to correct obvious errors. *See Research Prods. Corp. v. Honeywell Int'l Inc.*, IPR2018-00976, Paper 6 at 13 (PTAB Nov. 7, 2018) (citing *Novo*, 350 F.3d at 1354; *Apple Inc. v. Achates Reference Publ'g, Inc.*, IPR2013-00080, Paper 22 at 10–12 (PTAB June 3, 2013)).

On this record, and especially in the absence of guidance by the parties, we do not view the situation before us as falling within the ambit of *Novo*. On the one hand, the formatting structure of claim 1 places "a Player" at the same indentation level as several other limitations that appear to be sub-limitations of "an authoring tool configured to." Petitioner made the same observation before the district court, and argued that "the claim as structured makes no sense if no verb precedes 'a Player.'" Ex. 2009, 27. On the other hand, however, the punctuation of the claim gives a different impression. In particular, the action-verb sub-limitations following "an authoring tool configured to" are each separated by a *comma*—whereas the recitation of "a Player" is separated from the previous limitation by a *semicolon*. In the structure of claim 1, this punctuation suggests that "a Player" is not a sub-limitation of "an authoring tool configured to," but is instead a component *at the same level* as an authoring tool. We base this observation on the consistent use of commas in claim 1 to differentiate sub-limitations, as opposed to the consistent use of semicolons to separate limitations at a higher level. At this juncture, and on this record, we find this differentiation in punctuation to be a more compelling indication of the hierarchy of the limitations than the relative indentation of the clauses. And, under these circumstances, we find no error satisfying the *Novo* test to be evident. Rather, we determine that, on its face, claim 1 is subject to a reasonable reading in which no action verb must precede "a Player."

In short, on the record before us, we decline to interpret "a Player," as recited in claim 1, as something that the authoring tool must "produce." No further construction is necessary at this stage.

Our claim construction analysis in this Decision is preliminary, and does not preclude the parties from arguing proposed constructions of the claims during trial. Indeed, the parties are hereby given notice that claim construction, in general, is an issue to be addressed at trial. Claim construction will be determined at the close of all the evidence and after any hearing. The parties are expected to assert all their claim construction arguments and evidence in the Petition, Patent Owner's Response, Petitioner's Reply, or otherwise during trial, as permitted by our rules.

We determine, at this stage, that no explicit construction of any other terms is needed to resolve the issues presented by the arguments and evidence of record. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (per curiam) (claim terms need to be construed "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))).

E. Overview of Cited References

1. Anderson (Ex. 1003)

Anderson is an excerpt from a book published in 2006. Ex. 1003, 4.[11] Petitioner asserts that Anderson is prior art because it was published before April 2008, as shown by its "public accessibility and indexing in public libraries." Pet. 15. Petitioner also relies on the Declaration of Sylvia Hall-Ellis, Ph.D., who provides additional details as to the publication and public availability of Anderson. Ex. 1015 ¶¶ 43–49. Patent Owner does not, at this stage, dispute the prior-art status of Anderson.

Petitioner bears the burden at the institution stage "to identify with particularity evidence sufficient to establish a reasonable likelihood that the reference was publicly accessible before the critical date of the challenged patent, and therefore that there is a reasonable likelihood that it qualifies as a printed publication." *Hulu, LLC v. Sound View Innovations, LLC*, IPR2018-01039, Paper 29 at 16 (PTAB Dec. 20, 2019) (precedential). We determine that, at this point in the case and for purposes of institution, and based on the totality of the evidence to date, Petitioner has submitted sufficient evidence to establish a reasonable likelihood that Anderson was publicly accessible before the critical date of the challenged patent, and, thus Petitioner has established a reasonable likelihood that the reference qualifies as a printed publication. In particular, the face of Anderson bears a copyright date of 2006, a publication date of May 2006, an ISBN ("International Standard Book Number"), and identifies an established publisher (Prentice Hall).

[11] References to page numbers for this exhibit are to the page numbers as provided by Petitioner on the exhibit.

Ex. 1003, 4. *See Hulu* at 19 (noting that Petitioner had met its burden sufficiently for institution by submitting a copy of a reference that bore "a copyright date of 1990, a printing date of November 1992, and an ISBN date of 8/94," and was part of "a textbook from an established publisher . . . and a well-known book series"). Dr. Hall-Ellis testifies that Exhibit 1003 is a true and correct copy of the selected pages from the book as held by the Library of Congress, and also attests that the book was publicly available by or shortly after June 28, 2007, because by that date it had been received, cataloged, and indexed at the Library of Congress and made part of the Online Computer Library Center's ("OCLC's") bibliographic database. Ex. 1015 ¶¶ 43, 49; *see id.* ¶ 28 (describing OCLC).

Anderson is an excerpt from a book describing a software development tool known as Java Studio Creator (or "Creator"), which developers may use to create web applications. Ex. 1003, 35. For example, Anderson describes a Java application (Google1) that allows users to enter a search query and then access a Google web service (using a method called doGoogleSearch) to perform a search based on that query. *Id.* at 285, 289. Anderson also describes how to design the user interface for such application, including incorporating the Google logo, a text field entry box, and a search button, and to interface with Google's servers to utilize Google web services to perform the search and display the results. *Id.* at 286–288.

2. Bowers (Ex. 1004)

Bowers is an excerpt from a book titled "Advanced FileMaker® Pro 6 Web Development," published in 2003. Ex. 1004, 3.[12] Petitioner asserts that Bowers is prior art because it was published before April 2008, as shown by its "public accessibility and indexing in public libraries." Pet. 15. Petitioner also relies on the Declaration of Dr. Hall-Ellis, who provides additional details as to the publication and public availability of Bowers. Ex. 1015 ¶¶ 50–57. Patent Owner does not, at this stage, dispute the prior-art status of Bowers.

We determine that, at this point in the case and for purposes of institution, and based on the totality of the evidence to date, Petitioner has submitted sufficient evidence to establish a reasonable likelihood that Bowers was publicly accessible before the critical date of the challenged patent, and, thus Petitioner has established a reasonable likelihood that the reference qualifies as a printed publication. In particular, the face of Bowers bears a copyright date of 2003, a publication date of 2003, and an ISBN, and identifies an established publisher (Wordware Publishing). Ex. 1004, 3. *See Hulu* at 19. Dr. Hall-Ellis testifies that Exhibit 1004 is a true and correct copy of the selected pages from the book as held by the Library of Congress, and also attests that the book was publicly available by or shortly after April 19, 2004, because by that date it had been received, cataloged, and indexed at the Library of Congress and made part of the OCLC bibliographic database. Ex. 1015 ¶¶ 50, 56.

[12] References to page numbers for this exhibit are to the page numbers as provided by Petitioner on the exhibit.

Bowers describes techniques for developing web applications using FileMaker Pro, a commercially-available database software program. Ex. 1004, 68–121. Bowers describes the GoogleSearch web service—the same service discussed in Anderson—as a concrete example of how to incorporate a web service into an application. *Id.* at 97–110.

3. Jacobs (Ex. 1005)

Jacobs is an excerpt from a book titled "Foundation XML for Flash," published in 2006. Ex. 1005, 3.[13] Petitioner asserts that Jacobs is prior art because it was published before April 2008, as shown by its "public accessibility and indexing in public libraries." Pet. 15. Petitioner also relies on the Declaration of Dr. Hall-Ellis, who provides additional details as to the publication and public availability of Jacobs. Ex. 1015 ¶¶ 58–65. Patent Owner does not, at this stage, dispute the prior-art status of Jacobs.

We determine that, at this point in the case and for purposes of institution, and based on the totality of the evidence to date, Petitioner has submitted sufficient evidence to establish a reasonable likelihood that Jacobs was publicly accessible before the critical date of the challenged patent, and, thus Petitioner has established a reasonable likelihood that the reference qualifies as a printed publication. In particular, the face of Jacobs bears a copyright date of 2006, an ISBN, and identifies an established publisher (Springer-Verlag). Ex. 1005, 3. *See Hulu* at 19. Dr. Hall-Ellis testifies that Exhibit 1005 is a true and correct copy of the selected pages from the book as held by the Library of Congress, and also attests that the book was

[13] References to page numbers for this exhibit are to the page numbers as provided by Petitioner on the exhibit.

publicly accessible shortly after November 15, 2005, when it was received, catalogued, and indexed in the Langsam Library at the University of Cincinnati. Ex. 1015 ¶¶ 58–65.

Jacobs describes techniques for developing web applications using a technology known as Flash. Ex. 1005, 16. Like Anderson and Bowers, Jacobs includes a discussion of how to incorporate web services into a web application. *Id.* at 101–106. Jacobs explains that an industry standard technology known as the Simple Object Access Protocol ("SOAP") is used to specify the messages exchanged between a web service and a third party application that uses the web service. *Id.* at 65, 102. Jacobs shows an example of messages exchanged using SOAP in connection with the GoogleSearch web service, and in particular, to run a search. *Id.* at 102–106.

4. Ambrose-Haynes (Ex. 1006)

Ambrose-Haynes is an excerpt from a book titled "Professional ColdFusion 5.0," published in 2001. Ex. 1006, 6.[14] Petitioner asserts that Ambrose-Haynes is prior art because it was published before April 2008, as shown by its "public accessibility and indexing in public libraries." Pet. 15. Petitioner also relies on the Declaration of Dr. Hall-Ellis, who provides additional details as to the publication and public availability of Ambrose-Haynes. Ex. 1015 ¶¶ 66–73. Patent Owner does not, at this stage, dispute the prior-art status of Ambrose-Haynes.

[14] References to page numbers for this exhibit are to the page numbers as provided by Petitioner on the exhibit.

We determine that, at this point in the case and for purposes of institution, and based on the totality of the evidence to date, Petitioner has submitted sufficient evidence to establish a reasonable likelihood that Ambrose-Haynes was publicly accessible before the critical date of the challenged patent, and, thus Petitioner has established a reasonable likelihood that the reference qualifies as a printed publication. In particular, the face of Ambrose-Haynes bears a copyright date of 2001 and an ISBN. Ex. 1006, 6. *See Hulu* at 19. Dr. Hall-Ellis testifies that Exhibit 1006 is a true and correct copy of the selected pages from the book as held by the Library of Congress, and also attests that the book was publicly available by or shortly after August 2, 2001, when it was received, catalogued, and indexed in the Multnomah County Library in Portland, Oregon. Ex. 1015 ¶¶ 66–72.

Ambrose-Haynes describes a Java Virtual Machine ("JVM"), which is responsible for carrying out execution of Java applications. Ex. 1006, 55. Ambrose-Haynes explains that "JVMs are written for each operating platform"; therefore, "[e]xecuting the code through a JVM solves the problem of portability, since JVMs are written for each operating platform. In this way it is the JVM which is linked to the operating system, and handles all of the platform-specific details, while the source code [written] for interpretation is platform independent." *Id.*

5. NFS Administration (Ex. 1007)

NFS Administration comprises selections from a book titled "UNIX® System V NFS Administration." Ex. 1007, 1–3.[15] Petitioner asserts that NFS Administration is prior art because it was published before April 2008, as shown by its "public accessibility and indexing in public libraries." Pet. 15. Petitioner also relies on the Declaration of Dr. Hall-Ellis, who provides additional details as to the publication and public availability of NFS Administration. Ex. 1015 ¶¶ 74–81. Patent Owner does not, at this stage, dispute the prior-art status of NFS Administration.

We determine that, at this point in the case and for purposes of institution, and based on the totality of the evidence to date, Petitioner has submitted sufficient evidence to establish a reasonable likelihood that NFS Administration was publicly accessible before the critical date of the challenged patent, and, thus Petitioner has established a reasonable likelihood that the reference qualifies as a printed publication. In particular, the face of NFS Administration bears a copyright date of 1993, Library of Congress publication data, including an ISBN, and identifies an established publisher (Prentice Hall). Ex. 1007, 3. *See Hulu* at 19. Dr. Hall-Ellis testifies that Exhibit 1007 is a true and correct copy of the selected pages from the book as held by the Library of Congress, and also attests that the book was publicly available by or shortly after November 23, 1992, because by that date it had been received, cataloged, and indexed in the Library of

[15] References to page numbers for this exhibit are to the page numbers as provided by Petitioner on the exhibit.

Congress and made a part of the OCLC bibliographic database. Ex. 1015 ¶¶ 74–80.

NFS Administration describes a technology known as the Network File System ("NFS"). Ex. 1007, 10. NFS provides for a distributed file system that allows a computer to access files from remote storage over a computer network, such as a server connected to the Internet. *Id.* "NFS enables machines of different architectures running different operating systems to share resources across a network." *Id.*

6. Geary (Ex. 1011)

Geary is an excerpt from a book titled "Core JavaServer Faces," published in 2004. Ex. 1011, 3.[16] Petitioner asserts that Geary is prior art because it was published before April 2008, as shown by its "public accessibility and indexing in public libraries." Pet. 15. Petitioner also relies on the Declaration of Dr. Hall-Ellis, who provides additional details as to the publication and public availability of Geary. Ex. 1015 ¶¶ 82–89. Patent Owner does not, at this stage, dispute the prior-art status of Geary.

We determine that, at this point in the case and for purposes of institution, and based on the totality of the evidence to date, Petitioner has submitted sufficient evidence to establish a reasonable likelihood that Geary was publicly accessible before the critical date of the challenged patent, and, thus Petitioner has established a reasonable likelihood that the reference qualifies as a printed publication. In particular, the face of Geary bears a copyright date of 2004 and an ISBN, and identifies an established publisher

[16] References to page numbers for this exhibit are to the page numbers as provided by Petitioner on the exhibit.

(Prentice Hall). Ex. 1011, 3. *See Hulu* at 19. Dr. Hall-Ellis testifies that Exhibit 1011 is a true and correct copy of the selected pages from the book as held by the Library of Congress, and also attests that the book was publicly available by or shortly after October 20, 2004, because by that date it had been received, catalogued, and indexed in The Chinese University of Hong Kong and made a part of the OCLC bibliographic database. Ex. 1015 ¶¶ 82, 88.

Geary describes a technology known as JavaServer Faces ("JSF"). Ex. 1011, 1. Geary explains that JSF provides a specification for building and incorporating user interface components into web applications using Java, including text fields, checkboxes, and buttons. *Id.* at 81. Geary also discloses standard Java classes for building user interfaces for web applications, including UIOutput and UIInput which, respectively, handle output to and input from the user. *Id.* at 29.

F. Asserted Obviousness

1. Ground 1: Obviousness of Claims 1, 2, 5–7, and 12 over Anderson, Bowers, Jacobs, Ambrose-Haynes, and Geary

Petitioner contends that claims 1, 2, 5–7, and 12 are unpatentable under 35 U.S.C. § 103(a) over the combined teachings of Anderson, Bowers, Jacobs, Ambrose-Haynes, and Geary. Pet. 15–81. For the reasons explained herein, we are persuaded that Petitioner has established a reasonable likelihood of establishing unpatentability of at least one of these claims on this basis.

Patent Owner does not address Petitioner's arguments on a claim-by-claim basis, but presents several arguments against institution as to all claims and all grounds: (1) the Board should exercise its discretion to deny institution in view of prior petitions filed by other petitioners on the ’287 patent (Prelim. Resp. 27–35); (2) the Petition’s proposed combination fails to teach several limitations as recited in claim 1, namely (a) “access said computer memory to select the symbolic name” (*id.* at 7–13), (b) “produce an application including the selected symbolic name” (*id.* at 13–15), and (c) “each symbolic name has an associated data format class type corresponding to a subclass of user interface (UI) objects that support the data format type of the symbolic name” (*id.* at 17–22); (3) Petitioner’s proposed combination fails to teach “an authoring tool configured to [produce] a player,” as Petitioner has asserted in district court is required by claim 1 (*id.* at 15–17); (4) the Petition fails to demonstrate that it would have been obvious to combine Anderson with Ambrose-Haynes (*id.* at 22–25); and (5) the Petition improperly attempts to rely on a prior art system (*id.* at 25–26).

We have addressed Patent Owner’s arguments with regard to discretionary denial of review *supra* in Section II.A. We address the remainder of Patent Owner’s arguments herein in the context of evaluating the sufficiency of Petitioner’s showing as to each claim element.

a. Claim 1

(1) Preamble: “[a] system for generating code to provide content on a display of a device”

Petitioner contends Anderson discloses the subject matter of the preamble by disclosing Java Studio Creator, a software development and authoring tool for building web applications. Pet. 15 (citing Ex. 1003, 35). Petitioner asserts that the claimed “code” reads on either (1) Java source code generated by the developer for the web application using Creator, or

(2) compiled Java "bytecode" that, when executed, implements the web application. *Id.*

Petitioner also contends that Anderson discloses generating code "to provide content on a display of a device," as recited. *Id.* at 16. In particular, Petitioner contends that "[a] key feature of Java is the ability to access web services provided by third parties, such as Google." *Id.* Such web services include software application programming interfaces ("APIs") that are accessible over a network in a heterogeneous (cross-platform) environment. *Id.* (citing Ex. 1003, 285); *see also* Ex. 1003, 50, 52.

In connection with claim 1, Petitioner generally relies upon the teachings of Anderson in connection with a specific example of "how to create an application that uses the Google Web Service API," which is used for receiving and displaying search results in response to a user request. Pet. 16. Petitioner states that Anderson describes how to design the user interface for the application, and provides exemplary code (written in Java) for accessing Google web services to receive and display search results based on a user-provided search query. *Id.* Anderson then shows how such an application provides content on the display of a device, e.g. search results provided through the Google search web services. *Id.* at 16–17 (citing Ex. 1003, 307, Fig. 10-9).

Petitioner's arguments and evidence as to Anderson are persuasive on this record. Patent Owner does not, at this stage, present evidence or arguments addressing Petitioner's showing as to the preamble.

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that Anderson teaches "[a] system for generating code to provide content on a display of a device," as recited in the preamble of claim 1.[17]

(2) "computer memory storing a registry of:"

Claim 1 also recites "computer memory storing a registry of . . . a) symbolic names . . . and b) an address of the web service." Ex. 1001, 37:50–62.

Petitioner contends that Anderson teaches or suggests a registry stored in computer memory, arguing that the claimed "registry" corresponds in Anderson to an XML file—namely, a Web Services Description Language ("WSDL") file describing Google web services. Pet. 18. Petitioner further contends that the claimed "computer memory" corresponds in Anderson to the memory that stores the WSDL file. *Id.* at 17–18. Petitioner asserts that this memory may be memory on Google's servers that stores the GoogleSearch.wsdl file (*id.* at 18 (citing Ex. 1002 ¶ 78)); Petitioner also asserts that "it would also have been obvious that the WSDL file could have been stored in the memory of the developer computer running Creator." *Id.* at 19 (citing Ex. 1002 ¶ 79). Petitioner reasons that "Creator as disclosed in Anderson allows the developer to access WSDL information from local storage instead of a remote server." *Id.* Petitioner additionally relies on Bowers to confirm that the WSDL file described in Anderson is an XML file. *Id.* at 19–20 (citing Ex. 1004, 98–102). Because an XML file is within the Specification's description of a "registry" (Ex. 1001, 8:26–30), Petitioner

[17] Neither party takes a position at this stage as to whether the preamble is limiting. As Petitioner persuasively shows that Anderson teaches the subject matter of the preamble, we need not determine whether the preamble is limiting.

contends that the GoogleSearch.wsdl file described in both Anderson and Bowers is a "registry" as recited in claim 1. Pet. 20.

Petitioner's arguments and evidence are persuasive on this record. Patent Owner does not, at this stage, particularly challenge Petitioner's showing as to this limitation. Patent Owner does challenge certain aspects of Petitioner's showing as to when and how computer memory is accessed to select a symbolic name (*see* Prelim. Resp. 7–13), which is recited in a separate limitation and is discussed further below (Section II.F.1.a.(3)(c)).

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that Anderson (alone or combined with Bowers) teaches a "computer memory storing a registry," as recited in claim 1. We address the particular components recited to be stored in the registry in the next two sections.

(a) "symbolic names required for evoking one or more web components . . ."

Claim 1 recites that the registry stored in computer memory contains

> a) symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service[.]

Ex. 1001, 37:51–61.

Petitioner contends that the '287 patent specification does not define "symbolic name," but states that, "in the context of computer programming, this term generally refers to a name (usually in the form of an alphabetical character string) used to identify an entity or object in a computer system, such as a particular process, function, device, or data." Pet. 22; Ex. 1002

¶ 87 (citing IBM Dictionary of Computing (1994) (Ex. 1008), 5 (“**symbolic name** (1) In a programming language, a unique name used to represent an entity such as a field, file, data structure, or label.”)). Petitioner further contends that the claimed “symbolic names” correspond to at least the following names contained in the GoogleSearch.wsdl file (corresponding to the claimed “registry”) as described in Anderson: doGoogleSearch, doSpellingSuggestion, and doGetCachedPage. Pet. 22 (citing Ex. 1004, 105 (“The WSDL file tells us that Google supports messages such as doGoogleSearch() and doSpellingSuggestion().”)). Petitioner asserts that each of these symbolic names corresponds to a particular function or method (corresponding to the claimed “one or more web components”) provided by the GoogleSearch web service. *Id.* For example, doGoogleSearch invokes the Google web service to perform a web search based on a search query. *Id.* According to Petitioner, these symbolic names are also “required for evoking one or more web components,” as claimed, because the web application must use them to identify the particular method of the Google web services being requested. *Id.* at 23. The GoogleSearch.wsdl file (which Petitioner maps to the claimed “registry”) uses these names to uniquely identify these methods (e.g. <message name=“doGoogleSearch”>), which in turn identify the corresponding feature of the Google web service available to a web application. *Id.* at 24 (citing Ex. 1004, 102).

Petitioner further contends that the symbolic names described in Anderson are also “each related to a set of inputs and outputs of a web service obtainable over a network,” as also recited in claim 1. *Id.* To elaborate on this point, Petitioner points to the more detailed description of doGoogleSearch from Bowers, which identifies an “input message” and an "output message" type for each of the three methods provided by the Google web service (doGetCachedPage, doSpellingSuggestion, and doGoogleSearch). Pet. 24–25 (citing Ex. 1004, 100, 102); Ex. 1002 ¶ 97.

Petitioner further contends that the Google web service is "obtainable over a network," as also required by claim 1. Pet. 31 (emphasis omitted). In particular, Petitioner points to Bowers, which explains that the GoogleSearch.wsdl file "contains a <service> element that 'points to a URL. This URL is the address of the actual web service . . . ." *Id.* (citing Ex. 1004, 102). Petitioner explains that the "<service> URL identified in the version of GoogleSearch.wsdl in Bowers is <http://api.google.com/search/beta2>." *Id.* (citing Ex. 1004, 101). Petitioner reasons that "[a] web service available from a URL that points to 'google.com' is 'obtainable over a network,'" as claimed. *Id.* (emphasis omitted). Petitioner additionally notes that the symbolic names described in Anderson and Bowers (e.g., "doGoogleSearch") are character strings that "merely provide labels for the available methods; they do not contain a 'persistent address' or a 'pointer,' and do not by themselves identify the location (for storage or otherwise) of the web service or any output returned from it." *Id.* at 31–32.

Petitioner's arguments are persuasive and supported by the cited evidentiary disclosures.

Patent Owner does not challenge Petitioner's showing as to this particular limitation, but argues more generally that Petitioner's showing as to claim 1 is deficient because Petitioner fails to show that the prior art teaches or suggests a "selected symbolic name" and a distinct "input symbolic name." Prelim. Resp. 5–7. We discuss this argument *infra*

Section II.F.1.a.(4)(a) in connection with Petitioner's showing as to "input symbolic name."

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that the combination of Anderson and Bowers teaches or suggests

> symbolic names required for evoking one or more web components each related to a set of inputs and outputs of a web service obtainable over a network, where symbolic names are character strings that do not contain either a persistent address or pointer to an output value accessible to the web service[,]

as recited in claim 1.

(b) "each symbolic name has an associated data format class . . ."

Claim 1 also recites, with regard to the symbolic names, that "each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and has a preferred UI object." Ex. 1001, 37:57–61.

Petitioner contends that "[t]his limitation does little more than recite characteristics built into object-oriented programming systems such as Java – the end result of building the exemplary Java web application in Anderson for accessing the GoogleSearch web service." Pet. 32. Relying in part on the expert testimony of Dr. Madisetti, Petitioner then describes Java's object-oriented programming system, which is the system also described in Anderson and Bowers, in which procedures, known as "methods," can be accessed ("called") by other software. *Id.* at 32–33 (citing Ex. 1002 ¶ 101). In such an object-oriented system, Petitioner continues, the term "class" generally refers to the template for an object, and the term "subclass" is "simply a class that inherits [information and methods] from its parent class (referred to as a superclass).” *Id.* at 33 (citing Ex. 1002 ¶ 102; Ex. 1003, 43).

Petitioner then explains this limitation reads on Anderson’s description of an exemplary Java web application that accesses the “doGoogleSearch” method of the Google web service. *Id.* at 34–35. In particular, Petitioner explains that the claimed “symbolic name” corresponds to the symbolic name from the GoogleSearch.wsdl file, e.g., “doGoogleSearch.” *Id.* at 35. Petitioner also explains that this symbolic name also “has an associated data format class type corresponding to a subclass of User Interface (UI) objects” (*id.*), as claimed—that is, claimed “User Interface (UI) objects” correspond to the “Text Field” and “Grid Panel” as shown in Figure 10-1 of Anderson (*id.*), reproduced below:

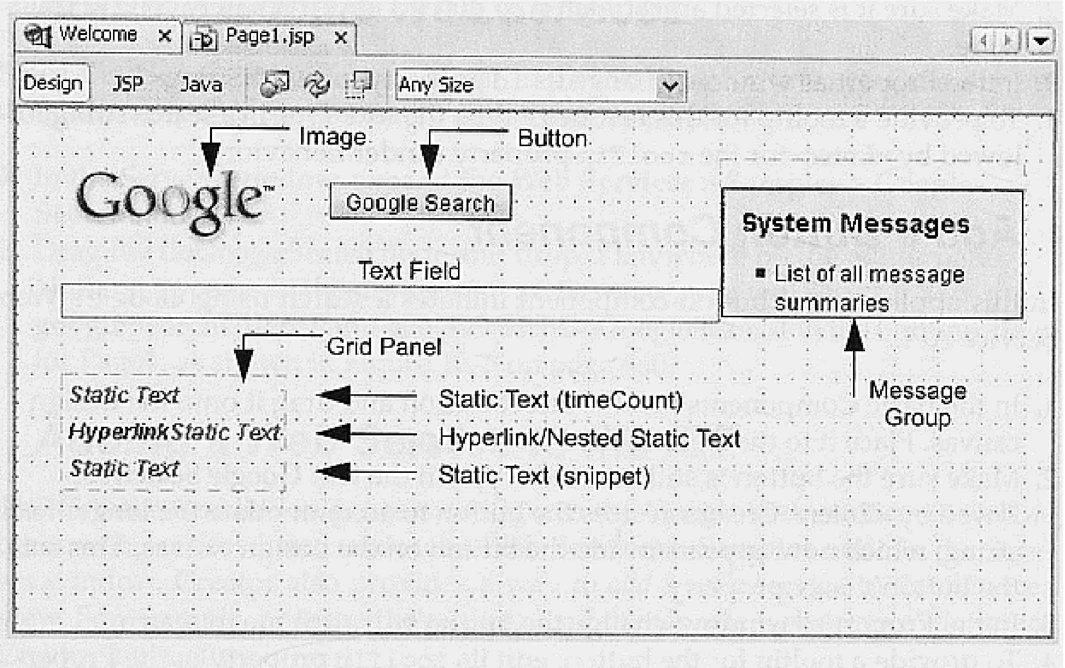


Figure 10–1 Creator’s design canvas view showing project Google1’s components

Anderson’s Figure 10-1 depicts Creator’s design canvas view.
Ex. 1003, 287.

Petitioner notes that Figure 10-1, above, shows several user interface ("UI") objects, including a "Text Field" for receiving a query for a search, and a "Grid Panel" containing "three text output components for displaying results of the search." Pet. 34–35 (citing Ex. 1003, 286). These UI objects "provide the input and output for the doGoogleSearch method." *Id.* at 35.

As for the claimed "data format class type," Petitioner notes that "at least two can be identified based on standard Java classes," as described in Geary. *Id.* at 36. In particular, Petitioner contends that Geary teaches standard Java classes for building user interfaces for web applications, including output classes, which belong to the "UIOutput class," and input components, that belong to the "UIInput class." *Id.* (citing Ex. 1011, 29, 87). Petitioner then asserts that

> [a]s applied to the exemplary web application in Anderson, therefore, it would have been obvious that when building the exemplary web application that accesses the GoogleSearch web service, the symbolic name doGoogleSearch (the name of the search method offered by the web service) "has an associated data format class type corresponding to a subclass of User Interface (UI) objects," for example, the UIInput or UIOutput classes which would handle, respectively, user input in the "Text Field" for the search query, and the static textual output for the "Grid Panel" of textual elements for displaying search results.

Id. at 37–38 (emphasis omitted) (citing Ex. 1003 ¶ 107).

Patent Owner contends that the Petition is deficient as to this limitation because "it only identifies an alleged data format class and then avers that the alleged data format class type is itself a subclass of UI objects." Prelim. Resp. 17. In particular, Patent Owner contends that the Petition asserts that the symbolic name doGoogleSearch has associated data format class types (UIInput and UIOutput), and then argues that these classes each qualify as a subclass of User Interface (UI) objects. *Id.* at 19 (citing Pet. 38–39). Patent Owner asserts that "[a]s the UIOutput and UIInput classes cannot be both a data format class type and a subclass of UI objects, the Petitioner necessarily fails." *Id.* at 20. Patent Owner also asserts that the Petition "provides no explanation or reason how the UIOutput and UIInput classes are each a 'data format class type.'" *Id.* at 20.

First, we note that the '287 Specification does not mention "data format class type" or "data format type" except in the claims themselves. The Specification does, however, refer repeatedly to various aspects of Java programming when describing the various components of the claimed invention and, in particular, specifically references Java when discussing the claimed classes. *E.g.*, Ex. 1001, 29:50–55 ("If a particular JAVA method is called, which requires an instance of a certain JAVA class to be executed by the run time engine, then that JAVA class is flagged, as well as any supporting methods, variables and/or object definitions."). Petitioner relies on Anderson, Bowers, and Geary—all of which describe building web applications using Java—as describing User Interface objects that support the data format type of the symbolic name. Pet. 35–40. In particular, Petitioner contends that the example web component with the symbolic name "doGoogleSearch" has an associated data format class type corresponding to a subclass of User Interface objects—namely, "Text Field" and "Grid Panel" are each user interface objects associated with doGoogleSearch. *Id.* at 34–35. In turn, "Text Field" and "Grid Panel" each correspond to a UIInput or a UIOutput class, which, respectively, handle user input for the "Text Field" search query and the static textual output for the "Grid Panel" elements of doGoogleSearch. *Id.* at 35–39.

Patent Owner's arguments regarding deficiencies in the Petition as to this limitation are unavailing. In particular, we disagree with Patent Owner's assertion that Petitioner is arguing that UIOutput and UIInput are both a "data format class type" and a "subclass of UI objects." *See* Prelim. Resp. 19–20. Rather, based on Petitioner's arguments, it appears Petitioner contends that the "data format class type" for the UIInput subclass of UI objects associated with doGoogleSearch is "user input," and for the UIOutput subclass of UI objects associated with doGoogleSearch is "static textual output." *See* Pet. 37–38.

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that the combination of Anderson, Bowers, and Geary teaches or suggests "each symbolic name has an associated data format class type corresponding to a subclass of User Interface (UI) objects that support the data format type of the symbolic name, and has a preferred UI object," as recited in claim 1.

(c) "an address of the web service"

Claim 1 also recites that the registry stored in computer memory includes "an address of the web service." Ex. 1001, 37:63.

Petitioner contends Anderson teaches this limitation by disclosing that the GoogleSearch.wsdl file (which Petitioner maps to the claimed "registry," as discussed *supra* in Section II.F.1.a.(2)) "contains a <service> element pointing to a URL," and that this URL provides "the address of the web service," as claimed, "because it provides the 'address of the actual web service and tells us where to send the request.'" Pet. 42 (citing Ex. 1004, 102). Petitioner notes that, "[i]n this example, the address is <http://api.google.com/search/beta2>." *Id.* (citing Ex. 1004, 101; Ex. 1002

¶¶ 98, 115).

Petitioner's arguments and evidence as to this limitation are persuasive on this record. Patent Owner does not, at this stage, particularly challenge Petitioner's showing as to this limitation.

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that Anderson teaches that the registry stored in computer memory includes "an address of the web service," as recited in claim 1.

(3) "an authoring tool configured to"

Claim 1 also recites "an authoring tool configured" to perform several recited functions. Ex. 1001, 37:63.

Petitioner contends that Anderson teaches "an authoring tool" by disclosing "a web application development system known as Creator." Pet. 42 (citing Ex. 1003, 35). Petitioner contends that "Anderson discloses using the Creator to create a new project for a web application, add graphics and text to the application, and add user interface controls such as a Text Field, Grid Panel, and button for initiating a Google search." *Id.* at 42–43 (citing Ex. 1003, 286–287). According to Petitioner, "Anderson further describes how to incorporate Google web services by specifying the WSDL file, and incorporating its functionality (including displaying search results) into the web application." *Id.* at 43 (citing Ex. 1003, 289–296; Ex. 1002 ¶ 116). Petitioner asserts that "Creator qualifies as 'an authoring tool' because it provides a user interface that enables a developer to create, generate, and/or modify content and/or functionality within a web application." *Id.* at 43 (citing Ex. 1002 ¶ 117; Ex. 1009, 6).

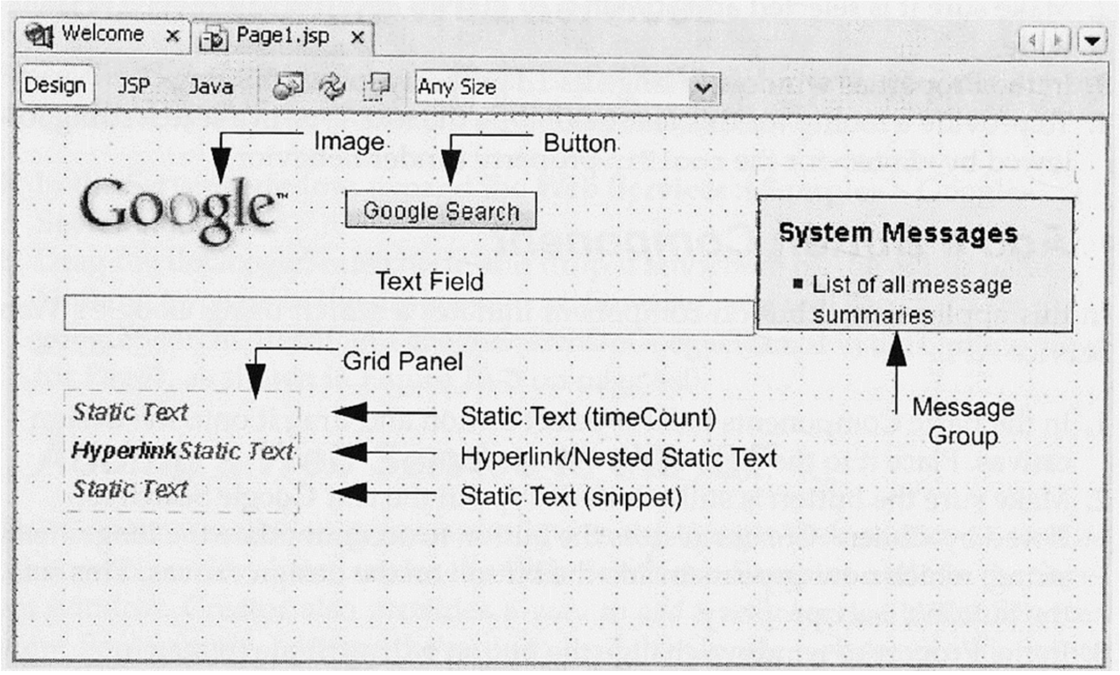
Petitioner's arguments and evidence as to this limitation are persuasive on this record. Patent Owner does not, at this stage, particularly challenge Petitioner's showing as to this limitation.

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that Anderson teaches "an authoring tool," as recited in claim 1. Petitioner's showing as to the additional limitations reciting the functionalities of the authoring tool are discussed below.

(a) "define a (UI) object for presentation on the display . . . "

Claim 1 also recites that the authoring tool is configured to "define a (UI) object for presentation on the display, where said defined UI object corresponds to a web component included in said registry selected from a group consisting of an input of the web service and an output of the web service." Ex. 1001, 37:64–38:1.

Petitioner contends that Anderson teaches this limitation in connection with its Figure 10-1, reproduced below:



Anderson's Figure 10-1 depicts Creator's design canvas view. Ex. 1003, 287.

Petitioner argues that "Figure 10-1 shows several user interface (UI) objects, including a 'Text Field' for receiving the query for a search, and a 'Grid Panel' containing three static text output components for displaying results of the search." Pet. 44 (emphasis omitted) (citing Ex. 1003, 286; Ex. 1002 ¶ 118). Petitioner asserts that each "Text Field" or "Grid Panel" would be a "user interface (UI) object." *Id*. at 45 (emphasis omitted) (citing Ex. 1002 ¶¶ 118–119).

Petitioner further argues that the UI objects "correspond to the web component included in said registry selected from the group consisting of an input of the web service and an output of the web service," as claimed, because "both the Text Field and Grid Panel correspond to the doGoogleSearch method in the GoogleSearch.wsdl file." Pet. 45. For confirmation, Petitioner points to Anderson's Figure 10-9, which shows the "user interface (UI) objects in action executing in a web application that obtains input in the form of a search query, and displays output in the form of search results." *Id*. at 46.

Petitioner's arguments and evidence as to defining a UI object are persuasive on this record. Patent Owner does not, at this stage, particularly challenge Petitioner's showing as to this limitation.

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that Anderson teaches the authoring tool being configured to "define a (UI) object for presentation on the display, where said defined UI object corresponds to a web component included in said registry selected from a group consisting of an input of the web service and an output of the web service," as recited in claim 1.

(b) "where each defined UI object is . . .
selected by a user of the authoring tool"

Claim 1 also recites

> where each defined UI object is either: 1) selected by a user of the authoring tool; or 2) automatically selected by the system as the preferred UI object corresponding to the symbolic name of the web component selected by the user of the authoring tool[.]

Ex. 1001, 38:1–6.

Petitioner contends this limitation is taught by Anderson because it describes how "[t]he Text Field is selected by the user of the authoring tool when the user navigates to the Basic Components palette in Creator," the user "select[s] component Text field and drag[s] it onto the design canvas," and the user "[p]lace[es] it below the Google logo." Pet. 48 (emphasis omitted) (citing Ex. 1003, 287). Petitioner similarly argues that, "[w]ith respect to the Grid Panel," the user "select[s] Grid Panel and place[s] it on the page below the text field component." *Id*. (citing Ex. 1003, 288).

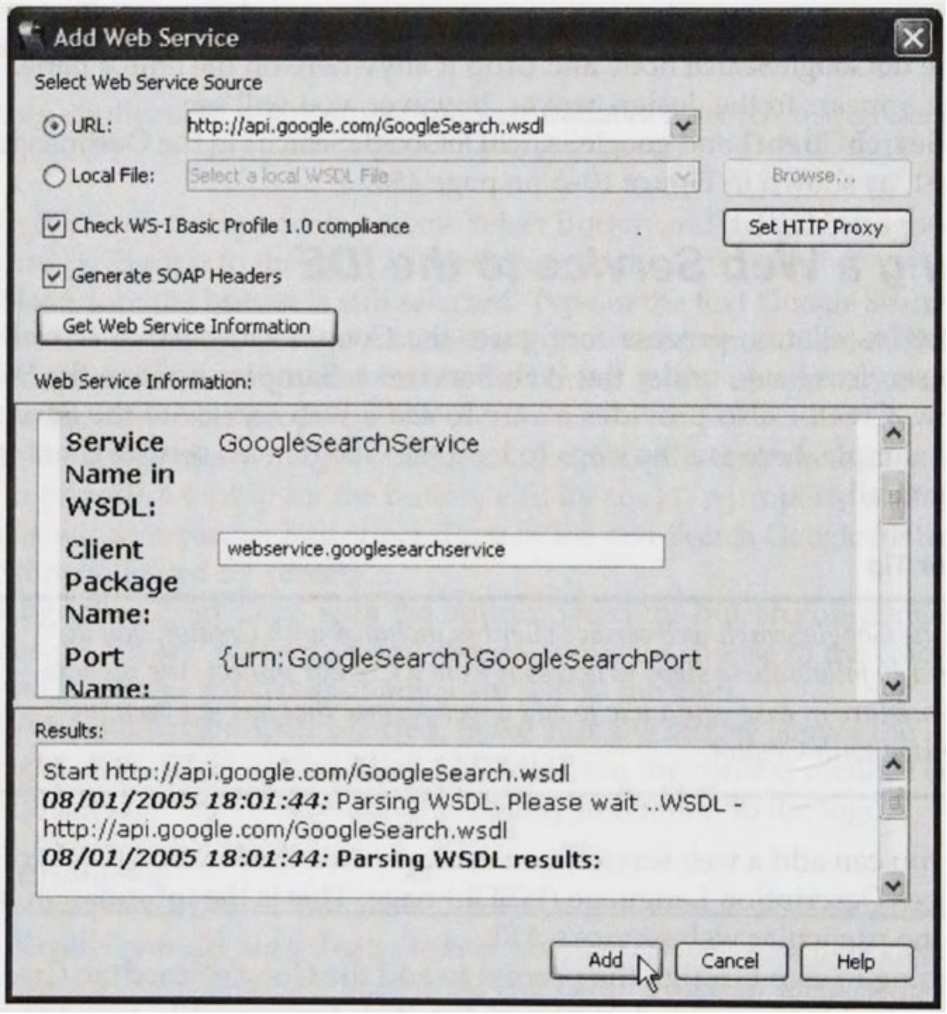
Petitioner's arguments and evidence as to selecting a defined UI object are persuasive on this record. Patent Owner does not, at this stage, particularly challenge Petitioner's showing as to this limitation.

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that Anderson teaches that "each defined UI object is . . . selected by a user of the authoring tool," as recited in claim 1.

(c) "access said computer memory to select the symbolic name . . ."

Claim 1 also recites that the authoring tool is configured to "access said computer memory to select the symbolic name corresponding to the web component of the defined UI object." Ex. 1001, 38:7–9.

Petitioner argues that "[t]he step of 'access[ing] said computer memory' occurs in Anderson when the developer using the authoring tool adds the web service to the current project by (for example) specifying the URL for the GoogleSearch.wsdl file using the Add Web Service dialog in Creator." Pet. 49 (emphasis omitted). Petitioner contends that this is depicted in Anderson's Figure 10-2, reproduced below.



Select Web Service Source

● URL: http://api.google.com/GoogleSearch.wsdl

○ Local File: Select a local WSDL File

☑ Check WS-I Basic Profile 1.0 compliance

☑ Generate SOAP Headers

Web Service Information:

Service Name in WSDL: GoogleSearchService

Client Package Name: webservice.googlesearchservice

Port Name: {urn:GoogleSearch}GoogleSearchPort

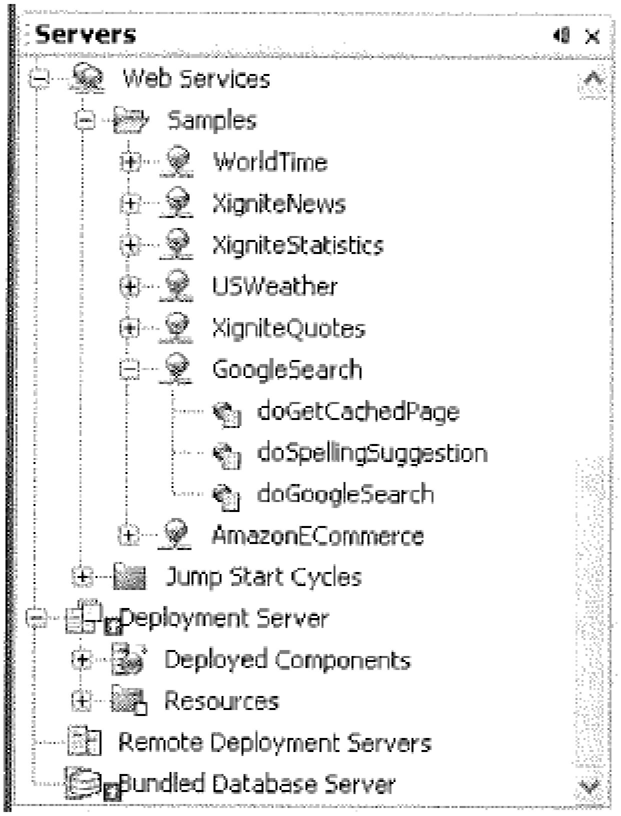
Results:

```
Start http://api.google.com/GoogleSearch.wsdl
08/01/2005 18:01:44: Parsing WSDL. Please wait ..WSDL -
http://api.google.com/GoogleSearch.wsdl
08/01/2005 18:01:44: Parsing WSDL results:
```

Anderson's Figure 10-2 depicts
Creator's Add Web Service dialog. Ex. 1003, 290.

Petitioner argues that "[i]n this example, under 'Select Web Service Source,' the developer inputs the URL for Google's WSDL file, which "is the GoogleSearch.wsdl file identified for the 'registry' limitation of claim 1[a]." Pet. 50–51 (emphasis omitted) (citing Ex. 1002 ¶¶ 77–78, 127). Petitioner argues that Figure 10-2 "further confirms that the authoring tool 'access[es] said computer memory,' because it reads the WSDL file stored in the memory and parses it to obtain and display information." *Id.* at 51 (citing Ex. 1002 ¶ 128). Petitioner further argues that "[t]he access of the

GoogleSearch.wsdl file . . . occurs in Anderson in order ‘to select the symbolic name corresponding to the web component of the defined UI object,’ as claimed,” because the “Add” button “makes the web service available to Creator through another window called Servers.” *Id*. (emphasis omitted) (citing Ex. 1003, 80 & Fig. 2-27; Ex. 1002 ¶¶ 129–130). The Servers window is shown in Figure 2-27, reproduced below:



Anderson’s Figure 2-27 depicts Creator’s Services window. Ex. 1003, 80.

Patent Owner argues “[t]he Petition fails to show that the alleged authoring tool of the proposed combination (‘Java Studio Creator’) selects any symbolic name from the alleged computer memory (i.e., some memory, be it Google servers or elsewhere, that stores the GoogleSearch.wsdl file).” Prelim. Resp. 7. Patent Owner contends that “Anderson teaches that Creator accesses the alleged computer memory to download the WSDL file" and that "[a]fter the alleged symbolic name is downloaded and no longer stored in the alleged computer memory, a user may perhaps go into Creator, open a separate window, and then select the alleged symbolic name." *Id*. at 7–8 (emphasis omitted). Patent Owner asserts that "[a]fter the Google web service is loaded into Creator from the alleged computer memory, a user can begin to use the web service" and that, for example, "once the GoogleSearch web service's WSDL fil (GoogleSearch.wsdl) . . . is parsed and its information is loaded into Creator, a user can use the information in a Creator project." *Id.* at 10 (citing Ex. 2001 ¶ 37). Patent Owner further asserts that "a user may then select a method (i.e., an alleged symbolic name), but the selected symbolic name is no longer within the WSDL file stored on the alleged computer memory." *Id*. (emphasis omitted). Thus, argues Patent Owner, "there is no teaching of Anderson that the extracted information is stored on the alleged computer memory" and "indeed, a POSITA would recognize that it would make no sense to again store that same information in the same memory, including within Google's server." *Id*. at 11 (citing Ex. 2001 ¶ 48).

We are not persuaded by Patent Owner's argument. Petitioner argues that "[t]he step of 'access[ing] said computer memory' occurs in Anderson when the developer using the authoring tool adds the web service to the current project by (for example) specifying the URL for the GoogleSearch.wsdl file using the Add Web Service dialog in Creator" (Pet. 49), but explains that the process continues through to selecting the "doGoogleSearch" method from the Server dialog box. *See* Pet. 49–53 (emphasis omitted). We see no reason why the accessing cannot include multiple steps, as Petitioner describes. Patent Owner only addresses the last step of the process.

Petitioner's arguments and evidence as to accessing computer memory to select the symbolic name are persuasive. Accordingly, based on the current record, for the purposes of this Decision, we are persuaded that Petitioner has demonstrated sufficiently that Anderson teaches "access[ing] . . . computer memory to select the symbolic name corresponding to the web component of the defined UI object," as recited in claim 1.[18]

(d) "associate the selected symbolic name with the defined UI object . . ."

Claim 1 also recites that the authoring tool is configured to "associate the selected symbolic name with the defined UI object, where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name." Ex. 1001, 38:10–14.

Petitioner argues that this limitation is taught in Anderson because the Grid Panel "is associated with doGoogleSearch because it displays the output of that method" and the Text Field "is also associated with doGoogleSearch because the text input by the user will provide parameter 'q,' i.e., the textual search query for doGoogleSearch." Pet. 53–54 (emphasis omitted) (citing Ex. 1003, 287–88, 303; Ex. 1002 ¶ 132).

[18] We could also find that the user is "accessing computer memory" to select the symbolic name when using Creator's Server dialog window when the WSDL file is local. The information that is displayed in the dialog box (e.g., the names of the services) must be stored in the computer memory in order to be listed in the dialog box, and the dialog box would allow the user to "access" that stored information (e.g., to select a service). Additionally, it appears that, when viewing the image displayed on the monitor or screen showing the dialog box, the user would be "accessing" the computer's video memory, which is also "computer memory," in order to select a name.

Petitioner further argues that "[t]he symbolic name is 'available' to those UI objects because . . . the software links these objects to the doGoogleSearch symbolic name by assigning them roles in the input and output of the doGoogleSearch method as explained above." *Id.* at 54 (emphasis omitted). Petitioner also argues that "a skilled artisan would have found this limitation to state nothing more than a basic principle of software design in an object-oriented system, *i.e.*, that the only information that should be made available to an object is the specific information it needs to carry out its purposes," and that it would have been obvious for the doGoogleSearch symbolic name to have been available only to UI objects that "play a role in submitting input to, or presenting output from, the doGoogleSearch method . . . as there would be no reason for the name to be available to other UI objects that lack any such role." *Id.* at 57–58 (citing Ex. 1002 ¶ 35).

Petitioner's arguments and evidence as to associating the symbolic name with the defined UI object are persuasive on this record. Patent Owner does not, at this stage, particularly challenge Petitioner's showing as to this limitation.

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that Anderson teaches "associat[ing] the selected symbolic name with the defined UI object, where the selected symbolic name is only available to UI objects that support the defined data format associated with that symbolic name," as recited in claim 1.

(e) "produce an application including the selected symbolic name of the defined UI object . . ."

Claim 1 also recites that the authoring tool is configured to "produce an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code." Ex. 1001, 38:15–17.

Petitioner argues that this is described in Anderson, which "explains that upon selecting the 'Run' option from the user interface, 'Creator builds the application, deploys it, and brings up a browser with the [developer's] web application running.'" Pet. 58 (emphasis omitted) (quoting Ex. 1003, 87). Petitioner argues that the application includes the selected symbolic name because "the whole purpose of the application is to invoke doGoogleSearch ('the selected symbolic name') to perform a search." *Id.* (emphasis omitted) (citing Ex. 1002 ¶¶ 136–137). Petitioner also argues it "would have been obvious that the application would include the selected symbolic name because the application must include the doGoogleSearch symbolic name to construct and issue a proper message to the GoogleSearch web service requesting performance of the method." *Id.* at 59 (emphasis, internal quotation marks, and brackets omitted) (citing Ex. 1002 ¶ 137).

Petitioner further argues that the Java application "qualifies as 'a device-independent code,' because bytecode instructions can run on any processor that has a Java Virtual Machine (JVM)." Pet. 60 (emphasis omitted) (citing Ex. 1002 ¶ 139).

Patent Owner argues that "Petitioner fails to show that the alleged Application includes 'doGoogleSearch'" because "the Petition only points to . . . a different method, 'googleSearchDoGoogleSearch1.'" Prelim. Resp.

13. Patent Owner contends that "[i]n Anderson, the Google web service, and *not* the alleged application includes the alleged selected symbolic name" and that "[a]s Anderson provides, googlesearchDoGoogleSearch1—and not the alleged selected symbolic name—is "use[d] to make calls to Google's web service API.'" *Id.* at 14 (citing Ex. 2001 ¶ 52). Patent Owner also argues that "Anderson teaches that the alleged selected symbolic name is not required to send a call to the Google web service, so it would not be obvious to modify the disclosures the Anderson to include that functionality." *Id.*

We find Patent Owner's arguments to be unavailing. The application includes "googleSearchDoGoogleSearch1," which "include[es] the selected symbolic name" "doGoogleSearch." We further agree with Petitioner that "the application must include the doGoogleSearch symbolic name to construct and issue a proper message to the GoogleSearch web service requesting performance of that method." *See* Pet. 59 (citing Ex. 1002 ¶ 137).

We find persuasive Petitioner's arguments and evidence as to producing an application. Accordingly, based on the current record, for the purposes of this Decision, we are persuaded that Petitioner has demonstrated sufficiently that Anderson teaches "produc[ing] an Application including the selected symbolic name of the defined UI object, where said Application is a device-independent code," as recited in claim 1.

(4) "a Player, where said Player is device-dependent code . . ."

Claim 1 also recites "a Player, where said Player is device-dependent code." Ex. 1001, 38:19–20.

Petitioner argues that this limitation reads on a Java Virtual Machine ("JVM"), as taught by Anderson and Ambrose-Haynes. Pet. 61–63. In particular, Petitioner contends that "the authoring tool under the proposed combination produces a Java application (in the form of device-independent bytecode) which is interpreted and executed by the JVM." Pet. 61. Petitioner asserts that this is confirmed by Ambrose-Haynes, which Petitioner states "provides additional detail about the JVM mentioned in Anderson." *Id.* Ambrose-Haynes describes JVMs as linked to different operating systems:

> Code compiled from Java is not executable code; rather, it's byte code. Byte code is a set of instructions that is interpreted within a Java Runtime Environment (JRE), consisting of the core Java class and support files, as well as a Java Virtual Machine (JVM), which actually executes the byte code. Executing the code through a JVM solves the problem of portability, since *JVMs are written for each operating system*. In this way *it is the JVM which is linked to the operating system*, and handles all of the platform-specific details, while the source code you write for interpretation is platform independent.

Ex. 1006, 55 (emphases added). Petitioner asserts that the JVM, as described in both Anderson and Ambrose-Haynes, is "device-dependent code," because it is the software directly executing on the computer and is tailored according to particular operating systems for different computers. *See* Pet. 61–62.

On this record, we find persuasive Petitioner's arguments and evidence as to this limitation.

Patent Owner does not challenge Petitioner's foregoing arguments as such. Rather, Patent Owner argues that the Petition is deficient because it fails to show that the Player is *produced by the authoring tool*, as Petitioner has argued in claim construction arguments in underlying district court litigation. Prelim. Resp. 15 (citing Ex. 2009 (Defendants' Responsive Claim

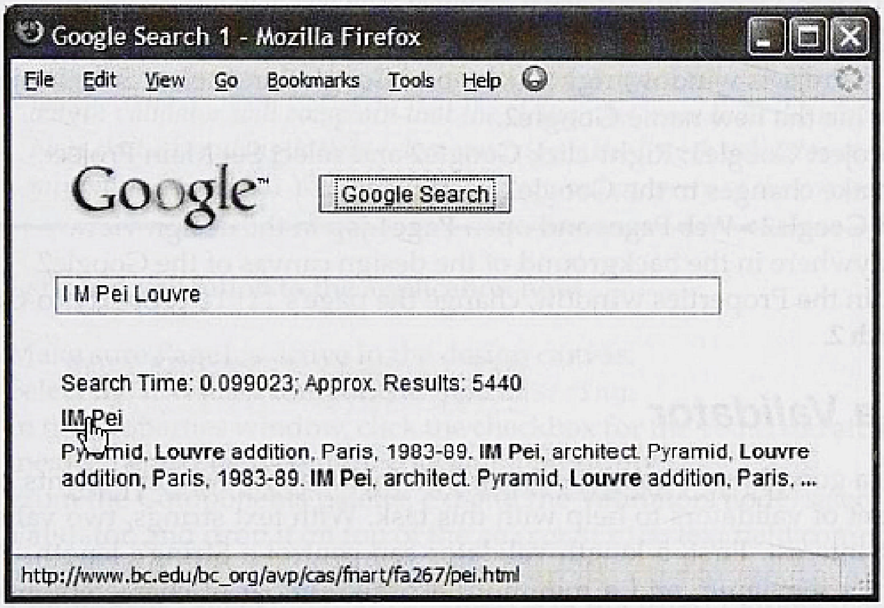
Constr. Br.), 26–30). We address this argument *supra* in Section II.D. As explained therein, we decline on this record effectively to add words to claim 1 and thus require that the authoring tool must "produce" the claimed Player.

On this record, and for purposes of this decision, we are persuaded that Petitioner has sufficiently shown that the cited art teaches or suggests "a Player, where said Player is device-dependent code," as recited in claim 1.

(a) "wherein . . . when the user of the device provides one or more input values associated with an input symbolic name . . ."

Claim 1 also recites, in connection with the claimed Player, "wherein, when the Application and Player are provided to the device and executed on the device, and when the user of the device provides one or more input values associated with an input symbolic name to an input of the defined UI object," several listed events happen. Ex. 1001, 38:20–25.

Petitioner argues that this limitation is taught by the proposed combination of Anderson and Ambrose-Haynes, in which the Java application of Anderson, built under the proposed combination (corresponding to the claimed "Application"), executes on a device using the JVM described in both Anderson and Ambrose-Haynes (corresponding to the claimed "Player"), in order to provide a search feature using doGoogleSearch. Pet. 63–64 (citing Ex. 1003, 76–77 (describing building application using Creator and running application on same machine (or different machine)), 303 (describing using same machine ("localhost") to run GoogleSearch application)). Petitioner asserts that Figure 10-9 of Anderson, reproduced below, shows a user providing a search query in the text box (corresponding to “one or more input values . . . to an input of the defined UI object”):



I M Pei Louvre

Search Time: 0.099023; Approx. Results: 5440

IM Pei

Pyramid, Louvre addition, Paris, 1983-89. IM Pei, architect. Pyramid, Louvre addition, Paris, 1983-89. IM Pei, architect. Pyramid, Louvre addition, Paris, ...

http://www.bc.edu/bc_org/avp/cas/fnart/fa267/pei.html

Figure 10–9 **First version of the Google Web Search application**

Anderson’s Figure 10-9 depicts a
Google Web Search application. Ex. 1003, 307.

Referring to Figure 10-9, above, Petitioner asserts that the search query “I M Pei Louvre” corresponds to the claimed “one or more input values,” and is provided to the Text Field (corresponding to the claimed “defined UI object”). Pet. 64. Petitioner further asserts that these input values are “associated with an input symbolic name” because they are associated with doGoogleSearch, the name of the method that receives the query and carries out the search. *Id.* (citing Ex. 1002 ¶¶ 122, 152). Petitioner’s arguments are persuasive and are supported by the cited evidence.

Patent Owner argues that Petitioner’s showing is lacking as to this limitation because the claims require *both* a “selected symbolic name” (addressed above in connection with the "produce an Application" limitation (Section II.F.1.a.(3)(d))) *and* an "input symbolic name" (as recited in the instant limitation), but Petitioner allegedly attempts to read both limitations on the same entity in the prior art. *See* Prelim. Resp. 5–7. In particular, Patent Owner contends that the Petition identifies Anderson's disclosure of "doGoogleSearch" as teaching the "selected symbolic name" claim element but Petitioner also (allegedly) identifies the identical "doGoogleSearch" as teaching the distinct "input symbolic name" claim element. *Id.* at 6–7 (citing Pet. 22 ("The claimed 'symbolic names' correspond to at least the following names in the GoogleSearch.wsdl file (the "registry"): doGoogleSearch, doSpellingSuggestion, and doGetCachedPage."), 68 ("'doGoogleSearch' (the 'corresponding input symbolic name to the web service')"); *see also id.* at 73 ("doGoogleSearch and the search query ('the input symbolic name and the user provided one or more input values')") (emphasis omitted). Patent Owner then contends that because "Anderson's disclosure of 'doGoogleSearch' cannot be reused to satisfy both the 'selected symbolic name' and 'input symbolic name' claim elements, the Petition[] necessarily fails." *Id.* at 7.

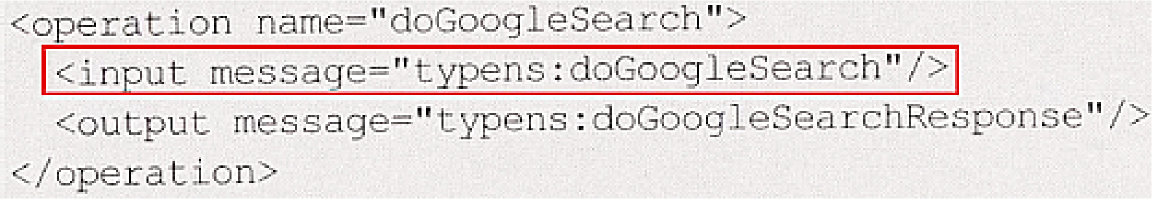
Patent Owner's argument is unavailing. On our reading, Petitioner does not map the claimed "selected symbolic name" and "input symbolic name" to the same entity; rather, the entities identified by Petitioner are different entities, albeit with the same name. That is, in Petitioner's proposed combination, "doGoogleSearch" is both a symbolic name for the search method as well as the symbolic name for the input message. *See* Pet. 65–66; Ex. 1004, 100. We discuss this point in more detail in connection with Petitioner's showing as to the next limitation.

On this record, and for purposes of this decision, we are persuaded that the proposed combination teaches or suggests "wherein . . . when the user of the device provides one or more input values associated with an input symbolic name," as recited in claim 1.

(b) "1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service . . ."

Claim 1 also recites, in connection with the claimed Player, one of the listed events is that "1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service." Ex. 1001, 38:26–28.

Petitioner argues that this limitation is taught by the disclosure in Anderson, and in particular Anderson's description of the method doGoogleSearch, which teaches that when the user of the Java application enters a search query into the text box and presses "Google Search" (as depicted in Figure 10-9 of Anderson), the search query is provided to the doGoogleSearch method of the web service. Pet. 65. Relying on code reproduced below from Bowers (which also describes the same doGoogleSearch method as described in Anderson), Petitioner contends that the GoogleSearch.wsdl file specifies that the search query is provided via an "input message" for the doGoogleSearch method.



```
<operation name="doGoogleSearch">
  <input message="typens:doGoogleSearch"/>
  <output message="typens:doGoogleSearchResponse"/>
</operation>
```

*The image above, as annotated by Petitioner, is a portion of a WSDL document describing Google Web APIs.
See Pet. 65; Ex. 1004, 102.*

As depicted above, the “doGoogleSearch” method allows a search query to be input via an “input message” (also named “doGoogleSearch”), annotated above with a red box, and provides a corresponding “output message” (named “doGoogleSearchResponse”). As Petitioner notes, the “doGoogleSearch” message type in the WSDL file, in turn, “defines a number of input parameters, including ‘**q**’ which . . contains the desired search query string.” Pet. 65–66 (citing Ex. 1004, 102). As Bowers explains, “[t]his definition [for doGoogleSearch] tells us the message name and the names and data types of each parameter . . . . So the WSDL file, among other things, defines the format and input types of all the messages the service listens to.” Ex. 1004, 102. Based on this disclosure, Petitioner asserts that “[w]hen invoking the 'doGoogleSearch method, therefore, the device provides an input message to the web service, the message identifying (among other things) the search query ‘q’ (‘the user provided one or more input values’) and the name of the ‘doGoogleSearch’ message (‘and corresponding input symbolic name’).” Pet. 66 (emphasis omitted) (citing Ex. 1002 ¶ 154).

Petitioner additionally points to Jacobs as confirming the foregoing by providing “a concrete example of a doGoogleSearch input message communicated to the Google web service in order to run an actual search.” *Id.* at 66–67 (citing Ex. 1005, 104; Ex. 1002 ¶¶ 154–155).

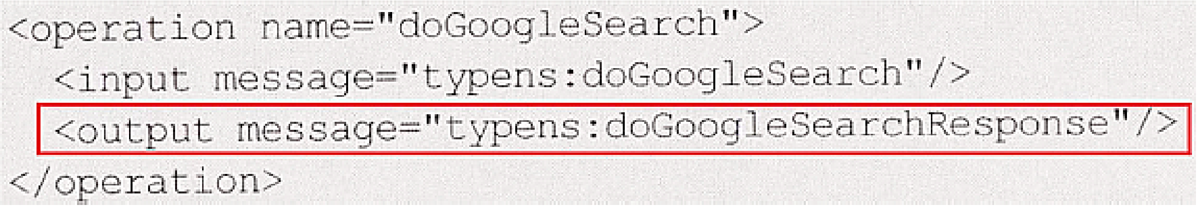
As noted above, Patent Owner challenges Petitioner’s showing as to claim 1 on the basis that Petitioner is allegedly reading both the “selected symbolic name” and the “input symbolic name” on the same entity (“doGoogleSearch”). Prelim. Resp. 5–7. Patent Owner’s argument is unavailing, as, on our reading, it misapprehends Petitioner’s arguments and supporting evidence. As shown in Bowers, the GoogleSearch.wsdl file contains *both* an "operation name" called "doGoogleSearch" *and* an "input message" for that operation, also called "doGoogleSearch." *See* Ex. 1004, 102. Petitioner reads the claimed "selected symbolic name" onto the "operation name," and reads the claimed "input symbolic name" onto the separate "input message." *See* Pet. 65–66. Although, in this particular example depicted in Bowers, the operation name and the input message appear to both have the *same name*, we are sufficiently persuaded for purposes of institution that they are, in fact, *distinct entities*. We note, however, that the record would benefit from additional clarity on this point, and we invite the parties to address this issue at trial.

On this record, and for purposes of this decision, we are persuaded that Petitioner has shown sufficiently that Anderson, in light of Bowers and Jacobs, teaches or suggests "1) the device provides the user provided one or more input values and corresponding input symbolic name to the web service," as recited in claim 1.

(c) "2) the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name . . ."

Claim 1 also recites, in connection with the claimed Player, one of the listed events is that "2) the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name." Ex. 1001, 38:29–32.

Petitioner argues that this limitation is taught by the disclosure in Anderson and Bowers regarding the GoogleSearch web service that utilizes the doGoogleSearch input message (which Petitioner contends contains both the input symbolic name and the user provided input values) to generate an output message that includes output search results based on the search query (which Petitioner contends corresponds to the claimed "one or more output values"). Pet. 70–75. In particular, Petitioner contends that "the GoogleSearch.wsdl file specifies an output message type used to communicate . . . search results to the application, *i.e.*, 'doGoogleSearchResponse.'" *Id.* at 70. Petitioner provides an excerpt from Bowers, reproduced below with annotation provided by Petitioner, which illustrates this point:



```
<operation name="doGoogleSearch">
  <input message="typens:doGoogleSearch"/>
  <output message="typens:doGoogleSearchResponse"/>
</operation>
```

The image above, as annotated by Petitioner, is a portion of a WSDL document describing Google Web APIs.
See Pet. 70; Ex. 1004, 102.

Petitioner notes that the "output message" depicted above, in the red box, is identified as "doGoogleSearchResponse," which Petitioner contends corresponds to the claimed "output symbolic name," and is associated with the output search results from the doGoogleSearch method. Pet. 70–71 (citing Ex. 1003, 297; Ex. 1004, 100). Petitioner further notes that, as described in Anderson, the search returns a "GoogleSearchResult object," which is associated with "a number of output values." *Id.* at 71–72 (citing Ex. 1003, 297, 299, Table 10.2). Petitioner then summarizes:

> Anderson thus confirms that the web service utilizes doGoogleSearch and the search query ("the input symbolic name and the user provided one or more input values") to generate a doGoogleSearchResponse output message containing search

> results ("for generating one or more output values").) Because the output message is specifically named in the GoogleSearch.wsdl file by its symbolic name (doGoogleSearchResponse), the "one or more output values hav[e] an associated output symbolic name," as recited in the claim.

Pet. 73 (emphasis omitted) (citing Ex. 1002 ¶¶ 160–161).

Petitioner additionally points to Jacobs as confirming the foregoing by providing a concrete example of a doGoogleSearch output message communicated from the Google web service to the application using SOAP. *Id.* at 73–74 (citing Ex. 1005, 105; Ex. 1003, 299; Ex. 1002 ¶¶ 156–158, 162).

Petitioner's arguments and evidence as to this limitation are persuasive on this record. Patent Owner does not, at this stage, particularly challenge Petitioner's showing as to this limitation.

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that Anderson teaches that "the web service utilizes the input symbolic name and the user provided one or more input values for generating one or more output values having an associated output symbolic name," as recited in claim 1.

(d) "3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object."

Claim 1 finally recites, also in connection with the claimed Player, one of the listed events is that "3) said Player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object." Ex. 1001, 38:33–36.

Petitioner argues that this limitation is taught by the combined disclosures of Anderson and Bowers with regard to the "doGoogleSearch" method described above. Pet. 75–78. In particular, Petitioner contends that after the search has been performed, the GoogleSearch web service sends the doGoogleSearchResponse output message to the Java application. *Id.* at 75. Petitioner notes that "[t]hat output message . . . identifies itself using its symbolic name and includes the search results, thus confirming that the Java application 'receives the output symbolic name and corresponding one or more output values.'" *Id.* (emphasis omitted). Petitioner further reasons that because it is the JVM (corresponding to the claimed "said Player") that executes the Java application on the device, the output message object is received by the JVM in order to be made available to the Java application. *Id.* (citing Ex. 1006, 55 ("[T]he JVM on any given platform will interpret and execute the [Java application's] code for you.")). Thus, according to Petitioner, "[t]he Player (as well as the Application) thus 'receives the output symbolic name and corresponding one or more output values,' as claimed." *Id.* (emphasis omitted) (citing Ex. 1002 ¶ 163).

Petitioner further alleges that the Player described in Anderson and Bowers also "provide[s] instructions for the display of the device to present an output value in the defined UI object," because "it carries out the functions of the Java application relating to displaying the user interface." *Id.* at 75–76 (emphasis omitted). Petitioner refers to Figure 10-1 of Anderson, reproduced below as annotated by Petitioner:

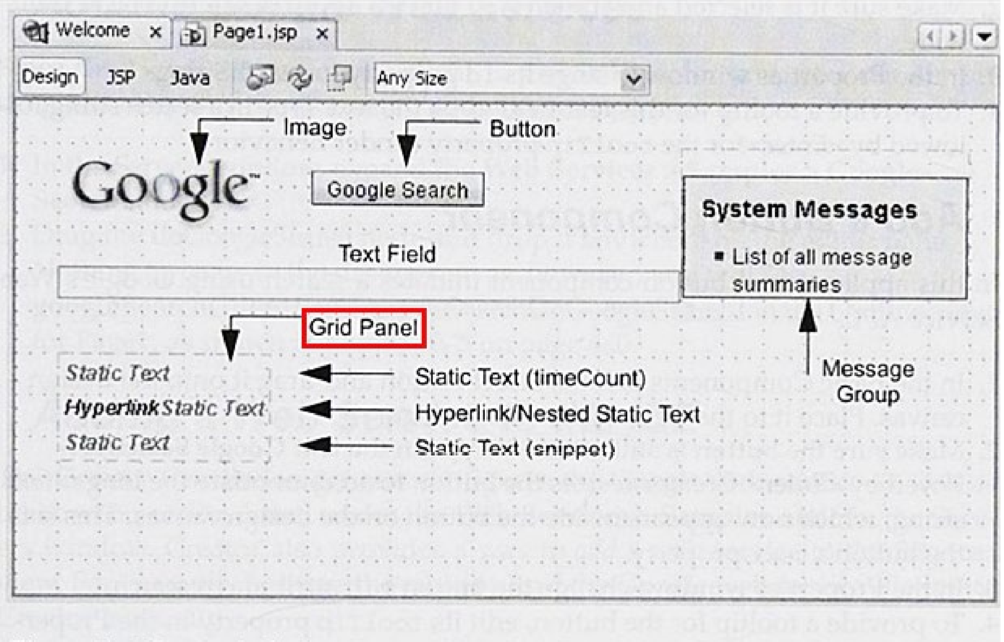
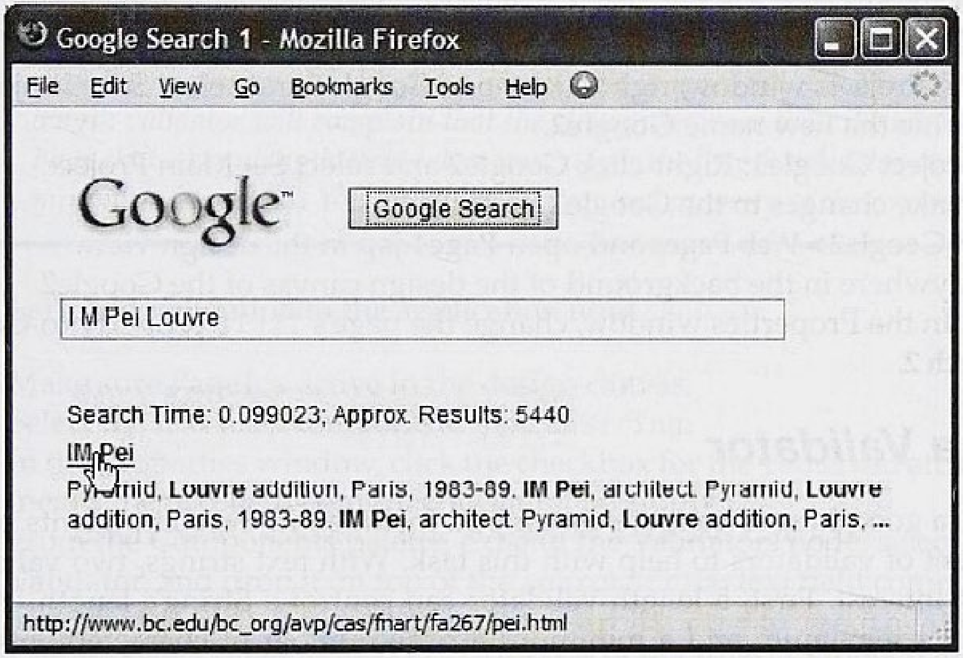


Figure 10–1 **Creator's design canvas view showing project Google1's components**

*Anderson's Figure 10-1 depicts Creator's design canvas view.
Ex. 1003, 287.*

Petitioner asserts that the Grid Panel shown in Figure 10-1 (corresponding to the claimed "defined UI object") presents the search results obtained from the doGoogleSearch method (corresponding to the claimed "present an output value in the defined UI object"). Pet. 76 (citing Ex. 1003, 303 ("The static text and hyperlink components that you nested inside the grid panel display portions of the result returned from the call to Google's search method.")) (emphasis omitted). Figure 10-9 of Anderson, reproduced below, shows the running Java application with the "Grid Panel" displaying these results.


Google Search 1 - Mozilla Firefox
File Edit View Go Bookmarks Tools Help
Google™
Google Search

I M Pei Louvre

Search Time: 0.099023; Approx. Results: 5440

IM Pei

Pyramid, Louvre addition, Paris, 1983-89. IM Pei, architect. Pyramid, Louvre addition, Paris, 1983-89. IM Pei, architect. Pyramid, Louvre addition, Paris, ...

http://www.bc.edu/bc_org/avp/cas/fnart/fa267/pei.html

Figure 10–9 **First version of the Google Web Search application**

*Anderson's Figure 10-9 depicts Creator's design canvas view.
Ex. 1003, 307.*

According to Petitioner:

> Under the proposed combination, the Java Virtual Machine (JVM) ("said Player") is responsible for carrying out all functions of the Java application. . . . It therefore would have been obvious that because the Java application requires the JVM to execute, the JVM ("said Player"), as well as the application, receives the output message from the web service ("output symbolic name and corresponding one or more output values") and provides instructions to display the search results on the display of the device ("provides instructions for the display of the device to present an output value in the defined UI object").

Pet. 77–78 (emphasis omitted) (citing Ex. 1006, 55; Ex. 1002 ¶ 165).

Petitioner's arguments and evidence as to this limitation are persuasive on this record. Patent Owner does not, at this stage, particularly challenge Petitioner's showing as to this limitation.

Based on the current record, for the purposes of this Decision, we are persuaded Petitioner has demonstrated sufficiently that the cited prior art teaches that "said Player receives the output symbolic name and corresponding one or more output values and provides instructions for the display of the device to present an output value in the defined UI object," as recited in claim 1.

(5) Motivation to Combine

Petitioner offers a series of reasons that it contends would have motivated the ordinarily skilled artisan to combine Anderson, Bowers, Jacobs, Ambrose-Haynes, and Geary. *See* Pet. 20–21 (Anderson and Bowers), 40–42 (Anderson and Bowers), 62–63 (Anderson and Ambrose-Haynes). Essentially, Petitioner uses Anderson as a base reference and asserts that it would have been obvious to add Bowers "for the contents of the Google WSDL file identified in Anderson to confirm what a skilled artisan would have already appreciated – that the GoogleSearch.wsdl file in fact, is an XML file and thus qualifies as a registry." Pet. 20. Petitioner then argues it would have been obvious to add Geary, which shows that a "Java web application built in accordance with the teachings of Anderson would have included classes/subclasses for user interface objects that receive textual input and handle textual output for the doGoogleSearch method." *Id*. at 40. Finally, Petitioner adds "Ambrose-Haynes largely to confirm what a skilled artisan would have understood about Java, *i.e.*, that the JVM is device-dependent code." *Id*. at 62. The secondary references are thus offered more to provide details that are omitted from Anderson than as the basis for any modification of Anderson. Petitioner also points to Jacobs as providing concrete examples of using the Google web service in a manner that Petitioner contends confirms Petitioner's contentions. *E.g.*, *id.* at 66–69, 73–75.

Patent Owner argues that "[t]he Petition fails to demonstrate that a POSITA would have found it obvious to combine Anderson and Ambrose-Haynes." Prelim. Resp. 22. Patent Owner asserts that the combination is improper because "the Petition generally alleges that it would be obvious to combine the two references because both discuss a Java virtual machine" and "[u]nder Petitioner's theory, a POSITA would find it obvious to combine a reference that mentions Java with any other number of references that also mention Java." *Id.* at 23. This argument is not persuasive. As noted, Ambrose-Haynes is cited merely as support for the assertion that "[a] skilled artisan would have recognized that a JVM is "device-dependent code." Pet. 61 (citing Ex. 1002 ¶ 141). We see no problem with relying on this reference to confirm that Java applications are device independent because they run on Java Virtual Machines written for each operating system.

Patent Owner also argues that Ambrose-Haynes "discourages the very combination that Petitioner proposes." Prelim. Resp. 23. Patent Owner contends that "Ambrose-Haynes, a book about ColdFusion, . . . tout[s] the advantages of ColdFusion, a server-side technology" over running Java applications locally. *Id*. at 24. We find also this argument to be unavailing, primarily because, as explained above, Petitioner is not relying on Ambrose-Haynes to support a modification of Anderson, but only to show that the Java code Anderson already discloses would have been known to be device independent. There is no combination or modification from which to teach away. Ambrose-Haynes simply confirms that applications written in Java have "all the advantages of the language, being stand-alone and platform independent." Ex. 1006, 43.

On this record, we agree with Petitioner's combination of these references and find a sufficient motivation to combine, to the extent one is needed.

Relatedly, Patent Owner also argues that Petitioner "improperly seeks to expand the disclosure of Anderson by relying on Bowers and Jacobs, and Geary, references which Petitioner[] argue[s] discuss this same system." Prelim. Resp. 25. Patent Owner then argues that, in effect, Petitioner "relies on the underlying *system* of Anderson as incorporating claim elements," which Patent Owner asserts is improper because an *inter partes* review must be based on "prior art constituting patents or printed publications." *Id.* (citing 35 U.S.C. § 311).

Patent Owner's contention is unavailing. As noted above, Petitioner relies on the secondary references essentially as evidence of what the ordinarily skilled artisan would have understood from Anderson's teachings, given that these references all describe techniques for creating Java web applications. As we note above, the secondary references are thus offered more to provide details that are not expressly stated in Anderson but would have been understood by the ordinarily skilled artisan. In cross-referencing teachings of several printed publications to create a full picture of a prior-art system, Petitioner does not rely on the underlying system itself, but is relying on printed publications describing that system as well as what the ordinarily skilled artisan would have understood from such printed publications. In other words, the fact that the printed publications describe the operation of an actual system does not mean that Petitioner is relying on the system itself, as opposed to printed publications describing the workings of such a system.

(6) Claim 1: Conclusion

For purposes of this Decision, based on the record presented, we are persuaded that Petitioner has cited sufficient evidence and provided sufficient articulated reasoning to explain why, at the time of the filing of the ’287 patent, one of ordinary skill would have been motivated to combine the teachings of Anderson, Bowers, Jacobs, Ambrose-Haynes, and Geary in the manner recited in claim 1.

b. Claims 2, 5–7, and 12

Petitioner additionally presents evidence that claims 2, 5–7, and 12, which depend from claim 1, are unpatentable over the combined teachings of Anderson, Bowers, Jacobs, Ambrose-Haynes, and Geary. Pet. 78–81. Patent Owner does not present arguments or evidence particularly directed to these claims, but relies on its arguments presented in connection with claim 1, which we address above.

For purposes of this Decision, based on the record presented, we are persuaded that Petitioner has cited sufficient evidence and provided sufficient articulated reasoning to explain why, at the time of the filing of the ’287 patent, one of ordinary skill would have been motivated to combine the teachings of Anderson, Bowers, Jacobs, Ambrose-Haynes, and Geary in the manner recited in claim 2, 5–7, and 12.

c. Ground 1: Conclusion

For the foregoing reasons, based on the current record, we determine that Petitioner has established a reasonable likelihood of prevailing in showing that claims 1, 2, 5–7, and 12 are unpatentable over the combination of Anderson, Bowers, Jacobs, Ambrose-Haynes, and Geary.

2. Ground 2: Claim 11 over Anderson, Bowers, Jacobs, Ambrose-Haynes, Geary, and NFS Administration

In Ground 2, Petitioner presents arguments and evidence that dependent claim 11 is unpatentable over the combination of Anderson, Bowers, Jacobs, Ambrose-Haynes, Geary, and NFS Administration. Pet. 81–85.

Claim 11 depends from claim 1, and recites "where said code is provided over said network." Ex. 1001, 39:6–7

Petitioner contends that "[i]t was well-known that users could store and retrieve any type of computer file (including files storing the Player and Application of claim 1) using remote storage provided over a network, such as the Internet." Pet. 81 (citing Ex. 1002 ¶¶ 174, 177). Petitioner relies in particular on NFS Administration as providing this teaching. *Id.* at 81–83. Relying in part on the testimony of Dr. Madisetti, Petitioner also provides arguments and evidence regarding a motivation to combine the art cited in Ground 1 with NFS Administration to add the ability "to obtain the code for the Application and the Player over the Internet using a distributed file system such as NFS." *Id.* at 83–85 (citing Ex. 1002 ¶¶ 178, 180–184).

Petitioner's contentions are persuasive and supported by cited teachings. NFS Administration explains, for example, that "NFS file sharing is used to make resources on a local system available to remote systems and, conversely, to access resources on remote systems from a local system. Using NFS, it is possible to share individual files, file hierarchies, and entire file systems across a network." Ex. 1007, 10.

At this stage, Patent Owner's arguments against Petitioner's showing are limited to those we have addressed above in connection with Ground 1. Patent Owner does not, at this stage, present separate arguments particularly against Petitioner's showing on this ground.

For purposes of this Decision, based on the record presented, we are persuaded that Petitioner has cited sufficient evidence and provided sufficient articulated reasoning to explain why, at the time of the filing of the '287 patent, one of ordinary skill would have been motivated to combine the teachings of Anderson, Bowers, Jacobs, Ambrose-Haynes, Geary, and NFS Administration in the manner recited in claim 11.

Based on the record before us, we determine Petitioner's arguments, evidence, and supporting testimony establish a reasonable likelihood that Petitioner would prevail in showing that claim 11 is unpatentable over the combination of Anderson, Bowers, Jacobs, Ambrose-Haynes, Geary, and NFS Administration.

III. CONCLUSION

For the foregoing reasons, we determine Petitioner has established a reasonable likelihood of prevailing in showing that one or more claims are unpatentable over the cited prior art.

IV. ORDER

For the reasons given, it is:

ORDERED that, pursuant to 35 U.S.C. § 314(a), an *inter partes* review is *instituted* on all challenged claims with respect to all grounds set forth in the Petition; and

FURTHER ORDERED that, pursuant to 35 U.S.C. § 314(a), *inter partes* review of the '287 patent is instituted commencing on the entry date of this Order, and pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is given of the institution of a trial.

PETITIONER:

Heidi Keefe
Philip Morton
Andrew Mace
Chih Yun Wu
Dustin Knight
COOLEY LLP
hkeefe@cooley.com
pmorton@cooley.com
amace@cooley.com
swu@cooley.com
dknight@cooley.com

PATENT OWNER:

Sal Lim
David L. Alberti
Russell
S. Tonkovich
Hong S. Lin
FEINBERG DAY KRAMER ALBERTI LIM
TONKOVICH & BELLOLI LLP
slim@feinday.com
dalberti@feinday.com
rtonkovich@feinday.com
hlin@feinday.com

Kenneth J. Weatherwax
Parham Hendifar
LOWENSTEIN & WEATHERWAX LLP
weatherwax@lowensteinweatherwax.com
hendifar@lowensteinweatherwax.com