

Petition for *Inter Partes* Review of
U.S. Patent No. 7,640,160 B2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

GOOGLE LLC,
Petitioner

v.

DIALECT, LLC,
Patent Owner

Case IPR2024-00753
U.S. Patent No. 7,640,160 B2
Issue Date: December 29, 2009

Title: SYSTEMS AND METHODS FOR RESPONDING TO
NATURAL LANGUAGE SPEECH UTTERANCE

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 7,640,160 B2**

Table of Contents

	Page
I. MANDATORY NOTICES UNDER §42.8(A)(1)	5
A. Real Party-In-Interest under §42.8(b)(1).....	5
B. Related Matters under §42.8(b)(2)	5
C. Lead and Back-Up Counsel under §42.8(b)(3)	6
D. Service Information	7
II. FEE PAYMENT	7
III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND CONSIDERATIONS UNDER §§ 314(A) AND 325(D).....	7
A. Grounds for Standing	7
B. Identification of Challenge and Statement of Precise Relief Requested	7
C. Considerations Under §§ 314(a) and 325(d).....	8
IV. OVERVIEW OF THE CHALLENGED PATENT.....	10
A. Person of Ordinary Skill in the Art (POSITA)	10
B. Specification Overview	10
V. CLAIM CONSTRUCTION	12
VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE.....	12
A. Overview of Grounds	12
B. Ground 1: Claim 12 Is Obvious Over Kennewick in View of Ross.	13
1. Claim 12: “A method for interpreting natural language utterances using knowledge-enhanced speech recognition engine, wherein the knowledge-enhanced speech recognition engine is configured to determine an intent and correct false recognitions of the natural language utterances, comprising:” (12[pre])	13

Table of Contents
(continued)

	Page
(a) “receiving a transcription of a natural language utterance at a computer comprising the knowledge-enhanced speech recognition engine;” (12[a])	14
(b) “identifying one or more contexts that completely or partially match one or more text combinations contained in the transcription, wherein identifying the matching contexts includes comparing the text combinations against the [sic] grammar expression entries in the [sic] context description grammar and against one or more expected contexts stored in a context stack;” (12[b])	23
(c) “scoring each of the identified matching contexts;” (12[c])	42
(d) “selecting the matching context having a highest score to determine a most likely context for the utterance; and” (12[d]).....	43
(e) “communicating a request to a domain agent configured to process requests in the most likely context for the utterance, the request formulated using at least one grammar expression entry in the context description grammar.” (12[e])	45
VII. CONCLUSION.....	47
CERTIFICATE OF SERVICE	49

Petition for *Inter Partes* Review of
U.S. Patent No. 7,640,160 B2

List of Exhibits

Exhibit No.	Description of Document
1001	U.S. Patent No. 7,640,160 B2 to Philippe Di Cristo et al. (filed August 15, 2005, issued December 29, 2009) (“ 160 ” or “ 160 patent ”)
1002	Declaration of Christopher M. Schmandt (“ Schmandt ”)
1003	<i>RESERVED</i>
1004	U.S. Patent App. Pub. No. 2004/0044516 A1 to Robert A. Kennewick et al. (filed June 3, 2003, published March 4, 2004) (“ Kennewick ”)
1005	U.S. Patent App. Pub. No. 2002/0133354 A1 to Steven I. Ross (filed August 16, 2001, published September 19, 2002) (“ Ross ”)
1006	<i>RESERVED</i>
1007	Redline comparison between Kennewick and the ’160 patent
1008	<i>RESERVED</i>
1009	Excerpts from <i>Microsoft Computer Dictionary</i> (5th ed. 2002)
1010	Excerpts from <i>Merriam-Webster’s Collegiate Dictionary</i> (10th ed. 1994)
1011	<i>RESERVED</i>
1012	Prosecution history of U.S. Patent No. 7,640,160 B2
1013	Proof of Service of Complaint, <i>Dialect, LLC v. Google LLC</i> , Case No. 1:23-cv-00378 (D. Del.)

I. MANDATORY NOTICES UNDER §42.8(A)(1)

A. Real Party-In-Interest under §42.8(b)(1)

Google LLC (“Petitioner”) is the real party-in-interest to this IPR petition.¹

B. Related Matters under §42.8(b)(2)

The ’160 patent is the subject of pending litigation in which Petitioner is named as a defendant, *Dialect, LLC v. Google LLC*, Case No. 1:23-cv-00378 (D. Del.) (filed April 3, 2023), and *Dialect, LLC v. Bank of America, NA*, Case No. 2:24-cv-00207 (E.D. Tex.) (filed March 22, 2024). Petitioner was first served with a complaint alleging infringement of the ’160 patent on April 6, 2023. (EX1013, p.002.) On March 28, 2024, Petitioner’s motion to transfer to the Northern District of California was granted. *See* Dkt. 32, Case No. 1:23-cv-00378 (D. Del.).

Petitioner is filing IPR petitions against eight patents asserted by Patent Owner in the pending litigation against Petitioner:

U.S. Patent No.	IPR Case
7,398,209 B2	IPR2024-00746
7,502,738 B2	IPR2024-00747
8,015,006 B2	IPR2024-00748
7,693,720 B2	IPR2024-00749

¹ Google LLC is a subsidiary of XXVI Holdings Inc., which is a subsidiary of Alphabet Inc. XXVI Holdings Inc. and Alphabet Inc. are not real parties in interest to this proceeding.

Petition for *Inter Partes* Review of
U.S. Patent No. 7,640,160 B2

U.S. Patent No.	IPR Case
9,031,845 B2	IPR2024-00750
8,447,607 B2	IPR2024-00751
8,849,652 B2	IPR2024-00752
7,640,160 B2	IPR2024-00753

C. Lead and Back-Up Counsel under §42.8(b)(3)

Petitioner provides the following designation of counsel.

LEAD COUNSEL	BACK-UP COUNSEL
Heidi L. Keefe (Reg. No. 40,673) hkeefe@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington, DC 20004 Tel: (650) 843-5001 Fax: (650) 849-7400	Andrew C. Mace (Reg. No. 63,342) amace@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Ave. NW, Suite 700 Washington D.C. 20004 Tel: (650) 843-5287 Fax: (650) 849-7400
	Mark R. Weinstein (Admission <i>pro hac vice</i> to be requested) mweinstein@cooley.com COOLEY LLP ATTN: Patent Group 1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (650) 843-5007 Fax: (650) 849-7400
	Reuben Chen (Admission <i>pro hac vice</i> to be requested) rchen@cooley.com COOLEY LLP ATTN: Patent Group

LEAD COUNSEL	BACK-UP COUNSEL
	1299 Pennsylvania Avenue NW Suite 700 Washington D.C. 20004 Tel: (650) 843-5007 Fax: (650) 849-7400

D. Service Information

This Petition is being served by Federal Express to the correspondence address of record for the '160 patent, 174899 - The Law Offices of David A. Gerasimow, P.C., 211 W. Wacker Dr., Ste. 1717, Chicago, IL 60606. Petitioner consents to electronic service at the addresses provided above for lead and back-up counsel.

II. FEE PAYMENT

Petitioner requests review of 1 claim, with a \$41,500 payment.

**III. REQUIREMENTS UNDER §§ 42.104 AND 42.108 AND
CONSIDERATIONS UNDER §§ 314(A) AND 325(D)**

A. Grounds for Standing

Petitioner certifies that the '160 patent is available for IPR and that Petitioner is not barred or otherwise estopped.

**B. Identification of Challenge and Statement of Precise Relief
Requested**

Petitioner requests IPR institution based on:

Ground	Claim	Basis for Challenge under §103
1	12	Kennewick (EX1004) and Ross (EX1005)

C. Considerations Under §§ 314(a) and 325(d)

§314(a): Petitioner is unaware of another post-grant challenge against the '160 patent. The pending litigation against Petitioner is still in early stages, with no discovery or claim construction having occurred and no trial date set, and as noted, was recently transferred to the Northern District of California. If IPR is instituted, Petitioner intends to bring a motion to stay the litigation.

§325(d): During prosecution of the '160 patent, the Examiner rejected pending claims as anticipated by U.S. Patent No. 7,137,126 to Coffman and obvious over Coffman in view of Kennewick. (EX1012, pp.00196-00210.) In response, the applicant canceled the pending claims and submitted a set of new claims. (EX1012, pp.00220-00227.) The new claims included a claim 45, which ultimately issued in amended form as claim 12. (EX1012, pp.00224-00225, 00369-00370.) The Examiner did not cite Coffman or Kennewick against claim 45. The Examiner distinguished Coffman, but did not discuss Kennewick, stating:

Each of independent claims 30, 45, 46, and 48 require identifying matching contexts in a transcribed utterance by comparing the text combinations against grammar expression entries in the context description grammar and against one or more expected contexts stored in a context stack. Dudemaine et al. and the additional prior art used in

Petition for *Inter Partes* Review of
U.S. Patent No. 7,640,160 B2

this Office action do not disclose or suggest such a feature. Additionally, Coffman et al. (used in a previous Office Action for teaching similar features) include a context stack, however, Coffman et al. do not disclose or suggest comparing text combinations against both grammar expression entries in the context description grammar and against one or more expected contexts stored in a context stack.

(EX1012, p.00305; *see also id.*, p.00382.) Why the Examiner did not appreciate Kennewick's significance is unclear—as explained below, Kennewick and the '160 patent describe substantially the same subject matter, since the '160 patent incorporates much of Kennewick nearly verbatim. (*See* EX1007.) But the Examiner did not have the **Ross** reference that this Petition also relies on, which provides further context and detail for understanding Kennewick and how its disclosures provide a compelling ground of obviousness. Petitioner respectfully submits that the Examiner materially erred by not rejecting the claims as obvious over Kennewick. As explained below, the combination of Kennewick and Ross discloses and renders obvious claim 12, including the features that the Examiner identified as not disclosed or suggested by Kennewick—i.e., comparing text combinations against both grammar expression entries in the context description grammar and against one or more expected contexts stored in a context stack.

IV. OVERVIEW OF THE CHALLENGED PATENT

A. Person of Ordinary Skill in the Art (POSITA)

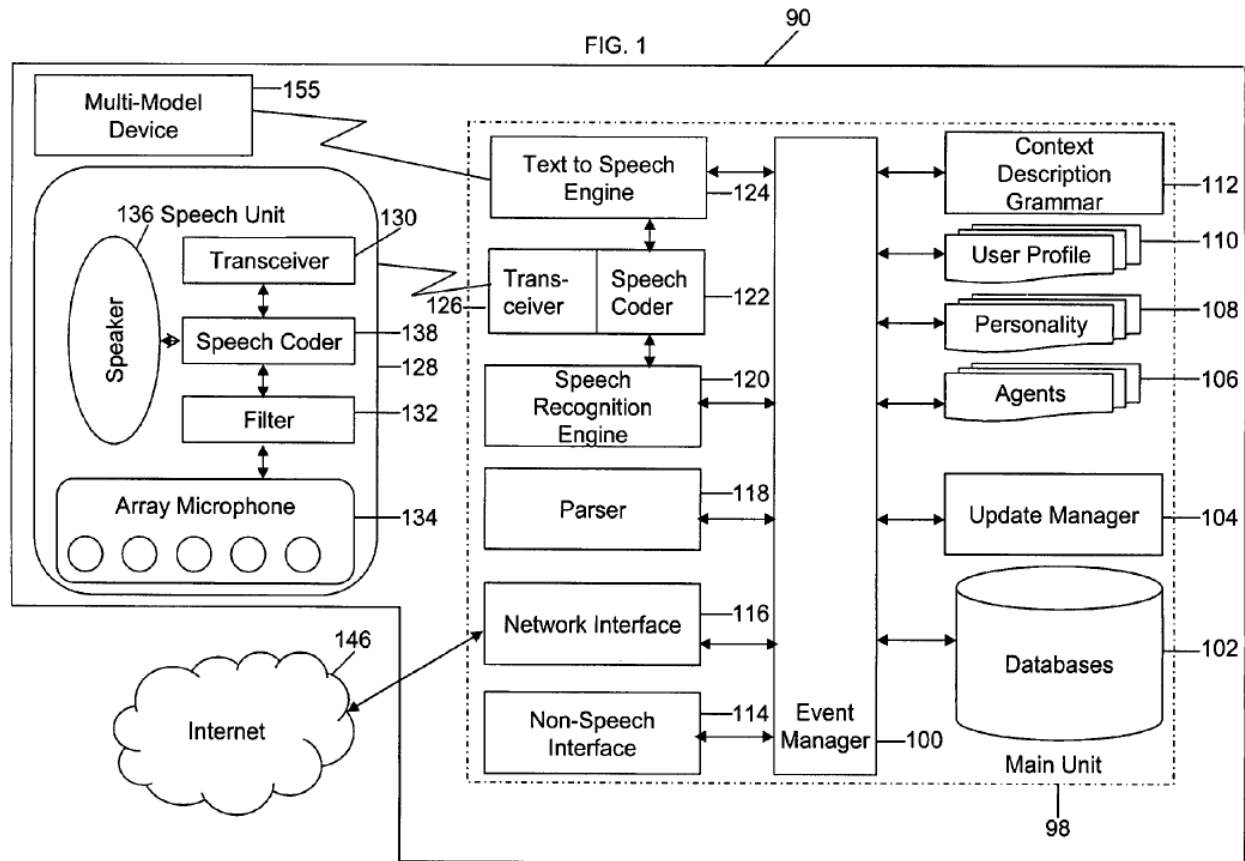
A POSITA as of August 5, 2005 would have possessed a bachelor's degree in electrical engineering or computer science, and two years of work experience in the field of speech- and natural language-interfaces for interacting with and directing computer systems and software, including speech recognition and processing (and more experience could substitute for less formal education, and vice versa). (EX1002 ("Schmandt"), ¶¶11-15.)

B. Specification Overview

The '160 patent states that it "relates to retrieval of information or processing of commands through a speech interface and/or a combination of a speech interface and a non-speech interface." ('160, 1:8-10.)

The '160 patent admits that "[s]peech recognition has steadily improved in accuracy and today is successfully used in a wide range of applications," and that "[n]atural language processing has been applied to the parsing of speech queries." ('160, 1:35-38.) According to the '160 patent, however, "current systems do not reliably provide a complete environment for users to submit verbal and/or textual communications through natural language queries that are processed to provide natural responses." ('160, 1:38-41.) The '160 patent purports to "overcome...drawbacks of prior systems." ('160, 1:41-44.)

Figure 1 of the '160 patent, reproduced below, “is an overall diagrammatic view according to one embodiment of the invention”:



(’160, Fig. 1, 9:63-64.) As shown in the figure above, the system includes, among other things, speech unit **128**, speech recognition engine **120**, context description grammar module **112**, parser **118**, and agents **106**. Speech unit **128** includes a microphone to receive a spoken utterance from a user. (’160, 11:55-57.) After initial processing, the speech is passed to speech recognition engine **120** for processing using the context description grammar module **112**. (’160, 12:23-25.) Then, “[a]ny recognized information may be processed by the parser **118**, which transforms

information into complete algorithms and questions using data supplied by knowledge agents.” (’160, 12:25-28.) “The knowledge agents may then process the commands or questions.” (’160, 12:32-36.) In order to select the correct agent to process the command or question, “[t]he parser **118** uses a scoring system to determine the most likely context or domain for a user’s question and/or command. The score is determined from weighing a number of factors including, the user profile **110**, the domain agent’s data content and previous context. Based on this scoring, the system **90** invokes the correct agent.” (’160, 21:10-15.)

V. CLAIM CONSTRUCTION

Express claim construction is unnecessary because the prior art renders the challenged claims obvious under any reasonable construction. Petitioner reserves its right to respond to Patent Owner’s claim construction arguments.

VI. THE CHALLENGED CLAIMS ARE UNPATENTABLE

A. Overview of Grounds

The challenged claim attempts to lay claim to well-known techniques for speech-enabled interfaces for interacting with and directing computer systems. The claim employs standard techniques that were known long before the ’160 patent, such as using speech recognition and grammars to select an application for carrying out a command or question included in a spoken utterance.

The Petition relies primarily on **Kennewick** (**EX1004**). Kennewick and the '160 patent share a common named inventor, Robert A. Kennewick, and the '160 patent incorporates much of Kennewick nearly verbatim. (*See* EX1007.) As a result, not surprisingly, Kennewick and the '160 patent describe substantially the same subject matter. The Petition combines Kennewick with **Ross** (**EX1005**) for its teachings about using grammars, contexts and a context list to select an application to process a spoken command or question.

Both Kennewick and Ross are prior art under at least pre-AIA §102(b).

B. Ground 1: Claim 12 Is Obvious Over Kennewick in View of Ross.

- 1. Claim 12: “A method for interpreting natural language utterances using knowledge-enhanced speech recognition engine, wherein the knowledge-enhanced speech recognition engine is configured to determine an intent and correct false recognitions of the natural language utterances, comprising:” (12[pre])**

As explained in detail for 12[a], Kennewick discloses a “**knowledge-enhanced speech recognition engine**” in the form of a parser that interprets a transcribed natural language utterance by (1) determining the intent of a request in a transcribed natural language utterance to select an application to invoke to process a command or question in the utterance, and (2) identifying potentially false recognitions and prompting the user to correct them. (Kennewick, e.g., ¶¶0031, 0092, 0152-0153; Schmandt, ¶51.) As shown below, the claimed method is disclosed and rendered obvious by the prior art.

(a) **“receiving a transcription of a natural language utterance at a computer comprising the knowledge-enhanced speech recognition engine;” (12[a])**

Kennewick discloses receiving **“a natural language utterance”** from a user. (Kennewick, ¶0010 (“[T]he system comprises a speech unit interface device that receives spoken natural language queries, commands and/or other utterances from a user.”).)² With respect to a **“transcription,”** transcribing (i.e. generating a “transcription”) is broadly understood as some form of recording, such as making a copy, whether verbatim or not, or converting information or data from one form to another. (Schmandt, ¶52 (citing EX1010, p.006 (entry for “transcribe”)).) Consistent with this ordinary understanding, Kennewick discloses the natural language utterance input from the user is passed to a speech recognition engine that generates a **“transcription”** of the utterance by recognizing words and phrases. (Kennewick, ¶¶0142 (“The speech received by the array microphone 134 and passed through the filter **132** may be sent to the speech digitizer or coder 138.”), 0147 (“The output of the speech coder **122** may be fed to the speech recognition engine **120**. The speech recognition engine **120** recognizes words and phrases, using information in the dictionary and phrase tables 112, and passes these to the parser 118 for interpretation.”); *see also id.*, ¶¶0186 (“The process **400** may begin when a user’s utterance (i.e., user query and/or command) is captured through accurate speech

² All underlining added to quoted material unless noted.

recognition operating in a variety of real-world environments at **402.**”), 0187 (“The process 450 may begin when a user’s utterance (i.e., user query and/or command) is captured through accurate speech recognition operating in a variety of real-world environments at 452.”); Schmandt, ¶52.) The speech recognition engine’s “**transcription**” comprising the recognized words and phrases is then passed to a parser for interpretation. (Kennewick, ¶0147 (“The speech recognition engine **120**...passes [recognized words and phrases] to the parser **118** for interpretation.”); *see also id.*, ¶¶0152 (“Once the words and phrases have been recognized by the speech recognition engine **120**, the tokens and user identification is passed to the parser **118**.”), (“The decoded speech is processed by the speech recognition engine **120**....The recognized words and phrases are processed by the parser **118**...”).) As explained below, Kennewick’s parser corresponds to the claimed “**knowledge-enhanced speech recognition engine**”; and because the parser runs on a computer, the step of receiving the transcription “**at a computer comprising the knowledge-enhanced speech recognition engine**” is disclosed by the parser receiving the transcription. (Schmandt, ¶52.)

Kennewick’s parser corresponds to the claimed “**knowledge-enhanced speech recognition engine**.” The written description of the ’160 patent describes a “knowledge-enhanced speech recognition engine” in terms of the functionality it provides, and largely tracks the language of claim 12. (’160, 13:62-14:6; Schmandt,

¶53.) The '160 patent states that the knowledge-enhanced speech recognition engine “may be used to determine the intent of the request and/or to correct false recognitions,” that it “may access a set of expected contexts that are stored in a context stack to determine a most likely context” and that it “may use context specific matchers that are able to identify context such as time, location, numbers, dates, categories (e.g., music, movies, television, addresses, etc.) and other context.” ('160, 13:64-14:2.) These descriptions are consistent with the term itself, which suggests software that uses additional/contextual information in connection with performing speech recognition/understanding—for example, to recognize the intent of a user’s request (and thus determine that the speech was likely recognized correctly) or to determine that the recognized speech may be incorrect. (Schmandt, ¶53.)

Petitioner notes that the written description of the '160 patent uses permissive language (e.g., “may”) when describing the knowledge-enhanced speech recognition engine’s functionality and does not require it to be able to perform all, or even any, of the specific functions described. (Schmandt, ¶53.) In any event, as explained below, Kennewick’s parser provides the same functionality. (Schmandt, ¶53.) In fact, during prosecution of the '160 patent, the application identified descriptions of the parser (which also appear in Kennewick) as providing written description support for the “knowledge-enhanced speech recognition engine.” (Schmandt, ¶53 (*comparing* EX1012, p.00281 (“For example, the specification even specifically

indicates that ‘the possible contexts are scored and the top one or few are used for further processing’ (pg. 43, paragraph 92). As such, the specification as originally filed does provide support for the features relating to the ‘knowledge-enhanced speech recognition engine’....”) *with* Kennewick, ¶0152.)

To start, Kennewick’s parser is used to determine the intent of a request in the transcribed natural language utterance. (Schmandt, ¶54.) It does this by examining the words of the transcribed utterance, as well as by using user profile information and dialog history, to determine a “context.” (Kennewick, ¶0152 (“The parser **118** determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history.”); Schmandt, ¶54.) “For example,” Kennewick explains, “a question with the keywords ‘temperature’ implies a context value of weather for the question.” (Kennewick, ¶0152.) In other words, in this example, the parser determines that the intent of a user’s question with the keywords “temperature” is to ask about the weather. (Schmandt, ¶54.) By determining the user’s intent, the parser can identify the appropriate application and query to execute in order to try to provide the user with the requested information. (Kennewick, ¶¶0152 (“The context of a question or command determines the domain and thereby, the domain agent **156**, if any, to be evoked.”), 0092 (“Based on keywords in the questions and commands and the structures of the questions and commands, the parser invokes the required

agent[s].”); Schmandt, ¶54.) And by being able to identify the context, the parser also provides an indication that the speech was recognized correctly. (Schmandt, ¶54.)

Kennewick’s parser is also used to correct false recognitions. As explained in more detail for 12[c]-[d], Kennewick discloses that the parser “uses a scoring system to determine the mostly likely context or domain for a user’s question and/or command.” (Kennewick, ¶0152.) But “[i]f the confidence level of the score is not high enough to ensure a reliable response, the system 90 may ask the user to verify whether the question and/or command is correctly understood.” (Kennewick, ¶0153.) Kennewick further explains:

If the confidence level of the domain or context score is not high enough to ensure a reliable response, the system can request that the user verify the question or command is correctly understood. In general, the question may be phrased to indicate the context of the question including all criteria or parameters. If the user confirms that the question is correct the system may proceed to produce a response. Otherwise, either the user can rephrase the original question, perhaps adding additional information to remove ambiguity, or the system may ask one or more questions to attempt to resolve the ambiguity or other actions may taken.

(Kennewick, ¶0031.) By causing the system to prompt the user to verify whether the input was correctly understood, the parser provides the ability to correct false

recognitions. (Schmandt, ¶55.)

Next, as explained in detail for 12[b]-[c], a person of ordinary skill in the art would have understood and found it obvious that Kennewick's parser would access a set of expected contexts that are stored in a context stack to determine a most likely context. (Schmandt, ¶56.) Kennewick discloses the "set of expected contexts" in a few different ways. First, Kennewick states that, as part of determining the most likely context, multiple "possible contexts are scored and the top one or few are used for further processing." (Kennewick, ¶0152.) Therefore, because these "possible contexts" are recognized as possibly matching the user's intent, this collection of "possible contexts" is the first way that Kennewick discloses the "set of expected contexts." (Schmandt, ¶56.) Kennewick further discloses that the possible contexts are stored in a context stack, describing an example of how "the contents of the context stack," such as keywords and criteria corresponding to past contexts, may be used in determining the context of a user's next request (e.g., "a following-on question"):

[A voice query language] provides a grammar to clearly specify the keyword used to determine the context and a set of one or criteria or parameters....The voice query language may be sensitive to the contents of the context stack. Thus, a following-on question can be asked using an abbreviated grammar, since key words and criteria can be inherited from the stack. For example, the user can simply asked

[sic] about another keyword if the criteria of the question remain constant.

(Kennewick, ¶¶0144-0145; *see also id.*, ¶0166 (“Part of the dialog history is maintained in a context stack.”).) In the excerpt above, the context keyword(s) stored in the context stack and associated with an earlier question is an example of an “expected context.” (Schmandt, ¶56.) The set of contexts associated with earlier question(s) is another way that Kennewick discloses and renders obvious the “set of expected contexts.” (Schmandt, ¶56.) To the extent there is any question that Kennewick alone discloses and renders obvious accessing expected contexts stored in a context stack in connection with determining a most likely context, it would have been obvious over Ross, as explained for 12[b].

Lastly, Kennewick’s parser, like the ’160 patent’s “knowledge-enhanced speech recognition engine,” also uses “context specific matchers that are able to identify context such as time, location, numbers, dates, categories (e.g., music, movies, television, addresses, etc.) and other context.” (’160, 13:64-14:2; Schmandt, ¶57.) For example, Kennewick describes “specific parameters or values (criteria) used to determine context for questions and commands.” (Kennewick, ¶0093.) These criteria, like the “context specific matchers” of the ’160 patent, include time, location, dates and other information. (Kennewick, ¶0154 (“For a weather context, examples of criteria include, location, date and time. Other criteria

can include command criteria (i.e., yes/no, on/off, pause, stop), and spelling.”); *see also id.*, ¶0189 (describing parameters and criteria).)

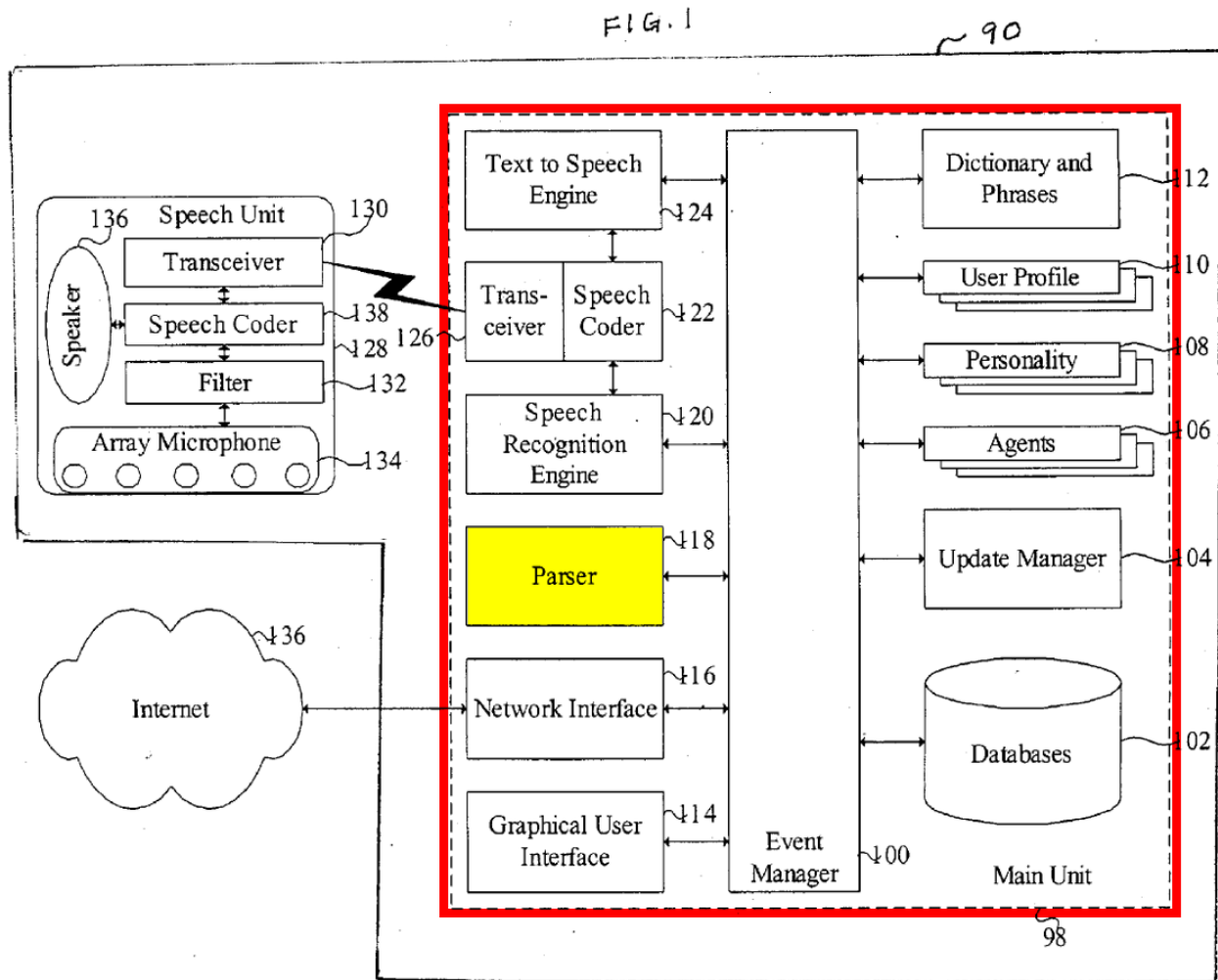
Therefore, because Kennewick’s parser includes the specific functionality ascribed to a **“knowledge-enhanced speech recognition engine”** in the ’160 patent’s written description—and more generally is software that uses additional/contextual information in connection with performing speech recognition/understanding—Kennewick’s parser discloses and renders obvious the claimed **“knowledge-enhanced speech recognition engine.”** (Schmandt, ¶58.)

Kennewick also discloses a **“computer comprising the knowledge-enhanced speech recognition engine,”** describing how “[o]ne aspect of the invention comprises software that is installed onto the computer, where the software comprises one or more of the following modules:...a parser for parsing the input”:

[T]he computer device or system may comprise a stand alone or networked PC, a PDA or other portable computer device, or other computer device or system. For convenience, these and other computer alternatives shall be simple referred to as a computer. One aspect of the invention comprises software that is installed onto the computer, where the software comprises one or more of the following modules: a speech recognition module for capturing the user input; a parser for parsing the input, a text to speech engine module for converting text to speech; a network interface for enabling the computer to interface with one or more networks; a graphical user interface module and an event manager

for managing events.

(Kennewick, ¶0012; Schmandt, ¶59.) Figure 1 of Kennewick illustrates a “main unit 98” (“computer”) that comprises “parser 118” (“the knowledge-enhanced speech recognition engine”):



(Kennewick, Fig. 1 (annotations added).) Therefore, Kennewick’s disclosure of the parser receiving a transcription from the speech recognition engine discloses “receiving a transcription of a natural language utterance at a computer comprising the knowledge-enhanced speech recognition engine.” (Schmandt,

¶59.)

- (b) “identifying one or more contexts that completely or partially match one or more text combinations contained in the transcription, wherein identifying the matching contexts includes comparing the text combinations against the [sic] grammar expression entries in the [sic] context description grammar and against one or more expected contexts stored in a context stack;” (12[b])

As explained below, 12[b] would have been obvious over Kennewick in view of Ross. (Schmandt, ¶¶60-82.)

As discussed above, Kennewick discloses “**identifying one or more contexts that...match...text in the transcription,**” describing matching tokens (e.g., words) in a recognized utterance to contexts—such as the text “temperature” to the context “weather”:

Once the words and phrases have been recognized by the speech recognition engine 120, the tokens and user identification is passed to the parser 118. The parser 118 examines the tokens for the questions or commands, context and criteria. The parser 118 determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile 110, and dialog history. The context of a question or command determines the domain and thereby, the domain agent 156, if any, to be evoked. For example, a question with the keywords “temperature” implies a context value of weather for the question. The parser dynamically receives keyword and associated prior probability or fuzzy possibility updates from the system agent 150

or an already active domain agent **156**. Based on these probabilities or possibilities the possible contexts are scored and the top one or few are used for further processing.

(Kennewick, ¶0152.) Petitioner notes that, although in this example, the keyword “temperature” does not verbatim match the word “weather” for a weather context, nothing in claim 12 requires this sort of narrow, verbatim matching. Claim 12 merely recites “one or more contexts that completely or partially match one or more text combinations,” which could be satisfied by recognizing that a text combination is associated with a context, and therefore matches the context. In fact, both Kennewick, as shown in the excerpt above, and the ’160 patent state how “[t]he parser **118** determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history.” (Kennewick, ¶0152 & ’160, 20:65-67.) A person of ordinary skill in the art would have therefore understood that the text “temperature” “**matches**” the “weather” context. (Schmandt, ¶61.) But even if claim 12 required a verbatim match, a person of ordinary skill in the art would have understood and found it obvious, based on Kennewick’s teachings, to identify a context that matches verbatim a text combination in the transcription. (Schmandt, ¶61.) For example, it would have been obvious that a keyword “weather” would identify a “weather” context. It would not make sense, for example, for a user’s question, “what’s the weather?” to not match

a weather context for processing the question. (Schmandt, ¶61.) Moreover, as discussed below in connection with the proposed combination based on Ross, Ross discloses verbatim matching of a user's utterance to a context. The combination of Kennewick and Ross therefore also discloses and renders obvious verbatim matching. (Schmandt, ¶61.)

With respect to “**text combinations**,” in addition to matching a single word, such as keyword “temperature” mentioned in the excerpt above, Kennewick also makes clear that “**text combinations**” are matched, in at least two distinct ways. First, Kennewick discloses examples of not only single-word keywords, but also multi-word keywords, such as “get the value” and “my stock portfolio” in the user utterance, “please get the value of my stock portfolio.” (Kennewick, ¶0186 (“For instance, suppose a user is interested in retrieving the value of her stock portfolio. The user may utter ‘please get the value of my stock portfolio.’ The system 90 may review this request together with stored data such as the user’s profile and determine keywords such as ‘get the value’ and ‘my stock portfolio.’”); *see also id.*, ¶¶0187 (“The user may utter ‘please record my favorite TV program.’ The system 90 may review this request together with stored data such as the user’s profile and determine keywords such as ‘record’ and ‘my favorite TV program.’”), 0098 (“new keywords for a domain, which can include names of politicians, athletes, entertainers, names of new movies or songs, etc. who have achieved recent prominence;”); Schmandt,

¶62.) A person of ordinary skill in the art would have understood and found it obvious to identify context(s) by matching multi-word keywords (i.e., “**text combinations**”) in a recognized utterance—for example, to identify a stock portfolio context (and ultimately invoke a stock portfolio agent) by matching the text combinations “get the value” and/or “my stock portfolio.” (Kennewick, e.g., ¶0186 (“a domain [agent] which accesses stock pricing sources to determine the answers to these questions”); Schmandt, ¶62.)

Second, Kennewick makes clear that the evaluation of a context can involve matching not only keyword(s) in the transcription, but criteria or parameters in the transcription as well. (Schmandt, ¶63.) Kennewick discloses that “[i]n general, the question may be phrased to indicate the context of the question including all criteria or parameters.” (Kennewick, ¶0031.) The criteria or parameters can be “used to determine the context for questions and commands.” (Kennewick, ¶0093 (“The system agent **150** manages the criteria handlers **152**, which handle specific parameters or values (criteria) used to determine context for questions and commands.”).)

12[b] continues by reciting that the claimed “**identifying**” includes: (1) “**comparing the text combinations against the grammar expression entries in the context description grammar**”; and (2) “**comparing the text**

combinations...against one or more expected contexts stored in a context stack.”

Turning to the first recited action, to start, Petitioner notes that 12[b] recites “**the grammar expression entries in the context description grammar**” without any antecedent basis for either “the grammar expression entries” or “the context description grammar.” For purposes of this Petition, Petitioner has interpreted “the grammar expression entries in the context description grammar” to mean “**grammar expression entries in a context description grammar.**” As explained below, identifying a context by “**comparing the text combinations against grammar expression entries in a context description grammar**” is disclosed by and obvious over Kennewick, alone or in view of Ross.

Kennewick explains that the matching process described above uses a grammar. (Schmandt, ¶66.) Kennewick discloses that an agent “passes a regular grammar expression to the parser **118** for evaluation of a context or question.” (Kennewick, ¶0108.) Kennewick further describes how, in one example, a grammar is used “to clearly specify the keyword used to determine the context and a set of one or [sic] criteria or parameters.” (Kennewick, ¶0144 (“[T]he voice query language helps users clearly specify the keywords or contexts of the question or command along with the parameters or criteria. The language provides a grammar

to clearly specify the keyword used to determine the context and a set of one or criteria or parameters.”).³

Kennewick also discloses that the grammar would include multiple grammar expressions. (Schmandt, ¶67.) Kennewick explains how a “content package” for an agent “include[s] questions or commands” and that each question or command has an associated regular grammar expression:

Content packages include questions or commands. Each command or question or group of commands or questions includes contexts used for creation of one or more queries. The agent **156** passes a regular grammar expression to the parser **118** for evaluation of a context or question. An initial or default context is typically supplied for each command or question. The command or question includes a grammar for the management and evaluation of the context stack.

(Kennewick, ¶0108.)

Therefore, a collection of grammar expressions used to identify a context in Kennewick discloses “**grammar expression entries in a context description grammar**” as claimed. (Schmandt, ¶68.) The collection discloses “**a context description grammar**” because it comprises a grammar that describes context(s). (Schmandt, ¶68.) This is consistent with the descriptions of a “context description

³ Kennewick explains that the “voice query language” is intended to assist and improve processing of natural language utterances that the system receives. (Schmandt, ¶66 n.2 (citing Kennewick, ¶¶0144, 0032).)

grammar” in the written description of the ’160 patent. (Schmandt, ¶68.) For example, the ’160 patent includes a nearly verbatim passage to the one excerpted above from Kennewick, largely simply replacing “regular grammar expression” in Kennewick with “context description grammar expression,” as shown in the side-by-side table below, with underlining added to show differences:

’160, 16:46-54	Kennewick, ¶0108
Content packages include questions or commands. Each command or question or group of commands or questions includes contexts used for creation of one or more requests. The <u>domain</u> agent 156 passes a <u>context description</u> grammar expression to the parser 118 for evaluation of a context or question. An initial or default context is typically supplied for each command or question. The command or question <u>may</u> include a grammar for the management and evaluation of the context stack.	Content packages include questions or commands. Each command or question or group of commands or questions includes contexts used for creation of one or more queries. The agent 156 passes a <u>regular</u> grammar expression to the parser 118 for evaluation of a context or question. An initial or default context is typically supplied for each command or question. The command or question includes a grammar for the management and evaluation of the context stack.

As shown in the excerpt above, the “regular grammar expression” in Kennewick serves the same purpose as the “context description grammar expression” in the ’160 patent: it is passed to the parser for evaluation of a context or question. (Schmandt, ¶68.) With respect to evaluating a context with the context description grammar, the

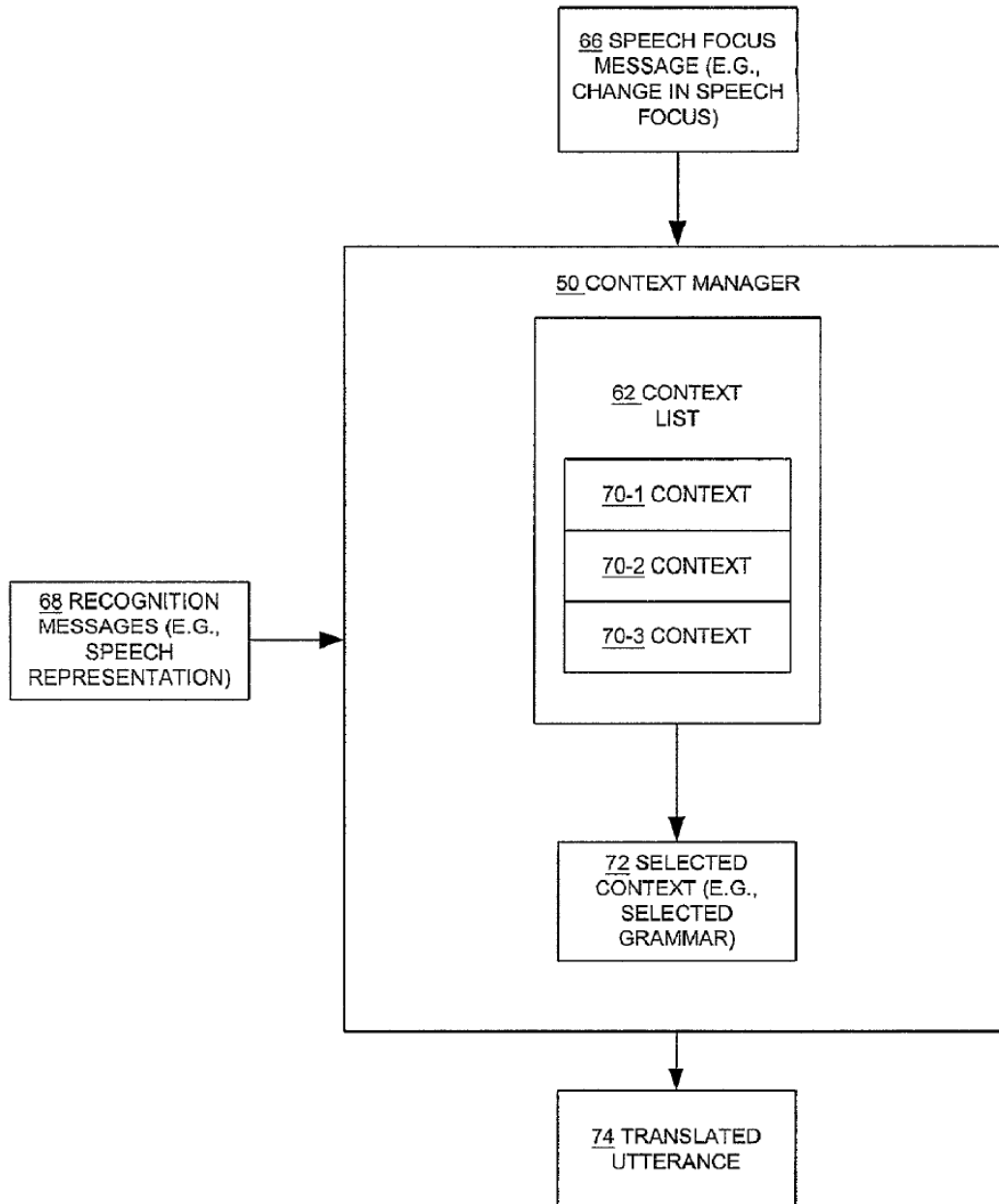
written description of the '160 patent largely tracks the language of claim 12, stating that “[t]he text combination [of a text message] may be compared against entries in a context description grammar that is associated with each agent **106**. If a match is identified between an active grammar in the context description grammar and the command and/or request, then the match may be scored.” (’160, 13:47-50; Schmandt, ¶68.) Although Kennewick does not also include a nearly-verbatim passage to this one, Kennewick does make clear, and it would have been obvious to a person of ordinary skill in the art, that using a “regular grammar” for evaluation of a context would involve comparing grammar expression entries in the regular grammar to text combinations in a transcribed utterance. (Schmandt, ¶68.) This is because, as explained above, Kennewick describes matching text in a transcription, such as keywords and criteria, to grammar expressions to identify context(s) for scoring. (Kennewick, e.g., ¶¶0144 (“[T]he voice query language helps users clearly specify the keywords or contexts of the question or command along with the parameters or criteria. The language provides a grammar to clearly specify the keyword used to determine the context and a set of one or criteria or parameters.”), 0152 (“The parser **118** determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history.”); Schmandt, ¶68.) Based on Kennewick’s descriptions, a person of ordinary skill in the art would have found it obvious, for example, that matching

“temperature” to identify a “weather” context or matching “stock portfolio” to identify a “stock portfolio” context would involve matching grammar expressions in a grammar. (Schmandt, ¶68.)

To the extent there is any question that “**grammar expression entries in a context description grammar**” as claimed would have been obvious over Kennewick alone, it would have also been obvious in view of **Ross** (**EX1005**).

Ross, U.S. Patent Application Publication No. 2002/0133354 A1, is entitled “System and Method for Determining Utterance Context in a Multi-Context Speech Application.” (Ross, face page.) Ross, as its title suggests, describes techniques for determining a context associated with a user’s spoken command or question in order to determine an application to invoke to process the command or question, just like Kennewick and the ’160 patent. (Ross, e.g., ¶¶0010, 0013.)

Ross describes a technique where determining a context involves evaluating grammars associated with potential contexts against a user’s recognized speech input—Figure 4 of Ross illustrates a general overview:



(Ross, Fig. 4.) As shown above, Ross includes a context manager **50** that includes a context list **62**. (Ross, ¶0033.) Context list **62**, in turn, “includes contexts **70** (e.g., **70-1**, **70-2**, **70-3**, etc.) for speech-enabled applications **26**, which represent grammars for the applications **26**.” (Ross, ¶0033.) A user’s transcribed utterance (a “recognition message”) is tested against the grammars in the context list in order to

determine an application to invoke:

The context manager **50** maintains the priority and state of the various grammars in the context list **62** in the system.... Recognition messages **68** from the speech engine interface **30** are tested by the context manager **50** against the active grammars in the context list **62** in priority order. When a successful match is found, the corresponding translation **74** is dispatched to the script engine **38** for execution, and the priority of the matching grammar (i.e., the selected context **72**) is raised.

(Ross, ¶0034.)

Ross further makes clear that “test[ing]...against the active grammars” and finding a successful “match” involves matching text combinations in a user’s transcribed utterance to grammar expressions in a grammar. (Schmandt, ¶72.) Ross explains that “a grammar is defined for each application **26**” and describes two example grammars, one for an electronic mail application:

```
<mail> = do I have any messages |  
        open <message> |  
        create a message |  
        send <message>|  
        print <message>.  
<message> = the? <nth> message | it | this.  
<nth> = first | second | third | fourth | fifth | . . .
```

(Ross, ¶¶0035, 0040) and one for a calendar application:

```
<appointment> = do I have any appointments |  
    open <appointment> |  
    create an appointment |  
    print <appointment>.  
<appointment> = the? <nth> appointment | it | this.
```

(Ross, ¶0046). A grammar expression can generally comprise word tokens, rule references, and grammar operators. (Schmandt, ¶72.) For example, in the grammar for the electronic mail application shown above, the line “<message> = the? <nth> message | it | this,” defines a rule named “message.” (Schmandt, ¶72.) The rule specifies a grammar expression that includes word tokens (e.g., “the,” “message”), a reference to another rule (“<nth>”) and a grammar operator (“?”). (Schmandt, ¶72.) As shown above, each grammar includes a number of grammar expressions. Therefore, as Ross explains, the grammar for the electronic mail application allows spoken phrases such as “open the first message,” “create a message,” “send this,” and “print it” to be matched, while the grammar for the calendar application matches phrases such as “open the first appointment,” “create an appointment,” “print the fourth appointment,” and “print it.” (Ross, ¶¶0041-0051; Schmandt, ¶72.) The phrases that can be matched can be easily recognized by substituting the term in angle brackets with text (or expression) to which it can correspond. For example, in the grammar for the electronic mail application, “<message>” can correspond to “the? <nth> message,” “it,” or “this.” Therefore, the grammar expression “print

<message>” can match “print it” (as well as “print this” or “print the first message” and so on). Ross discloses, therefore, that a text combination in a user’s transcribed utterance, such as “print it,” matches the “print <message>” expression defined by the “mail” rule as well as the “it” expression defined by the “message” rule in the grammar for the electronic mail application. (Schmandt, ¶72.) With respect to the calendar application grammar, “print it” would also match the “print <appointment>” and “it” expressions in the “appointment” rules. (Schmandt, ¶72.)

Using the example electronic mail and calendar application grammars shown above, Ross further describes how identifying context(s) involves matching transcribed text to the grammar expressions in the grammars:

If both of these grammars were loaded into the context manager **50**, (ignoring foreground/background issues in this example, assuming that the grammars are the same for both foreground and background), the speech center system **20** is listening for any of the phrases accepted by either grammar. If the speech center **20** hears a phrase such as “print the first message” or “print the first appointment,” the context manager **50** can readily figure out the intended target application **26** for the uttered sentence. Only one grammar will accept the phrase, which thus indicates the selected context **72** for that phrase and that associated application **26** is the one that should be targeted to receive the corresponding command.

If the sentence is “print it” however, both grammars are capable of accepting the utterance. The context manager **50** therefore has to make

a choice by referring to the context list **62** of applications in order of recency of access. The context manager **50** tests the utterance against these grammars (indicated by the contexts **70** in the context list **62**) in priority order, and passes the commands on to the first application **26** that has a grammar that will accept the phrase.

(Ross, ¶¶0052-0053; Schmandt, ¶73.)

Rationale and Motivation to Combine (Kennewick and Ross). It would have been obvious to a person of ordinary skill in the art to combine Kennewick with Ross's teachings regarding comparing transcribed user utterances to grammar expressions in a grammar to identify a context. (Schmandt, ¶74.) The combination would have predictably resulted in Kennewick's parser identifying context(s) matching text combination(s) contained in a transcribed user utterance by **“comparing the text combinations against the [sic] grammar expression entries in the [sic; a] context description grammar”** as claimed. (Schmandt, ¶74.)

Both Kennewick and Ross are analogous references in the same field as the '160 patent and would have been pertinent to a number of problems facing the '160 patent inventors, including techniques for using a grammar to identify a context for processing a user command or question and techniques for choosing an application to invoke to process a command or question in a user's transcribed utterance. ('160, 1:8-14; Kennewick, ¶0003; Ross, e.g., ¶¶0010, 0013; Schmandt, ¶75.)

A person of ordinary skill in the art would have been motivated to combine Kennewick and Ross in the manner described. (Schmandt, ¶76.) For starters, as discussed above, a person of ordinary skill in the art would have already understood and found it obvious based on Kennewick’s disclosures and background knowledge that Kennewick’s use of grammars would involve comparing text combinations against grammar expression entries in a context description grammar as claimed. (Schmandt, ¶76.) Ross confirms this understanding by detailing a remarkably similar technique. (Schmandt, ¶76.) Moreover, because of the similarity of Ross’s technique, a person of ordinary skill in the art, seeking to further understand or implement Kennewick’s disclosures, would have appreciated that Ross picks up where Kennewick leaves off and would have turned to Ross for its more detailed teachings. (Schmandt, ¶76.)

A person of ordinary skill in the art would have had a reasonable expectation of success, as combining Ross and Kennewick would have involved the straightforward application of conventional techniques and technologies without undue experimentation. (Schmandt, ¶77 (citing Ross, ¶0060 (“In a preferred embodiment, the grammar used for each speech-enabled application 26 may be a Backus Naur Form (BNF) grammar as shown above.”); EX1009, p.008 (dictionary entry for “Backus Naur Form”))).)

12[b] further recites that identifying the one or more matching contexts also **“includes comparing the text combinations...against one or more expected contexts stored in a context stack.”** As discussed for 12[a], a person of ordinary skill in the art would have understood and found it obvious that Kennewick’s parser would access one or more expected contexts that are stored in a context stack to determine a most likely context. Kennewick discloses the **“one or more expected contexts”** in a few different ways. First, Kennewick states that, as part of determining the most likely context, multiple “possible contexts are scored and the top one or few are used for further processing.” (Kennewick, ¶0152.) Therefore, because these “possible contexts” are recognized as possibly matching the user’s intent, the “possible contexts” are a first way that Kennewick discloses the **“one or more expected contexts.”** (Schmandt, ¶78.) Kennewick further discloses that possible contexts matching a transcription are stored in a context stack, describing an example of how “the contents of the context stack,” such as keywords and criteria corresponding to past contexts, may be used in determining the context of a user’s next request (e.g., “a following-on question”):

[A voice query language] provides a grammar to clearly specify the keyword used to determine the context and a set of one or criteria or parameters....The voice query language may be sensitive to the contents of the context stack. Thus, a following-on question can be asked using an abbreviated grammar, since key words and criteria can

be inherited from the stack. For example, the user can simply asked [sic] about another keyword if the criteria of the question remain constant.

(Kennewick, ¶¶0144-0145; *see also id.*, ¶0166 (“Part of the dialog history is maintained in a context stack.”).) In the excerpt above, the context keyword(s) stored in the context stack and associated with an earlier question is an example of an “**expected context.**” The one or more contexts associated with earlier question(s) is another way that Kennewick discloses and renders obvious the “**one or more expected contexts.**” (Schmandt, ¶78.) As Kennewick makes clear, the transcribed text of a user’s utterance (e.g., text of “a following-on question”) (“**text combinations**”) can be compared to context keywords in the context stack (“**against one or more expected contexts stored in a context stack**”). (Schmandt, ¶78.)

To the extent there is any question that Kennewick alone discloses and renders obvious identifying the one or more matching contexts by “**comparing the text combinations...against one or more expected contexts stored in a context stack,**” it would have been obvious in view of Ross, as noted for 12[a].

As discussed above, Ross describes a context manager **50** that includes a context list **62**. (Ross, e.g., Fig. 4, ¶0033.) The context list **62**, in turn, includes contexts **70**. The contexts in the context list **62** are sorted in priority order, such that the most-recently-accessed context will be the first context evaluated when

processing the next user utterance. (Ross, ¶¶0035 (“[W]henver an application 26 which is not the top priority application 26 is chosen as the target for a speech command, the application 26 indicated by the selected context 72 will move to the head of the list 62. In this way, the last application 26 that the user touched or talked to will get the first opportunity at interpreting the next user utterance, and the other applications 26 will be ordered in a most-recently-accessed way.”), 0037 (“[T]he context manager 50 receives the recognition message 68 and selects a context 70 for a speech enabled application 26 by checking the speech representation against contexts 70, starting at top of context list 62 to determine the highest priority context 70 that can process the speech representation”).) Ross’s priority ordering description of a context list is consistent with the well-known concept of a “stack,” in which the most recent item added to the stack will be the first item considered from the stack in a later operation. (Schmandt, ¶80 (citing EX1009, p.0022 (definition of “last in, first out”))).) This “last in, first out” (“LIFO”) behavior of a stack contrasts with, for example, the “first in, first out” (“FIFO”) behavior of a queue. (Schmandt, ¶80 (citing EX1009, pp.0017 & 0022 (definitions of “first in, first out” and “last in, first out”))).)

Rationale and Motivation to Combine (Kennewick and Ross). The rationale and motivation to combine Kennewick and Ross discussed above applies here as well. (Schmandt, ¶81.) Moreover, with respect to this portion of 12[b], it

would have been obvious to combine Kennewick with Ross’s disclosures regarding storing contexts associated with accessed applications on a context stack. (Schmandt, ¶81.) The combination would have predictably resulted in “**one or more expected contexts**”—such as the contexts associated with applications that had already been invoked—“**stored in a context stack.**” (Schmandt, ¶81.)

A person of ordinary skill in the art would have been motivated to combine. (Schmandt, ¶82.) The combination would have involved the application of known techniques to improve a similar system in the same way. (Schmandt, ¶82.) As discussed above, Kennewick already describes using a context stack storing context keywords in connection with determining a context for a user’s question. Kennewick further suggests that the context keywords in the context stack correspond to previously identified contexts, by describing how “key words and criteria can be inherited from the stack” when evaluating “a following-on question.” (Kennewick, ¶¶0144-0145; Schmandt, ¶82.) Ross’s a similar system confirms storage of “**expected contexts**” in a “**context stack,**” and that doing so enables comparing text against the expected contexts to efficiently identify a matching context. (Ross, ¶0035 (“The speech focus priority is established by maintaining an ordered list of applications (e.g., context list **62**). Whenever an application **26** gains window focus, it will move to the head of the list **62**. Likewise, whenever an application **26** which is not the top priority application **26** is chosen as the target for

a speech command, the application **26** indicated by the selected context **72** will move to the head of the list **62**. In this way, the last application **26** that the user touched or talked to will get the first opportunity at interpreting the next user utterance, and the other applications **26** will be ordered in a most-recently-accessed way.”); Schmandt, ¶82.)

**(c) “scoring each of the identified matching contexts;”
(12[c])**

Kennewick discloses “**scoring each of the identified matching contexts.**” As discussed for 12[b], Kennewick describes identifying possible contexts that match a user’s utterance “by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history.” (Kennewick, ¶0152.) “Based on these probabilities or possibilities the possible contexts are scored and the top one or few are used for further processing.” (Kennewick, ¶0152; *see also id.*, ¶¶0153 (“The parser **118** uses a scoring system to determine the mostly likely context or domain for a user’s question and/or command.”), 0030 (“The system may determine the mostly likely context or domain for a user’s question or command, for example, by using a real-time scoring system or other technique.”).)

Petitioner further notes that, during prosecution of the ’160 patent, the applicant identified a substantially verbatim passage in the ’160 application as supporting the written description disclosure of this limitation. (EX1012, pp.00275 (pending claim 45 reciting “scoring each of the identified matching contexts”),

00280-00281 (applicant response purporting to identify written description support);

Schmandt, ¶84.) The applicant stated:

The specification further describes how when a matching context is identified, “then the match may be scored,” wherein domain agents associated with various respective contexts may be selected or otherwise “ranked based on the determined score.” The scoring process used at various stages of a dialog is described throughout the specification (e.g., pgs. 10-11, paragraphs 23-24, pg. 42, paragraph 91, pg. 46, paragraph 101, etc.). For example, the specification even specifically indicates that “the possible contexts are scored and the top one or few are used for further processing” (pg. 43, paragraph 92). As such, the specification as originally filed does provide support for the features relating to the “knowledge-enhanced speech recognition engine,” including identifying matching contexts, scoring the matching contexts, and selecting a context with the highest score as the “most likely” context.

(EX1012, p.00281.)

(d) “selecting the matching context having a highest score to determine a most likely context for the utterance; and” (12[d])

As discussed for 12[c], Kennewick discloses that “the possible contexts are scored and the top one or few are used for further processing.” (Kennewick, ¶0152.)

Kennewick further describes how “[t]he parser **118** uses a scoring system to determine the mostly likely context or domain for a user’s question and/or

command” and that “[b]ased on this scoring, the system **90** invokes the correct agent.” (Kennewick, ¶0153.) A person of ordinary skill in the art would have therefore understood and found it obvious that the matching context “**having a highest score**” would determine the “**most likely context**”—first, because Kennewick discloses selecting only the context with the “top” score, and second, because it would be quite sensible to select the context having a highest score as a mostly likely context even where “the top...few are used for further processing.” (Schmandt, ¶85.)

Moreover, as with 12[c], during prosecution of the ’160 patent, the applicant identified a passage in the ’160 application that is substantially verbatim paragraph 0152 in Kennewick as supporting the written description disclosure of this limitation. (EX1012, pp.00275 (pending claim 45 reciting “selecting the matching context having a highest score to determine a most likely context for the utterance”), 00280-00281 (applicant response purporting to identify written description support, include paragraph 92 reciting “the possible contexts are scored and the top one or few are used for further processing”); Schmandt, ¶86.)

- (e) **“communicating a request to a domain agent configured to process requests in the most likely context for the utterance, the request formulated using at least one grammar expression entry in the context description grammar.”** (12[e])

Kennewick discloses **“communicating a request to a domain agent configured to process requests in the most likely context for the utterance.”**

Kennewick explains, for example, that “[b]oth system agents **150** and domain agents **156** receive questions and commands from the parser **118**” and that “[a]fter selecting the agents **106**, a properly formatted query and/or command that will be submitted to the agents **106** may be determined at **608**.” (Kennewick, ¶¶0092, 0189; *see also id.*, ¶¶0091 (describing domain agents), 0185 (“Once, the properly formatted queries and/or commands are generated, the appropriate action may be taken at **308**. This may require that the properly formatted queries and/or commands be sent to specific domain agent[s]...that can fulfill the requirements of the query and/or command.”).)

Kennewick therefore teaches **“communicating a request”** (submitting a “properly formatted query and/or command” from the parser) **“to a domain agent configured to process requests in the most likely context for the utterance”** (selected domain agent(s) **156**). (Schmandt, ¶87.)

Kennewick further discloses and renders obvious that the request communicated to a domain agent would be **“formulated using at least one grammar expression entry in the context description grammar.”** As discussed

for 12[b], Kennewick discloses and renders obvious a context description grammar that includes one or more grammar expression entries. As noted, Kennewick refers to a “regular grammar” that serves the same purpose of the “context description grammar” in the ’160 patent. (*E.g.*, Kennewick, ¶0108; ’160, 16:46-54.) As for 12[e], Kennewick refers back to the “regular grammar,” similarly disclosing “formulat[ing] a question or command in the regular grammar used by agents.” (Kennewick, ¶0154 (“To formulate a question or command in the regular grammar used by agents, the parser **118** may determine required and optional values for the criteria or parameters.”); *see also id.*, ¶0189 (“In order to formulate a question or command in the regular grammar used by agents **106**, a determination may be made as to the required and optional values for the criteria or parameters of the queries and/or commands.”).) Kennewick, alone or in view of Ross, therefore discloses formulating a request to a domain agent “**using at least one grammar expression entry in the context description grammar.**” (Schmandt, ¶88.) Moreover, because grammar expression entries are used when matching a transcription to determine a context and application to invoke, as taught by both Kennewick and Ross, a person of ordinary skill in the art would have understood that the resulting request communicated to the domain agent would have been formulated “**using at least one grammar expression entry in the context description grammar.**” (Schmandt, ¶88.)

Petition for *Inter Partes* Review of
U.S. Patent No. 7,640,160 B2

VII. CONCLUSION

Petitioner respectfully requests IPR institution.

Dated: April 5, 2024

Respectfully submitted,

COOLEY LLP
ATTN: Patent Group
1299 Pennsylvania Avenue NW
Suite 700
Washington, DC 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

By: / Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673
Counsel for Petitioner

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), I certify that this petition complies with the type-volume limits of 37 C.F.R. § 42.24(a)(1)(i) because it contains 8,322 words, according to the word-processing system used to prepare this petition, excluding the parts of this petition that are exempted by 37 C.F.R. § 42.24(a) (including the table of contents, a table of authorities, mandatory notices, a certificate of service or this certificate word count, appendix of exhibits, and claim listings).

DATED: April 5, 2024

COOLEY LLP
ATTN: Patent Docketing
1299 Pennsylvania Avenue NW
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001
Fax: (650) 849-7400

/ Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673

CERTIFICATE OF SERVICE

I hereby certify, pursuant to 37 C.F.R. Sections 42.6 and 42.105, that a complete copy of the attached **PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 7,640,160 B2**, including all exhibits (**Nos. 1001-1002, 1004-1005, 1007, 1009-1010, 1012-1013**) and related documents, are being served via Federal Express on the 5th day of April, 2024, the same day as the filing of the above-identified document in the United States Patent and Trademark Office/Patent Trial and Appeal Board, upon Patent Owner by serving the correspondence address of record with the USPTO as follows:

174899 - The Law Offices of David A. Gerasimow, P.C.
211 W. Wacker Dr.
Ste. 1717
Chicago, IL 60606
UNITED STATES

And, via Federal Express upon counsel of record for Patent Owner in the litigation pending before the U.S. District Court for the District of Delaware entitled *Dialect, LLC v. Google LLC*, Case No. 1:23-cv-00378 (D. Del.) as follows:

Garland Stephens
BLUE PEAK LAW GROUP LLP
3139 W Holcombe Blvd, PMB 8160
Houston, TX 77025
Telephone: 281-972-3036
garland@bluepeak.law

DATED: April 5, 2024

/ Heidi L. Keefe /
Heidi L. Keefe
Reg. No. 40,673
COOLEY LLP
1299 Pennsylvania Ave. NW
Suite 700
Washington, D.C. 20004
Tel: (650) 843-5001