

Exhibit A-Coffman
Invalidity Chart for U.S. Patent No. 7,640,160 in view of U.S. Patent No. 7,137,126 (“Coffman”)

U.S. Patent No. 7,137,126 (“Coffman”) was filed Jul. 2, 2001 and the PCT was published Apr. 13, 2000. Coffman qualifies as prior art to U.S. Patent No. 7,640,160 (the “’160 Patent”) at least under one or more applicable provisions of 35 U.S.C. §§ 102(a), (b), (e), and/or (g). Coffman discloses and/or renders obvious one or more of claims 12 and 13 of the ’160 Patent alone and/or in combination with other references, as set forth in the chart below. To the extent Coffman does not disclose one or more limitations of the claims, such limitations would have been inherent and/or it would have been obvious to combine the disclosures of Coffman with the knowledge of one of ordinary skill in the art and/or with one or more other references, including as identified for the reasons identified herein or in the Cover Pleading, to render the claims at-issue in the ’160 Patent invalid.

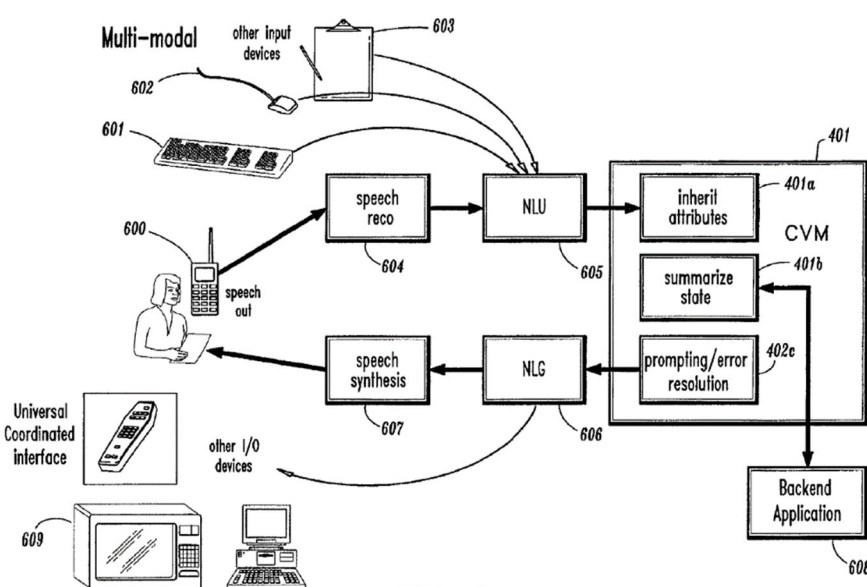
Defendant Bank of America, N.A. (“Bank”) incorporates in this chart all applicable qualifications, clarifications, and other statements made in Bank’s Invalidity Contentions. This invalidity claim chart is based on Bank’s present understanding of claims 12, 13, and Plaintiff Dialect, LLC (“Dialect”) apparent construction of the claims, as set forth in Dialect’s Infringement Contentions. These invalidity assertions are not an admission that Bank concedes or acquiesces to any claim construction(s) implied or suggested by Dialect the associated infringement claim charts. Nor is Bank asserting any claim construction positions through these charts, including whether the preamble is a limitation. Bank also does not concede or acquiesce that any asserted claim satisfies the requirements of 35 U.S.C. §§ 112 or 101, and submits these invalidity contentions only to the extent Dialect’s assertions may be understood.

The excerpts cited herein are exemplary. For any claim limitation, Bank may rely on excerpts cited for any other limitation and/or additional excerpts not set forth fully herein to the extent necessary to provide a more comprehensive explanation for a reference’s disclosure of a limitation. Where an excerpt refers to or discusses a figure or figure items, that figure and any additional descriptions of that figure should be understood to be incorporated by reference as if set forth fully herein. To the extent any cell lacks citations to the charted reference(s), this should not be taken as an admission that the reference(s) do(es) not disclose the corresponding limitation but rather indicates that Bank does not presently have sufficient information regarding the charted reference(s) based on Bank’s present understanding of the claim limitation.

These invalidity contentions are not an admission by Bank that the accused products or components, including any current or past version of the products or components, are covered by, or infringe the asserted claims, particularly when these claims are properly construed and applied.

1. Claim 12

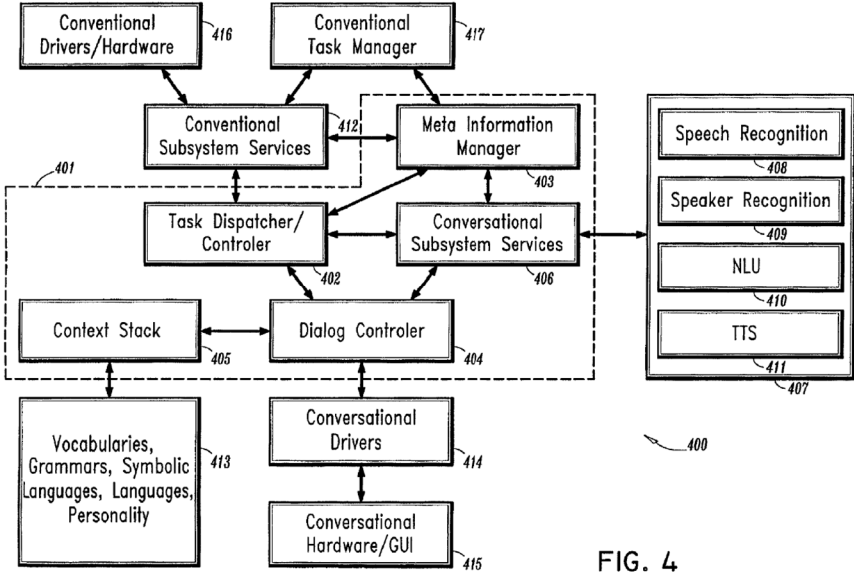
Claim 12 of the '160 Patent	Exemplary Citations to Coffman
[p] A method for interpreting natural language utterances using knowledge-enhanced speech recognition engine, wherein the knowledge-enhanced speech recognition engine is configured to determine an intent and correct false recognitions of the natural language utterances, comprising:	To the extent the preamble is limiting, Coffman discloses 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[.]” As one example, the system of Coffman that performs one or more of the method steps of claim 12 can be the knowledge-enhanced speech recognition engine that is configured to determine an intent and correct false recognitions of the natural language utterances. <i>E.g.</i>, Coffman at Fig. 1: FIG. 1

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	Coffman at Fig. 6:  FIG. 6 Coffman at 3:57-4:5: In one aspect of the present invention, the CVM utilizes conversational subsystems (which may be local or distributed) including speech recognition, speaker recognition, text-to-speech, natural language understanding and natural dialog generation engines to understand and generate dialog between and user and machine. These subsystem are accessed through the CVM. The engines are hidden to the application through the conversational application APIs. The CVM may control such engines through the conversational engine APIs. In addition, the conversational application APIs may include the conversational engine APIs. Typically, CVM includes direct exposure of these engine APIs to the application developer. This may be done by having the conversational engine

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	APIs included in the conversation application APIs or by emulating similar calls and functionalities at the level of the conversational application APIs. Coffman at 6:47-55: In another aspect, conversational selection capabilities are provided at the resource manager level or within any application relying on meta-information, abstraction and conversational queries/mixed initiative/correction. Such conversational selection capabilities avoid long sequences of elementary selections and provide natural shortcuts and correction of the selection. In addition, mechanisms are provided to access and present immediately the skeleton of objects with hierarchical structures. Coffman at 21:45-64: The conversational system 400 further comprises a plurality of conversational resource subsystems (engines) 407 including, for example, a speech recognition system 408, a speaker recognition system 409, a natural language understanding and natural language parsing system 410 and a text-to-speech synthesis (TTS) system 411. It is to be understood that the conversational resources 407 may also include other systems such as a NLG (natural language generation) engine and an audio subsystem. As explained above, each of these conversational subsystems 407 may be accessed through API calls to the CVM 401. The CVM 401 will locate the requested conversational subsystem 407 (via the conversational subsystem services manager 406), drive its execution and return appropriately the results. It is to be appreciated that these conversational subsystem 407 can be local or distributed over a network and that all conversational subsystem calls are hidden to the application (although the engine APIs are always available to the application if the developer wants a to implement a specific behavior of the engines 407). Coffman at 28:22-29: The key conversational aspect is in the input procedure wherein commands can be entered in NLU (to provide natural language understanding of input queries) or in FSG mode (for constrained

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	input according to rules: grammar and vocabulary, as opposed to free natural input). Commands or queries can be completed or corrected by completing missing fields or by correcting incorrect fields for the active task. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[a] receiving a transcription of a natural language utterance at a computer comprising the knowledge-enhanced speech recognition engine;	Coffman discloses “receiving a transcription of a natural language utterance at a computer comprising the knowledge-enhanced speech recognition engine[.]” <i>E.g.</i>, Coffman at 4:47-64: In another aspect, the CVM provides the capability to have natural dialog with NLU, NLG, contexts and mixed-initiatives sorted across multiple tasks, processes and discourses (with multiple domains). A conversational input interface is provided whereby a set of multi-mode input streams are each transcribed into an ASCII command or query (i.e., lists of attribute-value pairs or n-uples). Each input entity (command, NLU query field or argument unit (isolated letter, word, etc.) is associated with time-marks and appended accordingly to a compounded input stream. Two or more stream having the same time-marks are prioritized based on when each input stream contributed previously or the priority that each application/input stream received on the basis of the context history. Compounded inputs are checked against possible FSG and dictionaries and optionally fed back to the user. Each resource exchanges their conversational capabilities and the input stream is tailored to only exchange relevant information.

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	Coffman at 28:39-57: In accordance with the present invention, the input problem is treated as a merge of the output of multiple decoders, ASCII transcription or a list of attribute value n-uples. Each input stream is converted into its ASCII transcription and aligned with input time-marks via speech recognition processing 604. When different input stream are associated to the same task, the transcripts are merged as follows. First, commands and queries are sorted based on the time marks and appended to a single data stream. Command formulation can be checked against FSG rules and re-sorted to satisfy the grammar rules. NLU queries do not necessarily require re-sorting. For NLU queries, the symbolic fields are filled for each stream, then compounded at the level of the final input stream. Arguments such as spelling and alpha-numeric code do not exploit grammar rules or NLU to solve ordering ambiguity. Time-marks are used similarly to build a unique stream. However, the input is fed back to the user for confirmation, with possible pre-filtering using a dictionary or FSG rule book which is application-dependent. Coffman at 30:36-50: The task dispatcher/controller 402 divides each command/process into multiple actions, starts the associated threads/processes with the appropriate priority and relates/inserts them within the context stack 405. The task dispatcher 402 allocates each resource and shares them between the different spawned actions, and controls handles and streams to and from the resources. Based on the modality (pointer, keyboard, file, speech), the task dispatcher 402 redirects the stream to the appropriate conversational subsystems or conventional subsystem with speech inputs being transcribed/understood. The output of these subsystems is run down the context stack 405 to extract the active query and complete it. On the other hand, outputs are queued based on the priority levels of each task and dispatched sequentially to the output resource. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[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 grammar expression entries in the context description grammar and against one or more expected contexts stored in a context stack;	Coffman discloses “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 grammar expression entries in the context description grammar and against one or more expected contexts stored in a context stack[.]” <i>E.g.</i>, Coffman at Fig. 4:  FIG. 4 Coffman at 4:6-26:

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	In another aspect, a CVM kernel layer (or CVM controller) comprises a meta-information manager, a resource manager, a context stack, a global history, a dialog manager and a task dispatcher, for managing the dialog and selecting the active dialog, context, and application. The context stack accumulates the context (full query arguments and state/mode—i.e. query arguments already introduced, any I/O event, and event produced by an application) of each active process with an activated discourse along with any data needed for input understanding (e.g. active FSG, topic, vocabulary or possible queries for a speech input). The CVM kernel coordinates the different tasks and processes that are spawned on local and networked conventional and conversational resources. The CVM kernel layer keeps track of these resources, transmit input to the appropriate conversational subsystems and arbitrate between devices, state and applications. The CVM kernel layer also coordinates the output generation and prioritization according to the active conversation and conversation history, delayed returns, delegation across network resources and task delegation and memorization. Coffman at 22:5-46: The core of the CVM 401 is the context stack 405 which operates and is managed under the control of the dialog controller 404 (it is to be understood that the context stack 405 id directly related to the global history and meta information repository discussed above). In general, the context stack 405 accumulates the context (i.e., full query arguments list of attribute value n-uples, and state/mode) of each active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread) along with any data files 413 (or at least identifiers of such conversational arguments) for the different engines that may be needed for input understanding (e.g., files or arguments that the engines use for performing their respective tasks such as active FSG, topic, vocabulary, HMM (hidden markov models), voiceprints, language models or possible queries for a speech input). In other words, the term “context” refers to the state of each discourse (whether active or normative), which keeps track of the past history of the discourse, its current state, and the specific characteristics and full query arguments of the corresponding task (e.g, vocabulary file, language model, parsing, tags, voiceprint, TTS rules, grammar, NLU etc. of each active task/process) along with any data needed for input

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	understanding. It is to be appreciated that, in a distributed CVM (as explained below), the context stack (as well as other CVM components) may be directly associated with networked services (i.e., distributed over the network) (as described above with respect to the context and global history). More specifically, each new task, process, or thread creates a new stack entry and is associated with a discourse. Each application may be associated with multiple discourses (e.g. the application management discourse and the application content navigation discourses). Each context associated with a given discourse comprises the latest requests made to the corresponding process/task/thread as well as the latest output. The context of a given discourse is also associated with, e.g., any active grammars, vocabularies and symbolic language which maps the actual query. Again, the latest information is stored in the history and context stacks. Past history and context and other information is managed by the meta information manager 403 and stored as part of the meta information. Coffman at 31:23-49: Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active. Until the active command is completed, or until a new command is provided, the selected context is marked active, and pushed to the top of the context stack 405. When a message is returned to the user, the context is updated and then pushed to the top of the context stack 405 under the active context. The active context is updated to inform of the existence of a returned value. This can also be done at the level of the superseding CVM session discourse, which can be in the stack or always besides the stack and then searched right after the active context, before going down the stack. Simultaneously completed tasks result in contexts that are arranged under the active context according to CVM priorities (e.g. FIFO or FILO). Active contexts sufficiently completed to generate a task will be pushed down the stack under the next or all the returned contexts. Or it could become the active

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	discourse. This may be done automatically or when commanded by the user. This stack structure allows to maintain non-ambiguous conversational interactions with multiple tasks, threads or processes. Coffman at 42:58-43:6: With CVM running on the desktop and on the PVA and CVM aware smart phone and PDA, the application developer must only hook to the CVM API. It involves registering all its conversational capabilities and requirements: 1. Active vocabulary, finite state grammar and language models to control the application; 2 Symbolic mapping if NLU is supported or at list a context state list; 3 Associated relevant meta-information/categories in particular to allow categorization of the for the output; 4 Conversational I/O information: does it directly control the input/output or is it a silent partner, client to a conversational I/O provider; and 5 CVM capabilities/state: does it run a CVM; is it a CVM client; is it a master, slave or partner CVM. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[c] scoring each of the identified matching contexts;	Coffman discloses “scoring each of the identified matching contexts[.]”

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	<i>E.g.</i>, Coffman at 4:6-26: In another aspect, a CVM kernel layer (or CVM controller) comprises a meta-information manager, a resource manager, a context stack, a global history, a dialog manager and a task dispatcher, for managing the dialog and selecting the active dialog, context, and application. The context stack accumulates the context (full query arguments and state/mode—i.e. query arguments already introduced, any I/O event, and event produced by an application) of each active process with an activated discourse along with any data needed for input understanding (e.g. active FSG, topic, vocabulary or possible queries for a speech input). The CVM kernel coordinates the different tasks and processes that are spawned on local and networked conventional and conversational resources. The CVM kernel layer keeps track of these resources, transmit input to the appropriate conversational subsystems and arbitrate between devices, state and applications. The CVM kernel layer also coordinates the output generation and prioritization according to the active conversation and conversation history, delayed returns, delegation across network resources and task delegation and memorization. Coffman at 22:5-46: The core of the CVM 401 is the context stack 405 which operates and is managed under the control of the dialog controller 404 (it is to be understood that the context stack 405 id directly related to the global history and meta information repository discussed above). In general, the context stack 405 accumulates the context (i.e., full query arguments list of attribute value n-uples, and state/mode) of each active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread) along with any data files 413 (or at least identifiers of such conversational arguments) for the different engines that may be needed for input understanding (e.g., files or arguments that the engines use for performing their respective tasks such as active FSG, topic, vocabulary, HMM (hidden markov models), voiceprints, language models or possible queries for a speech input). In other words, the term “context” refers to the state of each discourse (whether active or normative), which keeps track of the past history of the discourse, its current state, and the specific characteristics and full query arguments of the corresponding task (e.g, vocabulary file, language model, parsing, tags, voiceprint, TTS rules,

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	grammar, NLU etc. of each active task/process) along with any data needed for input understanding. It is to be appreciated that, in a distributed CVM (as explained below), the context stack (as well as other CVM components) may be directly associated with networked services (i.e., distributed over the network) (as described above with respect to the context and global history). More specifically, each new task, process, or thread creates a new stack entry and is associated with a discourse. Each application may be associated with multiple discourses (e.g. the application management discourse and the application content navigation discourses). Each context associated with a given discourse comprises the latest requests made to the corresponding process/task/thread as well as the latest output. The context of a given discourse is also associated with, e.g., any active grammars, vocabularies and symbolic language which maps the actual query. Again, the latest information is stored in the history and context stacks. Past history and context and other information is managed by the meta information manager 403 and stored as part of the meta information. Coffman at 31:23-49: Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active. Until the active command is completed, or until a new command is provided, the selected context is marked active, and pushed to the top of the context stack 405. When a message is returned to the user, the context is updated and then pushed to the top of the context stack 405 under the active context. The active context is updated to inform of the existence of a returned value. This can also be done at the level of the superseding CVM session discourse, which can be in the stack or always besides the stack and then searched right after the active context, before going down the stack. Simultaneously completed tasks result in contexts that are arranged under the active context according to CVM priorities (e.g. FIFO or FILO). Active contexts sufficiently completed to generate a task will be

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	pushed down the stack under the next or all the returned contexts. Or it could become the active discourse. This may be done automatically or when commanded by the user. This stack structure allows to maintain non-ambiguous conversational interactions with multiple tasks, threads or processes. Coffman at 42:58-43:6: With CVM running on the desktop and on the PVA and CVM aware smart phone and PDA, the application developer must only hook to the CVM API. It involves registering all its conversational capabilities and requirements: 1. Active vocabulary, finite state grammar and language models to control the application; 2 Symbolic mapping if NLU is supported or at list a context state list; 3 Associated relevant meta-information/categories in particular to allow categorization of the for the output; 4 Conversational I/O information: does it directly control the input/output or is it a silent partner, client to a conversational I/O provider; and 5 CVM capabilities/state: does it run a CVM; is it a CVM client; is it a master, slave or partner CVM. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
[d] selecting the matching context having a highest score to determine a most likely context for the utterance; and	Coffman discloses “selecting the matching context having a highest score to determine a most likely context for the utterance[.]” <i>E.g.</i>, Coffman at 4:6-26: In another aspect, a CVM kernel layer (or CVM controller) comprises a meta-information manager, a resource manager, a context stack, a global history, a dialog manager and a task dispatcher, for managing the dialog and selecting the active dialog, context, and application. The context stack accumulates the context (full query arguments and state/mode—i.e. query arguments already introduced, any I/O event, and event produced by an application) of each active process with an activated discourse along with any data needed for input understanding (e.g. active FSG, topic, vocabulary or possible queries for a speech input). The CVM kernel coordinates the different tasks and processes that are spawned on local and networked conventional and conversational resources. The CVM kernel layer keeps track of these resources, transmit input to the appropriate conversational subsystems and arbitrate between devices, state and applications. The CVM kernel layer also coordinates the output generation and prioritization according to the active conversation and conversation history, delayed returns, delegation across network resources and task delegation and memorization. Coffman at 22:5-46: The core of the CVM 401 is the context stack 405 which operates and is managed under the control of the dialog controller 404 (it is to be understood that the context stack 405 id directly related to the global history and meta information repository discussed above). In general, the context stack 405 accumulates the context (i.e., full query arguments list of attribute value n-uples, and state/mode) of each active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread) along with any data files 413 (or at least identifiers of such conversational arguments) for the different engines that may be needed for input understanding (e.g., files or arguments that the engines use for performing their respective tasks such as active FSG, topic, vocabulary, HMM (hidden markov models), voiceprints, language models or possible queries for a speech input). In other words, the term “context” refers to the

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	state of each discourse (whether active or normative), which keeps track of the past history of the discourse, its current state, and the specific characteristics and full query arguments of the corresponding task (e.g. vocabulary file, language model, parsing, tags, voiceprint, TTS rules, grammar, NLU etc. of each active task/process) along with any data needed for input understanding. It is to be appreciated that, in a distributed CVM (as explained below), the context stack (as well as other CVM components) may be directly associated with networked services (i.e., distributed over the network) (as described above with respect to the context and global history). More specifically, each new task, process, or thread creates a new stack entry and is associated with a discourse. Each application may be associated with multiple discourses (e.g. the application management discourse and the application content navigation discourses). Each context associated with a given discourse comprises the latest requests made to the corresponding process/task/thread as well as the latest output. The context of a given discourse is also associated with, e.g., any active grammars, vocabularies and symbolic language which maps the actual query. Again, the latest information is stored in the history and context stacks. Past history and context and other information is managed by the meta information manager 403 and stored as part of the meta information. Coffman at 31:23-49: Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active. Until the active command is completed, or until a new command is provided, the selected context is marked active, and pushed to the top of the context stack 405. When a message is returned to the user, the context is updated and then pushed to the top of the context stack 405 under the active context. The active context is updated to inform of the existence of a returned value. This can also be done at the level of the superseding CVM session discourse, which can be in the stack or always besides the stack

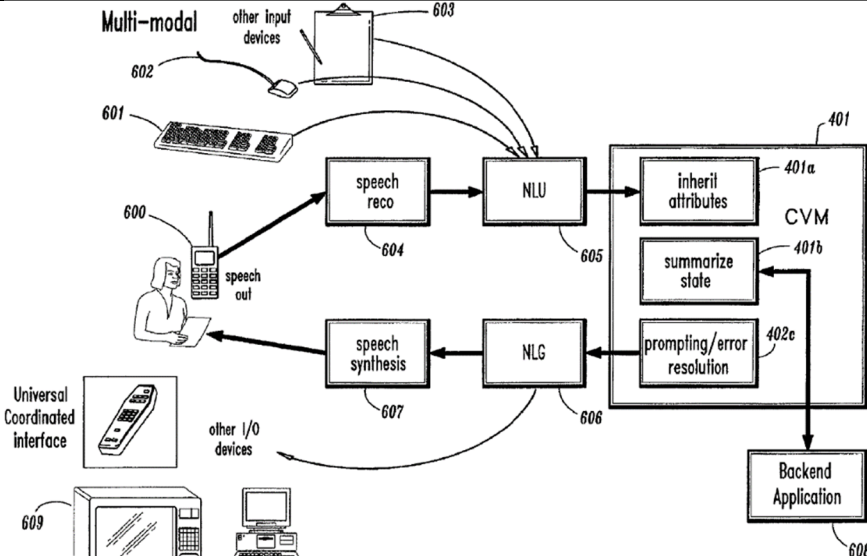
Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	and then searched right after the active context, before going down the stack. Simultaneously completed tasks result in contexts that are arranged under the active context according to CVM priorities (e.g. FIFO or FILO). Active contexts sufficiently completed to generate a task will be pushed down the stack under the next or all the returned contexts. Or it could become the active discourse. This may be done automatically or when commanded by the user. This stack structure allows to maintain non-ambiguous conversational interactions with multiple tasks, threads or processes. Coffman at 42:58-43:6: With CVM running on the desktop and on the PVA and CVM aware smart phone and PDA, the application developer must only hook to the CVM API. It involves registering all its conversational capabilities and requirements: 1. Active vocabulary, finite state grammar and language models to control the application; 2 Symbolic mapping if NLU is supported or at list a context state list; 3 Associated relevant meta-information/categories in particular to allow categorization of the for the output; 4 Conversational I/O information: does it directly control the input/output or is it a silent partner, client to a conversational I/O provider; and 5 CVM capabilities/state: does it run a CVM; is it a CVM client; is it a master, slave or partner CVM. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	endeavor to address similar known problems, according to known methods which would yield predictable results.
[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.	Coffman discloses “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[.]” <i>E.g.</i>, Coffman at 22:5-46: The core of the CVM 401 is the context stack 405 which operates and is managed under the control of the dialog controller 404 (it is to be understood that the context stack 405 id directly related to the global history and meta information repository discussed above). In general, the context stack 405 accumulates the context (i.e., full query arguments list of attribute value n-uples, and state/mode) of each active process with an activated discourse (i.e., conversational interaction associated with a given task/process/thread) along with any data files 413 (or at least identifiers of such conversational arguments) for the different engines that may be needed for input understanding (e.g., files or arguments that the engines use for performing their respective tasks such as active FSG, topic, vocabulary, HMM (hidden markov models), voiceprints, language models or possible queries for a speech input). In other words, the term “context” refers to the state of each discourse (whether active or normative), which keeps track of the past history of the discourse, its current state, and the specific characteristics and full query arguments of the corresponding task (e.g, vocabulary file, language model, parsing, tags, voiceprint, TTS rules, grammar, NLU etc. of each active task/process) along with any data needed for input understanding. It is to be appreciated that, in a distributed CVM (as explained below), the context stack (as well as other CVM components) may be directly associated with networked services (i.e., distributed over the network) (as described above with respect to the context and global history). More specifically, each new task, process, or thread creates a new stack entry and is associated with a discourse. Each application may be associated with multiple discourses (e.g. the application management discourse and the application content navigation discourses). Each context associated with a given discourse comprises the latest requests made to the corresponding process/task/thread

Claim 12 of the '160 Patent	Exemplary Citations to Coffman
	as well as the latest output. The context of a given discourse is also associated with, e.g., any active grammars, vocabularies and symbolic language which maps the actual query. Again, the latest information is stored in the history and context stacks. Past history and context and other information is managed by the meta information manager 403 and stored as part of the meta information. Coffman at 3:33-56: In another aspect, the CVM provides the capability to have natural dialog with NLU, NLG, contexts and mixed-initiatives sorted across multiple tasks, processes and discourses (with multiple domains). A conversational input interface is provided whereby a set of multi-mode input streams are each transcribed into an ASCII command or query (i.e., lists of attribute-value pairs or n-uples). Each input entity (command, NLU query field or argument unit (isolated letter, word, etc.) is associated with time-marks and appended accordingly to a compounded input stream. Two or more stream having the same time-marks are prioritized based on when each input stream contributed previously or the priority that each application/input stream received on the basis of the context history. Compounded inputs are checked against possible FSG and dictionaries and optionally fed back to the user. Each resource exchanges their conversational capabilities and the input stream is tailored to only exchange relevant information. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.

2. Claim 13

Claim 13 of the '160 Patent	Exemplary Citations to Coffman
[p] A method for processing natural language utterances, comprising:	To the extent the preamble is limiting, Coffman discloses a “method for processing natural language utterances[.]” See citations of 12[p]. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[a] receiving a natural language utterance at a computer comprising a multi-pass speech recognition module;	Coffman discloses “receiving a natural language utterance at a computer comprising a multi-pass speech recognition module[.]” <i>E.g.</i>, Coffman at Fig. 6:

Claim 13 of the '160 Patent	Exemplary Citations to Coffman
	 FIG. 6 Coffman at 4:47-64: In another aspect, the CVM provides the capability to have natural dialog with NLU, NLG, contexts and mixed-initiatives sorted across multiple tasks, processes and discourses (with multiple domains). A conversational input interface is provided whereby a set of multi-mode input streams are each transcribed into an ASCII command or query (i.e., lists of attribute-value pairs or n-uples). Each input entity (command, NLU query field or argument unit (isolated letter, word, etc.) is associated with time-marks and appended accordingly to a compounded input stream. Two or more stream having the same time-marks are prioritized based on when each input stream contributed previously or the priority that each application/input stream received on the basis of the context history. Compounded inputs are checked against possible FSG and dictionaries and optionally fed back to the user. Each resource exchanges their conversational capabilities and the input stream is tailored to only exchange relevant information.

Claim 13 of the '160 Patent	Exemplary Citations to Coffman
	Coffman at 8:64-9:15: Referring now to FIG. 1, a block diagram illustrates a conversational computing system (or CVM system) according to an embodiment of the present invention, which may be implemented on a client device or a server. In general, the CVM provides a universal coordinated multi-modal conversational user interface (CUI) 10. The “multi-modality” aspect of the CUI implies that various I/O resources such as voice, keyboard, pen, and pointing device (mouse), keypads, touch screens, etc can be used in conjunction with the CVM platform. The “universality” aspect of the CUI 10 implies that the CVM system provides the same UI to a user whether the CVM is implemented in connection with a desktop computer, a PDA with limited display capabilities, or with a phone where no display is provided. In other words, universality implies that the CVM system can appropriately handle the UI of devices with capabilities ranging from speech only to speech to multi-modal, i.e., speech+GUI, to purely GUI. Therefore, the universal CUI provides the same UI for all user interactions, regardless of the access modality. Coffman at 27:32-50: The interaction between the task dispatcher/controller 402, dialog controller 404 and context stack 405 of the CVM 401 in processing input and output data streams will now be explained in greater detail. It is to be appreciated that the present invention provides NLU interfaces with contexts and mixed-initiatives sorted across multiple tasks (with multiple domains). More specifically, the present invention provides the capability to have a natural dialog with NLU, NLG and mixed initiative across multiple applications, with multiple domains. In this regard, each application will provide the CVM 401 its own parsing and translation arguments. As explained in greater detail below, the NLU engine 410 can either tag a query sequentially (form filing) or in parallel (e.g., procedural threads or parallel conversational objects/procedures or parallel forms). The first task to have its dialog completed by producing a non-ambiguous query is executed and the corresponding query as interpreted by the other application is stored to activate if the recognized query is rejected by the user. Coffman at 30:36-50:

Claim 13 of the '160 Patent	Exemplary Citations to Coffman
	The task dispatcher/controller 402 divides each command/process into multiple actions, starts the associated threads/processes with the appropriate priority and relates/inserts them within the context stack 405. The task dispatcher 402 allocates each resource and shares them between the different spawned actions, and controls handles and streams to and from the resources. Based on the modality (pointer, keyboard, file, speech), the task dispatcher 402 redirects the stream to the appropriate conversational subsystems or conventional subsystem with speech inputs being transcribed/understood. The output of these subsystems is run down the context stack 405 to extract the active query and complete it. On the other hand, outputs are queued based on the priority levels of each task and dispatched sequentially to the output resource. Coffman at 31:23-49: Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally confirmation is obtained from the user. As soon a context is found that fits the utterance at the level of the NLU symbolic language, the context becomes active and the corresponding process becomes active. Until the active command is completed, or until a new command is provided, the selected context is marked active, and pushed to the top of the context stack 405. When a message is returned to the user, the context is updated and then pushed to the top of the context stack 405 under the active context. The active context is updated to inform of the existence of a returned value. This can also be done at the level of the superseding CVM session discourse, which can be in the stack or always besides the stack and then searched right after the active context, before going down the stack. Simultaneously completed tasks result in contexts that are arranged under the active context according to CVM priorities (e.g. FIFO or FILO). Active contexts sufficiently completed to generate a task will be pushed down the stack under the next or all the returned contexts. Or it could become the active discourse. This may be done automatically or when commanded by the user. This stack structure allows to maintain non-ambiguous conversational interactions with multiple tasks, threads or processes.

Claim 13 of the '160 Patent	Exemplary Citations to Coffman
	Coffman at 41:35-43: When a user interacts with the collection of devices, the interaction may always be via a central unit such as a PVA (personal vehicle assistant) 710 in a car, or a speech browser 713. The task dispatcher and context stack accumulates the contexts associated to all the devices and will parse and dispatch commands to each device accordingly. If the user interacts with the entire collection of devices, then a device is always active (the last activated context). This devices check if a new command fits its context stack. If not, it passes to a neighboring device that becomes active. The process is iterated until a match is found, and possibly confirmed by the user, or the request bounces back to the first device. In that case, an error or confusion message is returned to the user. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[b] transcribing the utterance using the multi-pass speech recognition module, the multi-pass speech recognition module configured to transcribe the utterance into text;	Coffman discloses “transcribing the utterance using the multi-pass speech recognition module, the multi-pass speech recognition module configured to transcribe the utterance into text[.]” See citations of 12[a]. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of

Claim 13 of the '160 Patent	Exemplary Citations to Coffman
	ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[c] identifying one or more contexts that completely or partially match one or more text combinations contained in the text of the transcribed utterance, wherein identifying the matching contexts includes comparing the text combinations against the grammar expression entries in the context description grammar and against one or more expected contexts stored in a context stack;	Coffman discloses “identifying one or more contexts that completely or partially match one or more text combinations contained in the text of the transcribed utterance, wherein identifying the matching contexts includes comparing the text combinations against the grammar expression entries in the context description grammar and against one or more expected contexts stored in a context stack[.]” See citations of 12[b]. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[d] scoring each of the identified matching contexts;	Coffman discloses “scoring each of the identified matching contexts[.]” See citations of 12[c]. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover

Claim 13 of the '160 Patent	Exemplary Citations to Coffman
	Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[e] selecting the matching context having a highest score to determine a most likely context for the utterance; and	Coffman discloses “selecting the matching context having a highest score to determine a most likely context for the utterance[.]” See citations of 12[d]. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.
[f] 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.	Coffman discloses “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[.]” See citations of 12[e]. To the extent it is found that Coffman does not expressly disclose this limitation, the limitation is inherent. And to the extent it is found that Coffman does not anticipate this claim, Coffman renders it obvious, standing alone, in combination with the knowledge of a person of ordinary skill in the art, and/or in combination with one or more other prior art references identified in the Cover

Dialect, LLC v Bank of America, N.A.
Eastern District of Texas No. 2:24-cv-00207-JRG

Claim 13 of the '160 Patent	Exemplary Citations to Coffman
	Pleading or herein, for the reasons identified herein or in the Cover Pleading. A person of ordinary skill in the art would have been motivated to combine Coffman with any of the other prior art references identified in the Cover Pleading or herein with a reasonable expectation of success because each of these references contains familiar elements within the similar fields of endeavor to address similar known problems, according to known methods which would yield predictable results.