

Exhibit E-Coffman

Invalidity Chart for U.S. Patent No. 9,495,957 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. 9,495,957 (the “’957 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 7 and 8 of the ’957 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 herein and/or in the Cover Pleading, for the reasons identified herein and/or in the Cover Pleading, to render the claims at-issue in the ’957 Patent invalid.

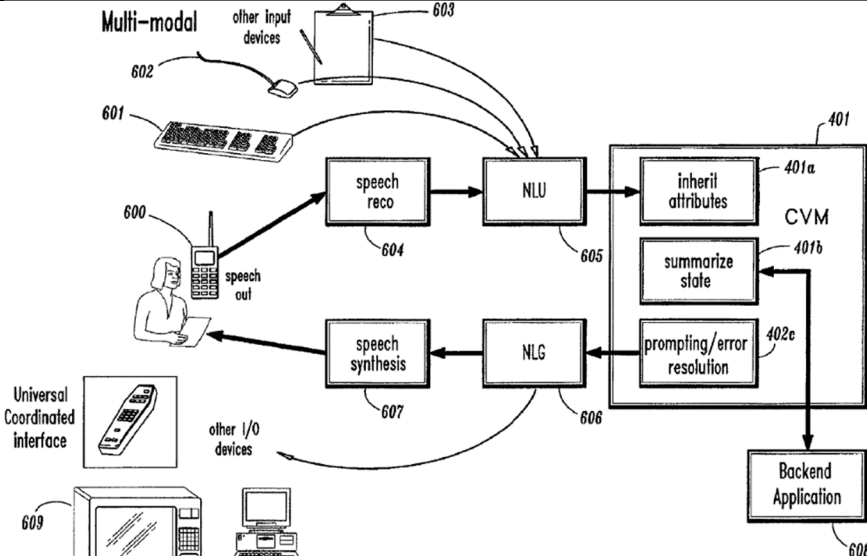
Defendant Bank of America, N.A. (“Bank”) incorporates in this chart all applicable qualifications, clarifications, and other statements made in Bank’s Invalidity Contentions. This invalidity claim chart is based on Bank’s present understanding of claims 7, 8, and Plaintiff Dialect, LLC (“Dialect”) apparent 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 [Coffman] 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.

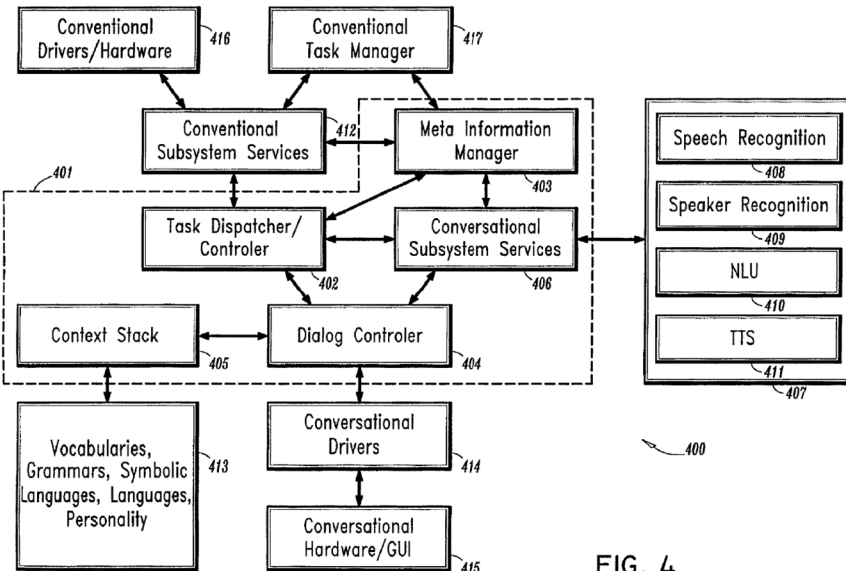
These invalidity contentions are not an admission by Bank that the accused products or components, including any current or past version of the products or components, are covered by, or infringe the asserted claims, particularly when these claims are properly construed and applied.

1. Claim 7

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
<p>[p] A computer-implemented method of processing a natural language utterance, the method comprising:</p>	<p>To the extent the preamble is limiting, Coffman discloses a “computer-implemented method of processing a natural language utterance[.]”</p> <p><i>E.g.</i>, Coffman at Fig. 1:</p> <div data-bbox="630 535 1449 1185"> <p>FIG. 1</p> </div> <p>Coffman at Fig. 6:</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	 <p style="text-align: center;">FIG. 6</p> <p>Coffman at 3:57-4:5:</p> <p>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 APIs included in the conversation application APIs or by emulating similar calls and functionalities at the level of the conversational application APIs.</p> <p>Coffman at 6:47-55:</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>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.</p> <p>Coffman at 21:45-64:</p> <p>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).</p> <p>Coffman at 28:22-29:</p> <p>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 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.</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<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.</p>
<p>[a] generating a context stack comprising context information that corresponds to a plurality of prior utterances, wherein the context stack includes a plurality of context entries;</p>	<p>Coffman discloses “generating a context stack comprising context information that corresponds to a plurality of prior utterances, wherein the context stack includes a plurality of context entries[.]”</p> <p><i>E.g., Coffman at Fig. 4:</i></p>  <p style="text-align: right;">FIG. 4</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>Coffman at 4:6-26:</p> <p>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.</p> <p>Coffman at 22:5-46:</p> <p>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</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>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).</p> <p>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.</p> <p>Coffman at 31:23-49:</p> <p>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</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>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.</p> <p>Coffman at 42:58-43:6:</p> <p>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:</p> <ol style="list-style-type: none">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; and5 CVM capabilities/state: does it run a CVM; is it a CVM client; is it a master, slave or partner CVM. <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.</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
<p>[b] receiving, by one or more computer processors, the natural language utterance, wherein the natural language utterance is associated with a command or is associated with a request;</p>	<p>Coffman discloses “receiving, by one or more computer processors, the natural language utterance, wherein the natural language utterance is associated with a command or is associated with a request[.]”</p> <p><i>E.g.</i>, Coffman at 30:36-50:</p> <p>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.</p> <p>Coffman at 31:23-49:</p> <p>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</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>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.</p> <p>Coffman at 42:17-32:</p> <p>You can instruct the PVA to dial in your PC using the phone. Once the connection is established, you are authenticated by voice and you find by voice the memo by simply requesting the “grocery list” that you had previously created, without having to remember the file name or the directory or browse through your directory to eventually select the appropriate file. You may need to confirm the selection if your PC CVM requests it. You can issue another query—“it should be synchronized with my PDA!—and the file is appropriately linked to be transferred to your PDA at the next synchronization. One last command—“Do it!”—and your PC gives up and lets the PVA handle that ambiguous query. The PVA understands your desire to synchronize the PDA and the PC based on your previous conversation. After possible confirmation, the synchronization is performed and the grocery list is stored on your PDA, ready for later use.</p> <p>Coffman at 42:33-39:</p> <p>You now instruct the PVA to guide you turn by turn to the store. Your position is computed, the location of the store is fetched, locally or from a server, and an itinerary is computed to take into account the latest traffic information. At any time, you can request navigation information about where you are, what to do next, how far to go or even request a different itinerary.</p> <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</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>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.</p>
<p>[c] determining one or more words of the natural language utterance by performing speech recognition on the natural language utterance;</p>	<p>Coffman discloses “determining one or more words of the natural language utterance by performing speech recognition on the natural language utterance[.]”</p> <p><i>E.g.</i>, Coffman at 4:47-64:</p> <p>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.</p> <p>Coffman at 28:39-57:</p> <p>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.</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>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.</p> <p>Coffman at 30:36-50:</p> <p>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.</p> <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.</p>
[d] identifying, from among the plurality of context entries, one or more context entries that correspond to the one or	Coffman discloses “identifying, from among the plurality of context entries, one or more context entries that correspond to the one or more words, wherein the context information includes the one or more context entries[.]”

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
<p>more words, wherein the context information includes the one or more context entries, wherein identifying the one or more context entries comprises:</p>	<p><i>E.g.</i>, Coffman at 4:6-26:</p> <p>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.</p> <p>Coffman at 22:5-46:</p> <p>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,</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>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).</p> <p>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.</p> <p>Coffman at 31:23-49:</p> <p>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</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>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.</p> <p>Coffman at 42:58-43:6:</p> <p>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:</p> <ol style="list-style-type: none">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; and5 CVM capabilities/state: does it run a CVM; is it a CVM client; is it a master, slave or partner CVM. <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.</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
<p>[e] comparing the plurality of context entries to the one or more words;</p>	<p>Coffman discloses “comparing the plurality of context entries to the one or more words[.]”</p> <p><i>E.g.</i>, Coffman at 31:23-49:</p> <p>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.</p> <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</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	endeavor to address similar known problems, according to known methods which would yield predictable results.
[f] generating, based on the comparison, one or more rank scores for individual context entries of the plurality of context entries; and	<p>Coffman discloses “generating, based on the comparison, one or more rank scores for individual context entries of the plurality of context entries[.]”</p> <p><i>E.g.</i>, Coffman at 31:23-30:</p> <p>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.</p> <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.</p>
[g] identifying, based on the one or more rank scores, the one or more context entries from among the plurality of context entries; and	<p>Coffman discloses “identifying, based on the one or more rank scores, the one or more context entries from among the plurality of context entries[.]”</p> <p><i>E.g.</i>, Coffman at 31:23-30:</p> <p>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</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>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.</p> <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.</p>
<p>[h] determining based on the determined one or more words and the context information, the command or the request associated with the natural language utterance</p>	<p>Coffman discloses “determining based on the determined one or more words and the context information, the command or the request associated with the natural language utterance[.]”</p> <p><i>E.g.</i>, Coffman at 30:36-50:</p> <p>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.</p> <p>Coffman at 31:23-49:</p>

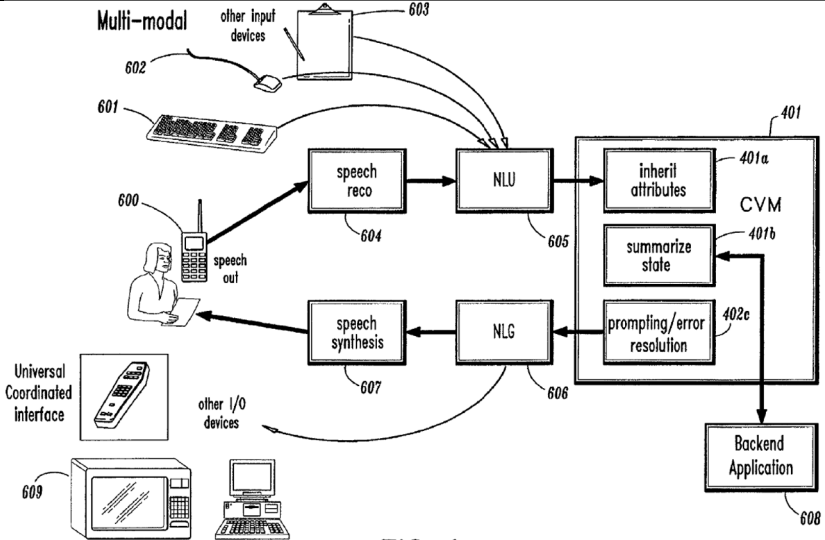
Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>Completion/search for the active context is performed from context to context down the stack. That is, new queries or arguments are compared by the dialog engine by going down the stack until an acceptable match is obtained and optionally 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.</p> <p>Coffman at 42:17-32:</p> <p>You can instruct the PVA to dial in your PC using the phone. Once the connection is established, you are authenticated by voice and you find by voice the memo by simply requesting the “grocery list” that you had previously created, without having to remember the file name or the directory or browse through your directory to eventually select the appropriate file. You may need to confirm the selection if your PC CVM requests it. You can issue another query—“it should be synchronized with my PDA!—and the file is appropriately linked to be transferred to your PDA at the next synchronization. One last command—“Do it!”—and your PC gives up and lets the PVA handle that ambiguous query. The PVA understands your desire to synchronize the PDA and the PC based on your previous conversation. After possible confirmation, the synchronization is performed and the grocery list is stored on your PDA, ready for later use.</p>

Claim 7 of the '957 Patent	Exemplary Citations to Coffman
	<p>Coffman at 42:33-39:</p> <p>You now instruct the PVA to guide you turn by turn to the store. Your position is computed, the location of the store is fetched, locally or from a server, and an itinerary is computed to take into account the latest traffic information. At any time, you can request navigation information about where you are, what to do next, how far to go or even request a different itinerary.</p> <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.</p>

2. Claim 8

Claim 8 of the '957 Patent	Exemplary Citations to Coffman
<p>[a] The method of claim 7, further comprising: providing a prompt regarding the command or request, wherein the prompt is a prompt for additional information or a prompt for confirmation; and</p>	<p>Coffman discloses “providing a prompt regarding the command or request, wherein the prompt is a prompt for additional information or a prompt for confirmation[.]”</p> <p><i>E.g.</i>, Coffman at 42:17-32:</p> <p>You can instruct the PVA to dial in your PC using the phone. Once the connection is established, you are authenticated by voice and you find by voice the memo by simply requesting the “grocery list” that you had previously created, without having to remember the file name or the directory or browse through your directory to eventually select the appropriate file. You may need to confirm</p>

Claim 8 of the '957 Patent	Exemplary Citations to Coffman
	<p>the selection if your PC CVM requests it. You can issue another query—“it should be synchronized with my PDA!”—and the file is appropriately linked to be transferred to your PDA at the next synchronization. One last command—“Do it!”—and your PC gives up and lets the PVA handle that ambiguous query. The PVA understands your desire to synchronize the PDA and the PC based on your previous conversation. After possible confirmation, the synchronization is performed and the grocery list is stored on your PDA, ready for later use.</p> <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.</p>
[b] receiving, in response to providing the prompt, a non-speech input, wherein the command or request is determined further based on the non-speech input	<p>Coffman discloses “receiving, in response to providing the prompt, a non-speech input, wherein the command or request is determined further based on the non-speech input[.]”</p> <p><i>E.g.</i>, Coffman at Fig. 6:</p>

Claim 8 of the '957 Patent	Exemplary Citations to Coffman
	 <p style="text-align: center;">FIG. 6</p> <p>Coffman at 5:8-30:</p> <p>In another aspect, programming/script languages are utilized that allow the use of any available resources as input or output stream. Using the conversational sub-systems, each input is converted into a binary or ASCII input (lists of attribute-value pairs or n-uples), which can be directly processed by the programming language as built-in objects. Calls, flags and tags are automatically included to transmit between object and processes the conversational meta-information required to correctly interface with the different objects. Indeed, any input in any modality is captured by the dialog manager of the CVM kernel layer as an event that is added to the associated context or context stack. For example, a mouse click or pointer/stylus pointing action followed by the command “I would like to open this” is disambiguated into a set of attribute value pairs: Command: open, Object: Windows or task selected by the last mouse click. Output can be specially formatted according to the needs of the application or user. Multi-modal discourse processing can now be easily built using the new programming tools. In addition, such</p>

Claim 8 of the '957 Patent	Exemplary Citations to Coffman
	<p>programming languages and scripts encompasses conversational API between conversational enabled applications and the CVM, as well as CML (conversational markup language).</p> <p>Coffman at 8:33-63:</p> <p>Mixed initiative also implies that the CVM can actively help (take the initiative to help) and coach a user through a task, especially in speech-enabled applications, wherein the mixed initiative capability is a natural way of compensating for a display less system or system with limited display capabilities. In general, the CVM complements conventional interfaces and user input/output rather than replacing them. This is the notion of “multi-modality” whereby speech is used in parallel with mouse, keyboard, and other input devices such as a pen. Conventional interfaces can be replaced when device limitations constrain the implementation of certain interfaces. In addition, the ubiquity and uniformity of the resulting interface across devices, tiers and services is an additional mandatory characteristic. It is to be understood that CVM system can to a large extent function with conventional input and/or output media. Indeed, a computer with classical keyboard inputs and pointing devices coupled with traditional monitor display can profit significantly by utilizing the CVM according to the present invention. One example is described in provisional application U.S. Ser. No. 60/128,081, filed on Apr. 7, 1999, entitled “Multi-Modal Shell” which is commonly assigned and incorporated herein by reference (which describes a method for constructing a true multi-modal application with tight synchronization between a GUI modality and a speech modality). In other words, even users who do not want to talk to their computer can also realize a dramatic positive change to their interaction with the CVM enabled machine.</p> <p>Coffman at 14:42-56:</p> <p>The conversational computing system of FIG. 2 further comprises a conversational engine API layer 207 which provides an interface between core engines conversational engines 208 (e.g., speech recognition, NL parsing, NLU, TTS and speech compression/decompression engines) and the applications using them. The engine API layer 207 also provides the protocols to communicate with core engines whether they be local or remote. An I/O API layer 210 provides an interface</p>

Claim 8 of the '957 Patent	Exemplary Citations to Coffman
	<p>with conventional I/O resources 211 such as a keyboard, mouse, touch screen, keypad, etc. (for providing a multi-modal conversational UI) and an audio subsystem for capturing speech I/O (audio in/audio out). The I/O API layer 210 provides device abstractions, I/O abstractions and UI abstractions. The I/O resources 211 will register with the CVM kernel layer 202 via the I/O API layer 210.</p> <p>Coffman at 27:65-28:38:</p> <p>Referring now to FIG. 6, a diagram illustrates a conversational input/output interface in accordance with one embodiment of the present invention. As illustrated, a conversational input interface according to an embodiment of the present invention can process multi-modal input, that is, files/streams/resources, speech via a phone 600, keyboard 601, pointing devices 602, handwriting devices 603, including natural interfaces. This means that all the input and output events across all the modalities are caught and transferred to the dialog manager (that also stores it appropriately in the context stack). Spoken input from a speech client (e.g., telephone 600) is subject to a speech recognition process 604 and other input (e.g., keyboard, mouse clicks etc) are subject to NLU processing 605. Each input is subject to attribute acquisition (401 a) whereby the attribute value n-uples are acquired from the input. A summarization process 401 b is performed whereby the attribute value n-uples are added to the context and then verifies with the syntax of the back-end application 608 whether the query is complete, incomplete, or ambiguous. The backend accesses are also tracked by the dialog manager and the context manager. It is sometimes possible to distribute some of the “intelligence” to the backend by loading some disambiguation capabilities (a feature of the dialog manager) to the backend. Individually, each input stream behaves the conventionally. 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 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. As such, the CVM introduces new issues not met with conventional OS: simultaneous input streams to be merged which create input ambiguity. For example, input may now combine input keyed on the keyboard, handwritten input and speech input, not to mention possible input from re-directed streams. Therefore, the present</p>

Claim 8 of the '957 Patent	Exemplary Citations to Coffman
	<p>invention provides a mechanism to resolve any ambiguity. This may be performed as explained in the above-incorporated application U.S. Ser. No. 60/128,081.</p> <p>Coffman at 42:17-32:</p> <p>You can instruct the PVA to dial in your PC using the phone. Once the connection is established, you are authenticated by voice and you find by voice the memo by simply requesting the “grocery list” that you had previously created, without having to remember the file name or the directory or browse through your directory to eventually select the appropriate file. You may need to confirm the selection if your PC CVM requests it. You can issue another query—“it should be synchronized with my PDA!—and the file is appropriately linked to be transferred to your PDA at the next synchronization. One last command—“Do it!”—and your PC gives up and lets the PVA handle that ambiguous query. The PVA understands your desire to synchronize the PDA and the PC based on your previous conversation. After possible confirmation, the synchronization is performed and the grocery list is stored on your PDA, ready for later use.</p> <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.</p>