

<p>Operand ↓</p>	<p>PM</p>	<p><u>Preferred embodiment Copy processing</u></p>
<p>209</p>	<p>C</p>	<p>Copying a broadcast email causes searching the source email system with search criteria of the email parameter string. The email parameter string can specify searching any email fields for any values including wildcarding (patterns for matching). Each field is referenced with a predefined name and then associated with a search criteria. For example, the email string of "subj:'personnel'; recip:'george@alltell.com'; body:'reduction in force'" causes searching all emails with a subject containing "personnel" and was sent to "george@alltell.com" and has a message body containing the string "reduction in force". To search for certain email containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:sent,inbox,company;" indicates to only search the email folders of sent, inbox, and company (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of email. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In one embodiment, all occurrences found in history are presented with at least their date/time stamps, subject line, sender and recipient, and perhaps other information, of the email and when it took place, when the ack parameter is set to true for user reconciliation. In a preferred embodiment, the most recent occurrence from searched folders is provided for the copy. An additional parameter may be specified for the target (specified folder), otherwise the object is copied to an assumed location (e.g. inbox, drafts, special named folder). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for doing the copy. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The copy is made to a special email folder of the target system, or another designated folder configured ahead of time, or as specified with a new parameter to copy processing, preferably with identifying information of the source and who did the copy.</p>

Fig. 69B-4

		Preferred embodiment Copy processing
<i>Operand</i> 211	PM C	<p>Copying a broadcast sms msg causes searching the source messaging system with search criteria of the sms message parameter string. The message parameter string can specify searching any message fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria. For example, the sms message string of "recip:'2144034071@nextel.com,9725397137@lboxsv.com,'" causes searching all messages to the sought recipients. To search for certain messaging containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:outgoing" indicates to only search the outgoing folder (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of messages. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In the one embodiment, all occurrences found in history are presented with at least their date/time stamps, message, sender and recipient, and perhaps other information, of the message and when it took place, when the ack parameter is set to true for user reconciliation. In a preferred embodiment, the most recent occurrence from searched folders is provided for copying. An additional parameter may be specified for the target (specified folder), otherwise the object is copied to an assumed location (e.g. inbox, drafts, special named folder). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The copy is made to a special messaging folder of the target system, or another designated folder configured ahead of time, or as specified with a new parameter to copy processing, preferably with identifying information of the source and who did the copy.</p>
213	C	<p>Copying an indicator searches appropriate source storage for the indicator (e.g. storage/memory used for indicators by other commands). The indicator parameter string specifies the indicator string being sought and wildcarding is supported. In one embodiment, appropriate MS storage/memory which contains the history of indicators sent to the source system is searched and all occurrences found in history are presented with at least their date/time stamps, the indicator, and perhaps other information, for user reconciliation at block 6942. In a preferred embodiment, the most recently delivered indicator is identified and used for the copy. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for the copy. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The copy is made so that the target system(s) are delivered the indicator(s) like delivering a new indicator for presentation.</p>

Fig. 69B-5

Operand ↓	PM	Preferred embodiment Copy processing
215	C	<p>Copying an application causes searching the source system for the application (and with the params parameter(s) if specified to get the params specified invocation of the application). The app parameter is preferably an executable name and may contain parameters that were passed. Providing a more defined partial or full path to the application parameter will limit the search result. The app parameter string preferably supports wildcarding. In one embodiment, all occurrences found at the source and their paths are presented to the user with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation at block 6942. In a preferred embodiment, the most recently executed instance of the matching application is determined for the copy. In one embodiment, the application itself is copied to the target systems, perhaps as directed by an additional parameter (e.g. directory location). In another embodiment, the executable path to run the application is placed into execution history at the system(s) so that a user can run it, albeit from a remote system (assumption that application available for running there already). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provide with the result for copy. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use.</p>
217	C	<p>Copying a document causes searching the source for the document. The doc parameter is a document name. The document parameter can be a wildcard (pattern) for matching. Providing a more defined partial or full path to the document name will narrow the search. In one embodiment, all occurrences found on the MS and their paths are presented to the user with at least their date/time stamps, size, and perhaps attributes information, for reconciliation at block 6942. In a preferred embodiment, the most recently accessed search result is provided for the copy. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Copying the document places a copy to each target system at a special shared folder, or configured folder for sharing, or as specified with a new destination parameter to copy processing.</p>

Fig. 69B-6

		<u>Preferred embodiment Copy processing</u>
Operand ↓ 219	PM C	<p>Copying a file causes searching the source path for the file. The path parameter is a file name. Providing a more defined partial or full path to the file will narrow the search result. The path parameter can be a wildcard (pattern) for matching. In one embodiment, all occurrences found on the MS and their paths are provided with at least their date/time stamps, size, and perhaps attributes information, for reconciliation at block 6942. In a preferred embodiment, the most recently accessed file meeting the search criteria is copied to the target systems. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provide with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In one embodiment, the copy results in overwriting an existing file with the same handle (e.g. name). In another embodiment, the copy results in writing a newly altered name of the file when there is an existing file with the same handle (e.g. name). An additional parameter may be specified for the target (specified folder), otherwise the object is copied to an assumed location).</p>
221	O	<p>Copying content causes searching the source for the content. The content parameter is a reference to the content. The content parameter can be a wildcard (pattern) for matching. In a preferred embodiment, the most recently accessed search result is provided for the copy. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Copying the content places a copy to each target system at a special shared destination, or configured destination for sharing, or as specified with a new parameter to copy processing.</p>

Fig. 69B-7

Operand	PM	<u>Preferred embodiment Copy processing</u>
223	O	<p>Copying a DB object causes searching the source for the database object value. The database object parameter is provided with a variety of syntaxes depending on the type of database object sought. For example, the DB-obj parameters is "T:tablename" to seek a table, "S:schemaname" to seek a particular schema, "C:columnname" to seek a particular column name, "D:DBname" to seek a particular DB name, "R:rolename" to seek a particular role set, "P:procname" to search for particular stored procedure, etc. There are unique syntaxes for every type of DB object being sought which maps to an appropriate SQL system tables query. The search criteria can be a wildcard (pattern) for matching. In one embodiment, all occurrences found on the source system and information about the occurrence is presented to the user for reconciliation at block 6942. In other embodiments, the best (e.g. most recently accessed) fit database object is identified for use in the copy, or a new parameter indicating how to search. The search takes place as though the user manually launched the search, entered the criteria or query for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The value of the DB object is copied to the value of the DB object with the same name and type at the destination system(s). If not found at a target system, then no action is performed at that system. Copying a database object copies the value to the same database object(s) at other system(s), or creates new copies of the DB objects there when names do not match. Copying a DB object is intended to keep DBs in synch in some uses. Value(s) are overwritten.</p>

Fig. 69B-8

		Preferred embodiment Copy processing
Operand ↓ 225	PM O	<p>Copying data causes searching the source system for the data. In the preferred embodiment, the data is a global system variable visible to all processes of a MS operating system. In other embodiments, the data may have limited scope which is made accessible to present disclosure processing (e.g. with extern). Depending on the embodiment, data may be that which is contained in a program data segment, stack segment, and/or extra segment. There can be unique syntaxes for specifying which type of data is being sought (e.g. "S:dataname" for data parameter). The search criteria can be a wildcard (pattern) for matching. In the preferred embodiment, a well known location of link symbol information files are consulted, and in another embodiment a new parameter specifies where to look, or which symbol file of information to use. In one embodiment, all occurrences found at the source system and information about the occurrence including its current value is presented to the user for reconciliation at block 6942. In a preferred embodiment, the best data value (e.g. most recently accessed if more than one matches) is provided for the copy. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The value of the copied data is copied to the data with the same name and type at the destination system(s). If not found at a target system, then no action is performed at that system, or an error is provided. Copying a data object copies the value to the same data object(s) at other system(s). Copying is intended to keep data in synch between systems in some uses. Value(s) are overwritten.</p>
227	O	<p>Copying a semaphore causes reading the current value of the semaphore at the source where the copy command action is being executed and then copying the current value to the same semaphore names at the target system(s). The semaphore param can be a wildcard (pattern) for matching. In the preferred embodiment, the semaphore is a global system semaphore visible to all processes of a MS operating system. In other embodiments, the semaphore may have limited scope which is made accessible to present disclosure processing. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The value (set or cleared) of the copied semaphore is copied to the semaphore with the same name and type at the destination system(s). If not found at a target system, then no action is performed at that system, or an error is provided. Copying a semaphore copies the value to the same semaphore at other system(s). Copying a semaphore is intended to keep systems in synch in some uses. Value(s) are overwritten.</p>

Fig. 69B-9

	Operand	PM	Preferred embodiment Copy processing
229	C	C	Copying a directory causes searching the source system for the directory. The path parameter is a directory name. Providing a more defined partial or full path to the directory parameter will narrow the search result. The path parameter can be a wildcard (pattern) for matching. In one embodiment, all occurrences found on the MS and their paths are provided with at least their date/time stamps, size, and perhaps attributes information, for reconciliation at block 6942. In a preferred embodiment, the most recently accessed directory meeting the search criteria is copied to the target systems. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In one embodiment, the copy results in overwriting an existing directory and files therein. In another embodiment, the copy results in writing a newly altered name of directory contents when there is a conflict (e.g. existing entity with same name). In another embodiment, an additional target path parameter is provided for where to place the directory.
231	C	C	Operand 215 (application object) is treated identically to this Operand 231 (application context) this LBX release (same params currently).
233	C	C	Copying a focused user interface object causes capturing the currently focused user interface object using the first parameter (e.g. Alt-Prtscm; can be changed with the param) string syntax for keystroke(s) to capture the image, and then copying the graphics file (file type in various embodiments) to a shared destination, or a configured destination at the target system(s) or as specified with a new parameter. The capture takes place as though the user manually performed the capture action, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the capture and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In one embodiment, the copy results in overwriting an existing file with the same handle (e.g. name), which will be seldom since the graphics file name preferably contains a date/time stamp portion. In another embodiment, the copy results in writing a newly altered name of the file when there is an existing file with the same handle (e.g. name).
235	C	C	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter). Also, the ack parameter provides a reconciliation option.
237	C	C	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter). Also, the ack parameter provides a reconciliation option.

Fig. 69B-10

		<u>Preferred embodiment Copy processing</u>
<u>PM</u>		
Operand ↓ 239	C	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter). Also, the ack parameter provides a reconciliation option.
241	C	Copying an alert causes searching the source for the alert. The alert parameter is the same parameter used to generate an alert (e.g. using another command), and can be wildcarded. In one embodiment, all occurrences found on the MS which is associated to the alert application in use at the MS, and which is used for other commands disclosed, are provided to the user for reconciliation at block 6942 with at least their date/time stamps, and perhaps other information. In other embodiments, the most recently generated alert matching the alert search criteria is used for copying, or the search occurs as specified with an additional parameter. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The copy is made so that the target system(s) are delivered the alert(s) like delivering a new alert to the systems.
243	C	Copying a process causes first finding all process names running at the source (e.g. MS) which contain the pname string parameter (e.g. in UNIX: "ps -ef grep pname"). In one embodiment, all occurrences found running at the MS are presented with interesting programmatic information such as when started, its size, etc for reconciliation at block 6942. In the preferred embodiment, one process running in the source system is to be found (i.e. >1 = ambiguous). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying. Results are useful statistics about the process which is running at the source. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, historical log, or other useful storage for subsequent use. Useful statistic(s) about the process (perhaps which statistics specified with an additional parameter) are copied to an appropriate destination of the target system(s) for informative purposes (e.g. a special log file). In another embodiment, the alerter process and/or indicator methodology can be used as the destination for the copy for alerting a user at the target system. In another embodiment, there is new parameter for which end result the copy will have (informative destination, handled like alert, handled like indicator).

Fig. 69B-11

Operand	PM	Preferred embodiment Copy processing
245	C	<p>Copying a container causes searching the source system for the container. The container parameter can be well defined to narrow the search result, and may be wildcarded (pattern) for matching. In one embodiment, all occurrences found on the MS and their references/handles are provided with at least their date/time stamps, size, and perhaps attributes information, for reconciliation at block 6942. In a preferred embodiment, the most recently accessed container meeting the search criteria is copied to the target systems. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In one embodiment, the copy results in overwriting an existing container. In another embodiment, the copy results in writing a newly altered reference/handle of the container when there is a conflict (e.g. existing entity with same name). An additional parameter may be specified for the target, otherwise the object is copied to an assumed location.</p>
247	C	<p>Copying a program object first causes searching the source for the program object. In the preferred embodiment, a unique syntax is used for which type of program object is being sought (similar to above). There can be unique syntaxes for specifying which type of program object is being sought (e.g. "S:dataname"). In one embodiment, all occurrences found on the MS and information about the occurrence including its current value is presented to the user for reconciliation at block 6942. In a preferred embodiment, one program object is to be found (e.g. >1 = ambiguous). In one embodiment, a well known location of link symbol information files are consulted, and in another embodiment a new parameter specifies where to look, or which symbol file of information to use. The search criteria can be a wildcard (pattern) for matching. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Useful statistic(s) about the program object (perhaps which statistics specified with an additional parameter) are copied to an appropriate destination of the target system(s) for informative purposes (e.g. a special log file). In another embodiment, the alterer process and/or indicator methodology can be used as the destination for the copy for alerting a user at the target system. In another embodiment, there is new parameter for which end result the copy will have (informative destination, handled like alert, handled like indicator). An alternate embodiment works like Operand 223 wherein copying is intended to keep program object(s) between systems in synch. Such embodiments require source and target system processing to have access to the object(s) (this may limit participating object(s)).</p>

Fig. 69B-12

Operand	<u>PM</u>	<u>Preferred embodiment Copy processing</u>
249	C	Copying a cursor causes searching the source system for the current cursor setting(s). In the preferred embodiment, provided in the search results is the current cursor information (e.g. image, animation, etc). The current cursor information of the source is then used to alter the cursor at the target system(s). Appropriate MS storage is updated and subsequently processed as though the user had manually performed the copy operation. Preferably, the copy command data is maintained to LBX-History, a historical log, or other useful storage for subsequent use.
251	C	Copying a calendar (CAL) object causes searching the source CAL system with search criteria of the calendar object parameter string. The calendar object parameter string can specify searching any calendar entry fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria (similar to email above). Those skilled in the art recognize many useful syntaxes for searching any characteristics of CAL objects. In one embodiment, all occurrences found in history are presented to the user for reconciliation at block 6942 with at least their date/time stamps, attendees, and perhaps other information, of the CAL object and when it was scheduled. In a preferred embodiment, the most recent occurrence from the calendaring system is provided for the copy. Wildcarding (pattern matching) is preferably inherent by searching for substrings. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX-History, a historical log, or other useful storage for subsequent use. The calendar entry is copied in its entirety to the target system calendaring system. In another embodiment, a new parameter is specified to copy the calendar item to a new schedule or time. A duplicate calendar entry may be created if one already exists.

Fig. 69B-13

	<u>PM</u>	<u>Preferred embodiment Copy processing</u>
Operand ↓ 253	C	<p>Copying an address book (AB) object causes searching the source AB system with search criteria of the AB object parameter string. The AB object parameter string can specify searching any AB entry fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria (similar to email above). Those skilled in the art recognize many useful syntaxes for searching any characteristics of AB objects/entries. In one embodiment, all occurrences found are presented to the user for reconciliation at block 6942 with appropriate AB information. In a preferred embodiment, the most recent occurrence from the AB system is provided for the copy. Wildcarding (pattern matching) is preferably inherent by searching for substrings. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for copying it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and copy. Preferably, the copy command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The AB entry is copied in its entirety to the target system AB system. In another embodiment, a new parameter is specified to copy the AB item to special destination. A duplicate AB entry may be created if one already exists.</p>
...		

Fig. 69B-14

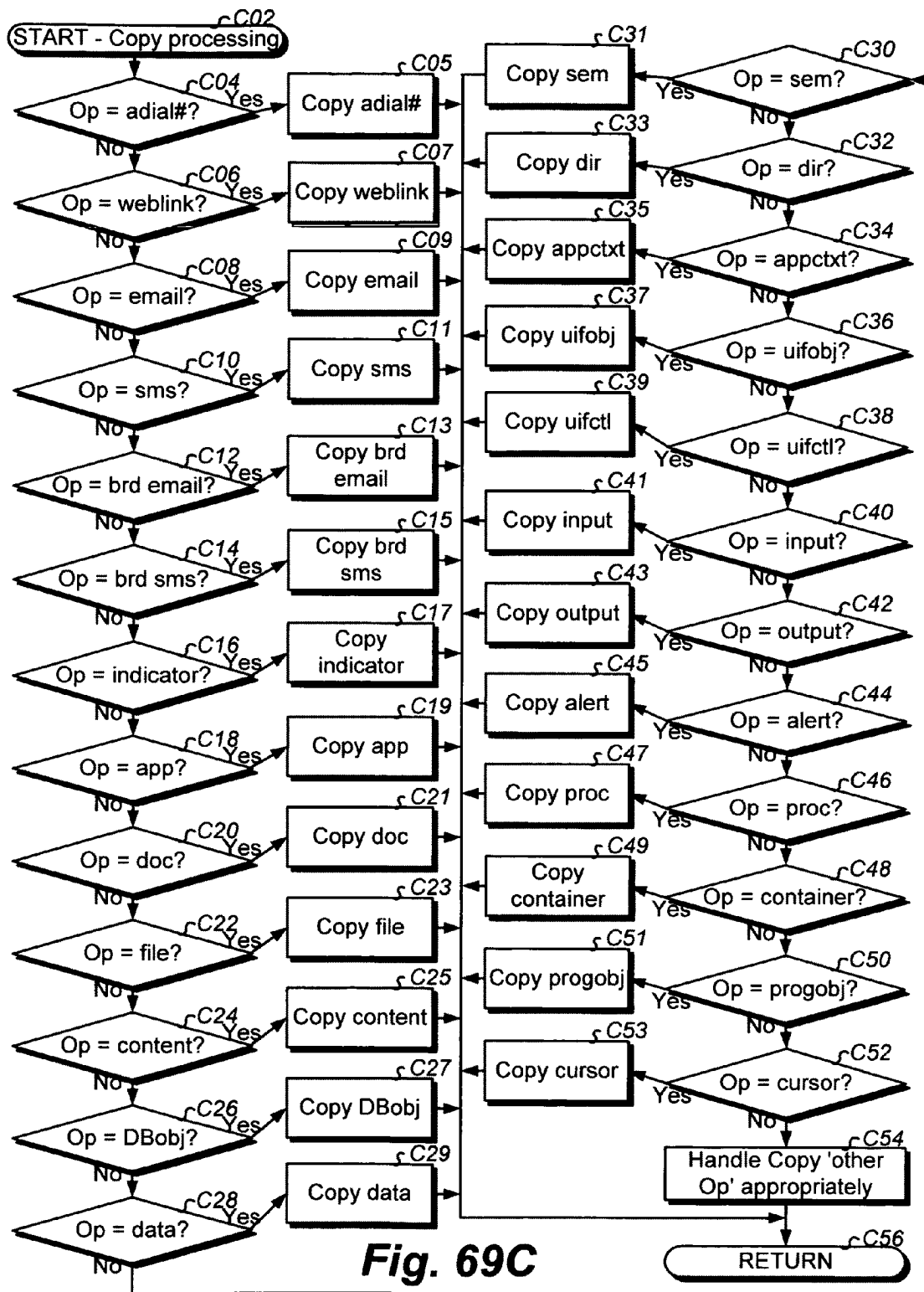


Fig. 69C

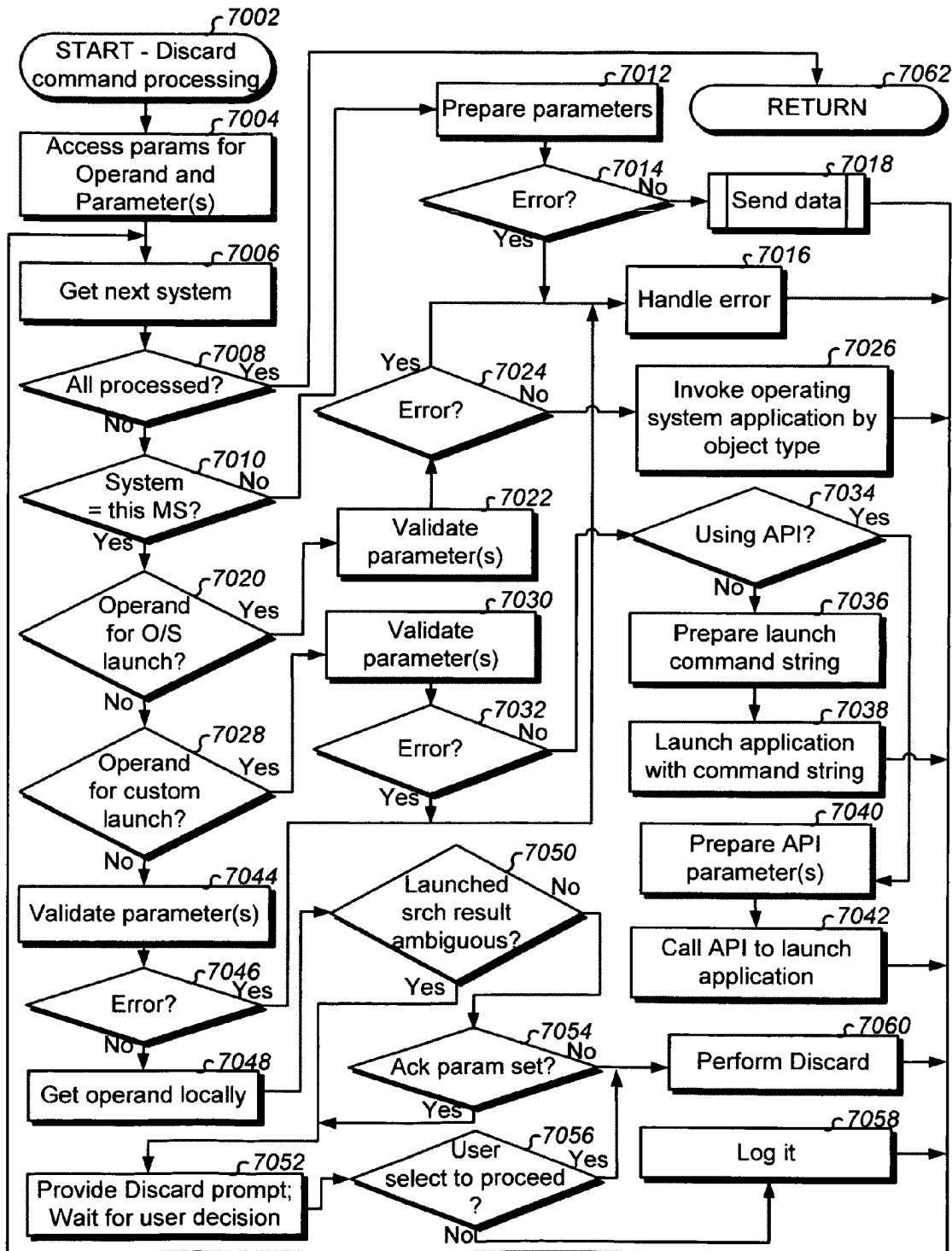


Fig. 70A

		<u>Preferred embodiment Discard processing</u>
<u>Operand</u> ↓	<u>PM</u>	
201	C	Discarding an auto-dial # launches a phone number log interface with the auto-dial # parameter (can be wildcarded) for searching the specified system (e.g. MS). Preferably, both the outgoing and incoming logs are searched. In an alternate embodiment, the log is specified with a parameter. In the preferred embodiment, the most recent occurrence from a particular log is to be discarded. In another embodiment, all occurrences found in history are presented with their date/time stamps, and perhaps other information, of the call and when it took place when the ack parameter is set and the user browses the results prior to accepting the discard of multiple items. The search takes place as though the user manually launched the search, entered the auto-dial # for the search, and then was provided with result(s) for the discard. Preferably, the discard shall take place if there are no ambiguities (e.g. more than one phone number returned per search criteria). Appropriate MS storage is updated and subsequently processed as though the user manually performed the search and discard of the result. Preferably, the discard cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The discard is made to system(s) logs.
203	S	Discarding a weblink launches a search to browser history with the weblink parameter (can be wildcarded) for searching the specified system. In one embodiment, all occurrences found in history are presented with their date/time stamps, and perhaps other information, of the link and when it was invoked, when the ack parameter is specified to true for presentation to the user prior to doing the discard. In the preferred embodiment, the most recent occurrence from a particular invocation is provided for the discard. The search takes place as though the user manually launched the search, entered the weblink for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard of the result. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The discard is made to a special browser favorites folder, another designated folder configured ahead of time, or as specified with an additional parameter.

Fig. 70B-1

<p>Operand ↓ 205</p>	<p>PM</p>	<p><u>Preferred embodiment Discard processing</u></p>
<p>Discarding an email causes searching the specified email system with search criteria of the email parameter string. The email parameter string can specify searching any email fields for any values including wildcarding (patterns for matching). Each field is referenced with a predefined name and then associated with a search criteria. For example, the email string of "subj:'personnel'; recip:'george@alltell.com'; body:'reduction in force'" causes searching all emails with a subject containing "personnel" and was sent to "george@alltell.com" and has a message body containing the string "reduction in force". To search for certain email containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:sent,inbox,company;" indicates to only search the email folders of sent, inbox, and company (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of email. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In one embodiment, all occurrences found in history are presented with at least their date/time stamps, subject line, sender and recipient, and perhaps other information, of the email and when it took place, when the ack parameter is set to true for user reconciliation. In another embodiment, the most recent occurrence from searched folders is provided for the discard. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for doing the discard. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The email is discarded from an email folder as specified with a syntax in the email parameter string.</p>		

Fig. 70B-2

	<u>PM</u>	<u>Preferred embodiment Discard processing</u>
<p>Operand ↓ 207</p>	<p>C</p>	<p>Discarding an sms message causes searching the specified messaging system with search criteria of the sms message parameter string. The message parameter string can specify searching any message fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria. For example, the sms message string of "recip:'2144034071@nextel.com,9725397137@lboxsv.com';" causes searching all messages to the sought recipients. To search for certain messaging containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:outgoing" indicates to only search the outgoing folder (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of messages. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In the one embodiment, all occurrences found in history are presented with at least their date/time stamps, message, sender and recipient, and perhaps other information, of the message and when it took place, when the ack parameter is set to true for user reconciliation. In another embodiment, the most recent occurrence from searched folders is provided for discarding. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The message is discarded from a folder as specified with a syntax in the sms message parameter string.</p>

Fig. 70B-3

	<u>PM</u>	<u>Preferred embodiment Discard processing</u>
<p>Operand ↓ 209</p>	C	<p>Discarding a broadcast email causes searching the specified email system with search criteria of the email parameter string. The email parameter string can specify searching any email fields for any values including wildcarding (patterns for matching). Each field is referenced with a predefined name and then associated with a search criteria. For example, the email string of "subj:'personnel'; recip:'george@alltell.com'; body:'reduction in force'" causes searching all emails with a subject containing "personnel" and was sent to "george@alltell.com" and has a message body containing the string "reduction in force". To search for certain email containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:sent,inbox,company;" indicates to only search the email folders of sent, inbox, and company (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of email. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In one embodiment, all occurrences found in history are presented with at least their date/time stamps, subject line, sender and recipient, and perhaps other information, of the email and when it took place, when the ack parameter is set to true for user reconciliation. In a preferred embodiment, the most recent occurrence from searched folders is provided for the discard. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for doing the discard. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The email s discarded from a folder as specified with a syntax in the email parameter string.</p>

Fig. 70B-4

Operand	<u>PM</u>	<u>Preferred embodiment Discard processing</u>
211	C	<p>Discarding a broadcast sms msg causes searching the specified messaging system with search criteria of the sms message parameter string. The message parameter string can specify searching any message fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria. For example, the sms message string of "recip:'2144034071@nextel.com,9725397137@bxsrv.com"; causes searching all messages to the sought recipients. To search for certain messaging containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:outgoing" indicates to only search the outgoing folder (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of messages. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In the one embodiment, all occurrences found in history are presented with at least their date/time stamps, message sender and recipient, and perhaps other information, of the message and when it took place, when the ack parameter is set to true for user reconciliation. In a preferred embodiment, the most recent occurrence from searched folders is provided for discard. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discard. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The message is discarded from a folder as specified with a syntax in the sms message parameter string.</p>
213	O	<p>Discarding an indicator searches appropriate specified system storage for the indicator (e.g. storage/memory used for indicators by other commands). The indicator parameter string specifies the indicator string being sought and wildcarding is supported. In one embodiment, appropriate MS storage/memory which contains the history of indicators sent to the source system is searched and all occurrences found in history are presented with at least their date/time stamps, the indicator, and perhaps other information, for user reconciliation. In one embodiment, the most recently delivered indicator is identified and used for the discard. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for the discard. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The discard is performed so that the target system(s) have the indicator(s) removed from the interface if currently presented and removed from history maintained for the user interface object presentation (preferably not from LBX history). An additional parameter may specify how to delete the indicator.</p>

Fig. 70B-5

		<u>Preferred embodiment Discard processing</u>
<u>Operand</u>	<u>PM</u>	
215	C	Discarding an application causes searching the specified system for the application (and with the params parameter(s) if specified to get the right invocation of the application). The app parameter is preferably an executable name. Providing a partial or full path to the application parameter will limit the search result. The app parameter string preferably supports wildcarding. In one embodiment, all occurrences found at the source and their paths are presented to the user with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In a preferred embodiment, the most recently executed instance of the matching application is determined for the discard. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provide with the result for discard. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The discard does not remove the application from the target system. It terminates the application by killing it at the operating system level. A Discard File operand command can be used to remove it from the system.
217	S	Discarding a document causes searching the specified system for the document. The doc parameter is a document name. The document parameter can be a wildcard (pattern) for matching. Providing a more defined partial or full path to the document name will narrow the search. In one embodiment, all occurrences found and their paths are presented to the user with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In a preferred embodiment, the most recently accessed search result is provided for the discard. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Discard of the document removes it from each target system at a special shared folder, or configured folder for sharing, or as specified with a new parameter to discard processing.
219	S	Discarding a file causes searching the source path for the file. The path parameter is a file name. Providing a more defined partial or full path to the file will narrow the search result. The path parameter can be a wildcard (pattern) for matching. In one embodiment, all occurrences found on the MS and their paths are provided with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In another embodiment, the most recently accessed file meeting the search criteria is discarded. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provide with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. File(s) are discarded at each target system.

Fig. 70B-6

Operand	PM	Preferred embodiment Discard processing
221	O	Discarding content causes searching the specified system for the content. The content parameter is a reference to the content. The content parameter can be a wildcard (pattern) for matching. In a preferred embodiment, the most recently accessed search result is provided for the discard. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Discarding the content removes it from each target system at a special shared destination, or configured destination for sharing, or as specified with a new parameter to discard processing.
223	C	Discarding a DB object causes searching the specified system for the database object value. The database object parameter is provided with a variety of syntaxes depending on the type of database object sought. For example, the DB-obj parameters is "T:tablename" to seek a table, "S:schemaname" to seek a particular schema, "C:columnname" to seek a particular column name, "D:DBname" to seek a particular DB name, "R:rolename" to seek a particular role set, "P:procname" to search for particular stored procedure, etc. There are unique syntaxes for every type of DB object being sought. The search criteria can be a wildcard (pattern) for matching. In one embodiment, all occurrences found on the source system and information about the occurrence is presented to the user for reconciliation. In a preferred embodiment, the best (e.g. most recently accessed) fit database object is identified for discard. The search takes place as though the user manually launched the search, entered the criteria or query for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The DB object is discarded (removed, deleted, dropped).
225	O	Same as Compose processing except modifies the value to 0 at each system.
227	O	Same as Compose processing except modifies the value to clear at each system.

Fig. 70B-7

		Preferred embodiment Discard processing
Operand	PM	
229	S	Discarding a directory causes searching the specified system for the directory. The path parameter is a directory name. Providing a more defined partial or full path to the directory parameter will narrow the search result. The path parameter can be a wildcard (pattern) for matching. In one embodiment, all occurrences found on the MS and their paths are provided with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In one embodiment, the most recently accessed directory meeting the search criteria is discarded. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The directory is discarded at each target system.
231	C	Operand 215 and 235 (application object) is treated identically to Operand 231 (application context) this LBX release (same params currently). The specified application is terminated, not removed.
233	S	Discarding a user interface object causes closing/terminating the focused object(s) at each specified system that contains the object parameter criteria in the titlebar. In a preferred embodiment, there is a unique syntax for which places of user interface objects that are currently active are to be search (e.g. title bar, entry fields, radio button options, window text, combinations thereof, etc). The search criteria can be a wildcard (pattern) for matching. The search takes place as though the user manually launched the search, entered the criteria for the search, and the object(s) are closed/terminated. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use.
235	C	Operand 215 and 231 (application object) is treated identically to Operand 235 (application context) this LBX release (same params currently). The specified application is terminated, not removed.
237	O	Discarding input causes reinitializing the iodev parameter input device stream of the specified system, so that any pending state is discarded. In one embodiment, a special input datastream is issued to reinitialize the I/O path. In another embodiment, the I/O path is terminated and restarted to reinitialize for the attached device(s). In another embodiment, the I/O path is flushed and then reinitialized. In another embodiment, an additional parameter indicates how to discard the iodev device stream. The iodev parameter specifies which Input/Output device to reinitialize. Preferably, the discard command data is maintained to LBX History, a log, or other useful storage for subsequent use.

Fig. 70B-8

	<u>PM</u>	<u>Preferred embodiment Discard processing</u>
<p><i>Operand</i></p> <p>239</p>	<p>O</p>	<p>Discarding output causes reinitializing the iodev parameter output device stream of the specified system, so that any pending state is discarded. In one embodiment, a special output datastream is issued to reinitialize the IO path. In another embodiment, the IO path is terminated and restarted to reinitialize for the attached device(s). In another embodiment, the I/O path is flushed and then reinitialized. In another embodiment, an additional parameter indicates how to discard the iodev device stream. The iodev parameter specifies which Input/Output device to reinitialize. Preferably, the discard command data is maintained to LBX History, a log, or other useful storage for subsequent use.</p>
<p>241</p>	<p>C</p>	<p>Discarding an alert causes searching the specified system for the alert and discarding it. The alert parameter is the same parameter used to generate an alert (e.g. using another command), and can be wildcarded. In one embodiment, all occurrences found which is associated to the alert application in use at the MS, and which is used for other commands disclosed, are provided to the user for reconciliation with at least their date/time stamps, and perhaps other information. In one embodiment, the most recently generated alert matching the alert search criteria is used for discarding. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use.</p>
<p>243</p>	<p>O</p>	<p>Discarding a process causes searching for and terminating (killing) all process names running at the specified system (e.g. MS) which contain the pname string parameter (e.g. in UNIX: "ps -ef grep pname"). In one embodiment, all occurrences found running at the MS are presented with interesting programmatic information such as when started, its size, etc for user reconciliation. In the preferred embodiment, one process running in the specified system is to be found (i.e. >1 = ambiguous). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, historical log, or other useful storage for subsequent use. The process is killed (terminated). It is not removed from the system.</p>

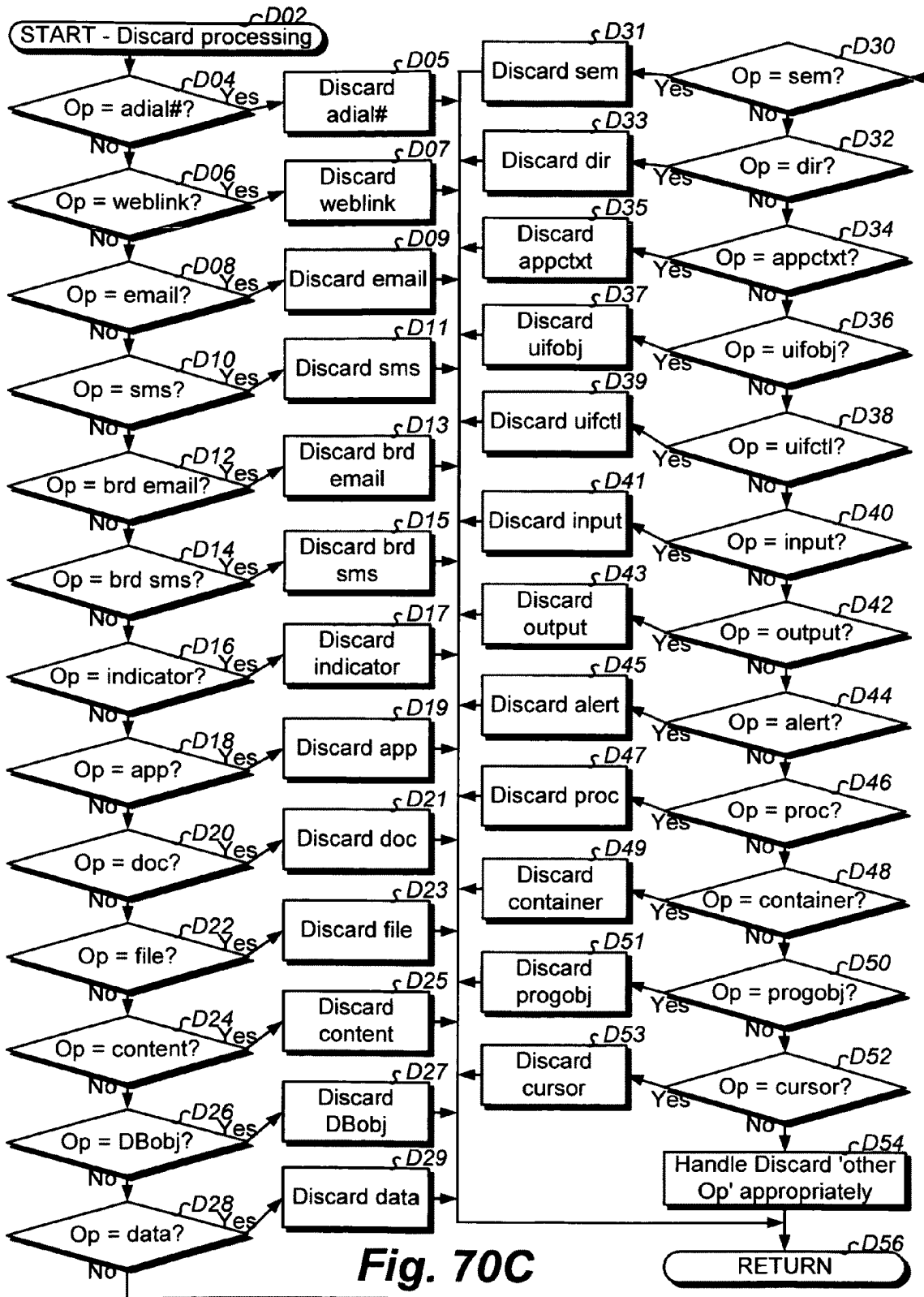
Fig. 70B-9

		Preferred embodiment Discard processing
Operand	PM	
245	S	Discarding a container causes searching the specified system for the container. The container parameter can be well defined to narrow the search result, and may be wildcarded (pattern) for matching. In one embodiment, all occurrences found on the MS and their references/handles are provided with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In one embodiment, the most recently accessed container meeting the search criteria is discarded at the target systems. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The container is discarded from each specified system.
247	O	Discarding a program object causes searching the specified system for the program object and initializing to a initial value i.e. discarding any current value). In the preferred embodiment, a unique syntax is used for which type of program object is being sought (similar to above). There can be unique syntaxes for specifying which type of program object is being sought (e.g. "S:dataname"). In one embodiment, all occurrences found on the MS and information about the occurrence including its current value is presented to the user for reconciliation. In a preferred embodiment, one program object is to be found (e.g. >1 = ambiguous). In one embodiment, a well known location of link symbol information files are consulted, and in another embodiment a new parameter specifies where to look, or which symbol file of information to use. The search criteria can be a wildcard (pattern) for matching. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. A reasonable reset value (e.g. 0 for data, clear for semaphore) is set to the program object. Program objects which cannot hold a value (procedure) are preferably not affected by the discard command. Local and remote processing must have programmatic visibility to affected program object(s).
249	O	Discarding a cursor causes resetting the cursor at the specified system. In the preferred embodiment, provided in the search results is the current cursor information (e.g. image, animation, etc) when user reconciliation is involved. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the discard operation. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In one embodiment, a system defaulted cursor is set. In another embodiment, the previous cursor setting is returned to, and multiple discard cursor actions can change the cursor continuously to historical settings. An additional parameter may provide how to discard the cursor.

Fig. 70B-10

Operand	PM	<u>Preferred embodiment Discard processing</u>
251	C	<p>Discarding a calendar object causes searching the specified calendar system with search criteria of the calendar object parameter string. The calendar object parameter string can specify searching any calendar entry fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria (similar to email above). Those skilled in the art recognize many useful syntaxes for searching any characteristics of calendar objects. In one embodiment, all occurrences found in history are presented to the user for reconciliation with at least their date/time stamps, sender and recipient, and perhaps other information, of the calendar object and when it was scheduled. In a preferred embodiment, the most recent occurrence from the calendaring system is discarded. Wildcarding (pattern matching) is preferably inherent by searching for substrings. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The calendar entry(s) are discarded from the target system calendaring system.</p>
253	C	<p>Discarding an address book (AB) object causes searching the specified AB system with search criteria of the AB object parameter string. The AB object parameter string can specify searching any AB entry fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria (similar to email above). Those skilled in the art recognize many useful syntaxes for searching any characteristics of AB objects/entries. In one embodiment, all occurrences found are presented to the user for reconciliation with appropriate AB information. In a preferred embodiment, the most recent occurrence from the AB system is discarded. Wildcarding (pattern matching) is preferably inherent by searching for substrings. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for discarding it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and discard. Preferably, the discard command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The AB entry(s) are discarded from the target system AB system.</p>
...		

Fig. 70B-11



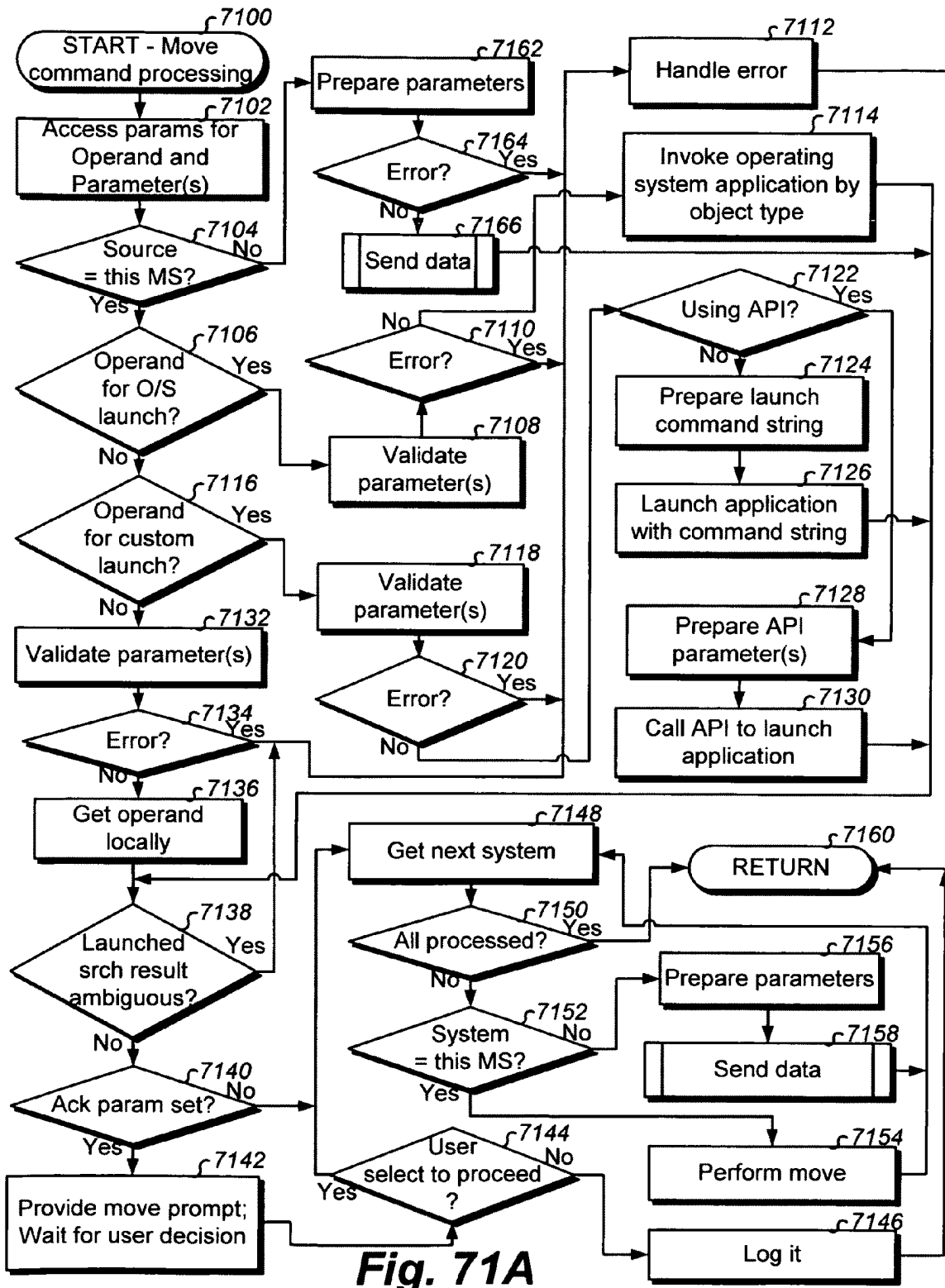


Fig. 71A

Preferred embodiment Move processing	
<p>Operand ↓ 201</p>	<p>PM C</p> <p>Moving an auto-dial # launches a phone number log interface with the auto-dial # parameter (can be wildcard) for searching the source system. Preferably, both the outgoing and incoming logs are searched. In an alternate embodiment, the log is specified with a parameter. In the preferred embodiment, the most recent occurrence from a particular log is provided. In another embodiment, all occurrences found in history are presented with their date/time stamps, and perhaps other information, of the call and when it took place (e.g. when the ack parameter is set) and the user browses the results prior to accepting the move. The search takes place as though the user manually launched the search, entered the auto-dial # for the search, and then was provided with result(s) for the move. Preferably, the move shall take place if there are no ambiguities (e.g. more than one phone number returned per search criteria). An additional parameter may be specified for the target (different log) of the move, otherwise the object is moved to an assumed location (e.g. same folder to more recent position). Appropriate MS storage is updated and subsequently processed as though the user manually performed the search and move of the result. Preferably, the move cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The move is made to system(s) logs, preferably with identifying information of the source and who did the move.</p>
<p>203</p>	<p>C</p> <p>Moving a weblink launches a search to MS browser history with the weblink parameter (can be wildcard) for searching the source system. In one embodiment, all occurrences found in history are presented with their date/time stamps, and perhaps other information, of the link and when it was invoked (e.g. when the ack parameter is specified to true) for presentation to the user prior to doing the move. In the preferred embodiment, the most recent occurrence from a particular invocation is provided for the move. An additional parameter may be specified for the target (specified favorites folder), otherwise the object is moved to an assumed location (e.g. highest level favorites folder). The search takes place as though the user manually launched the search, entered the weblink for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move of the result. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The move is made to a special browser favorites folder, or another designated folder configured ahead of time, or as specified with an additional parameter, preferably with identifying information of the source and who did the move.</p>

Fig. 71B-1

<p>Operand ↓</p>	<p>PM</p>	<p><u>Preferred embodiment Move processing</u></p>
<p>205</p>	<p>C</p>	<p>Moving an email causes searching the source email system with search criteria of the email parameter string. The email parameter string can specify searching any email fields for any values including wildcarding (patterns for matching). Each field is referenced with a predefined name and then associated with a search criteria. For example, the email string of "subj:'personnel'; recip:'george@alltell.com'; body:'reduction in force'" causes searching all emails with a subject containing "personnel" and was sent to "george@alltell.com" and has a message body containing the string "reduction in force". To search for certain email containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:sent,inbox,company;" indicates to only search the email folders of sent, inbox, and company (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of email. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In one embodiment, all occurrences found in history are presented with at least their date/time stamps, subject line, sender and recipient, and perhaps other information, of the email and when it took place, when the ack parameter is set to true for user reconciliation. In a preferred embodiment, the most recent occurrence from searched folders is provided for the move. An additional parameter may be specified for the target (specified folder), otherwise the object is moved to an assumed location (e.g. inbox, drafts, special named folder). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for doing the move. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The move is made to a special email folder of the target system, or another designated folder configured ahead of time, or as specified with a new parameter for move processing, preferably with identifying information of the source and who did the move (if supported in email application).</p>

Fig. 71B-2

<p>Operand ↓ 207</p>	<p>PM C</p>	<p style="text-align: center;"><u>Preferred embodiment Move processing</u></p> <p>Moving an sms message causes searching the source messaging system with search criteria of the sms message parameter string. The message parameter string can specify searching any message fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria. For example, the sms message string of "recip:'2144034071@nextel.com,9725397137@lboxsv.com';" causes searching all messages to the sought recipients. To search for certain messaging containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:outgoing" indicates to only search the outgoing folder (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of messages. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In the one embodiment, all occurrences found in history are presented with at least their date/time stamps, message, sender and recipient, and perhaps other information, of the message and when it took place, when the ack parameter is set to true for user reconciliation. In a preferred embodiment, the most recent occurrence from searched folders is provided for moving. An additional parameter may be specified for the target (specified folder), otherwise the object is moved to an assumed location (e.g. inbox, drafts, special named folder). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The move is made to a special messaging folder of the target system, or another designated folder configured ahead of time, or as specified with a new parameter to move processing, preferably with identifying information of the source and who did the move.</p>
-------------------------------------	-------------------------------	---

Fig. 71B-3

	<u>Preferred embodiment Move processing</u>
<p>Operand ↓ 209</p>	<p>PM</p> <p>C</p> <p>Moving a broadcast email causes searching the source email system with search criteria of the email parameter string. The email parameter string can specify searching any email fields for any values including wildcarding (patterns for matching). Each field is referenced with a predefined name and then associated with a search criteria. For example, the email string of "subj:'personnel'; recip:'george@alltell.com'; body:'reduction in force'" causes searching all emails with a subject containing "personnel" and was sent to "george@alltell.com" and has a message body containing the string "reduction in force". To search for certain email containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:sent,inbox,company," indicates to only search the email folders of sent, inbox, and company (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of email. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In one embodiment, all occurrences found in history are presented with at least their date/time stamps, subject line, sender and recipient, and perhaps other information, of the email and when it took place, when the ack parameter is set to true for user reconciliation. In a preferred embodiment, the most recent occurrence from searched folders is provided for the move. An additional parameter may be specified for the target (specified folder), otherwise the object is moved to an assumed location (e.g. inbox, drafts, special named folder). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for doing the move. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The move is made to a special email folder of the target system, or another designated folder configured ahead of time, or as specified with a new parameter to move processing, preferably with identifying information of the source and who did the move.</p>

Fig. 71B-4

Operand	PM	Preferred embodiment Move processing
211	C	<p>Moving a broadcast sms msg causes searching the source messaging system with search criteria of the sms message parameter string. The message parameter string can specify searching any message fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria. For example, the sms message string of "recip:2144034071@nextel.com,9725397137@lboxsv.com"; causes searching all messages to the sought recipients. To search for certain messaging containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:outgoing" indicates to only search the outgoing folder (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of messages. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In the one embodiment, all occurrences found in history are presented with at least their date/time stamps, message, sender and recipient, and perhaps other information, of the message and when it took place, when the ack parameter is set to true for user reconciliation. In a preferred embodiment, the most recent occurrence from searched folders is provided for moving. An additional parameter may be specified for the target (specified folder), otherwise the object is moved to an assumed location (e.g. inbox, drafts, special named folder). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The move is made to a special messaging folder of the target system, or another designated folder configured ahead of time, or as specified with a new parameter to move processing, preferably with identifying information of the source and who did the move.</p>
213	C	<p>Moving an indicator searches appropriate source storage for the indicator (e.g. storage/memory used for indicators by other commands). The indicator parameter string specifies the indicator string being sought and wildcarding is supported. In one embodiment, appropriate MS storage/memory which contains the history of indicators sent to the source system is searched and all occurrences found in history are presented with at least their date/time stamps, the indicator, and perhaps other information, for user reconciliation. In a preferred embodiment, the most recently delivered indicator is identified and used for the move. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for the move. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The move is made so that the target system(s) are delivered the indicators like delivering new indicator(s) for presentation.</p>

Fig. 71B-5

		<u>Preferred embodiment Move processing</u>
Operand ↓ 215	<u>PM</u> C	<p>Moving an application causes searching the source system for the application (and with the params parameter(s) if specified to get the param specified invocation of the application). The app parameter is preferably an executable name, and may contain parameters that were passed. Providing a more defined partial or full path to the application parameter will limit the search result. The app parameter string preferably supports wildcarding. In one embodiment, all occurrences found at the source and their paths are presented to the user with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In a preferred embodiment, the most recently executed instance of the matching application is determined for the move. In one embodiment, the application itself is moved to the target systems, perhaps as directed by an additional parameter (e.g. directory location). In another embodiment, the executable path to run the application is moved to execution history at the system(s) so that a user can run it, albeit from a remote system (assumption that application available for running there already). In another embodiment, the executable(s) are roved to the target system using methodologies of U.S. Patent 5,938,722 ("Method of executing programs in a network", Johnson). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for move. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use.</p>
217	C	<p>Moving a document causes searching the source system for the document. The doc parameter is a document name. The document parameter can be a wildcard (pattern) for matching. Providing a more defined partial or full path to the document name will narrow the search. In one embodiment, all occurrences found on the MS and their paths are presented to the user with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In a preferred embodiment, the most recently accessed search result is provided for the move. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Moving the document places the document to each target system at a special shared folder, or configured folder for sharing, or as specified with a new destination parameter to move processing.</p>

Fig. 71B-6

	PM	Preferred embodiment Move processing
<p>Operand ↓ 219</p>	<p>C</p>	<p>Moving a file causes searching the source path for the file. The path parameter is a file name. Providing a more defined partial or full path to the file will narrow the search result. The path parameter can be a wildcard (pattern) for matching. In one embodiment, all occurrences found on the MS and their paths are provided with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In a preferred embodiment, the most recently accessed file meeting the search criteria is moved to the target systems. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provide with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In one embodiment, the move results in overwriting an existing file with the same handle (e.g. name). In another embodiment, the move may result in writing a newly altered name of the file when there is an existing file with the same handle (e.g. name). An additional parameter may be specified for the target (specified folder), otherwise the object is moved to an assumed location.</p>
<p>221</p>	<p>O</p>	<p>Moving content causes searching the source for the content. The content parameter is a reference to the content. The content parameter can be a wildcard (pattern) for matching. In a preferred embodiment, the most recently accessed search result is provided for the move. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Moving the content places the content to each target system at a special shared destination, or configured destination for sharing, or as specified with a new parameter to move processing.</p>

Fig. 71B-7

Operand	PM	<u>Preferred embodiment Move processing</u>
223	O	<p>Moving a DB object causes searching the source for the database object value. The database object parameter is provided with a variety of syntaxes depending on the type of database object sought. For example, the DB-obj parameters is "T:tablename" to seek a table, "S:schemaname" to seek a particular schema, "C:columnname" to seek a particular column name, "D:DBname" to seek a particular DB name, "R:rolename" to seek a particular role set, "P:procname" to search for particular stored procedure, etc. There are unique syntaxes for every type of DB object being sought which maps to an appropriate SQL system tables query. The search criteria can be a wildcard (pattern) for matching. In one embodiment, all occurrences found on the source system and information about the occurrence is presented to the user for reconciliation. In other embodiments, the best (e.g. most recently accessed) fit database object is identified for use in the move, or a new parameter indicates how to search. The search takes place as though the user manually launched the search, entered the criteria or query for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The value of the DB object is moved to the value of the DB object with the same name and type at the destination system(s). If not found at a target system, then no action is performed at that system. Moving a database object moves the value to the same database object(s) at other system(s), or creates new ones when there is not match. Value(s) are overwritten.</p>

Fig. 71B-8

Preferred embodiment Move processing	
<p>Operand</p> <p>225</p>	<p>PM</p> <p>O</p> <p>Moving data causes searching the source system for the data. In the preferred embodiment, the data is a global system variable visible to all processes of a MS operating system. In other embodiments, the data may have limited scope which is made accessible to present disclosure processing (e.g. with extern). Depending on the embodiment, data may be that which is contained in a program data segment, stack segment, and/or extra segment. There can be unique syntaxes for specifying which type of data is being sought (e.g. "S:dataname" for data parameter). The search criteria can be a wildcard (pattern) for matching. In the preferred embodiment, a well known location of link symbol information files are consulted, and in another embodiment a new parameter specifies where to look, or which symbol file of information to use. In one embodiment, all occurrences found at the source system and information about the occurrence including its current value is presented to the user for reconciliation. In a preferred embodiment, the best data value (e.g. most recently accessed if more than one matches) is provided for the move. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The value of the data to be moved is copied to the data with the same name and type at the destination system(s). In another embodiment, the source data is reinitialized (e.g. 0), or reinitialized according to a new parameter, as part of the move operation. If not found at a target system, then no action is performed at that system, or an error is provided. Moving a data object at least copies the value to the same data object(s) at other system(s). Value(s) are overwritten.</p>
<p>227</p>	<p>O</p> <p>Moving a semaphore causes reading the current value of the semaphore at the source where the move command action is being executed and then moving the current value to the same semaphore names at the target system(s). The semaphore param can be a wildcard (pattern) for matching. In the preferred embodiment, the semaphore is a global system semaphore visible to all processes of a MS operating system. In other embodiments, the semaphore may have limited scope which is made accessible to present disclosure processing. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The value (set or cleared) of the copied semaphore is copied to the semaphore with the same name and type at the destination system(s). In another embodiment, the source data is reinitialized (e.g. 0), or reinitialized according to a new parameter, as part of the move operation. If not found at a target system, then no action is performed at that system, or an error is provided. Moving a semaphore at least copies the value to the same semaphore at other system(s). Value(s) are changed (clear or set).</p>

Fig. 71B-9

Operand	PM	<u>Preferred embodiment Move processing</u>
229	C	Moving a directory causes searching the source system for the directory. The path parameter is a directory name. Providing a more defined partial or full path to the directory parameter will narrow the search result. The path parameter can be a wildcard (pattern) for matching. In one embodiment, all occurrences found on the MS and their paths are provided with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In a preferred embodiment, the most recently accessed directory meeting the search criteria is copied to the target systems. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In one embodiment, the move results in overwriting an existing directory and files therein. In another embodiment, the move results in writing a newly altered name of directory contents when there is a conflict (e.g. existing entity with same name). In another embodiment, an additional target path parameter is provided for where to place the directory.
231	C	Operand 215 (application object) is treated identically to this Operand 231 (application context) this LBX release (same params currently).
233	C	Moving a focused user interface object causes capturing the currently focused user interface object using the first parameter (e.g. Alt-Prtscrn; can be changed with the param) string syntax for keystroke(s) to capture the image, and then moving the graphics file (file type in various embodiments) to a shared destination, or a configured destination at the target system(s), or as specified with a new parameter. The capture takes place as though the user manually performed the capture action, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the capture and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In one embodiment, the move results in overwriting an existing file with the same handle (e.g. name), which will be seldom since the graphics file name preferably contains a date/time stamp portion. In another embodiment, the move results in writing a newly altered name of the file when there is an existing file with the same handle (e.g. name).
235	C	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter). Also, the ack parameter provides a reconciliation option.
237	C	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter). Also, the ack parameter provides a reconciliation option.

Fig. 71B-10

	PM	Preferred embodiment Move processing
Operand 239	C	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter). Also, the ack parameter provides a reconciliation option.
241	C	Moving an alert causes searching the source system for the alert. The alert parameter is the same parameter used to generate an alert (e.g. using another command), and can be wildcarded. In one embodiment, all occurrences found on the MS which is associated to the alerter application in use at the MS, and which is used for other commands disclosed, are provided to the user for reconciliation with at least their date/time stamps, and perhaps other information. In other embodiments, the most recently generated alert matching the alert search criteria is used for moving, or the search occurs as specified with an additional parameter. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The move is made so that the target system(s) are delivered the alert(s) like delivering a new alert to the systems.
243	C	Moving a process causes first finding all process names running at the source (e.g. MS) which contain the pname string parameter (e.g. in UNIX: "ps -ef grep pname"). In one embodiment, all occurrences found running at the MS are presented with interesting programmatic information such as when started, its size, etc for user reconciliation. In the preferred embodiment, one process running in the source system is to be found (i.e. >1 = ambiguous). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving. Results are useful statistics about the process which is running at the source. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, historical log, or other useful storage for subsequent use. Useful statistic(s) about the process (perhaps which statistics specified with an additional parameter) are copied to an appropriate destination of the target system(s) for informative purposes (e.g. a special log file). In another embodiment, the alerter process and/or indicator methodology can be used as the destination for the move for alerting a user at the target system. In another embodiment, there is new parameter for which end result the move will have (informative destination, handled like alert, handled like indicator). In some embodiments, the process is roved to the target system using methodologies of U.S. Patent 5,938,722 ("Method of executing programs in a network", Johnson), as requested in a new parameter.

Fig. 71B-11

Operand	PM	Preferred embodiment Move processing
245	C	<p>Moving a container causes searching the source system for the container. The container parameter can be well defined to narrow the search result, and may be wildcarded (pattern) for matching. In one embodiment, all occurrences found on the MS and their references/handles are provided with at least their date/time stamps, size, and perhaps attributes information, for user reconciliation. In a preferred embodiment, the most recently accessed container meeting the search criteria is copied to the target systems. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In one embodiment, the move results in overwriting an existing container. In another embodiment, the move results in writing a newly altered reference/handle of the container when there is a conflict (e.g. existing entity with same name). An additional parameter may be specified for the target, otherwise the object is moved to an assumed location.</p>
247	C	<p>Moving a program object causes first searching the source for the program object. In the preferred embodiment, a unique syntax is used for which type of program object is being sought (similar to above). There can be unique syntaxes for specifying which type of program object is being sought (e.g. "S:dataname"). In one embodiment, all occurrences found on the MS and information about the occurrence including its current value is presented to the user for reconciliation. In a preferred embodiment, one program object is to be found (e.g. >1 = ambiguous). In one embodiment, a well known location of link symbol information files are consulted, and in another embodiment a new parameter specifies where to look, or which symbol file of information to use. The search criteria can be a wildcard (pattern) for matching. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Useful statistic(s) about the program object (perhaps which statistics specified with an additional parameter) are moved/copied to an appropriate destination of the target system(s) for informative purposes (e.g. a special log file). In another embodiment, the alterer process and/or indicator methodology can be used as the destination for the move for alerting a user at the target system. In another embodiment, there is new parameter for which end result the move will have (informative destination, handled like alert, handled like indicator). An alternate embodiment works like Operand 223 wherein moving is intended to keep program object(s) between systems in synch, albeit with a discard from the source system. Such embodiments require source and target system processing to have access to the object(s) (this may limit participating object(s)).</p>

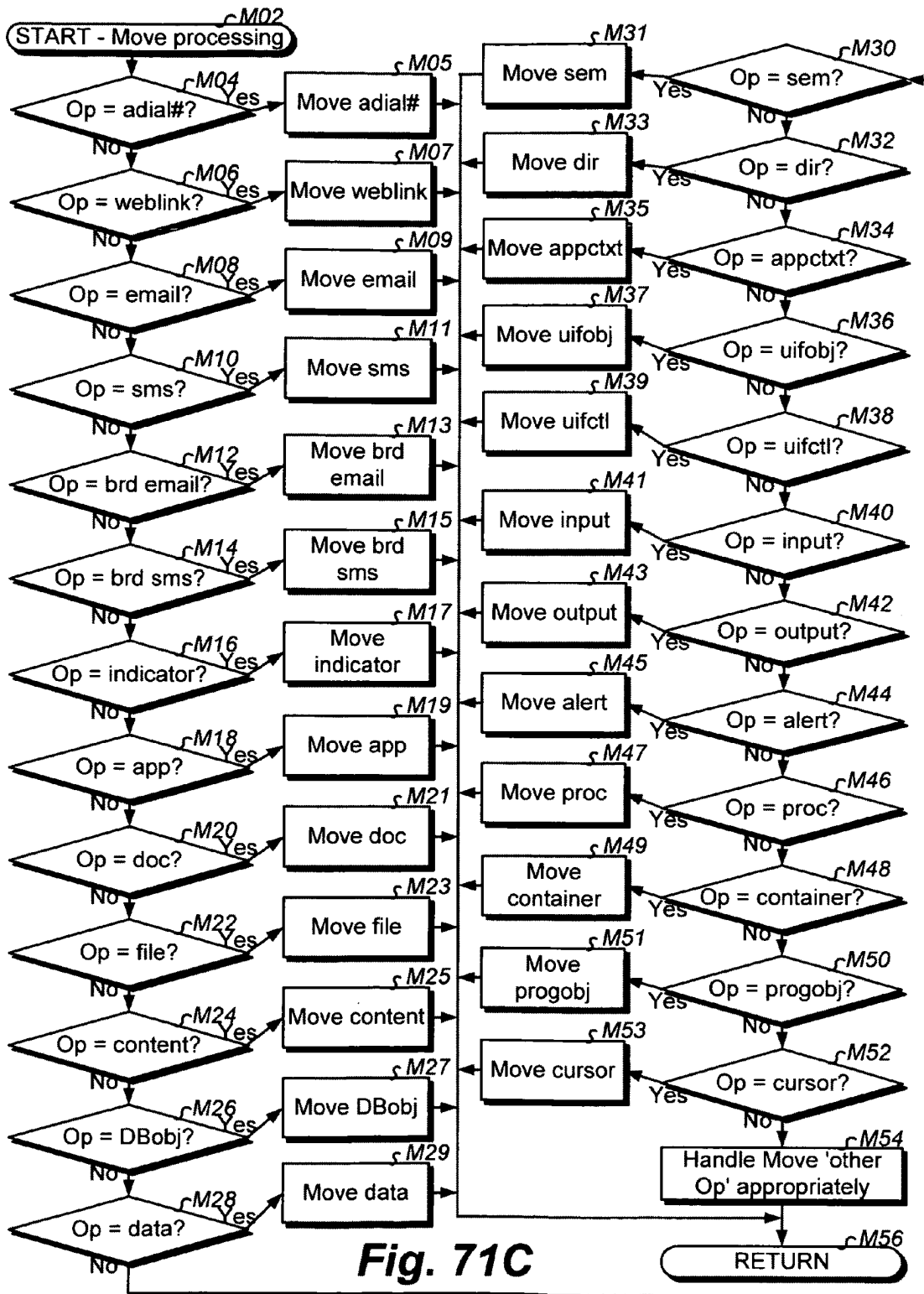
Fig. 71B-12

	<u>PM</u>	<u>Preferred embodiment Move processing</u>
<p>Operand ↓ 249</p>	<p>C</p>	<p>Moving a cursor causes searching the source system for the current cursor setting(s). In the preferred embodiment, provided in the search results is the current cursor information (e.g. image, animation, etc). The current cursor information of the source is then used to alter the cursor at the target system(s). Appropriate MS storage is updated and subsequently processed as though the user had manually performed the move operation. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In some embodiments, the source cursor is reset to a different (e.g. initialized) setting as resulting from the move.</p>
<p>251</p>	<p>C</p>	<p>Moving a calendar object causes searching the source calendar system with search criteria of the calendar object parameter string. The calendar object parameter string can specify searching any calendar entry fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria (similar to email above). Those skilled in the art recognize many useful syntaxes for searching any characteristics of calendar objects. In one embodiment, all occurrences found in history are presented to the user for reconciliation with at least their date/time stamps, attendees, and perhaps other information, of the calendar object and when it was scheduled. In a preferred embodiment, the most recent occurrence from the calendaring system is provided for the move. Wildcarding (pattern matching) is preferably inherent by searching for substrings. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The calendar entry is moved in its entirety to the target system calendaring system. In another embodiment, a new parameter is specified to move the calendar item(s) to a new schedule or time. A duplicate calendar entry may be created if one already exists.</p>

Fig. 71B-13

		<u>Preferred embodiment Move processing</u>
Operand ↓ 253	PM C	<p>Moving an address book (AB) object causes searching the source AB system with search criteria of the AB object parameter string. The AB object parameter string can specify searching any AB entry fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria (similar to email above). Those skilled in the art recognize many useful syntaxes for searching any characteristics of AB objects/entries. In one embodiment, all occurrences found are presented to the user for reconciliation with appropriate AB information. In a preferred embodiment, the most recent occurrence from the AB system is provided for the move. Wildcarding (pattern matching) is preferably inherent by searching for substrings. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was provided with the result for moving it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and move. Preferably, the move command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The AB entry is moved in its entirety to the target system AB system. In another embodiment, a new parameter is specified to move the AB item(s) to a special destination. A duplicate AB entry may be created if one already exists.</p>
	...	

Fig. 71B-14



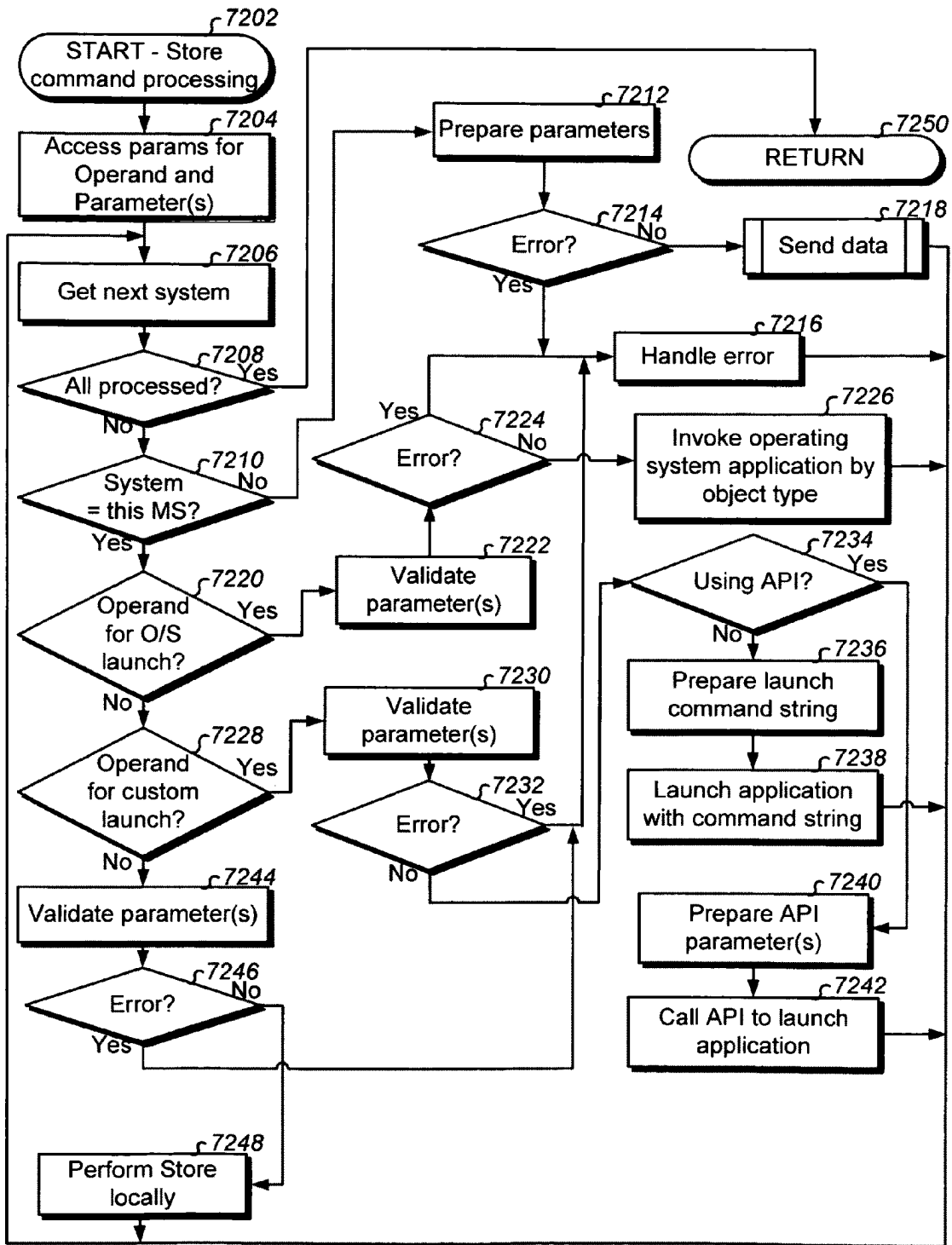


Fig. 72A

	PM	Preferred embodiment Store processing
Operand ↓ 201	C	Storing an auto-dial # stores the auto-dial # parameter to the system(s). Preferably, a certain log is used to store the auto-dial #, or an additional parameter can be provided for where to store the auto-dial # to. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored auto-dial #. An additional parameter may be specified for how/where exactly to store it (e.g. which log).
203	S	Storing a weblink stores the weblink parameter to the system(s). Preferably, a certain link folder is used to store it, or an additional parameter can be provided for where to store it. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored weblink. An additional parameter may be specified for how/where exactly to store it (e.g. which folder).
205	C	Storing an email causes storing the email object to the system(s)' email system. In one embodiment, the email parameter string is a string containing a syntax for defining an email item and used to create the email to a certain folder, configured folder, or as specified with an additional parameter. Each email field can be defined with values, and the store command may result in defaulting fields which are not specified in the email parameter. In another embodiment, the email parameter points to a file containing a syntax for creating the email object. Those skilled in the art recognize many useful syntaxes for setting data in a new email object. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored email.

Fig. 72B-1

		<u>Preferred embodiment Store processing</u>
Operand ↓	PM	
207	C	<p>Storing an sms message causes storing the sms message object to the system(s)' messaging system. In one embodiment, the sms message parameter string is a string containing a syntax for defining an sms message item and used to create the sms message to a certain folder, configured folder, or as specified with an additional parameter. Each sms message field can be defined with values, and the store command may result in defaulting fields which are not specified in the sms message parameter. In another embodiment, the sms message parameter points (e.g. path) to a file containing a syntax for creating the sms message object. Those skilled in the art recognize many useful syntaxes for setting data in a new sms message object. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored sms message.</p>
209	C	<p>Storing a broadcast email causes storing the email object to the system(s)' email system. In one embodiment, the email parameter string is a string containing a syntax for defining an email item and used to create the email to a certain folder, configured folder, or as specified with an additional parameter. Each email field can be defined with values, and the store command may result in defaulting fields which are not specified in the email parameter. In another embodiment, the email parameter points (e.g. path) to a file containing a syntax for creating the email object. Those skilled in the art recognize many useful syntaxes for setting data in a new email object. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored email.</p>

Fig. 72B-2

		<u>Preferred embodiment Store processing</u>
<u>Operand</u> ↓	<u>PM</u>	
211	C	Storing a broadcast sms message causes storing the sms message object to the system(s) messaging system. In one embodiment, the sms message parameter string is a string containing a syntax for defining an sms message item and used to create the sms message to a certain folder, configured folder, or as specified with an additional parameter. Each sms message field can be defined with values, and the store command may result in defaulting fields which are not specified in the sms message parameter. In another embodiment, the sms message parameter points to a file (e.g. path) containing a syntax for creating the sms message object. Those skilled in the art recognize many useful syntaxes for setting data in a new sms message object. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored sms message..
213	O	See Invoke Command for identical processing.
215	C	See Invoke Command for identical processing.
217	S	Storing a document stores the document parameter to the system(s). The document may be a self contained object parameter, or pointer (e.g. path) to a file defining the document. Preferably, a certain folder is used to store it, or an additional parameter can be provided for where to store it. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored document.
219	S	Storing a file stores the file from the path parameter to the system(s). Preferably, a certain folder is used to store it, or an additional parameter can be provided for where to store it. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored file.

Fig. 72B-3

	PM	Preferred embodiment Store processing
Operand 221	O	Storing content stores the content parameter to the system(s). The content may be a self contained object parameter, or pointer (e.g. path) to a file defining the content. Preferably, a certain destination is used to store it, or an additional parameter can be provided for where to store it. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored content.
223	C	See Invoke Command for identical processing.
225	O	See Invoke Command for identical processing.
227	O	See Invoke Command for identical processing.
229	S	Storing a directory stores the directory from the path parameter to the system(s). Preferably, a certain folder is used to store it, or an additional parameter can be provided for where to store it. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored directory.
231	C	See Invoke Command for identical processing.
233	S	Storing a focused user interface object causes a snapshot to be taken of the currently focused user interface object (to .jpg, .gif, alternate embodiments, etc) at the MS and then the snapshot file is stored as though the user manually captured the focused user interface object (e.g. Alt-Prtscm) and saved it. Preferably, a certain folder is used to store it, or an additional parameter can be provided for where to store it. The first parameter command syntax can be defaulted or changed. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored snapshot.
235	O	See Invoke Command for identical processing.
237	O	See Invoke Command for identical processing.
239	O	See Invoke Command for identical processing.

Fig. 72B-4

<i>Operand</i> ↓	PM	Preferred embodiment Store processing
241	S	Storing an alert causes storing the alert to the specified system(s). The alert parameter is the same parameter used to generate an alert (e.g. using another command). Storing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search and store. Preferably, the store command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. The store is performed so that the target system(s) are delivered the alert(s) like delivering a new alert to the systems.
243	O	Storing a process causes sending an operating system signal (see UNIX signaling) to the process name (after determining the Process ID (PID) of the pname parameter). A numeric value parameter (e.g. 0 or 1) may be communicated with the signal. An error is logged if the process is not found for signaling. Preferably, the store command data is maintained to LBX History, a log, or other useful storage for subsequent use.
245	S	Storing a container stores the container parameter to the system(s). The container may be a self contained object parameter, or pointer (e.g. path) to a file defining the container. Preferably, a certain destination is used to store it, or an additional parameter can be provided for where to store it. Store processing takes place as though the user manually launched it. Appropriate MS storage is updated and subsequently processed as though the user manually performed the store command. Preferably, the store cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. In a preferred embodiment, information about who, when, why is additionally maintained with the stored container.
247	O	See Invoke Command for identical processing.
249	O	See Invoke Command for identical processing.
251	C	See Invoke Command for identical processing.
253	C	See Invoke Command for identical processing.
...		

Fig. 72B-5

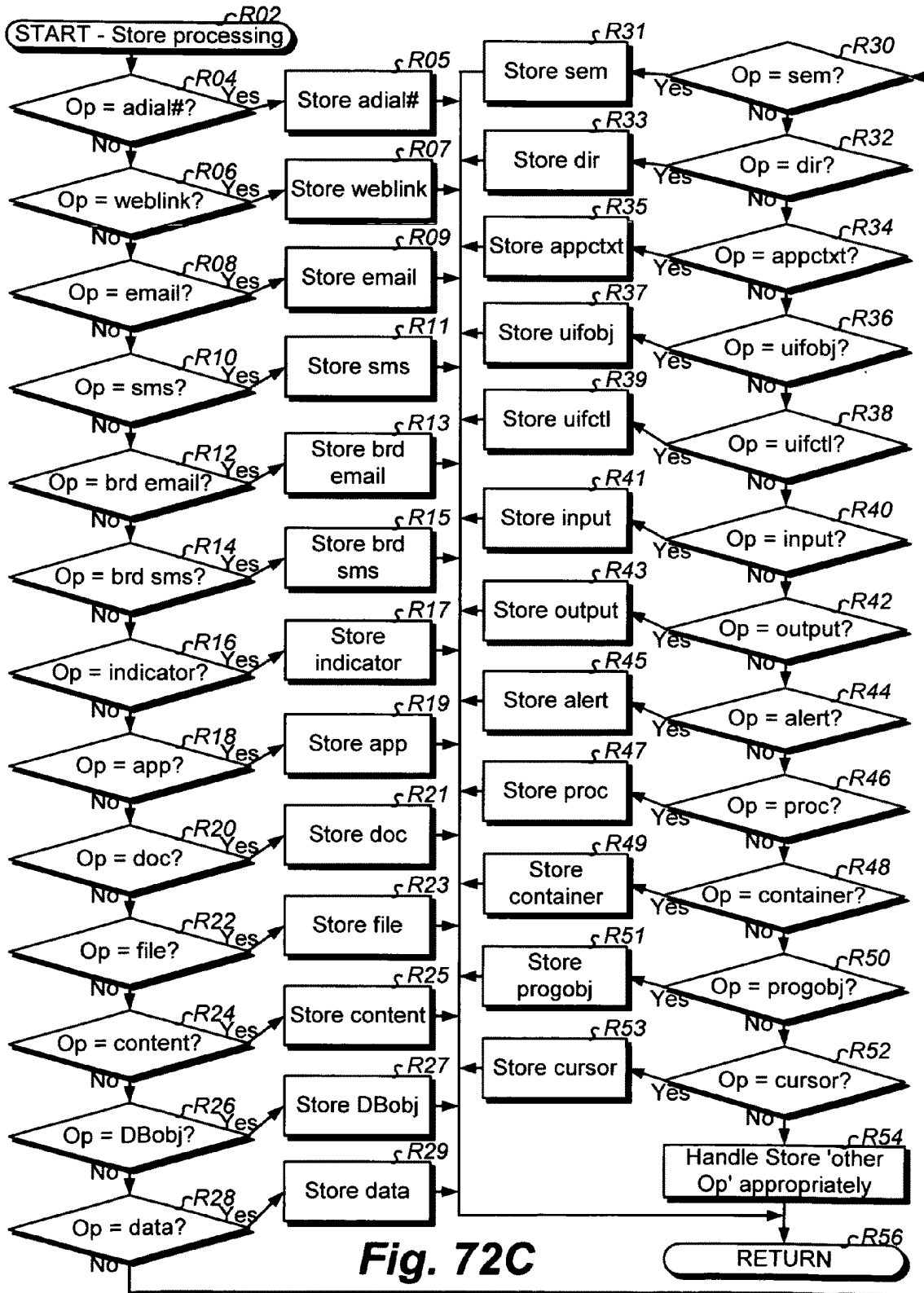


Fig. 72C

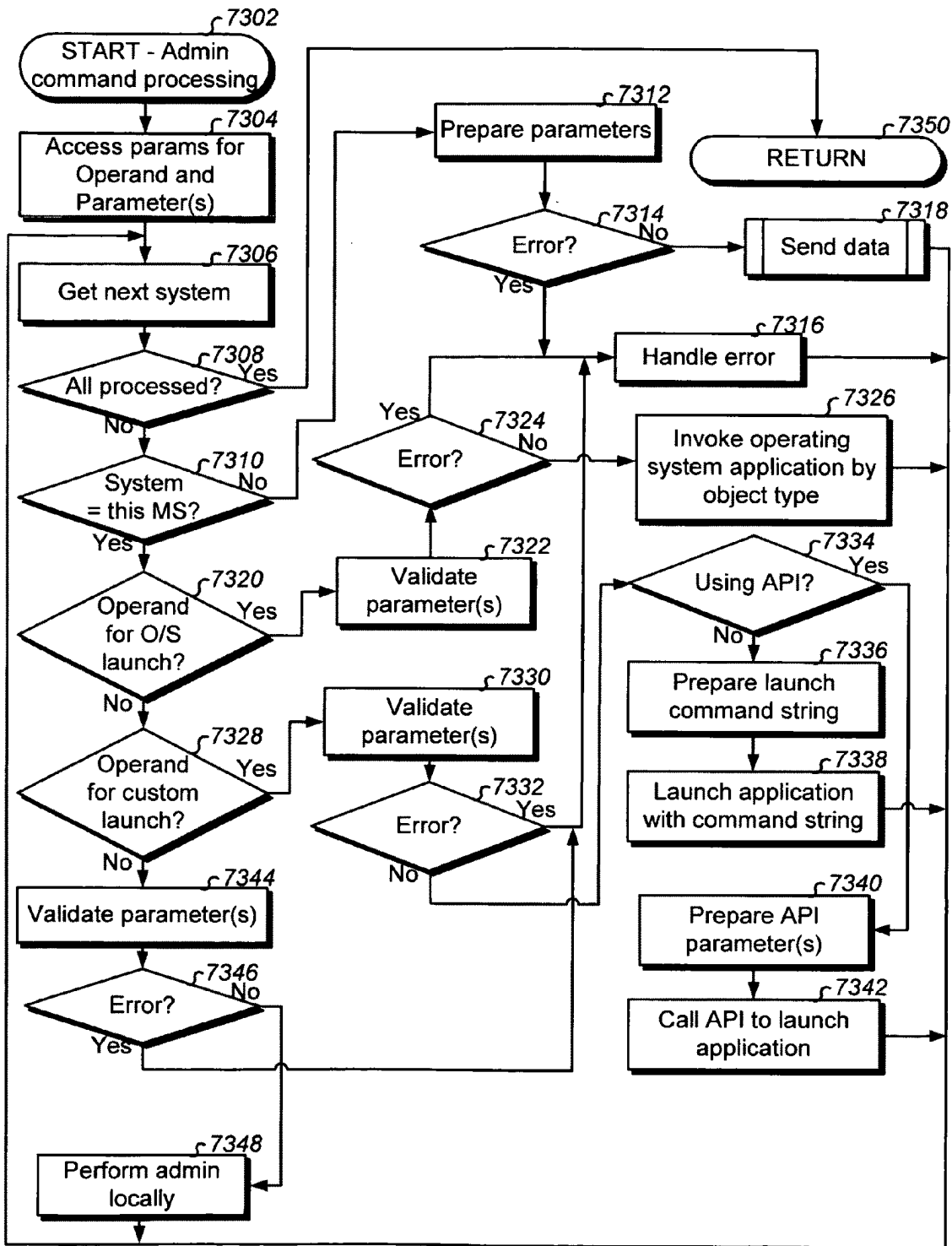


Fig. 73A

		<u>Preferred embodiment Administrate processing</u>
<u>Operand</u> ↓	PM	
201	C	<p>Administering an auto-dial # launches a MS phone number log interface with the auto-dial # parameter for searching. Preferably, both the outgoing and incoming logs are searched. The auto-dial # parameter can be a wildcard (pattern) for matching. In one embodiment, all occurrences found in history are presented with their date/time stamps, log found in, and perhaps other information, of the call and when it took place. In another embodiment, the most recent occurrence from a particular log is presented, and perhaps in an interface which enables calling the # with a minimal user action. The search takes place as though the user manually launched the search, entered the auto-dial # for the search, and then was presented with the result(s) in an appropriate administration interface. Appropriate MS storage is updated and subsequently processed as though the user manually performed the search. Preferably, the administration cmd data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Subsequent administration includes modifying the log entry(s) as desired (e.g. add a 1 prefix since caller id may not have maintained one when it is needed for auto-dial). A new parameter can be specified for which log(s) to search.</p>
203	S	<p>Administering a weblink launches a search to MS browser history with the weblink parameter (and with the params parameter if specified) for searching. The weblink parameter can be a wildcard (pattern) for matching. In one embodiment, all occurrences found in history are presented with their date/time stamps, folder found in, and perhaps other information, of the link and when it was invoked. In another embodiment, the most recent occurrence from a particular invocation is presented, and perhaps in an interface which enables invoking (transposing to) the weblink with a minimal user action. The search takes place as though the user manually launched the search, entered the weblink for the search, and then was presented with the result(s) in an appropriate administration interface. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search. Preferably, the administration command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Subsequent administration includes modifying the weblink(s) as desired (e.g. change description).</p>

Fig. 73B-1

<p>Operand ↓</p>	<p>PM</p>	<p><u>Preferred embodiment Administrative processing</u></p>
<p>205</p>	<p>C</p>	<p>Administering an email causes searching a MS email system with search criteria of the email parameter string. The email parameter string can specify searching any email fields for any values including wildcarding (patterns for matching). Each field is referenced with a predefined name and then associated with a search criteria. For example, the email string of "subj:'personnel'; recip:'george@alltell.com'; body:'reduction in force'" causes searching all emails with a subject containing "personnel" and was sent to "george@alltell.com" and has a message body containing the string "reduction in force". To search for certain email containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:sent;inbox,company;" indicates to only search the email folders of sent, inbox, and company (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of email. Wildcarding (pattern matching) is preferably inherent by searching for substrings. All occurrences found in history are presented with at least their date/time stamps, subject line, sender and recipient, and perhaps other information, of the email and when it took place. In another embodiment, the most recent occurrence from searched folders is presented, and perhaps in an interface which enables appropriate MS email system processing from that point forward (e.g. when processed at local MS). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was presented with the result(s) in an appropriate administration interface. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search. Preferably, the administration command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Subsequent administration includes modifying the email(s) as desired, and perhaps having an option to send/resend.</p>

Fig. 73B-2

Operand ↓	207	PM	<p align="center"><u>Preferred embodiment Administrate processing</u></p> <p>Administering an sms message causes searching a MS sms messaging system with search criteria of the sms message parameter string. The message parameter string can specify searching any message fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria. For example, the sms message string of "recip:'2144034071@nextel.com,9725397137@lboxsv.com':" causes searching all messages to the sought recipients. To search for certain messaging containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:outgoing" indicates to only search the outgoing folder (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of messages. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In the preferred embodiment, all occurrences found in history are presented with at least their date/time stamps, message, sender and recipient, and perhaps other information, of the message and when it took place. In another embodiment, the most recent occurrence from searched folders is presented, and perhaps in an interface which enables appropriate MS messaging system processing from that point forward (e.g. when processed at local MS). The search takes place as though the user manually launched the search, entered the criteria for the search, and then updated with the result(s) in an appropriate administration interface. Appropriate MS storage is preferably, the administration command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Subsequent administration includes modifying the sms message(s) as desired, and perhaps having an option to send/resend.</p>
--------------	------------	-----------	--

Fig. 73B-3

Operand ↓	PM	Preferred embodiment Administrate processing
209	C	<p>Administering a broadcast email causes searching a MS email system with search criteria of the email parameter string. The email parameter string can specify searching any email fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria. For example, the email string of "subj:'personnel';recip:'george@alltell.com';body:'reduction in force'" causes searching all emails with a subject containing "personnel" and was sent to "george@alltell.com" and has a message body containing the string "reduction in force". To search for certain email containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:sent,inbox,company;" indicates to only search the email folders of sent, inbox, and company (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of email. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In the preferred embodiment, all occurrences found in history are presented with at least their date/time stamps, subject line, sender and recipient, and perhaps other information, of the email and when it took place. In another embodiment, the most recent occurrence from searched folders is presented, and perhaps in an interface which enables appropriate MS email system processing from that point forward (e.g. when processed at local MS). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was presented with the result(s) in an appropriate administration interface. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search. Preferably, the administration command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Subsequent administration includes modifying the email(s) as desired, and perhaps having an option to send/resend.</p>

Fig. 73B-4

		<u>Preferred embodiment Administrative processing</u>
Operand ↓ 211	PM C	<p>Administrating a broadcast sms message causes searching a MS sms messaging system with search criteria of the sms message parameter string. The message parameter string can specify searching any message fields for any values including wildcarding. Each field is referenced with a predefined name and then associated with a search criteria. For example, the sms message string of "recip:2144034071@nextel.com,9725397137@lboxsv.com;" causes searching all messages to the sought recipients. To search for certain messaging containers/folders, a sub-search criteria of "folders" is used (e.g. "folders:outgoing" indicates to only search the outgoing folder (no specification preferably indicates to search all folders). Those skilled in the art recognize many useful syntaxes for searching any characteristics of messages. Wildcarding (pattern matching) is preferably inherent by searching for substrings. In the preferred embodiment, all occurrences found in history are presented with at least their date/time stamps, message, sender and recipient, and perhaps other information, of the message and when it took place. In another embodiment, the most recent occurrence from searched folders is presented, and perhaps in an interface which enables appropriate MS messaging system processing from that point forward (e.g. when processed at local MS). The search takes place as though the user manually launched the search, entered the criteria for the search, and then was presented with the result(s) in an appropriate administration interface. Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search. Preferably, the administration command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Subsequent administration includes modifying the message(s) as desired, and perhaps having an option to send/resend.</p>
213	O	See Find Command for identical processing this LBX release.

Fig. 73B-5

Operand	PM	Preferred embodiment Administrative processing
215	C	Administering an application causes searching the MS for application (and with the params parameter if specified). The app parameter is preferably an executable name. Providing a more defined partial or full path to the application parameter will validate that it is found there. The app parameter string preferably supports wildcarding. In the preferred embodiment, all occurrences found on the MS and their paths are presented to the user with at least their date/time stamps, size, and perhaps attributes information. In another embodiment, all parts which are linked to the executable are identified with their paths, date/time stamps, size, and perhaps attributes when a symbol file is specified with a new parameter. The symbol file is output from a link process and can be used to identify all executable parts such as dynamic link libraries, linked binaries, and any other executable binary file involved with the application. The search takes place as though the user manually launched the search, entered the criteria for the search, and then was presented with the result(s) in an appropriate administration interface (e.g. a properties edit user interface). Appropriate MS storage is updated and subsequently processed as though the user had manually performed the search. Preferably, the administration command data is maintained to LBX History, a historical log, or other useful storage for subsequent use. Subsequent administration includes modifying the configuration, startup parameters, or any other environmental variables of the application.
217	S	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
219	S	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
221	O	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
223	C	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
225	O	See Invoke Command for identical processing this LBX release.
227	O	See Invoke Command for identical processing this LBX release.
229	S	See Invoke Command for identical processing this LBX release.
231	C	See Invoke Command for identical processing this LBX release.
233	S	See Invoke Command for identical processing this LBX release.
235	O	See Invoke Command for identical processing this LBX release.
237	O	See Invoke Command for identical processing this LBX release.
239	O	See Invoke Command for identical processing this LBX release.

Fig. 73B-6

<i>Operand</i>	PM	<u>Preferred embodiment Administrate processing</u>
241	S	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
243	O	See Invoke Command for identical processing this LBX release.
245	S	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
247	O	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
249	O	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
251	C	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
253	C	See Compose Command for identical processing, except processing may take place locally and/or at privilege-providing remote MS(s) (system(s) parameter).
...		

Fig. 73B-7

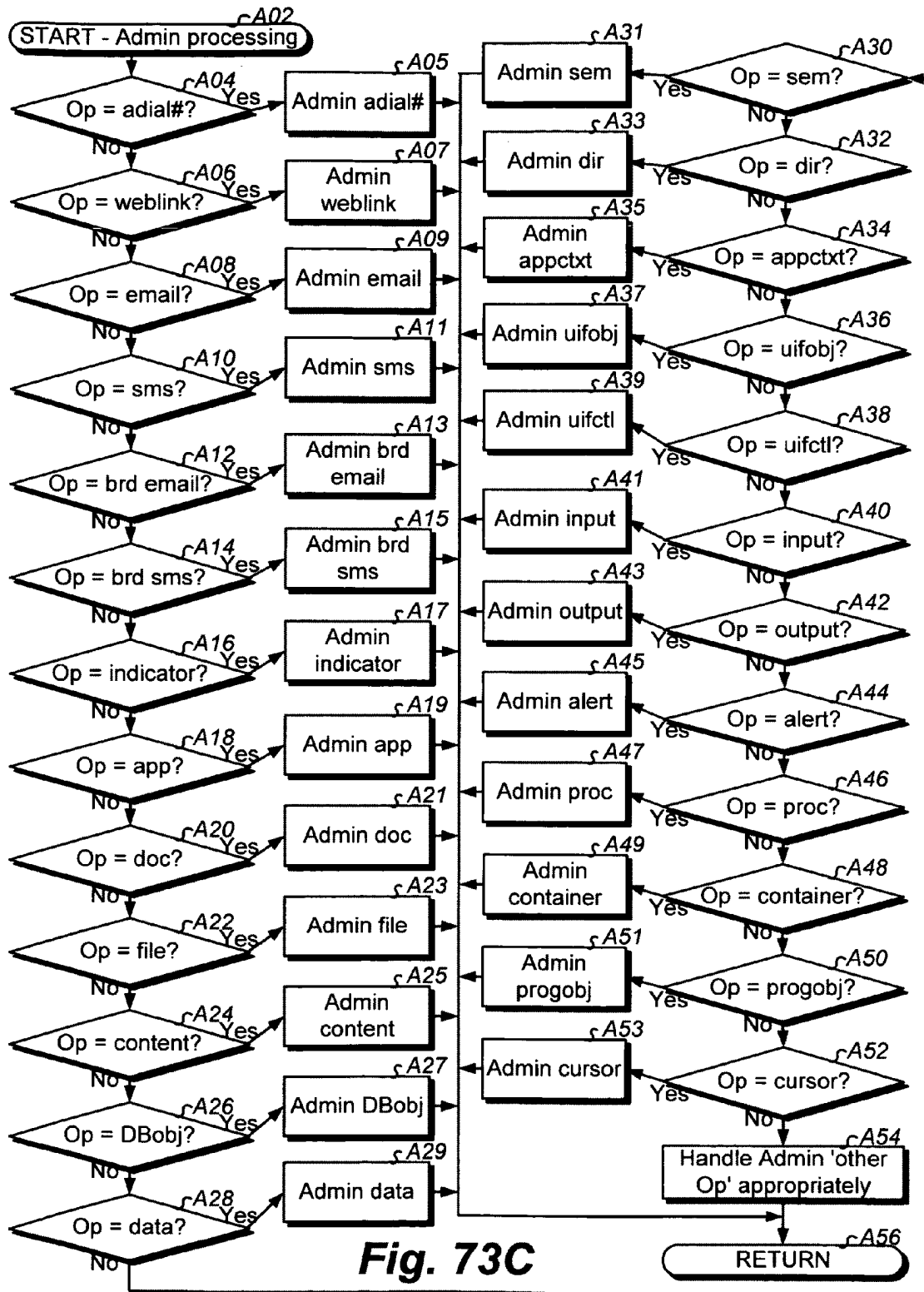


Fig. 73C

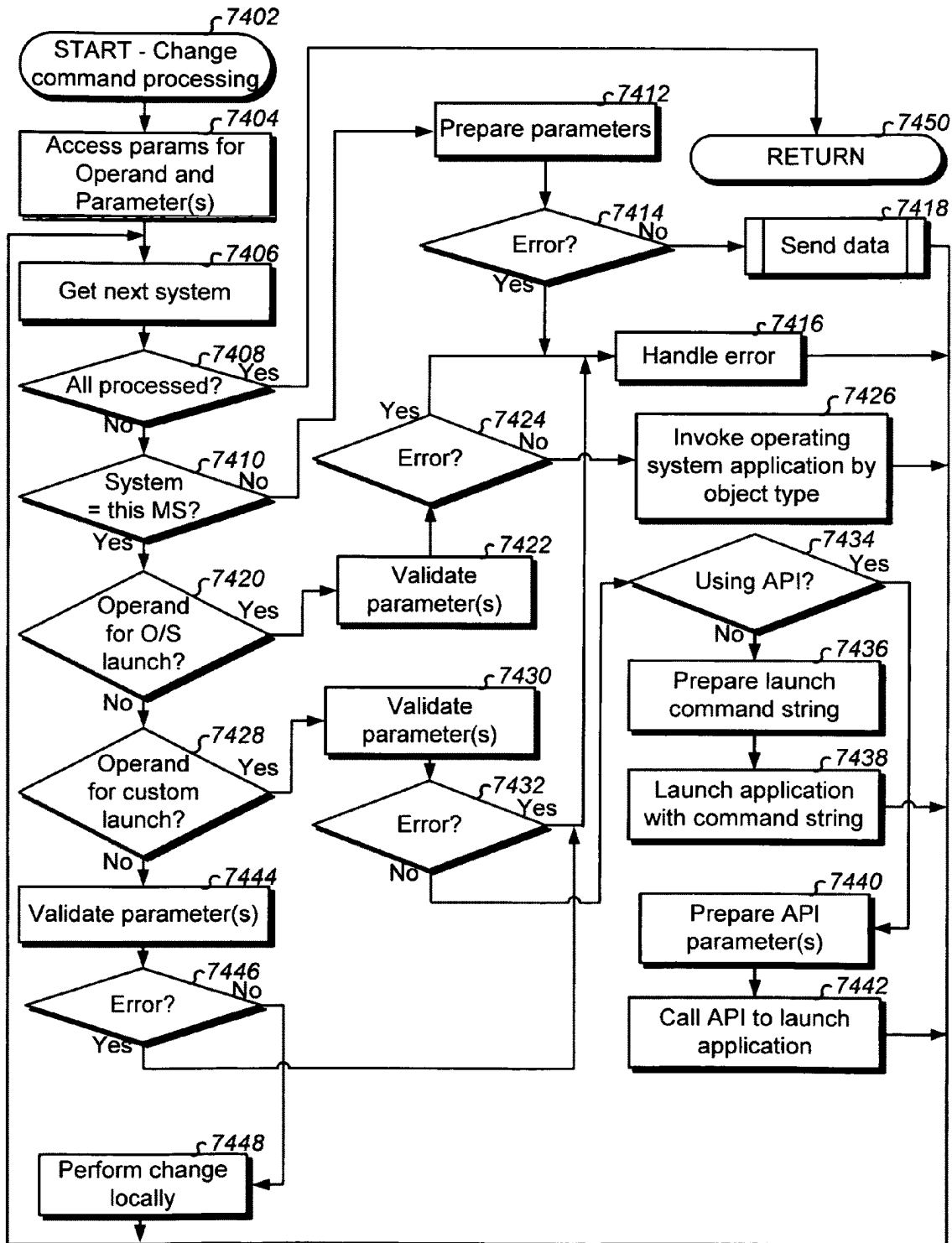


Fig. 74A

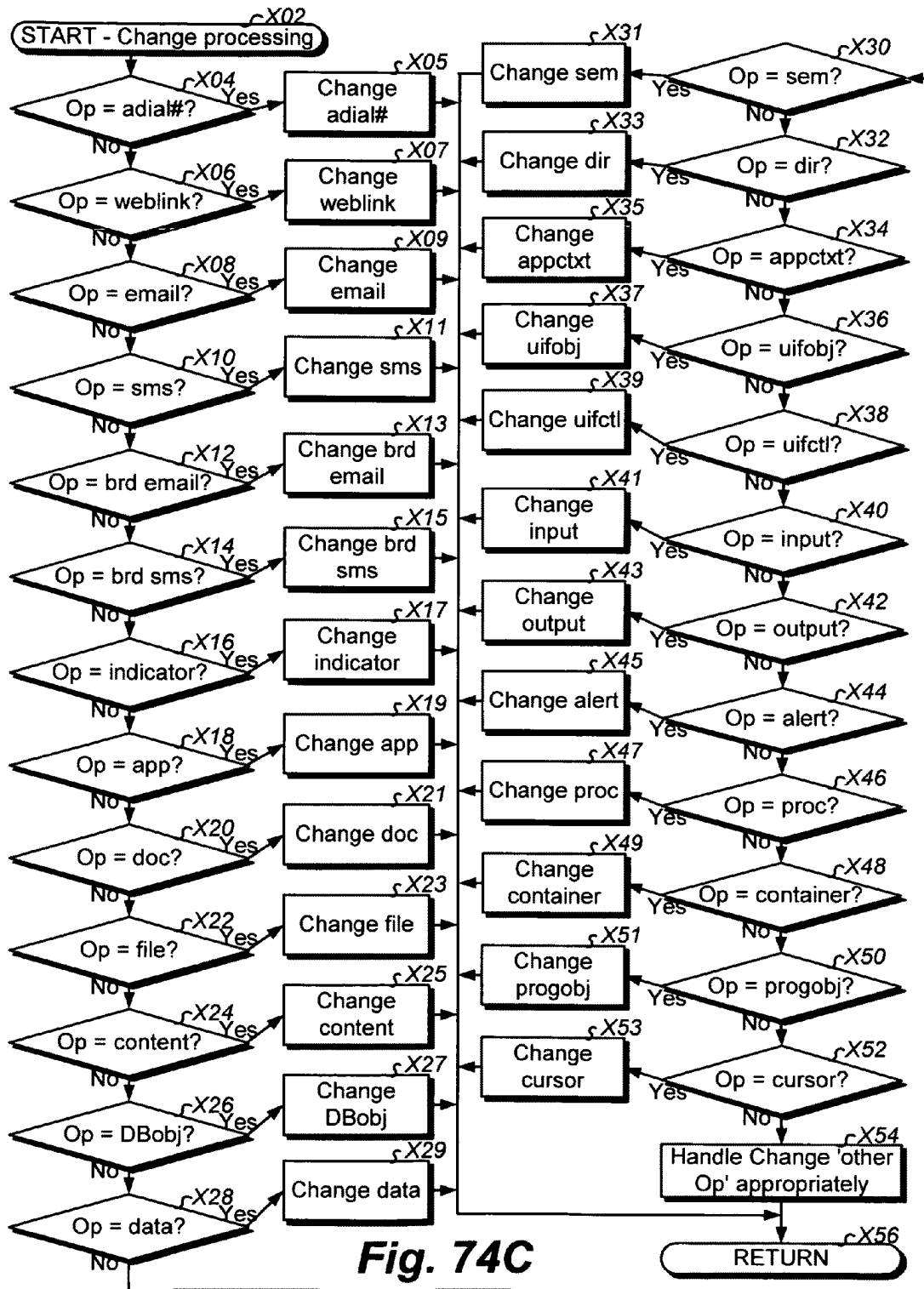


Fig. 74C

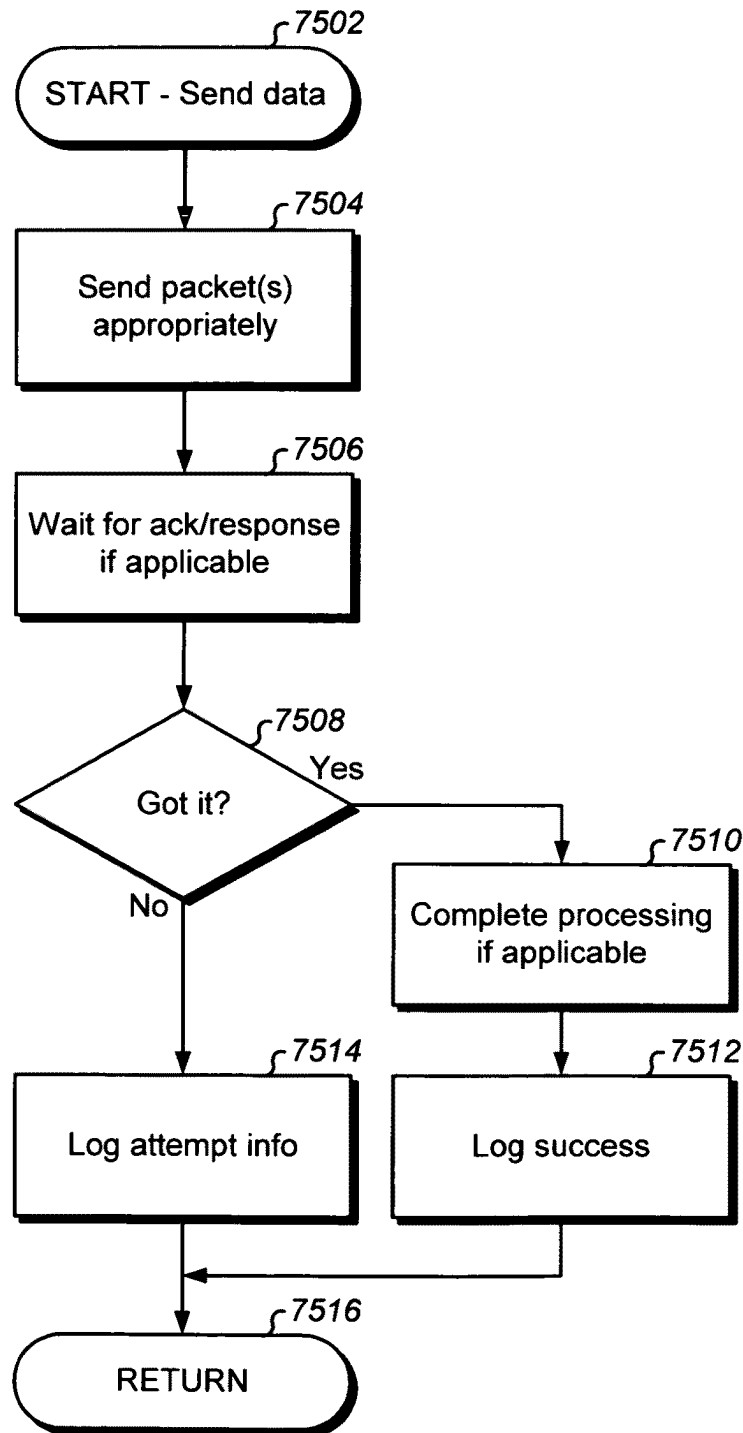


Fig. 75A

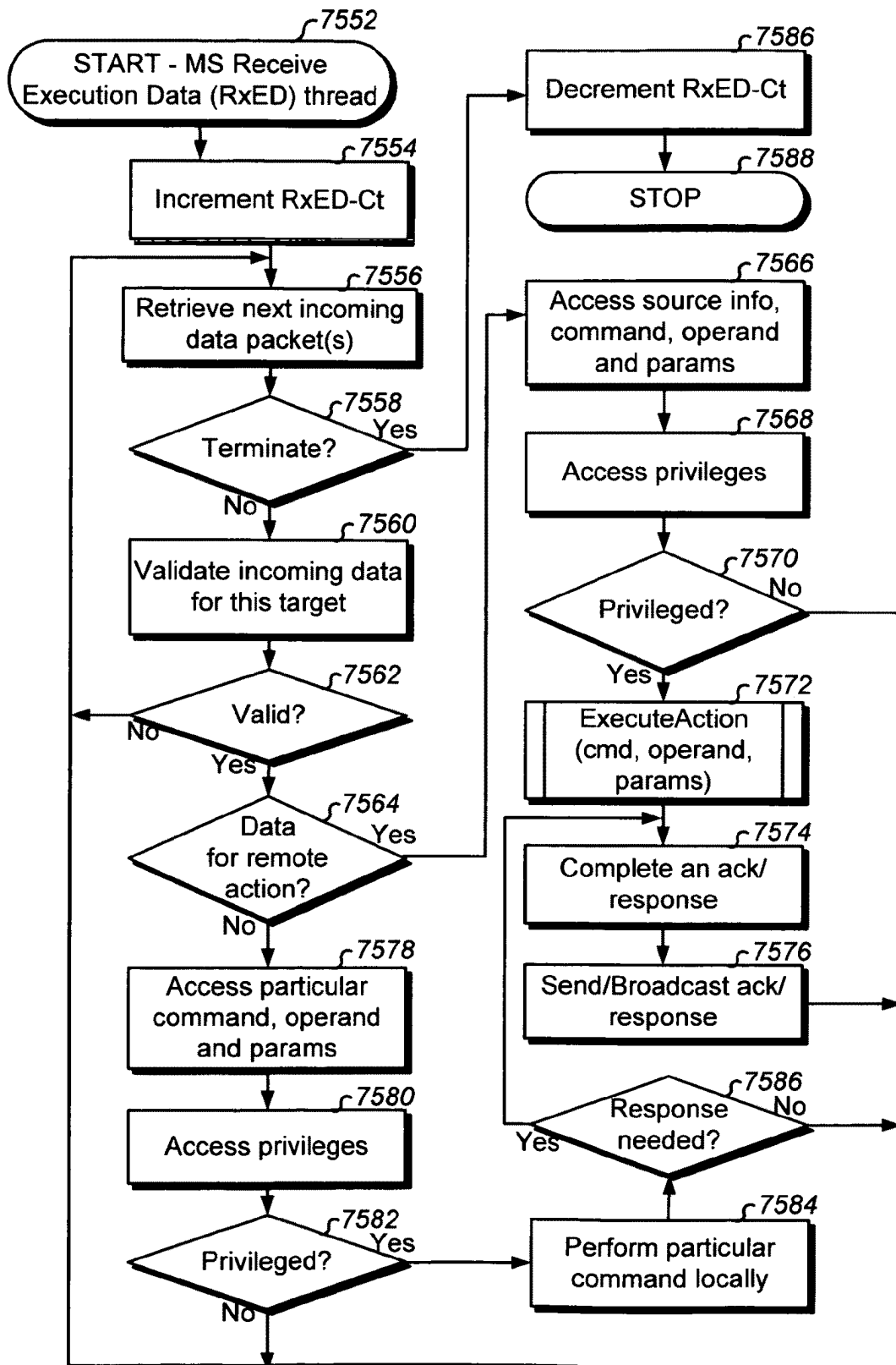


Fig. 75B

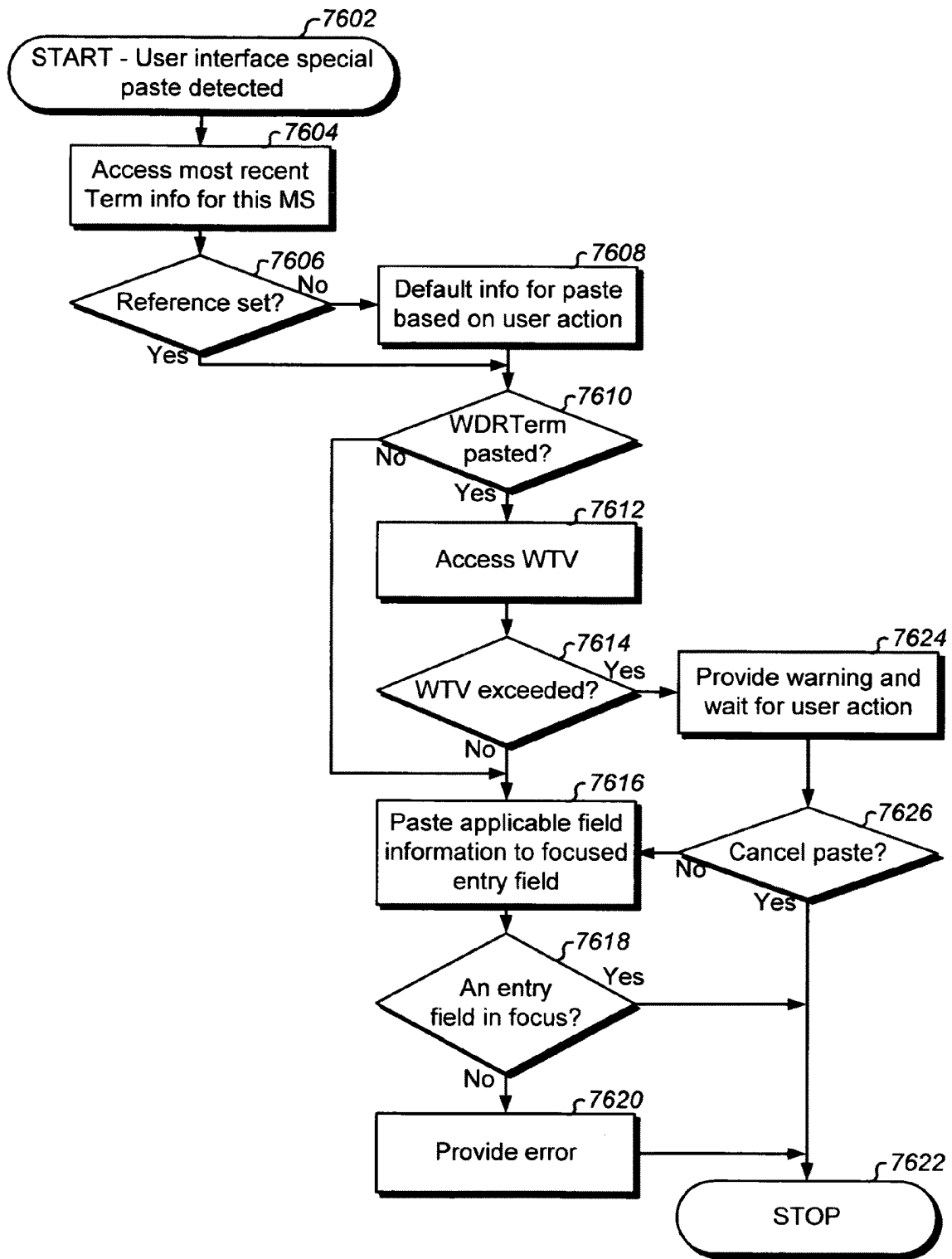


Fig. 76

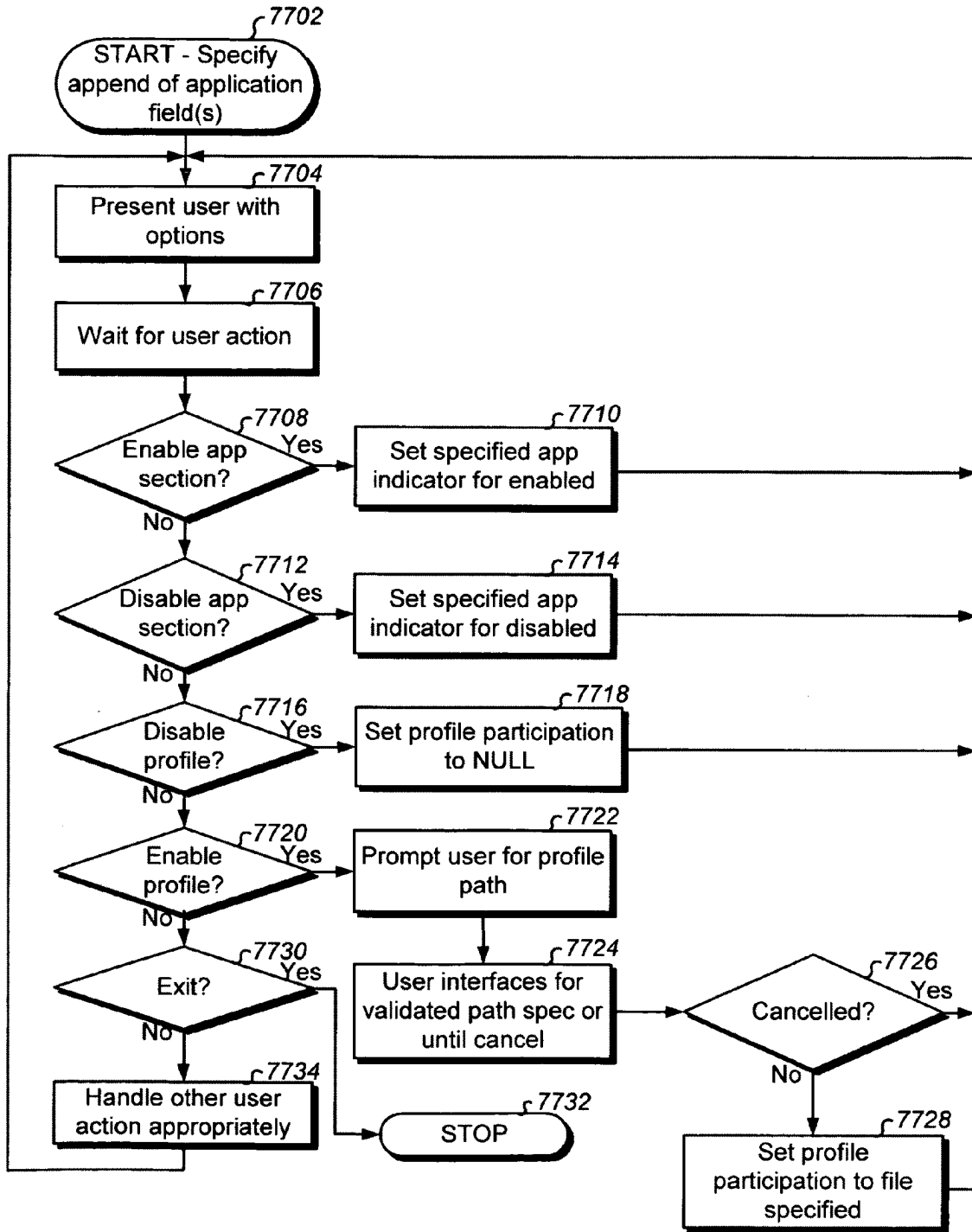


Fig. 77

```
...
<home>
  ...
  <city>Moorestown</city>
  <state>New Jersey</state>
  ...
</home>
...
<interests >
basketball;programming; running; football
</interests>
...
<hangouts>
  ...
  <morning></morning>
  <lunch>Jammin's;Mongolian Barbeque</lunch>
  <evening></evening>
  ...
</hangouts>
...
```

Fig. 78

1

**SYSTEM AND METHOD FOR LOCATION
BASED EXCHANGES OF DATA
FACILITATING DISTRIBUTED LOCATIONAL
APPLICATIONS**

CROSS-REFERENCES TO RELATED
APPLICATIONS

This application is a continuation in part of application Ser. No. 12/077,041 filed Mar. 14, 2008 and entitled "System and Method for Location Based Exchanges of Data Facilitating Distributed Locational Applications". Specification from the aforementioned application is included herein. FIGS. 30A through 78 and the associated descriptions are new specification.

FIELD OF THE INVENTION

The present disclosure relates generally to location based services for mobile data processing systems, and more particularly to location based exchanges of data between distributed mobile data processing systems for locational applications. A common connected service is not required for location based functionality and features. Location based exchanges of data between distributed mobile data processing systems enable location based features and functionality in a peer to peer manner.

BACKGROUND OF THE INVENTION

The internet has exploded with new service offerings. Websites yahoo.com, google.com, ebay.com, amazon.com, and iTunes.com have demonstrated well the ability to provide valuable services to a large dispersed geographic audience through the internet (ebay, yahoo, google, amazon and iTunes (Apple) are trademarks of the respective companies). Thousands of different types of web services are available for many kinds of functionality. Advantages of having a service as the intermediary point between clients, users, and systems, and their associated services, includes centralized processing, centralized maintaining of data, for example to have an all knowing database for scope of services provided, having a supervisory point of control, providing an administrator with access to data maintained by users of the web service, and other advantages associated with centralized control. The advantages are analogous to those provided by the traditional mainframe computer to its clients wherein the mainframe owns all resources, data, processing, and centralized control for all users and systems (clients) that access its services. However, as computers declined in price and adequate processing power was brought to more distributed systems, such as Open Systems (i.e. Windows, UNIX, Linux, and Mac environments), the mainframe was no longer necessary for many of the daily computing tasks. In fact, adequate processing power is incorporated in highly mobile devices, various handheld mobile data processing systems, and other mobile data processing systems. Technology continues to drive improved processing power and data storage capabilities in less physical space of a device. Just as Open Systems took much of the load of computing off of mainframe computers, so to can mobile data processing systems offload tasks usually performed by connected web services. As mobile data processing systems are more capable, there is no need for a service to middleman interactions possible between them.

While a centralized service has its advantages, there are also disadvantages. A service becomes a clearinghouse for all web service transactions. Regardless of the number of threads

2

of processing spread out over hardware and processor platforms, the web service itself can become a bottleneck causing poor performance for timely response, and can cause a large amount of data that must be kept for all connected users and/or systems. Even large web services mentioned above suffer from performance and maintenance overhead. A web service response will likely never be fast enough. Additionally, archives must be kept to ensure recovery in the event of a disaster because the service houses all data for its operations. Archives also require storage, processing power, planning, and maintenance. A significantly large and costly data center is necessary to accommodate millions of users and/or systems to connect to the service. There is a tremendous amount of overhead in providing such a service. Data center processing power, data capacity, data transmission bandwidth and speed, infrastructure entities, and various performance considerations are quite costly. Costs include real estate required, utility bills for electricity and cooling, system maintenance, personnel to operate a successful business with service(s), etc. A method is needed to prevent large data center costs while eliminating performance issues for features sought. It is inevitable that as users are hungry for more features and functionality on their mobile data processing systems, processing will be moved closer to the device for optimal performance and infrastructure cost savings.

Service delivered location dependent content was disclosed in U.S. Pat. Nos. 6,456,234; 6,731,238; 7,187,997 (Johnson). Anonymous location based services was disclosed in U.S. PTO Publication 2006/0022048 (Johnson). The Johnson patents and published application operate as most web services do in that the clients connecting to the service benefit from the service by having some connectivity to the service. U.S. Publication 2006/0022048 (Johnson) could cause large numbers of users to inundate the service with device heartbeats and data to maintain, depending on the configurations made. While this may be of little concern to a company that has successfully deployed substantially large web service resources, it may be of great concern to other more frugal companies. A method is needed for enabling location dependent features and functionality without the burden of requiring a service.

Users are skeptical about their privacy as internet services proliferate. A service by its very nature typically holds information for a user maintained in a centralized service database. The user's preferences, credential information, permissions, customizations, billing information, surfing habits, and other conceivable user configurations and activity monitoring, can be housed by the service at the service. Company insiders, as well as outside attackers, may get access. Most people are concerned with preventing personal information of any type being kept in a centralized database which may potentially become compromised from a security standpoint. Location based services are of even more concern, in particular when the locations of the user are to be known to a centralized service. A method and system is needed for making users comfortable with knowing that their personal information is at less risk of being compromised.

A reasonable requirement is to push intelligence out to the mobile data processing systems themselves, for example, in knowing their own locations and perhaps the locations of other nearby mobile data processing systems. Mobile data processing systems can intelligently handle many of their own application requirements without depending on some remote service. Just as two people in a business organization should not need a manager to speak to each other, no two mobile data processing systems should require a service middleman for useful location dependent features and func-

3

tionality. The knowing of its own location should not be the end of social interaction implementation local to the mobile data processing systems, but rather the starting place for a large number of useful distributed local applications that do not require a service.

Different users use different types of Mobile data processing Systems (MSs) which are also called mobile devices: laptops, tablet computers, Personal Computers (PCs), Personal Digital Assistants (PDAs), cell phones, automobile dashboard mounted data processing systems, shopping cart mounted data processing systems, mobile vehicle or apparatus mounted data processing systems, Personal Navigational Devices (PNDs), iPhones (iPhone is a trademark of Apple, Inc.), various handheld mobile data processing systems, etc. MSs move freely in the environment, and are unpredictably moveable (i.e. can be moved anywhere, anytime). Many of these Mobile data processing Systems (MSs) do not have capability of being automatically located, or are not using a service for being automatically located. Conventional methods use directly relative stationary references such as satellites, antennas, etc. to locate MSs. Stationary references are expensive to deploy, and risk obsolescence as new technologies are introduced to the marketplace. Stationary references have finite scope of support for locating MSs.

While the United States E911 mandate for cellular devices documents requirements for automatic location of a Mobile data processing System (MS) such as a cell phone, the mandate does not necessarily promote real time location and tracking of the MSs, nor does it define architecture for exploiting Location Based Services (LBS). We are in an era where Location Based Services (LBS), and location dependent features and functionality, are among the most promising technologies in the world. Automatic locating of every Mobile data processing System (MS) is an evolutionary trend. A method is needed to shorten the length of time for automatically locating every MS. Such a goal can be costly using prior art technologies such as GPS (Global Positioning System), radio wave triangulation, coming within range to a known located sensor, or the like. Complex system infrastructure, or added hardware costs to the MSs themselves, make such ventures costly and time constrained by schedules and costs involved in engineering, construction, and deployment.

A method is needed for enabling users to get location dependent features and functionality through having their mobile locations known, regardless of whether or not their MS is equipped for being located. Also, new and modern location dependent features and functionality can be provided to a MS unencumbered by a connected service.

BRIEF SUMMARY OF THE INVENTION

LBS (Location Based Services) is a term which has gained in popularity over the years as MSs incorporate various location capability. The word "Services" in that terminology plays a major role in location based features and functionality involving interaction between two or more users. This disclosure introduces a new terminology, system, and method referred to as Location Based eXchanges (LBX). LBX is an acronym used interchangeably/contextually throughout this disclosure for the singular term "Location Based Exchange" and for the plural term "Location Based Exchanges", much the same way LBS is used interchangeably/contextually for the single term "Location Based Service" and for the plural term "Location Based Services". LBX describes leveraging the distributed nature of connectivity between MSs in lieu of leveraging a common centralized service nature of connectivity between MSs. The line can become blurred between

4

LBS and LBX since the same or similar features and functionality are provided, and in some cases strengths from both may be used. The underlying architectural shift differentiates LBX from LBS for depending less on centralized services, and more on distributed interactions between MSs. LBX provide server-free and server-less location dependent features and functionality.

Disclosed are many different aspects to LBX, starting with the foundation requirement for each participating MS to know, at some point in time, their own whereabouts. LBX is enabled when an MS knows its own whereabouts. It is therefore a goal to first make as many MSs know their own whereabouts as possible. When two or more MSs know their own whereabouts, LBX enables distributed locational applications whereby a server is not required to middleman social interactions between the MSs. The MSs interact as peers. LBX disclosed include purely peer to peer interactions, peer to peer interactions for routing services, peer to peer interactions for delivering distributed services, and peer to peer interactions for location dependent features and functionality. One embodiment of an LBX enabled MS is referred to as an lbxPhone™.

It is an advantage herein to have no centralized service governing location based features and functionality among MSs. Avoiding a centralized service prevents performance issues, infrastructure costs, and solves many of the issues described above. No centralized service also prevents a user's information from being kept in one accessible place. LBS contain centralized data that is personal in nature to its users. This is a security concern. Having information for all users in one place increases the likelihood that a disaster to the data will affect more than a single user. LBX spreads data out across participating systems so that a disaster affecting one user does not affect any other user.

It is an advantage herein for enabling useful distributed applications without the necessity of having a service, and without the necessity of users and/or systems registering with a service. MSs interact as peers in preferred embodiments, rather than as clients to a common service (e.g. internet connected web service).

It is an advantage herein for locating as many MSs as possible in a wireless network, and without additional deployment costs on the MSs or the network. Conventional locating capability includes GPS (Global Positioning System) using stationary orbiting satellites, improved forms of GPS, for example AGPS (Adjusted GPS) and DGPS (Differential GPS) using stationary located ground stations, wireless communications to stationary located cell tower base stations, TDOA (Time Difference of Arrival) or AOA (Angle of Arrival) triangulation using stationary located antennas, presence detection in vicinity of a stationary located antenna, presence detection at a wired connectivity stationary network location, or other conventional locating systems and methods. Mobile data processing systems, referred to as Indirectly Located Mobile data processing systems (ILMs), are automatically located using automatically detected locations of Directly Located Mobile data processing systems (DLMs) and/or automatically detected locations of other ILMs. ILMs are provided with the ability to participate in the same LBS, or LBX, as a DLM (Directly Located Mobile data processing system). DLMs are located using conventional locating capability mentioned above. DLMs provide reference locations for automatically locating ILMs, regardless of where any one is currently located. DLMs and ILMs can be highly mobile, for example when in use by a user. There are a variety of novel methods for automatically locating ILMs, for example triangulating an ILM (Indirectly Located Mobile data processing

5

system) location using a plurality of DLMs, detecting the ILM being within the vicinity of at least one DLM, triangulating an ILM location using a plurality of other ILMs, detecting the ILM being within the vicinity of at least one other ILM, triangulating an ILM location using a mixed set of DLM(s) and ILM(s), determining the ILM location from heterogeneously located DLMs and/or ILMs, and other novel methods.

MSs are automatically located without using direct conventional means for being automatically located. The conventional locating capability (i.e. conventional locating methods) described above is also referred to as direct methods. Conventional methods are direct methods, but not all direct methods are conventional. There are new direct techniques disclosed below. Provided herein is an architecture, as well as systems and methods, for immediately bringing automatic location detection to every MS in the world, regardless of whether that MS is equipped for being directly located. MSs without capability of being directly located are located by leveraging the automatically detected locations of MSs that are directly located. This is referred to as being indirectly located. An MS which is directly located is hereinafter referred to as a Directly Located Mobile data processing system (DLM). For a plural acronym, MSs which are directly located are hereinafter referred to as Directly Located Mobile data processing systems (DLMs). MSs without capability of being directly located are located using the automatically detected locations of MSs that have already been located. An MS which is indirectly located is hereinafter referred to as an Indirectly Located Mobile data processing system (ILM). For a plural acronym, MSs which are indirectly located are hereinafter referred to as Indirectly Located Mobile data processing systems (ILMs). A DLM can be located in the following ways:

- A) New triangulated wave forms;
- B) Missing Part Triangulation (MPT) as disclosed below;
- C) Heterogeneous direct locating methods;
- D) Assisted Direct Location Technology (ADLT) using a combination of direct and indirect methods;
- E) Manually specified; and/or
- F) Any combinations of A) through E);

DLMs provide reference locations for automatically locating ILMs, regardless of where the DLMs are currently located. It is preferable to assure an accurate location of every DLM, or at least provide a confidence value of the accuracy. A confidence value of the accuracy is used by relative ILMs to determine which are the best set (e.g. which are of highest priority for use to determine ILM whereabouts) of relative DLMs (and/or ILMs) to use for automatically determining the location of the ILM.

In one example, the mobile locations of several MSs are automatically detected using their local GPS chips. Each is referred to as a DLM. The mobile location of a non-locatable MS is triangulated using radio waves between it and three (3) of the GPS equipped DLMs. The MS becomes an ILM upon having its location determined relative the DLMs. ILMs are automatically located using DLMs, or other already located ILMs. An ILM can be located in the following ways:

- G) Triangulating an ILM location using a plurality of DLMs with wave forms of any variety (e.g. AOA, TDOA, MPT (a heterogeneous location method));
- H) Detecting the ILM being within the reasonably close vicinity of at least one DLM;
- I) Triangulating an ILM location using a plurality of other ILMs with wave forms of any variety;
- J) Detecting the ILM being within the reasonable close vicinity of at least one other ILM;

6

K) Triangulating an ILM location using a mixed set of DLM(s) and ILM(s) with wave forms of any variety (referred to as ADLT);

L) Determining the ILM location from heterogeneously located DLMs and/or ILMs (i.e. heterogeneously located, as used here, implies having been located relative different location methodologies);

M) A) through F) Above; and/or

N) Any combinations of A) through M).

Locating functionality may leverage GPS functionality, including but not limited to GPS, AGPS (Adjusted GPS), DGPS, (Differential GPS), or any improved GPS embodiment to achieve higher accuracy using known locations, for example ground based reference locations. The Nextel GPS enabled iSeries cell phones provide excellent examples for use as DLMs (Nextel is a trademark of Sprint/Nextel). Locating functionality may incorporate triangulated locating of the MS, for example using a class of Radio Frequency (RF) wave spectrum (cellular, WiFi (some WiFi embodiments referred to as WiMax), bluetooth, etc), and may use measurements from different wave spectrums for a single location determination (depends on communications interface(s) 70 available). A MS may have its whereabouts determined using a plurality of wave spectrum classes available to it (cellular, WiFi, bluetooth, etc). The term "WiFi" used throughout this disclosure also refers to the industry term "WiMax". Locating functionality may include in-range proximity detection for detecting the presence of the MS. Wave forms for triangulated locating also include microwaves, infrared wave spectrum relative infrared sensors, visible light wave spectrum relative light visible light wave sensors, ultraviolet wave spectrum relative ultraviolet wave sensors, X-ray wave spectrum relative X-ray wave sensors, gamma ray wave spectrum relative gamma ray wave sensors, and longwave spectrum (below AM) relative longwave sensors. While there are certainly more common methods for automatically locating a MS (e.g. radio wave triangulation, GPS, in range proximity detection), those skilled in the art recognize there are methods for different wave spectrums being detected, measured, and used for carrying information between data processing systems.

Kubler et al (U.S. PTO publications 2004/0264442, 2004/0246940, 2004/0228330, 2004/0151151) disclosed methods for detecting presence of mobile entities as they come within range of a sensor. In Kubler et al, accuracy of the location of the detected MS is not well known, so an estimated area of the whereabouts of the MS is enough to accomplish intended functionality, for example in warehouse installations. A confidence value of this disclosure associated with Kubler et al tends to be low (i.e. not confident), with lower values for long range sensors and higher values for short range sensors.

GPS and the abundance of methods for improving GPS accuracy has led to many successful systems for located MSs with high accuracy. Triangulation provides high accuracies for locating MSs. A confidence value of this disclosure associated with GPS and triangulating location methods tends to be high (i.e. confident). It is preferred that DLMs use the highest possible accuracy method available so that relative ILMs are well located. Not all DLMs need to use the same location methods. An ILM can be located relative DLMs, or other ILMs, that each has different locating methodologies utilized.

Another advantage herein is to generically locate MSs using varieties and combinations of different technologies. MSs can be automatically located using direct conventional methods for accuracy to base on the locating of other MSs. MSs can be automatically located using indirect methods. Further, it is an advantage to indirectly locate a MS relative

heterogeneously located MSs. For example, one DLM may be automatically located using GPS. Another DLM may be automatically located using cell tower triangulation. A third DLM may be automatically located using within range proximity. An ILM can be automatically located at a single location, or different locations over time, relative these three differently located DLMs. The automatically detected location of the ILM may be determined using a form of triangulation relative the three DLMs just discussed, even though each DLM had a different direct location method used. In a preferred embodiment, industry standard IEEE 802.11 WiFi is used to locate (triangulate) an ILM relative a plurality of DLMs (e.g. TDOA in one embodiment). This standard is prolific among more compute trended MSs. Any of the family of 802.11 wave forms such as 802.11a, 802.11b, 802.11g, or any other similar class of wave spectrum can be used, and the same spectrum need not be used between a single ILM and multiple DLMs. 802.x used herein generally refers to the many 802.whatever variations.

Another advantage herein is to make use of existing marketplace communications hardware, communications software interfaces, and communications methods and location methods where possible to accomplish locating an MS relative one or more other MSs. While 802.x is widespread for WiFi communications, other RF wave forms can be used (e.g. cell phone to cell tower communications). In fact, any wave spectrum for carrying data applies herein.

Still another advantage is for support of heterogeneous locatable devices. Different people like different types of devices as described above. Complete automation of locating functionality can be provided to a device through local automatic location detection means, or by automatic location detection means remote to the device. Also, an ILM can be located relative a laptop, a cell phone, and a PDA (i.e. different device types).

Yet another advantage is to prevent the unnecessary storing of large amounts of positioning data for a network of MSs. Keeping positioning data for knowing the whereabouts of all devices can be expensive in terms of storage, infrastructure, performance, backup, and disaster recovery. A preferred embodiment simply uses a distributed approach to determining locations of MSs without the overhead of an all-knowing database maintained somewhere. Positions of MSs can be determined "on the fly" without storing information in a master database. However, there are embodiments for storing a master database, or a subset thereof, to configurable storage destinations, when it makes sense. A subset can be stored at a MS.

Another advantage includes making use of existing location equipped MSs to expand the network of locatable devices by locating non-equipped MSs relative the location of equipped MSs. MSs themselves help increase dimensions of the locatable network of MSs. The locatable network of MSs is referred to as an LN-Expanse (i.e. Location-Network Expanse). An LN-Expanse dynamically grows and shrinks based on where MSs are located at a particular time. For example, as users travel with their personal MSs, the personal MSs themselves define the LN-Expanse since the personal MSs are used to locate other MSs. An ILM simply needs location awareness relative located MSs (DLMs and/or ILMs).

Yet another advantage is a MS interchangeably taking on the role of a DLM or ILM as it travels. MSs are chameleons in this regard, in response to location technologies that happen to be available. A MS may be equipped for DLM capability, but may be in a location at some time where the capability is inoperable. In these situations the DLM takes on the role of an

ILM. When the MS again enters a location where it can be a DLM, it automatically takes on the role of the DLM. This is very important, in particular for emergency situations. A hiker has a serious accident in the mountains which prevents GPS equipped DLM capability from working. Fortunately, the MS automatically takes on the role of an ILM and is located within the vicinity of neighboring (nearby) MSs. This allows the hiker to communicate his location, operate useful locational application functions and features at his MS, and enable emergency help that can find him.

It is a further advantage that MS locations be triangulated using any wave forms (e.g. RF, microwaves, infrared, visible light, ultraviolet, X-ray, gamma ray). X-ray and gamma ray applications are special in that such waves are harmful to humans in short periods of times, and such applications should be well warranted to use such wave forms. In some medical embodiments, micro-machines may be deployed within a human body. Such micro-machines can be equipped as MSs. Wave spectrums available at the time of deployment can be used by the MSs for determining exact positions when traveling through a body.

It is another advantage to use TDOA (Time Difference Of Arrival), AOA (Angle Of Arrival), and Missing Part Triangulation (MPT) when locating a MS. TDOA uses time information to determine locations, for example for distances of sides of a triangle. AOA uses angles of arrival to antennas to geometrically assess where a MS is located by intersecting lines drawn from the antennas with detected angles. MPT is disclosed herein as using combinations of AOA and TDOA to determine a location. Exclusively using all AOA or exclusively using all TDOA is not necessary. MPT can be a direct method for locating MSs.

Yet another advantage is to locate MSs using Assisted Direct Location Technology (ADLT). ADLT is disclosed herein as using direct (conventional) location capability together with indirect location capability to confidently determine the location of a MS.

Still another advantage is to permit manual specification for identifying the location of a MS (a DLM). The manual location can then in turn be used to facilitate locating other MSs. A user interface may be used for specification of a DLM location. The user interface can be local, or remote, to the DLM. Various manual specification methods are disclosed. Manual specification is preferably used with less mobile MSs, or existing MSs such as those that use dodgeball.com (trademark of Google). The confidence value depends on how the location is specified, whether or not it was validated, and how it changes when the MS moves after being manually set. Manual specification should have limited scope in an LN-expanse unless inaccuracies can be avoided.

Another advantage herein is locating a MS using any of the methodologies above, any combinations of the methodologies above, and any combinations of direct and/or indirect location methods described.

Another advantage is providing synergy between different locating technologies for smooth operations as an MS travels. There are large numbers of methods and combinations of those methods for keeping an MS informed of its whereabouts. Keeping an MS informed of its whereabouts in a timely manner is critical in ensuring LBX operate optimally, and for ensuring nearby MSs without certain locating technologies can in turn be located.

It is another advantage for locating a MS with multiple location technologies during its travels, and in using the best of breed data from multiple location technologies to infer a MS location confidently. Confidence values are associated with reference location information to ensure an MS using the

location information can assess accuracy. A DLM is usually an “affirmifier”. An affirmifier is an MS with its whereabouts information having high confidence of accuracy and can serve as a reference for other MSs. An ILM can also be an affirmifier provided there is high confidence that the ILM location is known. An MS (e.g. ILM) may be a “pacifier”. A pacifier is an MS having location information for its whereabouts with a low confidence for accuracy. While it can serve as a reference to other ILMs, it can only do so by contributing a low confidence of accuracy.

It is an advantage to synergistically make use of the large number of locating technologies available to prevent one particular type of technology to dominate others while using the best features of each to assess accurate mobile locations of MSs.

A further advantage is to leverage a data processing system with capability of being located for co-locating another data processing system without any capability of being located. For example, a driver owns an older model automobile, has a useful second data processing system in the automobile without means for being automatically located. The driver also owns a cell phone, called a first data processing system, which does have means for being automatically located. The location of the first data processing system can be shared with the second data processing system for locating the second data processing system. Further still, the second data processing system without means for being automatically located is located relative a first set (plurality) of data processing systems which are not at the same location as the second data processing system. So, data processing systems are automatically located relative at least one other data processing which can be automatically located.

Another advantage is a LBX enabled MS includes a service informant component for keeping a supervisory service informed. This prevents an MS from operating in total isolation, and prevents an MS from operating in isolation with those MSs that are within its vicinity (e.g. within maximum range **1306**) at some point in time, but to also participate when the same MSs are great distances from each other. There are LBX which would fit well into an LBS model, but a preferred embodiment chooses to use the LBX model. For example, multiple MS users are seeking to carpool to and from a common destination. The service informant component can perform timely updates to a supervisory service for route comparisons between MSs, even though periods of information are maintained only at the MSs. For example, users find out that they go to the same church with similar schedules, or coworkers find out they live nearby and have identical work schedules. The service informant component can keep a service informant of MS whereabouts to facilitate novel LBX applications.

It is a further advantage in leveraging the vast amount of MS WiFi/WiMax deployment underway in the United States. More widespread WiFi/WiMax availability enhances the ability for well performing peer to peer types of features and functionality disclosed.

It is a further advantage to prevent unnecessary established connections from interfering with successfully triangulating a MS position. As the MS roams and encounters various wave spectrum signals, that is all that is required for determining the MS location. Broadcast signaling contains the necessary location information for automatically locating the MS.

Yet another advantage is to leverage Network Time Protocol (NTP) for eliminating bidirectional communications in determining Time of Arrival (TOA) and TDOA (Time Difference Of Arrival) measurements (TDOA as used in the disclosure generally refers to both TOA and TDOA). NTP enables

a single unidirectional transmission of data to carry all that is necessary in determining TDOA, provided the sending data processing system and the receiving data processing system are NTP synchronized to an adequate granulation of time.

It is an advantage of this disclosure to provide a competing superior alternative to server based mobile technologies such as that of U.S. Pat. Nos. 6,456,234; 6,731,238; 7,187,997; and U.S. PTO Publication 2006/0022048 (Johnson). It is also an advantage to leverage both LBX technology and LBS technology in the same MS in order to improve the user experience. The different technologies can be used to complement each other in certain embodiments.

A further advantage herein is to leverage existing “usual communications” data transmissions for carrying new data that is ignored by existing MS processing, but observed by new MS processing, for carrying out processing maximizing location functions and features across a large geography. Alternatively, new data can be transmitted between systems for the same functionality.

It is an advantage herein in providing peer to peer service propagation. ILMs are provided with the ability to participate in the same Location Based Services (LBS) or other services as DLM(s) in the vicinity. An MS may have access to services which are unavailable to other MSs. Any MS can share its accessible services for being accessible to any other MS, preferably in accordance with permissions. For example, an MS without internet access can get internet access via an MS in the vicinity with internet access. In a preferred embodiment, permissions are maintained in a peer to peer manner prior to lookup for proper service sharing. In another embodiment, permissions are specified and used at the time of granting access to the shared services. Once granted for sharing, services can be used in a mode as if the sharing user is using the services, or in a mode as if the user accepting the share is a new user to the service. Routing paths are dynamically reconfigured and transparently used as MSs travel. Hop counts dynamically change to strive for a minimal number of hops for an MS getting access to a desirable service. Route communications depend on where the MS needing the service is located relative a minimal number of hops through other MSs to get to the service. Services can be propagated from DLMs to DLMS, DLMs to ILMs, or ILMs to ILMs.

It is another advantage herein for providing peer to peer permissions, authentication, and access control. A service is not necessary for maintaining credentials and permissions between MSs. Permissions are maintained locally to a MS. In a centralized services model, a database can become massive in size when searching for needed permissions. Permission searching and validation of U.S. PTO Publication 2006/0022048 (Johnson) was costly in terms of database size and performance. There was overhead in maintaining who owned the permission configuration for every permission granted. Maintaining permissions locally, as described below, reduces the amount of data to represent the permission because the owner is understood to be the personal user of the MS. Additionally, permission searching is very fast because the MS only has to search its local data for permissions that apply to only its MS.

Yet another advantage is to provide a nearby, or nearness, status using a peer to peer system and method, rather than intelligence maintained in a centralized database for all participating MSs. There is lots of overhead in maintaining a large database containing locations of all known MSs. This disclosure removes such overhead through using nearby detection means of one MS when in the vicinity of another MS. There are varieties of controls for governing how to generate the nearby status. In one aspect, a MS automatically

11

calls the nearby MS thereby automatically connecting the parties to a conversation without user interaction to initiate the call. In another aspect, locally maintained configurations govern functionality when MSs are newly nearby, or are newly departing being nearby. Nearby status, alerts, and queries are achieved in a LBX manner.

It is yet another advantage for automatic call forwarding, call handling, and call processing based on the whereabouts of a MS, or whereabouts of a MS relative other MSs. The nearness condition of one MS to another MS can also affect the automatic call forwarding functionality.

Yet another advantage herein is for peer to peer content delivery and local MS configuration of that content. Users need no connectivity to a service. Users make local configurations to enjoy location based content delivery to other MSs. Content is delivered under a variety of circumstances for a variety of configurable reasons. Content maintained local to an MS is delivered asynchronously to other MSs for nearby alerts, arrival or departure to and from geofenced areas, and other predicated conditions of nearby MSs. While it may appear there are LBS made available to users of MSs, there are in fact LBX being made available to those users.

Another advantage herein is a LBX enabled MS can operate in a peer to peer manner to data processing systems which control environmental conditions. For example, automobile equipped (or driver kept) MSs encounter an intersection having a traffic light. Interactions between the MSs at the intersection and a data processing system in the vicinity for controlling the traffic light can automatically override light color changing for optimal traffic flow. In another embodiment, a parking lot search by a user with an MS is facilitated as he enters the parking lot, and in accordance with parking spaces currently occupied. In general, other nearby data processing systems can have their control logic processed for a user's preferences (as defined in the MS), a group of nearby user's preferences, and/or situational locations (see U.S. Pat. Nos. 6,456,234; 6,731,238; 7,187,997 (Johnson) for "situational location" terminology) of nearby MSs.

Another advantage herein is an MS maintains history of hotspot locations detected for providing graphical indication of hotspot whereabouts. This information can be used by the MS user in guiding where a user should travel in the future for access to services at the hotspot. Hotspot growth prevents a database in being timely configured with new locations. The MS can learn where hotspots are located, as relevant to the particular MS. The hotspot information is instantly available to the MS.

A further advantage is for peer to peer proximity detection for identifying a peer service target within the MS vicinity. A peer service target can be acted upon by an MS within range, using an application at the MS. The complementary whereabouts of the peer service target and MS automatically notify the user of service availability. The user can then use the MS application for making a payment, or for performing an account transfer, account deposit, account deduction, or any other transaction associated with the peer service target.

Yet another advantage is for a MS to provide new self management capability such as automatically marking photographs taken with location information, a date/time stamp, and who was with the person taking the picture.

Yet another advantage is being alerted to nearby people needing assistance and nearby fire engines or police cars that need access to roads.

A further advantage is providing a MS platform for which new LBX features and functionality can be brought quickly to the marketplace. The platform caters to a full spectrum of users including highly technical software developers, novice

12

users, and users between those ranges. A rich programming environment is provided wherein whereabouts (WDR) information interchanged with other MSs in the vicinity causes triggering of privileged actions configured by users. The programming environment can be embedded in, or "plugged into", an existing software development environment, or provided on its own. A syntax may be specified with source code statements, XML, SQL database definitions, a datastream, or any other derivative of a well defined BNF grammar. A user friendly configuration environment is provided wherein whereabouts information interchanged with other MSs in the vicinity causes triggering of privileged actions configured by users. The platform is an event based environment wherein WDRs containing certain configured sought information are recognized at strategic processing paths for causing novel processing of actions. Events can be defined with complex expressions, and actions can be defined using homegrown executables, APIs, scripts, applications, a set of commands provided with the LBX platform, or any other executable processing. The LBX platform includes a variety of embodiments for charter and permission definitions including an internalized programmatic form, a SQL database form, a data record form, a datastream form, and a well defined BNF grammar for deriving other useful implementations (e.g. lex and yacc).

It is another advantage to support a countless number of privileges that can be configured, managed, and processed in a peer to peer manner between MSs. Any peer to peer feature or set of functionality can have a privilege associated to it for being granted from one user to another. It is also an advantage for providing a variety of embodiments for how to manage and maintain privileges in a network of MSs.

It is another advantage to support a complete set of options for charters that can be configured, managed, and processed in a peer to peer manner between MSs. Charters can become effective under a comprehensive set of conditions, expressions, terms, and operators. It is also an advantage for providing a variety of embodiments for how to manage and maintain charters in a network of MSs.

It is a further advantage for providing multithreaded communications of permission and charter information and transactions between MSs for well performing peer to peer interactions. Any signal spectrum for carrying out transmission and reception is candidate, depending on the variety of MS. In fact, different signaling wave spectrums, types, and protocols may be used in interoperating communications, or even for a single transaction, between MSs.

It is yet another advantage for increasing the range of the LN-expanse from a wireless vicinity to potentially infinite vicinity through other data processing (e.g. routing) equipment. While wireless proximity is used for governing automatic location determination, whereabouts information may be communicated between MSs great distances from each other provided there are privileges and/or charters in place making such whereabouts information relevant for the MS. Whereabouts information of others will not be maintained unless there are privileges in place to maintain it. Whereabouts information may not be shared with others if there have been no privileges granted to a potential receiving MS. Privileges can provide relevance to what whereabouts (WDR) information is of use, or should be processed, maintained, or acted upon.

Further features and advantages of the disclosure, as well as the structure and operation of various embodiments of the disclosure, are described in detail below with reference to the accompanying drawings. In the drawings, like reference numbers generally indicate identical, functionally similar,

13

and/or structurally similar elements. The drawing in which an element first appears is indicated by the leftmost digit(s) in the corresponding reference number, except that reference numbers 1 through 99 may be found on the first 4 drawings of FIGS. 1A through 1D. None of the drawings, discussions, or materials herein is to be interpreted as limiting to a particular embodiment. The broadest interpretation is intended. Other embodiments accomplishing same functionality are within the spirit and scope of this disclosure. It should be understood that information is presented by example and many embodiments exist without departing from the spirit and scope of this disclosure.

BRIEF DESCRIPTION OF THE DRAWINGS

There is no guarantee that there are descriptions in this specification for explaining every novel feature found in the drawings. The present disclosure will be described with reference to the accompanying drawings, wherein:

FIG. 1A depicts a preferred embodiment high level example componentization of a MS in accordance with the present disclosure;

FIG. 1B depicts a Location Based eXchanges (LBX) architectural illustration for discussing the present disclosure;

FIG. 1C depicts a Location Based Services (LBS) architectural illustration for discussing prior art of the present disclosure;

FIG. 1D depicts a block diagram of a data processing system useful for implementing a MS, ILM, DLM, centralized server, or any other data processing system disclosed herein;

FIG. 1E depicts a network illustration for discussing various deployments of whereabouts processing aspects of the present disclosure;

FIG. 2A depicts an illustration for describing automatic location of a MS through the MS coming into range of a stationary cellular tower;

FIG. 2B depicts an illustration for describing automatic location of a MS through the MS coming into range of some stationary antenna;

FIG. 2C depicts an illustration for discussing an example of automatically locating a MS through the MS coming into range of some stationary antenna;

FIG. 2D depicts a flowchart for describing a preferred embodiment of a service whereabouts update event of an antenna in-range detected MS when MS location awareness is monitored by a stationary antenna or cell tower;

FIG. 2E depicts a flowchart for describing a preferred embodiment of an MS whereabouts update event of an antenna in-range detected MS when MS location awareness is monitored by the MS;

FIG. 2F depicts a flowchart for describing a preferred embodiment of a procedure for inserting a Whereabouts Data Record (WDR) to an MS whereabouts data queue;

FIG. 3A depicts a locating by triangulation illustration for discussing automatic location of a MS;

FIG. 3B depicts a flowchart for describing a preferred embodiment of the whereabouts update event of a triangulated MS when MS location awareness is monitored by some remote service;

FIG. 3C depicts a flowchart for describing a preferred embodiment of the whereabouts update event of a triangulated MS when MS location awareness is monitored by the MS;

FIG. 4A depicts a locating by GPS triangulation illustration for discussing automatic location of a MS;

14

FIG. 4B depicts a flowchart for describing a preferred embodiment of the whereabouts update event of a GPS triangulated MS;

FIG. 5A depicts a locating by stationary antenna triangulation illustration for discussing automatic location of a MS;

FIG. 5B depicts a flowchart for describing a preferred embodiment of the whereabouts update event of a stationary antenna triangulated MS;

FIG. 6A depicts a flowchart for describing a preferred embodiment of a service whereabouts update event of a physically or logically connected MS;

FIG. 6B depicts a flowchart for describing a preferred embodiment of a MS whereabouts update event of a physically or logically connected MS;

FIGS. 7A, 7B and 7C depict a locating by image sensory illustration for discussing automatic location of a MS;

FIG. 7D depicts a flowchart for describing a preferred embodiment of graphically locating a MS, for example as illustrated by FIGS. 7A through 7C;

FIG. 8A heterogeneously depicts a locating by arbitrary wave spectrum illustration for discussing automatic location of a MS;

FIG. 8B depicts a flowchart for describing a preferred embodiment of locating a MS through physically contacting the MS;

FIG. 8C depicts a flowchart for describing a preferred embodiment of locating a MS through a manually entered whereabouts of the MS;

FIG. 9A depicts a table for illustrating heterogeneously locating a MS;

FIG. 9B depicts a flowchart for describing a preferred embodiment of heterogeneously locating a MS;

FIGS. 10A and 10B depict an illustration of a Locatable Network expanse (LN-Expanse) for describing locating of an ILM with all DLMs;

FIG. 10C depicts an illustration of a Locatable Network expanse (LN-Expanse) for describing locating of an ILM with an ILM and DLM;

FIGS. 10D, 10E, and 10F depict an illustration of a Locatable Network expanse (LN-Expanse) for describing locating of an ILM with all ILMs;

FIGS. 10G and 10H depict an illustration for describing the infinite reach of a Locatable Network expanse (LN-Expanse) according to MSs;

FIG. 10I depicts an illustration of a Locatable Network expanse (LN-Expanse) for describing a supervisory service;

FIG. 11A depicts a preferred embodiment of a Whereabouts Data Record (WDR) 1100 for discussing operations of the present disclosure;

FIGS. 11B, 11C and 11D depict an illustration for describing various embodiments for determining the whereabouts of an MS;

FIG. 11E depicts an illustration for describing various embodiments for automatically determining the whereabouts of an MS;

FIG. 12 depicts a flowchart for describing an embodiment of MS initialization processing;

FIGS. 13A through 13C depict an illustration of data processing system wireless data transmissions over some wave spectrum;

FIG. 14A depicts a flowchart for describing a preferred embodiment of MS LBX configuration processing;

FIG. 14B depicts a continued portion flowchart of FIG. 14A for describing a preferred embodiment of MS LBX configuration processing;

FIG. 15A depicts a flowchart for describing a preferred embodiment of DLM role configuration processing;

15

FIG. 15B depicts a flowchart for describing a preferred embodiment of ILM role configuration processing;

FIG. 15C depicts a flowchart for describing a preferred embodiment of a procedure for Manage List processing;

FIG. 16 depicts a flowchart for describing a preferred embodiment of NTP use configuration processing;

FIG. 17 depicts a flowchart for describing a preferred embodiment of WDR maintenance processing;

FIG. 18 depicts a flowchart for describing a preferred embodiment of a procedure for variable configuration processing;

FIG. 19 depicts an illustration for describing a preferred embodiment multithreaded architecture of peer interaction processing of a MS in accordance with the present disclosure;

FIG. 20 depicts a flowchart for describing a preferred embodiment of MS whereabouts broadcast processing;

FIG. 21 depicts a flowchart for describing a preferred embodiment of MS whereabouts collection processing;

FIG. 22 depicts a flowchart for describing a preferred embodiment of MS whereabouts supervisor processing;

FIG. 23 depicts a flowchart for describing a preferred embodiment of MS timing determination processing;

FIG. 24A depicts an illustration for describing a preferred embodiment of a thread request queue record;

FIG. 24B depicts an illustration for describing a preferred embodiment of a correlation response queue record;

FIG. 24C depicts an illustration for describing a preferred embodiment of a WDR request record;

FIG. 25 depicts a flowchart for describing a preferred embodiment of MS WDR request processing;

FIG. 26A depicts a flowchart for describing a preferred embodiment of MS whereabouts determination processing;

FIG. 26B depicts a flowchart for describing a preferred embodiment of processing for determining a highest possible confidence whereabouts;

FIG. 27 depicts a flowchart for describing a preferred embodiment of queue prune processing;

FIG. 28 depicts a flowchart for describing a preferred embodiment of MS termination processing;

FIG. 29A depicts a flowchart for describing a preferred embodiment of a process for starting a specified number of threads in a specified thread pool;

FIG. 29B depicts a flowchart for describing a preferred embodiment of a procedure for terminating the process started by FIG. 29A;

FIGS. 30A through 30B depict a preferred embodiment BNF grammar for variables, variable instantiations and common grammar for BNF grammars of permissions, groups and charters;

FIG. 30C depicts a preferred embodiment BNF grammar for permissions and groups;

FIGS. 30D through 30E depict a preferred embodiment BNF grammar for charters;

FIGS. 31A through 31E depict a preferred embodiment set of command and operand candidates for Action Data Records (ADRs) facilitating discussing associated parameters of the ADRs of the present disclosure;

FIG. 32A depicts a preferred embodiment of a National Language Support (NLS) directive command cross reference;

FIG. 32B depicts a preferred embodiment of a NLS directive operand cross reference;

FIG. 33A depicts a preferred embodiment American National Standards Institute (ANSI) X.409 encoding of the BNF grammar of FIGS. 30A through 30B for variables, variable instantiations and common grammar for BNF grammars of permissions and charters;

16

FIG. 33B depicts a preferred embodiment ANSI X.409 encoding of the BNF grammar of FIG. 30C for permissions and groups;

FIG. 33C depicts a preferred embodiment ANSI X.409 encoding of the BNF grammar of FIGS. 30D through 30E for charters;

FIGS. 34A through 34G depict preferred embodiment C programming source code header file contents, derived from the grammar of FIGS. 30A through 30E;

FIG. 35A depicts a preferred embodiment of a Granting Data Record (GDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 35B depicts a preferred embodiment of a Grant Data Record (GRTDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 35C depicts a preferred embodiment of a Generic Assignment Data Record (GADR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 35D depicts a preferred embodiment of a Privilege Data Record (PDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 35E depicts a preferred embodiment of a Group Data Record (GRPDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 36A depicts a preferred embodiment of a Description Data Record (DDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 36B depicts a preferred embodiment of a History Data Record (HDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 36C depicts a preferred embodiment of a Time specification Data Record (TDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 36D depicts a preferred embodiment of a Variable Data Record (VDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 37A depicts a preferred embodiment of a Charter Data Record (CDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 37B depicts a preferred embodiment of an Action Data Record (ADR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 37C depicts a preferred embodiment of a Parameter Data Record (PARMDR) for discussing operations of the present disclosure, derived from the grammar of FIGS. 30A through 30E;

FIG. 38 depicts a flowchart for describing a preferred embodiment of MS permissions configuration processing;

FIGS. 39A through 39B depict flowcharts for describing a preferred embodiment of MS user interface processing for permissions configuration;

FIGS. 40A through 40B depict flowcharts for describing a preferred embodiment of MS user interface processing for grants configuration;

17

FIGS. 41A through 41B depict flowcharts for describing a preferred embodiment of MS user interface processing for groups configuration;

FIG. 42 depicts a flowchart for describing a preferred embodiment of a procedure for viewing MS configuration information of others;

FIG. 43 depicts a flowchart for describing a preferred embodiment of a procedure for configuring MS acceptance of data from other MSs;

FIG. 44A depicts a flowchart for describing a preferred embodiment of a procedure for sending MS data to another MS;

FIG. 44B depicts a flowchart for describing a preferred embodiment of receiving MS configuration data from another MS;

FIG. 45 depicts a flowchart for describing a preferred embodiment of MS charters configuration processing;

FIGS. 46A through 46B depict flowcharts for describing a preferred embodiment of MS user interface processing for charters configuration;

FIGS. 47A through 47B depict flowcharts for describing a preferred embodiment of MS user interface processing for actions configuration;

FIGS. 48A through 48B depict flowcharts for describing a preferred embodiment of MS user interface processing for parameter information configuration;

FIG. 49A depicts an illustration for preferred permission data characteristics in the present disclosure LBX architecture;

FIG. 49B depicts an illustration for preferred charter data characteristics in the present disclosure LBX architecture;

FIGS. 50A through 50C depict an illustration of data processing system wireless data transmissions over some wave spectrum;

FIG. 51A depicts an example of a source code syntactical encoding embodiment of permissions, derived from the grammar of FIGS. 30A through 30E;

FIG. 51B depicts an example of a source code syntactical encoding embodiment of charters, derived from the grammar of FIGS. 30A through 30E;

FIG. 52 depicts another preferred embodiment C programming source code header file contents, derived from the grammar of FIGS. 30A through 30E;

FIG. 53 depicts a preferred embodiment of a Prefix Registry Record (PRR) for discussing operations of the present disclosure;

FIG. 54 depicts an example of an XML syntactical encoding embodiment of permissions and charters, derived from the BNF grammar of FIGS. 30A through 30E;

FIG. 55A depicts a flowchart for describing a preferred embodiment of MS user interface processing for Prefix Registry Record (PRR) configuration;

FIG. 55B depicts a flowchart for describing a preferred embodiment of Application Term (AppTerm) data modification;

FIG. 56 depicts a flowchart for appropriately processing an encoding embodiment of the BNF grammar of FIGS. 30A through 30E, in context for a variety of parser processing embodiments;

FIG. 57 depicts a flowchart for describing a preferred embodiment of WDR In-process Triggering Smarts (WITS) processing;

FIG. 58 depicts an illustration for granted data characteristics in the present disclosure LBX architecture;

18

FIG. 59 depicts a flowchart for describing a preferred embodiment of a procedure for enabling LBX features and functionality in accordance with a certain type of permissions;

FIG. 60 depicts a flowchart for describing a preferred embodiment of a procedure for performing LBX actions in accordance with a certain type of permissions;

FIG. 61 depicts a flowchart for describing a preferred embodiment of performing processing in accordance with configured charters;

FIG. 62 depicts a flowchart for describing a preferred embodiment of a procedure for performing an action corresponding to a configured command;

FIG. 63A depicts a flowchart for describing a preferred embodiment of a procedure for Send command action processing;

FIGS. 63B-1 through 63B-7 depicts a matrix describing how to process some varieties of the Send command;

FIG. 63C depicts a flowchart for describing one embodiment of a procedure for Send command action processing, as derived from the processing of FIG. 63A;

FIG. 64A depicts a flowchart for describing a preferred embodiment of a procedure for Notify command action processing;

FIGS. 64B-1 through 64B-4 depicts a matrix describing how to process some varieties of the Notify command;

FIG. 64C depicts a flowchart for describing one embodiment of a procedure for Notify command action processing, as derived from the processing of FIG. 64A;

FIG. 65A depicts a flowchart for describing a preferred embodiment of a procedure for Compose command action processing;

FIGS. 65B-1 through 65B-7 depicts a matrix describing how to process some varieties of the Compose command;

FIG. 65C depicts a flowchart for describing one embodiment of a procedure for Compose command action processing, as derived from the processing of FIG. 65A;

FIG. 66A depicts a flowchart for describing a preferred embodiment of a procedure for Connect command action processing;

FIGS. 66B-1 through 66B-2 depicts a matrix describing how to process some varieties of the Connect command;

FIG. 66C depicts a flowchart for describing one embodiment of a procedure for Connect command action processing, as derived from the processing of FIG. 66A;

FIG. 67A depicts a flowchart for describing a preferred embodiment of a procedure for Find command action processing;

FIGS. 67B-1 through 67B-13 depicts a matrix describing how to process some varieties of the Find command;

FIG. 67C depicts a flowchart for describing one embodiment of a procedure for Find command action processing, as derived from the processing of FIG. 67A;

FIG. 68A depicts a flowchart for describing a preferred embodiment of a procedure for Invoke command action processing;

FIGS. 68B-1 through 68B-5 depicts a matrix describing how to process some varieties of the Invoke command;

FIG. 68C depicts a flowchart for describing one embodiment of a procedure for Invoke command action processing, as derived from the processing of FIG. 68A;

FIG. 69A depicts a flowchart for describing a preferred embodiment of a procedure for Copy command action processing;

FIGS. 69B-1 through 69B-14 depicts a matrix describing how to process some varieties of the Copy command;

FIG. 69C depicts a flowchart for describing one embodiment of a procedure for Copy command action processing, as derived from the processing of FIG. 69A;

FIG. 70A depicts a flowchart for describing a preferred embodiment of a procedure for Discard command action processing;

FIGS. 70B-1 through 70B-11 depicts a matrix describing how to process some varieties of the Discard command;

FIG. 70C depicts a flowchart for describing one embodiment of a procedure for Discard command action processing, as derived from the processing of FIG. 70A;

FIG. 71A depicts a flowchart for describing a preferred embodiment of a procedure for Move command action processing;

FIGS. 71B-1 through 71B-14 depicts a matrix describing how to process some varieties of the Move command;

FIG. 71C depicts a flowchart for describing one embodiment of a procedure for Move command action processing, as derived from the processing of FIG. 71A;

FIG. 72A depicts a flowchart for describing a preferred embodiment of a procedure for Store command action processing;

FIGS. 72B-1 through 72B-5 depicts a matrix describing how to process some varieties of the Store command;

FIG. 72C depicts a flowchart for describing one embodiment of a procedure for Store command action processing, as derived from the processing of FIG. 72A;

FIG. 73A depicts a flowchart for describing a preferred embodiment of a procedure for Administration command action processing;

FIGS. 73B-1 through 73B-7 depicts a matrix describing how to process some varieties of the Administration command;

FIG. 73C depicts a flowchart for describing one embodiment of a procedure for Administration command action processing, as derived from the processing of FIG. 73A;

FIG. 74A depicts a flowchart for describing a preferred embodiment of a procedure for Change command action processing;

FIG. 74C depicts a flowchart for describing one embodiment of a procedure for Change command action processing, as derived from the processing of FIG. 74A;

FIG. 75A depicts a flowchart for describing a preferred embodiment of a procedure for sending data to a remote MS;

FIG. 75B depicts a flowchart for describing a preferred embodiment of processing for receiving execution data from another MS;

FIG. 76 depicts a flowchart for describing a preferred embodiment of processing a special Term information paste action at a MS;

FIG. 77 depicts a flowchart for describing a preferred embodiment of configuring data to be maintained to WDR Application Fields; and

FIG. 78 depicts a simplified example of an XML syntactical encoding embodiment of a profile for the profile section of WDR Application Fields.

DETAILED DESCRIPTION OF THE INVENTION

With reference now to detail of the drawings, the present disclosure is described. Obvious error handling is omitted from the flowcharts in order to focus on the key aspects of the present disclosure. Obvious error handling includes database I/O errors, field validation errors, errors as the result of database table/data constraints or unique keys, data access errors, communications interface errors or packet collision, hardware failures, checksum validations, bit error detections/cor-

rections, and any other error handling as well known to those skilled in the relevant art in context of this disclosure. A semicolon may be used in flowchart blocks to represent, and separate, multiple blocks of processing within a single physical block. This allows simpler flowcharts with less blocks in the drawings by placing multiple blocks of processing description in a single physical block of the flowchart. Flowchart processing is intended to be interpreted in the broadest sense by example, and not for limiting methods of accomplishing the same functionality. Preferably, field validation in the flowcharts checks for SQL injection attacks, communications protocol sniff and hack attacks, preventing of spoofing MS addresses, syntactical appropriateness, and semantics errors where appropriate. Disclosed user interface processing and/or screenshots are also preferred user interface processing examples that can be implemented in other ways without departing from the spirit and scope of this disclosure. Alternative user interfaces (since this disclosure is not to be limiting) will use similar mechanisms, but may use different mechanisms without departing from the spirit and scope of this disclosure.

Locational terms such as whereabouts, location, position, area, destination, perimeter, radius, geofence, situational location, or any other related two or three dimensional locational term used herein to described position(s) and/or locations and/or whereabouts is to be interpreted in the broadest sense. Location field 1100c may include an area (e.g. on earth), a point (e.g. on earth), or a three dimensional bounds in space. In another example, a radius may define a sphere in space, rather than a circle in a plane. In some embodiments, a planet field forms part of the location (e.g. Earth, Mars, etc as part of field 1100c) for which other location information (e.g. latitude and longitude on Mars also part of field 1100c) is relative. In some embodiments, elevations (or altitudes) from known locatable point(s), distances from origin(s) in the universe, etc. can denote where exactly is a point of three dimensional space, or three dimensional sphere, area, or solid, is located. That same point can provide a mathematical reference to other points of the solid area/region in space. Descriptions for angles, pitches, rotations, etc from some reference point(s) may be further provided. Three dimensional areas/regions include a conical shape, cubical shape, spherical shape, pyramidal shape, irregular shapes, or any other shape either manipulated with a three dimensional graphic interface, or with mathematical model descriptions. Areas/regions in space can be occupied by a MS, passed through (e.g. by a traveler) by a MS, or referenced through configuration by a MS. In a three dimensional embodiment, nearby/nearness is determined in terms of three dimensional information, for example, a spherical radius around one MS intersecting a spherical radius around another MS. In a two dimensional embodiment, nearby/nearness is determined in terms of two dimensional information, for example, a circular radius around one MS intersecting a circular radius around another MS. Points can be specified as a point in a x-y-z plane, a point in polar coordinates, or the like, perhaps the center of a planet (e.g. Earth) or the Sun, some origin in the Universe, or any other origin for distinctly locating three dimensional location(s), positions, or whereabouts in space. Elevation (e.g. for earth, or some other planet, etc) may be useful to the three dimensional point of origin, and/or for the three dimensional region in space. A region in space may also be specified with connecting x-y-z coordinates together to bound the three dimensional region in space. There are many methods for representing a location (field 1100c) without departing from the spirit and scope of this disclosure. MSs, for example as carried by users, can travel by airplane through three dimen-

21

sional areas/regions in space, or travel under the sea through three dimensional regions in space.

Various embodiments of communications between MSs, or an MS and service(s), will share channels (e.g. frequencies) to communicate, depending on when in effect. Sharing a channel will involve carrying recognizable and processable signature to distinguish transmissions for carrying data. Other embodiments of communications between MSs, or an MS and service(s), will use distinct channels to communicate, depending on when in effect. The number of channels that can be concurrently listened on and/or concurrently transmitted on by a data processing system will affect which embodiments are preferred. The number of usable channels will also affect which embodiments are preferred. This disclosure avoids unnecessary detail in different communication channel embodiments so as to not obfuscate novel material. Independent of various channel embodiments within the scope and spirit of the present disclosure, MSs communicate with other MSs in a peer to peer manner, in some aspects like automated walkie-talkies.

Novel features disclosed herein need not be provided as all or none. Certain features may be isolated in some MS embodiments, or may appear as any subset of features and functionality in other embodiments.

Location Based eXchanges (LBX) Architecture

FIG. 1A depicts a preferred embodiment high level example componentization of a MS in accordance with the present disclosure. A MS 2 includes processing behavior referred to as LBX Character 4 and Other Character 32. LBX character 4 provides processing behavior causing MS 2 to take on the character of a Location Based Exchange (LBX) MS according to the present disclosure. Other Character 32 provides processing behavior causing MS to take on character of prior art MSs in context of the type of MS. Other character 32 includes at least other processing code 34, other processing data 36, and other resources 38, all of which are well known to those skilled in the art for prior art MSs. In some embodiments, LBX character 4 components may, or may not, make use of other character 32 components 34, 36, and 38. Other character 32 components may, or may not, make use of LBX character 4 components 6 through 30.

LBX character 4 preferably includes at least Peer Interaction Processing (PIP) code 6, Peer Interaction Processing (PIP) data 8, self management processing code 18, self management processing data 20, WDR queue 22, send queue 24, receive queue 26, service informant code 28, and LBX history 30. Peer interaction processing (PIP) code 6 comprises executable code in software, firmware, or hardware form for carrying out LBX processing logic of the present disclosure when interacting with another MS. Peer interaction processing (PIP) data 8 comprises data maintained in any sort of memory of MS 2, for example hardware memory, flash memory, hard disk memory, a removable memory device, or any other memory means accessible to MS 2. PIP data 8 contains intelligence data for driving LBX processing logic of the present disclosure when interacting with other MSs. Self management processing code 18 comprises executable code in software, firmware, or hardware form for carrying out the local user interface LBX processing logic of the present disclosure. Self management processing data 20 contains intelligence data for driving processing logic of the present disclosure as disclosed for locally maintained LBX features. WDR queue 22 contains Whereabouts Data Records (WDRs) 1100, and is a First-In-First-Out (FIFO) queue when considering housekeeping for pruning the queue to a reasonable

22

trailing history of inserted entries (i.e. remove stale entries). WDR queue 22 is preferably designed with the ability of queue entry retrieval processing similar to Standard Query Language (SQL) querying, wherein one or more entries can be retrieved by querying with a conditional match on any data field(s) of WDR 1100 and returning lists of entries in order by an ascending or descending key on one or any ascending/descending ordered list of key fields.

All disclosed queues (e.g. 22, 24, 26, 1980 and 1990 (See FIG. 19)) are implemented with an appropriate thread-safe means of queue entry peeking (makes copy of sought queue entry without removing), discarding, retrieval, insertion, and queue entry field sorted search processing. Queues are understood to have an associated implicit semaphore to ensure appropriate synchronous access to queue data in a multi-threaded environment to prevent data corruption and misuse. Such queue interfaces are well known in popular operating systems. In MS operating system environments which do not have an implicit semaphore protected queue scheme, queue accesses in the present disclosure flowcharts are to be understood to have a previous request to a queue-assigned semaphore lock prior to queue access, and a following release of the semaphore lock after queue access. Operating systems without semaphore control may use methods to achieve similar thread-safe synchronization functionality. Queue functionality may be accomplished with lists, arrays, databases (e.g. SQL) and other methodologies without departing from the spirit and scope of queue descriptions herein.

Queue 22 alternate embodiments may maintain a plurality of WDR queues which segregate WDRs 1100 by field(s) values to facilitate timely processing. WDR queue 22 may be at least two (2) separate queues: one for maintaining the MS 2 whereabouts, and one for maintaining whereabouts of other MSs. WDR queue 22 may be a single instance WDR 1100 in some embodiments which always contains the most current MS 2 whereabouts for use by MS 2 applications (may use a sister queue 22 for maintaining WDRs from remote MSs). At least one entry is to be maintained to WDR queue 22 at all times for MS 2 whereabouts.

Send queue 24 (Transmit (Tx) queue) is used to send communications data, for example as intended for a peer MS within the vicinity (e.g. nearby as indicated by maximum range 1306) of the MS 2. Receive queue 26 (Receive (Rx) queue) is used to receive communications data, for example from peer MSs within the vicinity (e.g. nearby as indicated by maximum range 1306) of the MS 2. Queues 24 and 26 may also each comprise a plurality of queues for segregating data thereon to facilitate performance in interfacing to the queues, in particular when different queue entry types and/or sizes are placed on the queue. A queue interface for sending/receiving data to/from the MS is optimal in a multi-threaded implementation to isolate communications transport layers to processing behind the send/receive queue interfaces, but alternate embodiments may send/receive data directly from a processing thread disclosed herein. Queues 22, 24, and/or 26 may be embodied as a purely data form, or SQL database, maintained at MS 2 in persistent storage, memory, or any other storage means. In some embodiments, queues 24 and 26 are not necessary since other character 32 will already have accessible resources for carrying out some LBX character 4 processing.

Queue embodiments may contain fixed length records, varying length records, pointers to fixed length records, or pointers to varying length records. If pointers are used, it is assumed that pointers may be dynamically allocated for record storage on insertions and freed upon record use after discards or retrievals.

23

As well known to those skilled in the art, when a thread sends on a queue **24** in anticipation of a corresponding response, there is correlation data in the data sent which is sought in a response received by a thread at queue **26** so the sent data is correlated with the received data. In a preferred embodiment, correlation is built using a round-robin generated sequence number placed in data for sending along with a unique MS identifier (MS ID). If data is not already encrypted in communications, the correlation can be encrypted. While the unique MS identifier (MS ID) may help the MS identify which (e.g. wireless) data is destined for it, correlation helps identify which data at the MS caused the response. Upon receipt of data from a responder at queue **26**, correlation processing uses the returned correlation (e.g. field **1100m**) to correlate the sent and received data. In preferred embodiments, the sequence number is incremented each time prior to use to ensure a unique number, otherwise it may be difficult to know which data received is a response to which data was sent, in particular when many data packets are sent within seconds. When the sequence number reaches a maximum value (e.g. $2^{**32}-1$), then it is round-robbined to 0 and is incremented from there all over again. This assures proper correlation of data between the MS and responders over time. There are other correlation schemes (e.g. signatures, random number generation, checksum counting, bit patterns, date/time stamp derivatives) to accomplish correlation functionality. If send and receive queues of Other Character **32** are used, then correlation can be used in a similar manner to correlate a response with a request (i.e. a send with a receipt).

There may be good reason to conceal the MS ID when transmitting it wirelessly. In this embodiment, the MS ID is a dependable and recognizable derivative (e.g. a pseudo MS ID) that can be detected in communications traffic by the MS having the pseudo MS ID, while concealing the true MS ID. This would conceal the true MS ID from would-be hackers sniffing wireless protocol. The derivative can always be reliably the same for simplicity of being recognized by the MS while being difficult to associate to a particular MS. Further still, a more protected MS ID (from would-be hackers that take time to deduce how an MS ID is scrambled) can itself be a dynamically changing correlation anticipated in forthcoming communications traffic, thereby concealing the real MS ID (e.g. phone number or serial number), in particular when anticipating traffic in a response, yet still useful for directing responses back to the originating MS (with the pseudo MS ID (e.g. correlation)). A MS would know which correlation is anticipated in a response by saving it to local storage for use until it becomes used (i.e. correlated in a matching response), or becomes stale. In another embodiment, a correlation response queue (like CR queue **1990**) can be deployed to correlate responses with requests that contain different correlations for pseudo MS IDs. In all embodiments, the MS ID (or pseudo MS ID) of the present disclosure should enable targeting communications traffic to the MS.

Service informant code **28** comprises executable code in software, firmware, or hardware form for carrying out of informing a supervisory service. The present disclosure does not require a connected web service, but there are features for keeping a service informed with activities of MS LBX. Service informant code **28** can communicate as requested any data **8, 20, 22, 24, 26, 30, 36, 38**, or any other data processed at MS **2**.

LBX history **30** contains historical data useful in maintaining at MS **2**, and possibly useful for informing a supervisory service through service informant code **28**. LBX History **30** preferably has an associated thread of processing for keeping it pruned to the satisfaction of a user of MS **2** (e.g. prefers to

24

keep last 15 days of specified history data, and 30 days of another specified history data, etc). With a suitable user interface to MS **2**, a user may browse, manage, alter, delete, or add to LBX History **30** as is relevant to processing described herein. Service informant code **28** may be used to cause sending of an outbound email, SMS message, outbound data packet, or any other outbound communication in accordance with LBX of the MS.

PIP data **8** preferably includes at least permissions **10**, charters **12**, statistics **14**, and a service directory **16**. Permissions **10** are configured to grant permissions to other MS users for interacting the way the user of MS **2** desires for them to interact. Therefore, permissions **10** contain permissions granted from the MS **2** user to other MS users. In another embodiment, permissions **10** additionally, or alternatively, contain permissions granted from other MS users to the MS **2** user. Permissions are maintained completely local to the MS **2**. Charters **12** provide LBX behavior conditional expressions for how MSs should interact with MS **2**. Charters **12** are configured by the MS **2** user for other MS users. In another embodiment, charters **12** additionally, or alternatively, are configured by other MS users for the MS **2** user. Some charters expressions depend on permissions **10**. Statistics **14** are maintained at MS **2** for reflecting peer (MS) to peer (MS) interactions of interest that occurred at MS **2**. In another embodiment, statistics **14** additionally, or alternatively, reflect peer (MS) to peer (MS) interactions that occurred at other MSs, preferably depending on permissions **10**. Service informant code **28** may, or may not, inform a service of statistics **14** maintained. Service directory **16** includes routing entries for how MS **2** will find a sought service, or how another MS can find a sought service through MS **2**.

In some embodiments, any code (e.g. **6, 18, 28, 34, 38**) can access, manage, use, alter, or discard any data (e.g. **8, 20, 22, 24, 26, 30, 36, 38**) of any other component in MS **2**. Other embodiments may choose to keep processing of LBX character **4** and other character **32** disjoint from each other. Rectangular component boundaries are logical component representations and do not have to delineate who has access to what. MS (also MSs) references discussed herein in context for the new and useful features and functionality disclosed is understood to be an MS **2** (MSs **2**).

FIG. **1B** depicts a Location Based exchanges (LBX) architectural illustration for discussing the present disclosure. LBX MSs are peers to each other for locational features and functionality. An MS **2** communicates with other MSs without requiring a service for interaction. For example, FIG. **1B** depicts a wireless network **40** of five (5) MSs. Each is able to directly communicate with others that are in the vicinity (e.g. nearby as indicated by maximum range **1306**). In a preferred embodiment, communications are limited reliability wireless broadcast datagrams having recognizable data packet identifiers. In another embodiment, wireless communications are reliable transport protocols carried out by the MSs, such as TCP/IP. In other embodiments, usual communications data associated with other character **32** include new data (e.g. Communications Key **1304**) in transmissions for being recognized by MSs within the vicinity. For example, as an MS conventionally communicates, LBX data is added to the protocol so that other MSs in the vicinity can detect, access, and use the data. The advantage to this is that as MSs use wireless communications to carry out conventional behavior, new LBX behavior is provided by simply incorporating additional information (e.g. Communications Key **1304**) to existing communications.

Regardless of the embodiment, an MS **2** can communicate with any of its peers in the vicinity using methods described

25

below. Regardless of the embodiment, a communication path **42** between any two MSs is understood to be potentially bidirectional, but certainly at least unidirectional. The bidirectional path **42** may use one communications method for one direction and a completely different communications method for the other, but ultimately each can communicate to each other. When considering that a path **42** comprises two unidirectional communications paths, there are $N*(N-1)$ unidirectional paths for N MSs in a network **40**. For example, 10 MSs results in 90 (i.e. $10*9$) one way paths of communications between all 10 MSs for enabling them to talk to each other. Sharing of the same signaling channels is preferred to minimize the number of MS threads listening on distinct channels. Flowcharts are understood to process at incredibly high processing speeds, in particular for timely communications processing. While the MSs are communicating wirelessly to each other, path **42** embodiments may involve any number of intermediary systems or communications methods, for example as discussed below with FIG. 1E.

FIG. 1C depicts a Location Based Services (LBS) architectural illustration for discussing prior art of the present disclosure. In order for a MS to interact for LBS with another MS, there is service architecture **44** for accomplishing the interaction. For example, to detect that MS **1** is nearby MS N , the service is indispensably involved in maintaining data and carrying out processing. For example, to detect that MS **1** is arriving to, or departing from, a geofenced perimeter area configured by MS N , the service was indispensably involved in maintaining data and carrying out processing. For example, for MS N to locate MS **1** on a live map, the service was indispensably involved in maintaining data and carrying out processing. In another example, to grant and revoke permissions from MS **1** to MS N , the service was indispensably involved in maintaining data and carrying out processing. While it is advantageous to require a single bidirectional path **46** for each MS (i.e. two unidirectional communications paths; $(2*N)$ unidirectional paths for N MSs), there are severe requirements for service(s) when there are lots of MSs (i.e. when N is large). Wireless MSs have advanced beyond cell phones, and are capable of housing significant parallel processing, processing speed, increased wireless transmission speeds and distances, increased memory, and richer features.

FIG. 1D depicts a block diagram of a data processing system useful for implementing a MS, ILM, DLM, centralized server, or any other data processing system described herein. An MS **2** is a data processing system **50**. Data processing system **50** includes at least one processor **52** (e.g. Central Processing Unit (CPU)) coupled to a bus **54**. Bus **54** may include a switch, or may in fact be a switch **54** to provide dedicated connectivity between components of data processing system **50**. Bus (and/or switch) **54** is a preferred embodiment coupling interface between data processing system **50** components. The data processing system **50** also includes main memory **56**, for example, random access memory (RAM). Memory **56** may include multiple memory cards, types, interfaces, and/or technologies. The data processing system **50** may include secondary storage devices **58** such as persistent storage **60**, and/or removable storage device **62**, for example as a compact disk, floppy diskette, USB flash, or the like, also connected to bus (or switch) **54**. In some embodiments, persistent storage devices could be remote to the data processing system **50** and coupled through an appropriate communications interface. Persistent storage **60** may include flash memory, disk drive memory, magnetic, charged, or bubble storage, and/or multiple interfaces and/or technologies, perhaps in software interface form of variables, a database, shared memory, etc.

26

The data processing system **50** may also include a display device interface **64** for driving a connected display device (not shown). The data processing system **50** may further include one or more input peripheral interface(s) **66** to input devices such as a keyboard, keypad, Personal Digital Assistant (PDA) writing implements, touch interfaces, mouse, voice interface, or the like. User input (“user input”, “user events” and “user actions” used interchangeably) to the data processing system are inputs accepted by the input peripheral interface(s) **66**. The data processing system **50** may still further include one or more output peripheral interface(s) **68** to output devices such as a printer, facsimile device, or the like. Output peripherals may also be available via an appropriate interface.

Data processing system **50** will include a communications interface(s) **70** for communicating to another data processing system **72** via analog signal waves, digital signal waves, infrared proximity, copper wire, optical fiber, or other wave spectrums described herein. A MS may have multiple communications interfaces **70** (e.g. cellular connectivity, 802.x, etc). Other data processing system **72** may be an MS. Other data processing system **72** may be a service. Other data processing system **72** is a service data processing system when MS **50** communicates to other data processing system **72** by way of service informant code **28**. In any case, the MS and other data processing system are said to be interoperating when communicating.

Data processing system programs (also called control logic) may be completely inherent in the processor(s) **52** being a customized semiconductor, or may be stored in main memory **56** for execution by processor(s) **52** as the result of a read-only memory (ROM) load (not shown), or may be loaded from a secondary storage device into main memory **56** for execution by processor(s) **52**. Such programs, when executed, enable the data processing system **50** to perform features of the present disclosure as discussed herein. Accordingly, such data processing system programs represent controllers of the data processing system.

In some embodiments, the disclosure is directed to a control logic program product comprising at least one processor **52** having control logic (software, firmware, hardware microcode) stored therein. The control logic, when executed by processor(s) **52**, causes the processor(s) **52** to provide functions of the disclosure as described herein. In another embodiment, this disclosure is implemented primarily in hardware, for example, using a prefabricated component state machine (or multiple state machines) in a semiconductor element such as a processor **52**.

Those skilled in the art will appreciate various modifications to the data processing system **50** without departing from the spirit and scope of this disclosure. A data processing system, and more particularly a MS, preferably has capability for many threads of simultaneous processing which provide control logic and/or processing. These threads can be embodied as time sliced threads of processing on a single hardware processor, multiple processors, multi-core processors, Digital Signal Processors (DSPs), or the like, or combinations thereof. Such multi-threaded processing can concurrently serve large numbers of concurrent MS tasks. Concurrent processing may be provided with distinct hardware processing and/or as appropriate software driven time-sliced thread processing. Those skilled in the art recognize that having multiple threads of execution on an MS is accomplished in many different ways without departing from the spirit and scope of this disclosure. This disclosure strives to deploy software to

existing MS hardware configurations, but the disclosed software can be deployed as burned-in microcode to new hardware of MSs.

Data processing aspects of drawings/flowcharts are preferably multi-threaded so that many MSs and applicable data processing systems are interfaced with in a timely and optimal manner. Data processing system **50** may also include its own clock mechanism (not shown), if not an interface to an atomic clock or other clock mechanism, to ensure an appropriately accurate measurement of time in order to appropriately carry out processing described below. In some embodiments, Network Time Protocol (NTP) is used to keep a consistent universal time for MSs and other data processing systems in communications with MSs. This is most advantageous to prevent unnecessary round-tripping of data between data processing systems to determine timing (e.g. Time Difference of Arrival (TDOA)) measurements. A NTP synchronized date/time stamp maintained in communications is compared by a receiving data processing system for comparing with its own NTP date/time stamp to measure TOA (time of arrival (i.e. time taken to arrive)). Of course, in the absence of NTP used by the sender and receiver, TOA is also calculated in a bidirectional transmission using correlation. In this disclosure, TOA measurements from one location technology are used for triangulating with TOA measurements from another location technology, not just for determining “how close”. Therefore, TDOA terminology is generally used herein to refer to the most basic TOA measurement of a wave spectrum signal being the difference between when it was sent and when it was received. TDOA is also used to describe using the difference of such measurements to locate (triangulate). NTP use among participating systems has the advantage of a single unidirectional broadcast data packet containing all a receiving system requires to measure TDOA, by knowing when the data was sent (date/time stamp in packet) and when the data was received (signal detected and processed by receiving system). A NTP clock source (e.g. atomic clock) used in a network is to be reasonably granular to carry out measurements, and ensures participating MSs are updated timely according to anticipated time drifts of their own clocks. There are many well known methods for accomplishing NTP, some which require dedicated thread(s) for NTP processing, and some which use certain data transmitted to and from a source to keep time in synch.

Those skilled in the art recognize that NTP accuracy depends on participating MS clocks and processing timing, as well as time server source(s). Radio wave connected NTP time server(s) is typically accurate to as granular as 1 millisecond. Global Positioning System (GPS) time servers provide accuracy as granular as 50 microseconds. GPS timing receivers provide accuracy to around 100 nanoseconds, but this may be reduced by timing latencies in time server operating systems. With advancements in hardware, microcode, and software, obvious improvements are being made to NTP. In NTP use embodiments of this disclosure, an appropriate synchronization of time is used for functional interoperability between MSs and other data processing systems using NTP. NTP is not required in this disclosure, but it is an advantage when in use.

LBX Directly Located Mobile Data Processing Systems (DLMs)

FIG. 1E depicts a network illustration for discussing various deployments of whereabouts processing aspects of the present disclosure. In some embodiments, a cellular network cluster **102** and cellular network cluster **104** are parts of a

larger cellular network. Cellular network cluster **102** contains a controller **106** and a plurality of base stations, shown generally as base stations **108**. Each base station covers a single cell of the cellular network cluster, and each base station **108** communicates through a wireless connection with the controller **106** for call processing, as is well known in the art. Wireless devices communicate via the nearest base station (i.e. the cell the device currently resides in), for example base station **108b**. Roaming functionality is provided when a wireless device roams from one cell to another so that a session is properly maintained with proper signal strength. Controller **106** acts like a telephony switch when a wireless device roams across cells, and it communicates with controller **110** via a wireless connection so that a wireless device can also roam to other clusters over a larger geographical area. Controller **110** may be connected to a controller **112** in a cellular cluster through a physical connection, for example, copper wire, optical fiber, or the like. This enables cellular clusters to be great distances from each other. Controller **112** may in fact be connected with a physical connection to its base stations, shown generally as base stations **114**. Base stations may communicate directly with the controller **112**, for example, base station **114e**. Base stations may communicate indirectly to the controller **112**, for example base station **114a** by way of base station **114d**. It is well known in the art that many options exist for enabling interoperating communications between controllers and base stations for the purpose of managing a cellular network. A cellular network cluster **116** may be located in a different country. Base controller **118** may communicate with controller **110** through a Public Service Telephone Network (PSTN) by way of a telephony switch **120**, PSTN **122**, and telephony switch **124**, respectively. Telephony switch **120** and telephony switch **124** may be private or public. In one cellular network embodiment of the present disclosure, the services execute at controllers, for example controller **110**. In some embodiments, the MS includes processing that executes at a wireless device, for example mobile laptop computer **126**, wireless telephone **128**, a personal digital assistant (PDA) **130**, an iPhone **170**, or the like. As the MS moves about, positional attributes are monitored for determining location. The MS may be handheld, or installed in a moving vehicle. Locating a wireless device using wireless techniques such as Time Difference of Arrival (TDOA) and Angle Of Arrival (AOA) are well known in the art. The service may also execute on a server computer accessible to controllers, for example server computer **132**, provided an appropriate timely connection exists between cellular network controller(s) and the server computer **132**. Wireless devices (i.e. MSs) are preferably known by a unique identifier, for example a phone number, caller id, device identifier, or like appropriate unique handle.

In another embodiment of the present disclosure, GPS satellites such as satellite **134**, satellite **136**, and satellite **138** provide information, as is well known in the art, to GPS devices on earth for triangulation locating of the GPS device. In this embodiment, a MS has integrated GPS functionality so that the MS monitors its positions. The MS is preferably known by a unique identifier, for example a phone number, caller id, device identifier, or like appropriate unique handle.

In yet another embodiment of the present disclosure, a physically connected device, for example, telephone **140**, computer **142**, PDA **144**, telephone **146**, and fax machine **148**, may be newly physically connected to a network. Each is a MS, although the mobility is limited. Physical connections include copper wire, optical fiber, USB, or any other physical connection, by any communications protocol thereon. Devices are preferably known by a unique identifier, for

example a phone number, caller id, device identifier, physical or logical network address, or like appropriate unique handle. The MS is detected for being newly located when physically connected. A service can be communicated to upon detecting connectivity. The service may execute at an Automatic Response Unit (ARU) 150, a telephony switch, for example telephony switch 120, a web server 152 (for example, connected through a gateway 154), or a like data processing system that communicates with the MS in any of a variety of ways as well known to those skilled in the art. MS detection may be a result of the MS initiating a communication with the service directly or indirectly. Thus, a user may connect his laptop to a hotel network, initiate a communication with the service, and the service determines that the user is in a different location than the previous communication. A local area network (LAN) 156 may contain a variety of connected devices, each an MS that later becomes connected to a local area network 158 at a different location, such as a PDA 160, a server computer 162, a printer 164, an internet protocol telephone 166, a computer 168, or the like. Hard copy presentation could be made to printer 164 and fax 148.

Current technology enables devices to communicate with each other, and other systems, through a variety of heterogeneous system and communication methods. Current technology allows executable processing to run on diverse devices and systems. Current technology allows communications between the devices and/or systems over a plethora of methodologies at close or long distance. Many technologies also exist for automatic locating of devices. It is well known how to have an interoperating communications system that comprises a plurality of individual systems communicating with each other with one or more protocols. As is further known in the art of developing software, executable processing of the present disclosure may be developed to run on a particular target data processing system in a particular manner, or customized at install time to execute on a particular data processing system in a particular manner.

FIG. 2A depicts an illustration for describing automatic location of a MS, for example a DLM 200, through the MS coming into range of a stationary cellular tower. A DLM 200, or any of a variety of MSs, travels within range of a cell tower, for example cell tower 108b. The known cell tower location is used to automatically detect the location of the DLM 200. In fact, any DLM that travels within the cell served by cell tower 108b is identified as the location of cell tower 108b. The confidence of a location of a DLM 200 is low when the cell coverage of cell tower 108b is large. In contrast, the confidence of a location of a DLM 200 is higher when the cell coverage of cell tower 108b is smaller. However, depending on the applications locating DLMs using this method, the locating can be quite acceptable. Location confidence is improved with a TDOA measurement for the elapsed time of communication between DLM 200 and cell tower to determine how close the MS is to the cell tower. Cell tower 108b can process all locating by itself, or with interoperability to other services as connected to cell tower 108b in FIG. 1E. Cell tower 108b can communicate the location of DLM 200 to a service, to the DLM 200, to other MSs within its coverage area, any combination thereof, or to any connected data processing system, or MS, of FIG. 1E.

FIG. 2B depicts an illustration for describing automatic location of a MS, for example a DLM 200, through the MS coming into range of some stationary antenna. DLM 200, or any of a variety of MSs, travels within range of a stationary antenna 202 that may be mounted to a stationary object 204. The known antenna location is used to automatically detect the location of the DLM 200. In fact, any DLM that travels

within the coverage area served by antenna 202 is identified as the location of antenna 202. The confidence of a location of a DLM 200 is low when the antenna coverage area of antenna 202 is large. In contrast, the confidence of a location of a DLM 200 is higher when the antenna coverage area of antenna 202 is smaller. However, depending on the applications locating DLMs using this method, the locating can be quite acceptable. Location confidence is improved with a TDOA measurement for the elapsed time of communication between DLM 200 and a particular antenna to determine how close the MS is to the antenna. Antenna 202 can process all locating by itself (with connected data processing system (not shown) as well known to those skilled in the art), or with interoperability to other services as connected to antenna 202, for example with connectivity described in FIG. 1E. Antenna 202 can be used to communicate the location of DLM 200 to a service, to the DLM 200, to other MSs within its coverage area, any combination thereof, or to any connected data processing system, or MS, of FIG. 1E.

FIG. 2C depicts an illustration for discussing an example of automatically locating a MS, for example a DLM 200, through the MS coming into range of some stationary antenna. DLM 200, or any of a variety of MSs, travels within range of a stationary antenna 212 that may be mounted to a stationary object, such as building 210. The known antenna location is used to automatically detect the location of the DLM 200. In fact, any DLM that travels within the coverage area served by antenna 212 is identified as the location of antenna 212. The confidence of a location of a DLM 200 is low when the antenna coverage area of antenna 212 is large. In contrast, the confidence of a location of a DLM 200 is higher when the antenna coverage area of antenna 212 is smaller. However, depending on the applications locating DLMs using this method, the locating can be quite acceptable. Location confidence is improved with a TDOA measurement as described above. Antenna 212 can process all locating by itself (with connected data processing system (not shown) as well known to those skilled in the art), or with interoperability to other services as connected to antenna 212, for example with connectivity described in FIG. 1E. Antenna 212 can be used to communicate the location of DLM 200 to a service, to the DLM 200, to other MSs within its coverage area, any combination thereof, or to any connected data processing system, or MS, of FIG. 1E.

Once DLM 200 is within the building 210, a strategically placed antenna 216 with a desired detection range within the building is used to detect the DLM 200 coming into its proximity. Wall breakout 214 is used to see the antenna 216 through the building 210. The known antenna 216 location is used to automatically detect the location of the DLM 200. In fact, any DLM that travels within the coverage area served by antenna 216 is identified as the location of antenna 216. The confidence of a location of a DLM 200 is low when the antenna coverage area of antenna 216 is large. In contrast, the confidence of a location of a DLM 200 is higher when the antenna coverage area of antenna 216 is smaller. Travels of DLM 200 can be limited by objects, pathways, or other limiting circumstances of traffic, to provide a higher confidence of location of DLM 200 when located by antenna 216, or when located by any locating antenna described herein which detects MSs coming within range of its location. Location confidence is improved with a TDOA measurement as described above. Antenna 216 can process all locating by itself (with connected data processing system (not shown) as well known to those skilled in the art), or with interoperability to other services as connected to antenna 216, for example with connectivity described in FIG. 1E. Antenna 216 can be

31

used to communicate the location of DLM 200 to a service, to the DLM 200, to other MSs within its coverage area, any combination thereof, or to any connected data processing system, or MS, of FIG. 1E. Other in-range detection antennas of a FIG. 2C embodiment may be strategically placed to facilitate warehouse operations such as in Kubler et al.

FIG. 2D depicts a flowchart for describing a preferred embodiment of a service whereabouts update event of an antenna in-range detected MS, for example a DLM 200, when MS location awareness is monitored by a stationary antenna, or cell tower (i.e. the service thereof). FIGS. 2A through 2C location detection processing are well known in the art. FIG. 2D describes relevant processing for informing MSs of their own whereabouts. Processing begins at block 230 when a MS signal deserving a response has been received and continues to block 232 where the antenna or cell tower service has authenticated the MS signal. A MS signal can be received for processing by blocks 230 through 242 as the result of a continuous, or pulsed, broadcast or beaconing by the MS (FIG. 13A), perhaps as part of usual communication protocol in progress for the MS (FIG. 13A usual data 1302 with embedded Communications Key (CK) 1304), or an MS response to continuous, or pulsed, broadcast or beaconing via the service connected antenna (FIG. 13C). MS and/or service transmission can be appropriately correlated for a response (as described above) which additionally facilitates embodiments using TDOA measurements (time of communications between the MS and antenna, or cell tower) to determine at least how close is the MS in range (or use in conjunction with other data to triangulate the MS location). The MS is preferably authenticated by a unique MS identifier such as a phone number, address, name, serial number, or any other unique handle to the MS. In this, and any other embodiments disclosed, an MS may be authenticated using a group identifier handle indicating membership to a supported/known group deserving further processing. Authentication will preferably consult a database for authenticating that the MS is known. Block 232 continues to block 234 where the signal received is immediately responded back to the MS, via the antenna, containing at least correlation along with whereabouts information for a Whereabouts Data Record (WDR) 1100 associated with the antenna (or cell tower). Thereafter, the MS receives the correlated response containing new data at block 236 and completes a local whereabouts data record 1100 (i.e. WDR 1100) using data received along with other data determined by the MS.

In another embodiment, blocks 232 through 234 are not required. A service connected antenna (or cell tower) periodically broadcasts its whereabouts (WDR info (e.g. FIG. 13C)) and MSs in the vicinity use that directly at block 236. The MS can choose to use only the confidence and location provided, or may determine a TDOA measurement for determining how close it is. If the date/time stamp field 1100b indicates NTP is in use by the service, and the MS is also using NTP, then a TDOA measurement can be determined using the one unidirectional broadcast via the antenna by using the date/time stamp field 1100b received with when the WDR information was received by the MS (subtract time difference and use known wave spectrum for distance). If either the service or MS is not NTP enabled, then a bidirectional correlated data flow between the service and MS is used to assess a TDOA measurement in terms of time of the MS. One embodiment provides the TDOA measurement from the service to the MS. Another embodiment calculates the TDOA measurement at the MS.

Network Time protocol (NTP) can ensure MSs have the same atomic clock time as the data processing systems driv-

32

ing antennas (or cell towers) they will encounter. Then, date/time stamps can be used in a single direction (unidirectional) broadcast packet to determine how long it took to arrive to/from the MS. In an NTP embodiment, the MS (FIG. 13A) and/or the antenna (FIG. 13C) sends a date/time stamp in the pulse, beacon, or protocol. Upon receipt, the antenna (or cell tower) service data processing system communicates how long the packet took from an MS to the antenna (or cell tower) by comparing the date/time stamp in the packet and a date/time stamp of when it was received. The service may also set the confidence value, before sending WDR information to the MS. Similarly, an MS can compare a date/time stamp in the unidirectional broadcast packet sent from a locating service (FIG. 13C) with when received by the MS. So, NTP facilitates TDOA measurements in a single broadcast communication between systems through incorporation to usual communications data 1302 with a date/time stamp in Communications Key (CK) 1304, or alternatively in new data 1302. Similarly, NTP facilitates TDOA measurement in a single broadcast communication between systems through incorporation to usual communications data 1312 with a date/time stamp in Communications Key (CK) 1314, or alternatively in new data 1312.

The following template is used in this disclosure to highlight field settings. See FIG. 11A descriptions. Fields are set to the following upon exit from block 236:

MS ID field 1100a is preferably set with: Unique MS identifier of the MS invoking block 240. This field is used to uniquely distinguish this MS WDRs on queue 22 from other originated WDRs.

DATE/TIME STAMP field 1100b is preferably set with: Date/time stamp for WDR completion at block 236 to the finest granulation of time achievable by the MS. The NTP use indicator is set appropriately.

LOCATION field 1100c is preferably set with: Location of stationary antenna (or cell tower) as communicated by the service to the MS.

CONFIDENCE field 1100d is preferably set with: The same value (e.g. 76) for any range within the antenna (or cell tower), or may be adjusted using the TDOA measurement (e.g. amount of time detected by the MS for the response at block 234). The longer time it takes between the MS sending a signal detected at block 232 and the response with data back received by the MS (block 234), the less confidence there is for being located because the MS must be a larger distance from the antenna or cell tower. The less time it takes between the MS sending a signal detected at block 232 and the response with data back, the more confidence there is for being located because the MS must be a closer distance to the antenna or cell tower. Confidence values are standardized for all location technologies. In some embodiments of FIG. 2D processing, a confidence value can be set for 1 through 100 (1 being lowest confidence and 100 being highest confidence) wherein a unit of measurement between the MS and antenna (or cell tower) is used directly for the confidence value. For example, 20 meters is used as the unit of measurement. For each unit of 20 meters distance determined by the TDOA measurement, assign a value of 1, up to a worst case of 100 (i.e. 2000 meters). Round the 20 meter unit of distance such that 0 meters to <25 meters is 20 meters (i.e. 1 unit of measurement), 26 meters to <45 meters is 40 meters (i.e. 2 units of measurement), and so on. Once the number of units is determined, subtract that number from 101 for the confidence value (i.e. 1 unit=confidence value 100, 20 units=confidence value 81; 100 units or greater=confidence value of 1). Yet another embodiment will use a standard confidence value for this "coming in range" technology such as 76 and then further

increase or decrease the confidence using the TDOA measurement. Many embodiments exist for quantifying a higher versus lower confidence. In any case, a confidence value (e.g. 76) is determined by the MS, service, or both (e.g. MS uses TDOA measurement to modify confidence sent by service). LOCATION TECHNOLOGY field **1100e** is preferably set with: "Server Antenna Range" for an antenna detecting the MS, and is set to "Server Cell Range" for a cell tower detecting the MS. The originator indicator is set to DLM.

LOCATION REFERENCE INFO field **1100f** is preferably set with: The period of time for communications between the antenna and the MS (a TDOA measurement), if known; a communications signal strength, if available; wave spectrum used (e.g. from MS receive processing), if available; particular communications interface **70**, if available. The TDOA measurement may be converted to a distance using wave spectrum information. The values populated here should have already been factored into the confidence value at block **236**. COMMUNICATIONS REFERENCE INFO field **1100g** is preferably set with: Parameters uniquely identifying a/the service (e.g. antenna (or cell tower)) and how to best communicate with it again, if available. May not be set, regardless if received from the service.

SPEED field **1100h** is preferably set with: Data received by MS at block **234**, if available.

HEADING field **1100i** is preferably set with: Data received by MS at block **234**, if available.

ELEVATION field **1100j** is preferably set with: data received by MS at block **234**, if available. Elevation field **1100j** is preferably associated with the antenna (or cell tower) by the elevation/altitude of the antenna (or cell tower).

APPLICATION FIELDS field **1100k** is preferably set with: Data received at block **234** by the MS, or set by data available to the MS, or set by both the locating service for the antenna (or cell tower) and the MS itself. Application fields include, and are not limited to, MS navigation APIs in use, social web site identifying information, application information for applications used, accessed, or in use by the MS, or any other information complementing whereabouts of the MS.

CORRELATION FIELD **1100m** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

SENT DATE/TIME STAMP field **1100n** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

RECEIVED DATE/TIME STAMP field **1100p** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

A service connected to the antenna (or cell tower) preferably uses historical information and artificial intelligence interrogation of MS travels to determine fields **1100h** and **1100i**. Block **236** continues to block **238** where parameters are prepared for passing to FIG. 2F processing invoked at block **240**. Parameters are set for: WDRREF=a reference or pointer to the WDR; DELETEQ=FIG. 2D location queue discard processing; and SUPER=FIG. 2D supervisory notification processing. Thereafter, block **240** invokes FIG. 2F processing and FIG. 2D processing terminates at block **242**. FIG. 2F processing will insert to queue **22** so this MS knows at least its own whereabouts whenever possible. A single data instance embodiment of WDR queue **22** will cause FIG. 2F to update the single record of WDR information for being current upon exit from block **240** (this is true for all flowchart blocks invoking FIG. 2F processing).

With reference now to FIG. 2F, depicted is a flowchart for describing a preferred embodiment of a procedure for inserting a Whereabouts Data Record (WDR) **1100** to MS WDR queue **22**. Appropriate semaphores are used for variables which can be accessed simultaneously by another thread other than the caller. With reference now to FIG. 2F, proce-

sure processing starts at block **270** and continues to block **272** where parameters passed from the invoking block of processing, for example block **240**, are determined. The variable WDRREF is set by the caller to a reference or pointer to the WDR so subsequent blocks of FIG. 2F can access the WDR. The variable DELETEQ is set by the caller so that block **292** knows how to discard obsolete location queue entries. The DELETEQ variable can be a multi-field record (or reference thereof) for how to prune. The variable SUPER is set by the caller so that block **294** knows under what condition(s), and which data, to contact a supervisory service. The SUPER variable can be a multi-field record (or reference thereof) for instruction.

Block **272** continues to block **274** where the DLMV (see FIG. 12 and later discussions for DLMV (DLM role(s) List Variable)), or ILMV (see FIG. 12 and later discussions for ILMV (ILM role(s) List Variable)), is checked for an enabled role matching the WDR for insertion (e.g. DLM: location technology field **1100e** (technology and originator indicator) when MS ID=this MS; ILM: DLM or ILM indicator when MS ID not this MS). If no corresponding DLMV/ILMV role is enabled for the WDR to insert, then processing continues to block **294** (the WDR is not inserted to queue **22**). If the ILMV/DLMV role for the WDR is enabled, then processing continues to block **276** where the confidence of the WDR **1100** is validated prior to insertion. An alternate embodiment to FIG. 2F will not have block **274** (i.e. block **272** continues directly to block **276**) since appropriate DLM and/or ILM processing may be terminated anyway when DLM/ILM role(s) are disabled (see FIG. 14A/B).

If block **276** determines the data to be inserted is not of acceptable confidence (e.g. field **1100d**<confidence floor value (see FIG. 14A/B)), then processing continues to block **294** described below. If block **276** determines the data to be inserted is of acceptable confidence (e.g. field **1100d**>70), then processing continues to block **278** for checking the intent of the WDR insertion.

If block **278** determines the WDR for insert is a WDR describing whereabouts for this MS (i.e. MS ID matching MS of FIG. 2F processing (DLM: FIGS. 2A through 9B, or ILM: FIG. 26A/B)), then processing continues to block **280**. If block **278** determines the WDR for insert is from a remote ILM or DLM (i.e. MS ID does not match MS of FIG. 2F processing), then processing continues to block **290**. Block **280** peeks the WDR queue **22** for the most recent highest confidence entry for this MS whereabouts by searching queue **22** for: the MS ID field **1100a** matching the MS ID of FIG. 2F processing, and a confidence field **1100d** greater than or equal to the confidence floor value, and a most recent date/time stamp field **1100b**. Thereafter, if block **282** determines one was found, then processing continues to block **284**, otherwise processing continues to block **286** where a Last Whereabouts date/Time stamp (LWT) variable is set to field **1100b** of the WDR for insert (e.g. first MS whereabouts WDR), and processing continues to block **288**.

If block **284** determines the WDR for insertion has significantly moved (i.e. using a movement tolerance configuration (e.g. 3 meters) with fields **1100c** of the WDR for insert and the WDR peeked at block **280**), then block **286** sets the LWT (Last Whereabouts date/Time stamp) variable (with appropriate semaphore) to field **1100b** of the WDR for insert, and processing continues to block **288**, otherwise processing continues directly to block **288** (thereby keeping the LWT as its last setting). The LWT is to hold the most recent date/time stamp of when the MS significantly moved as defined by a movement tolerance. The movement tolerance can be system defined or configured, or user configured in FIG. 14 by an

35

option for configuration detected at block 1408, and then using the Configure Value procedure of FIG. 18 (like confidence floor value configuration).

Block 288 accesses the DLMV and updates it with a new DLM role if there is not one present for it. This ensures a correct list of DLMV roles are available for configuration by FIG. 14. Preferably, by default an unanticipated DLMV role is enabled (helps inform the user of its availability). Likewise in another embodiment, ILMV roles can be similarly updated, in particular if a more granulated list embodiment is maintained to the ILMV, or if unanticipated results help to identify another configurable role. By default, block 274 should allow unanticipated roles to continue with WDR insertion processing, and then block 288 can add the role, enable it, and a user can decide what to do with it in configuration (FIG. 14A/B).

Thereafter, the WDR 1100 is inserted to the WDR queue 22 at block 290, block 292 discards any obsolete records from the queue as directed by the caller (invoker), and processing continues to block 294. The WDR queue 22 preferably contains a list of historically MS maintained Whereabouts Data Records (WDRs) as the MS travels. When the MS needs its own location, for example from an application access, or to help locate an ILM, the queue is accessed for returning the WDR with the highest confidence value (field 1100d) in the most recent time (field 1100b) for the MS (field 1100a). Block 292 preferably discards by using fields 1100b and 1100d relative to other WDRs. The queue should not be allowed to get too large. This will affect memory (or storage) utilization at the MS as well as timeliness in accessing a sought queue entry. Block 292 also preferably discards WDRs from queue 22 by moving selected WDRs to LBX History 30.

As described above, queue interfaces assume an implicit semaphore for properly accessing queue 22. There may be ILMs requesting to be located, or local applications of the MS may request to access the MS whereabouts. Executable thread(s) at the MS can access the queue in a thread-safe manner for responding to those requests. The MS may also have multiple threads of processing for managing whereabouts information from DLMs, ILMs, or stationary location services. The more concurrently executable threads available to the MS, the better the MS is able to locate itself and respond to others (e.g. MSs). There can be many location systems and methods used to keeping a MS informed of its own whereabouts during travel. While the preferred embodiment is to maximize thread availability, the obvious minimum requirement is to have at least 1 executable thread available to the MS. As described above, in operating system environments without proper queue interfaces, queue access blocks are first preceded by an explicit request for a semaphore lock to access queue 22 (waits until obtained), and then followed by a block for releasing the semaphore lock to another thread for use. Also, in the present disclosure it is assumed in blocks which access data accessible to more than 1 concurrent thread (e.g. shared memory access to DLMV or ILMV at block 274) that an appropriate semaphore (created at block 1220) protect synchronous access.

If block 294 determines information (e.g. whereabouts) should be communicated by service informant code 28 to a supervisory service, for example a service 1050, then block 296 communicates specified data to the service and processing terminates at block 298 by returning to the invoker (caller). If block 294 determines a supervisory service is not to be informed, then processing terminates with an appropriate return to the caller at block 298. Service informant code

36

28, at block 296, can send information as data that is reliably acknowledged on receipt, or as a datagram which most likely (but unreliably) is received.

Depending on the SUPER variable, block 294 may opt to communicate every time a WDR is placed to the queue, or when a reasonable amount of time has passed since last communicating to the supervisory service, or when a WDR confidence reaches a certain sought value, or when any WDR field or fields contain certain sought information, or when a reasonably large number of entries exist in WDR queue 22, or for any processing condition encountered by blocks 270 through 298, or for any processing condition encountered by caller processing up to the invocation of FIG. 2F processing. Different embodiments will send a single WDR 1100 at block 296, a plurality of WDRs 1100, or any other data. Various SUPER parameter(s) embodiments for FIG. 2F caller parameters can indicate what, when, where and how to send certain data. Block 296 may send an email, an SMS message, or use other means for conveying data. Service informant code 28 may send LBX history 30, statistics 14 and/or any other data 8, data 20, queue data, data 36 or resources 38. Service informant code 28 may update data in history 30, statistics 14 or any other data 8, data 20, queue data, data 36 and/or resources 38, possibly using conditions of this data to determine what is updated. Blocks 294 and 296 may be omitted in some embodiments.

If a single WDR is sent at block 296 as passed to FIG. 2F processing, then the WDR parameter determined at block 272 is accessed. If a plurality of WDRs is sent at block 296, then block 296 appropriately interfaces in a thread-safe manner to queue 22, and sends the WDRs.

Some preferred embodiments do not incorporate blocks 278 through 286. (i.e. block 276 continues to block 288 if confidence ok). Blocks 278 through 286 are for the purpose of implementing maintaining a date/time stamp of last MS significant movement (using a movement tolerance). Architecture 1900 uses FIG. 2F, as does DLM processing. FIG. 2F must perform well for the preferred multithreaded architecture 1900. Block 280 performs a peek, and block 284 can be quite timely depending on embodiments used for location field 1100c. A movement tolerance incorporated at the MS is not necessary, but may be nice to have. Therefore, blocks 278 through 286 are optional blocks of processing.

FIG. 2F may also maintain (with appropriate semaphore) the most recent WDR describing whereabouts of the MS of FIG. 2F processing to a single data record every time a new one is to be inserted. This allows applications needing current whereabouts to simply access a current WDR, rather than interface to a plurality of WDRs at queue 22. For example, there could be a new block 289 for updating the single WDR 1100 ((just prior to block 290 such that incoming blocks to block 290 go to new block 289, and new block 289 continues to block 290).

With reference now to FIG. 2E, depicted is a flowchart for describing a preferred embodiment of an MS whereabouts update event of an antenna in-range detected MS, for example a DLM 200, when MS location awareness is monitored by the MS. FIG. 2E describes relevant processing for MSs to maintain their own whereabouts. Processing begins at block 250 when the MS receives a signal from an antenna (or cell tower) deserving a response and continues to block 252 where the antenna or cell tower signal is authenticated by the MS as being a legitimate signal for processing. The signal can be received for processing by blocks 250 through 264 as the result of a continuous, or pulsed, broadcast or beaconing by the antenna, or cell tower (FIG. 13C), or as part of usual communication protocol in progress with at least one MS

37

(FIG. 13C usual data 1312 with embedded Communications Key 1314), or as a response via antenna to a previous MS signal (FIG. 13A). The signal is preferably authenticated by a data parsed signature deserving further processing. Block 252 continues to block 254 where the MS sends an outbound request for soliciting an immediate response from the antenna (or cell tower) service. The request by the MS is appropriately correlated (e.g. as described above) for a response, which additionally facilitates embodiments using TDOA measurements (time of communications between the MS and antenna, or cell tower) to determine how close is the MS in range. Block 254 waits for a response, or waits until a reasonable timeout, whichever occurs first. There are also multithreaded embodiments to breaking up FIG. 2E where block 254 does not wait, but rather terminates FIG. 2E processing and depends on another thread to correlate the response and then continue processing blocks 256 through 260 (like architecture 1900).

Thereafter, if block 256 determines the request timed out, then processing terminates at block 264. If block 256 determines the response was received, then processing continues to block 258. Block 258 completes a WDR 1100 with appropriate response data received along with data set by the MS. See FIG. 11A descriptions. Fields are set to the following upon exit from block 258:

MS ID field 1100a is preferably set with: Same as was described for FIG. 2D (block 236) above.

DATE/TIME STAMP field 1100b is preferably set with: Same as was described for FIG. 2D (block 236) above.

LOCATION field 1100c is preferably set with: Same as was described for FIG. 2D (block 236) above.

CONFIDENCE field 1100d is preferably set with: Same as was described for FIG. 2D (block 236) above.

LOCATION TECHNOLOGY field 1100e is preferably set with: "Client Antenna Range" for an antenna detecting the MS, and is set to "Client Cell Range" for a cell tower detecting the MS. The originator indicator is set to DLM.

LOCATION REFERENCE INFO field 1100f is preferably set with: Same as was described for FIG. 2D (block 236) above.

COMMUNICATIONS REFERENCE INFO field 110g is preferably set with: Same as was described for FIG. 2D (block 236) above.

SPEED field 1100h is preferably set with: Same as was described for FIG. 2D (block 236) above.

HEADING field 1100i is preferably set with: Same as was described for FIG. 2D (block 236) above.

ELEVATION field 1100j is preferably set with: Same as was described for FIG. 2D (block 236) above.

APPLICATION FIELDS field 1100k is preferably set with: Same as was described for FIG. 2D (block 236) above.

CORRELATION FIELD 1100m is preferably set with: Not Applicable (i.e. not maintained to queue 22).

SENT DATE/TIME STAMP field 1100n is preferably set with: Not Applicable (i.e. not maintained to queue 22).

RECEIVED DATE/TIME STAMP field 1100p is preferably set with: Not Applicable (i.e. not maintained to queue 22).

The longer time it takes between sending a request and getting a response at block 254, the less confidence there is for being located because the MS must be a larger distance from the antenna or cell tower. The less time it takes, the more confidence there is for being located because the MS must be a closer distance to the antenna or cell tower. Confidence values are analogously determined as described for FIG. 2D. FIG. 2D NTP embodiments also apply here. NTP can be used so no bidirectional communications is required for TDOA measurement. In this embodiment, the antenna (or cell tower)

38

sets a NTP date/time stamp in the pulse, beacon, or protocol. Upon receipt, the MS instantly knows how long the packet took to be received by comparing the NTP date/time stamp in the packet and a MS NTP date/time stamp of when it was received (i.e. no request/response pair required). If location information is also present with the NTP date/time stamp in data received at block 252, then block 252 can continue directly to block 258.

An alternate MS embodiment determines its own (direction) heading and/or speed for WDR completion based on historical records maintained to the WDR queue 22 and/or LBX history 30.

Block 258 continues to block 260 for preparing parameters for: WDRREF=a reference or pointer to the WDR; DELETEQ=FIG. 2E location queue discard processing; and SUPER=FIG. 2E supervisory notification processing. Thereafter, block 262 invokes the procedure (FIG. 2F processing) to insert the WDR to queue 22. After FIG. 2F processing of block 262, FIG. 2E processing terminates at block 264.

In alternative "coming within range" (same as "in range", "in-range", "within range") embodiments, a unique MS identifier, or MS group identifier, for authenticating an MS for locating the MS is not necessary. An antenna emitting signals (FIG. 13C) will broadcast (in CK 1314 of data 1312) not only its own location information (e.g. location field 1100c), but also an NTP indicated date/time stamp field 1100b, which the receiving MS (also having NTP for time synchronization) uses to perform a TDOA measurement upon receipt. This will enable a MS to determine at least how close (e.g. radius 1318 range, radius 1320 range, radius 1322 range, or radius 1316 range) it is located to the location of the antenna by listening for and receiving the broadcast (e.g. of FIG. 13C). Similarly, in another embodiment, an NTP synchronized MS emits signals (FIG. 13A) and an NTP synchronized data processing system associated with a receiving antenna can make a TDOA measurement upon signal receipt. In other embodiments, more than a single unidirectional signal may be used while still preventing the requirement to recognize the MS to locate it. For example, an antenna emitting signals (e.g. FIG. 13C hotspot WiFi 802.x) will contain enough information for a MS to respond with correlation for being located, and visa-versa. In any case, there can be multi-directional exchanged signals for determining a TDOA measurement.

FIG. 3A depicts a locating by triangulation illustration for discussing automatic location of a MS, for example DLM 200. DLM 200 is located through triangulation, as is well known in the art. At least three base towers, for example, base tower 108b, base tower 108d, and base tower 108f, are used for locating the MS. A fourth base tower may be used if elevation (or altitude) was configured for use in locating DLM 200. There are cases where only two base towers are necessary given routes of travel are limited and known, for example, in spread out roadways or limited configured locations. Base towers may also be antennas 108b, 108d, and 108f in similar triangulation embodiments.

FIG. 3B depicts a flowchart for describing a preferred embodiment of the whereabouts update event of a triangulated MS, for example DLM 200, when MS location awareness is monitored by some remote service. While FIG. 3A location determination with TDOA and AOA is well known in the art, FIGS. 3B and 3C include relevant processing for MSs to maintain their own whereabouts. Processing begins at block 310 and continues to block 312 where base stations able to communicate to any degree with a MS continue reporting to their controller the MS signal strength with an MS identifier (i.e. a unique handle) and Time Difference of Arrival (TDOA) information, Angle of Arrival (AOA) information, or

heterogeneously both TDOA and AOA (i.e. MPT), depending on the embodiment. The MS can pick signals from base stations. In some embodiments, the MS monitors a paging channel, called a forward channel. There can be multiple forward channels. A forward channel is the transmission frequency from the base tower to the MS. Either the MS provides broadcast heartbeats (FIG. 13A) for base stations, or the base stations provide heartbeats (FIG. 13C) for a response from the MS, or usual MS use protocol signals are detected and used (incorporating CK 1304 in usual data 1302 by MS, or CK 1314 in "usual data" 1312 by service). Usual data is the usual communications traffic data in carrying out other character 32 processing. Communication from the MS to the base tower is on what is called the reverse channel. Forward channels and reverse channel are used to perform call setup for a created session channel.

TDOA is calculated from the time it takes for a communication to occur from the MS back to the MS via the base tower, or alternatively, from a base tower back to that base tower via the MS. NTP may also be used for time calculations in a unidirectional broadcast from a base tower (FIG. 13C) to the MS, or from the MS (FIG. 13A) to a base tower (as described above). AOA is performed through calculations of the angle by which a signal from the MS encounters the antenna. Triangle geometry is then used to calculate a location. The AOA antenna is typically of a phased array type.

See "Missing Part Triangulation (MPT)" section below with discussions for FIGS. 11A through 11E for details on heterogeneously locating the MS using both TDOA and AOA (i.e. Missing Part Triangulation (MPT)). Just as high school taught geometry for solving missing parts of a triangle, so to does MPT triangulate an MS location. Think of the length of a side of a triangle as a TDOA measurement—i.e. length of time, translatable to a distance. Think of the AOA of a signal to an antenna as one of the angles of a triangle vertice. Solving with MPT analogously uses geometric and trigonometric formulas to solve the triangulation, albeit at fast processing speeds.

Thereafter, if the MS is determined to be legitimate and deserving of processing (similar to above), then block 314 continues to block 316. If block 314 determines the MS is not participating with the service, in which case block 312 did little to process it, then processing continues back to block 312 to continue working on behalf of legitimate participating MSs. The controller at block 316 may communicate with other controllers when base stations in other cellular clusters are picking up a signal, for example, when the MS roams. In any case, at block 316, the controller(s) determines the strongest signal base stations needed for locating the MS, at block 316. The strongest signals that can accomplish whereabouts information of the MS are used. Thereafter, block 318 accesses base station location information for base stations determined at block 316. The base station provides stationary references used to (relatively) determine the location of the MS. Then, block 320 uses the TDOA, or AOA, or MPT (i.e. heterogeneously both AOA and TDOA) information together with known base station locations to calculate the MS location.

Thereafter, block 322 accesses historical MS location information, and block 324 performs housekeeping by pruning location history data for the MS by time, number of entries, or other criteria. Block 326 then determines a heading (direction) of the MS based on previous location information. Block 326 may perform Artificial Intelligence (AI) to determine where the MS may be going by consulting many or all of the location history data. Thereafter, block 328 completes a service side WDR 1100, block 330 appends the WDR

information to location history data and notifies a supervisory service if there is one outside of the service processing of FIG. 3B. Processing continues to block 332 where the service communicates the WDR to the located MS.

Thereafter, the MS completes its own WDR at block 334 for adding to WDR queue 22 to know its own whereabouts whenever possible, and block 336 prepares parameters for invoking WDR insertion processing at block 338. Parameters are set for: WDRREF=a reference or pointer to the MS WDR; DELETEQ=FIG. 3B location queue discard processing; and SUPER=FIG. 3B supervisory notification processing (e.g. no supervisory notification processing because it was already handled at block 330, or by being in context of the FIG. 3B service processing). At block 338, the MS invokes FIG. 2F processing already described. After block 338, processing continues back to block 312. Of course, block 332 continues directly to block 312 at the service(s) since there is no need to wait for MS(s) processing in blocks 334 through 338. FIG. 3B processing is continuous for every MS in the wireless network 7 days a week, 24 hours a day.

See FIG. 11A descriptions. Fields are set to the following upon exit from block 334:

MS ID field 1100a is preferably set with: Same as was described for FIG. 2D (block 236) above.

DATE/TIME STAMP field 1100b is preferably set with: Same as was described for FIG. 2D (block 236) above.

LOCATION field 1100c is preferably set with: The triangulated location of the MS as communicated by the service.

CONFIDENCE field 1100d is preferably set with: Confidence of triangulation determined by the service which is passed to the MS at block 332. The confidence value may be set with the same value (e.g. 85) regardless of how the MS was triangulated. In other embodiments, field 1100d will be determined (completely, or adjusting the value of 85) by the service for TDOA measurements used, AOA measurements, signal strengths, wave spectrum involved, and/or the abundance of particular MS signals available for processing by blocks 312 through 320. Higher confidences are assigned for smaller TDOA measurements (shorter distances), strong signal strengths, and numerous additional data points beyond what is necessary to locate the MS. Lower confidences are assigned for larger TDOA measurements, weak signal strengths, and minimal data points necessary to locate the MS. A reasonable confidence can be assigned using this information as guidelines where 1 is the lowest confidence and 100 is the highest confidence.

LOCATION TECHNOLOGY field 1100e is preferably set with: "Server Cell TDOA", "Server Cell AOA", "Server Cell MPT", "Server Antenna TDOA", "Server Antenna AOA", or "Server Antenna MPT", depending on how the MS was located and what flavor of service was used. The originator indicator is set to DLM.

LOCATION REFERENCE INFO field 1100f is preferably set with: null (not set) for indicating that all triangulation data was factored into determining confidence, and none is relevant for a single TDOA or AOA measurement in subsequent processing (i.e. service did all the work).

COMMUNICATIONS REFERENCE INFO field 1100g is preferably set with: Same as was described for FIG. 2D (block 236) above.

SPEED field 1100h is preferably set with: Service WDR information at block 332, wherein the service used historical information and artificial intelligence interrogation of MS travels to determine, if available.

HEADING field 1100i is preferably set with: Service WDR information at block 332, wherein the service used historical

41

information and artificial intelligence interrogation of MS travels to determine, if available.

ELEVATION field **1100j** is preferably set with: Elevation/altitude, if available.

APPLICATION FIELDS field **1100k** is preferably set with: Same as was described for FIG. 2D (block **236**) above.

CORRELATION FIELD **1100m** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

SENT DATE/TIME STAMP field **1100n** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

RECEIVED DATE/TIME STAMP field **1100p** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

FIG. 3C depicts a flowchart for describing a preferred embodiment of the whereabouts update event of a triangulated MS, for example a DLM **200**, when MS location awareness is monitored by the MS. Communications between the base stations and MS is similar to FIG. 3B processing except the MS receives information (FIG. 13C) for performing calculations and related processing. Processing begins at block **350** and continues to block **352** where the MS continues receiving (FIG. 13C) pulse reporting from base stations (or antennas). AOA, TDOA, and MPT (See “Missing Part Triangulation (MPT)” section below with discussions for FIGS. **11A** through **11E** for details on heterogeneously locating the MS using both TDOA and AOA) can be used to locate the MS, so there are many possible signal types received at block **352**. Then, block **354** determines the strongest signals which can accomplish a completed WDR, or at least a location, of the MS. Thereafter, block **356** parses base station location information from the pulse messages that are received by the MS. Block **358** communicates with base stations to perform TDOA and/or AOA measurements and calculations. The time it takes for a communication to occur from the MS back to the MS for TDOA, or alternatively, from a base tower back to that base tower can be used. NTP may also be used, as described above, so that base towers (or antennas) broadcast signals (FIG. 13C) picked up by the MS which already contain the base tower locations and NTP date/time stamps for TDOA calculations. Block **358** uses the TDOA and/or AOA information with the known base station information to determine the MS location. While AOA information from the base stations (or antennas) is used by the MS, various MS embodiments can use AOA information detected at an MS antenna provided the heading, yaw, pitch, and roll is known at the MS during the same time as signal reception by the MS. A 3-axis accelerometer (e.g. in iphone) may also provide yaw, pitch and roll means for proper AOA calculation.

Thereafter, block **360** accesses historical MS location information (e.g. WDR queue **22** and/or LBX history **30**) to prevent redundant information kept at the MS, and block **362** performs housekeeping by pruning the LBX history **30** for the MS by time, number of entries, or other criteria. Block **364** then determines a heading (direction) of the MS based on previous location information (unless already known from block **358** for AOA determination). Block **364** may perform Artificial Intelligence (AI) to determine where the MS may be going by consulting queue **22** and/or history **30**. Thereafter, block **366** completes a WDR **1100**, and block **368** prepares parameters for FIG. 2F processing: WDRREF=a reference or pointer to the MS WDR; DELETEQ=FIG. 3C location queue discard processing; and SUPER=FIG. 3B supervisory notification processing. Block **368** continues to block **370** for invoking FIG. 2F processing already described above. After block **370**, processing continues back to block **352**. FIG. 3C processing is continuous for the MS as long as the MS is enabled. In various multithreaded embodiments, many

42

threads at the MS work together for high speed processing at blocks **352** through **358** for concurrently communicating to many stationary references.

See FIG. 11A descriptions. Fields are set to the following upon exit from block **366**:

MS ID field **1100a** is preferably set with: Same as was described for FIG. 2D (block **236**) above.

DATE/TIME STAMP field **1100b** is preferably set with: Same as was described for FIG. 2D (block **236**) above.

LOCATION field **1100c** is preferably set with: The triangulated location of the MS as determined by the MS.

CONFIDENCE field **1100d** is preferably set with: The confidence of triangulation as determined by the MS. Confidence may be set with the same value (e.g. 80 since MS may be moving during triangulation) regardless of how the MS was triangulated. In other embodiments, field **1100d** will be determined (completely, or adjusting the value of 80) by the MS for TDOA measurements used, AOA measurements, signal strengths, wave spectrum involved, and/or the abundance of particular service signals available for processing. Higher confidences are assigned for smaller TDOA measurements (shorter distances), strong signal strengths, and numerous additional data points beyond what is necessary to locate the MS. Lower confidences are assigned for larger TDOA measurements, weak signal strengths, and minimal data points necessary to locate the MS. A reasonable confidence can be assigned using this information as guidelines where 1 is the lowest confidence and 100 is the highest confidence.

LOCATION TECHNOLOGY field **1100e** is preferably set with: “Client Cell TDOA”, “Client Cell AOA”, “Client Cell MPT”, “Client Antenna TDOA”, “Client Antenna AOA”, or “Client Antenna MPT”, depending on how the MS located itself. The originator indicator is set to DLM.

LOCATION REFERENCE INFO field **1100f** is preferably set with: Data associated with selected best stationary reference(s) used by the MS: the selection location/whereabouts, TDOA measurement to it, and wave spectrum (and/or particular communications interface **70**) used, if reasonable. The TDOA measurement may be converted to a distance using wave spectrum information. Also, preferably set herein is data associated with a selected best stationary reference used by the MS (may be same or different than for TDOA measurement): the selection location, AOA measurement to it, and heading, yaw, pitch, and roll values (or accelerometer readings), if reasonable. Values that may be populated here should have already been factored into the confidence value. There may be one or more stationary reference whereabouts with useful measurements maintained here for FIG. 26B processing of block **2652**.

COMMUNICATIONS REFERENCE INFO field **1100g** is preferably set with: Parameters referencing MS internals, if desired.

SPEED field **1100h** is preferably set with: Speed determined by the MS using historical information (queue **22** and/or history **30**) and artificial intelligence interrogation of MS travels to determine, if reasonable.

HEADING field **1100i** is preferably set with: Heading determined by the MS using historical information (queue **22** and/or history **30**) and artificial intelligence interrogation of MS travels to determine, if reasonable.

ELEVATION field **1100j** is preferably set with: Elevation/altitude, if available.

APPLICATION FIELDS field **1100k** is preferably set with: Same as was described for FIG. 2D (block **236**) above.

CORRELATION FIELD **1100m** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

SENT DATE/TIME STAMP field **1100n** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

RECEIVED DATE/TIME STAMP field **1100p** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

In alternative triangulation embodiments, a unique MS identifier, or MS group identifier, for authenticating an MS for locating the MS is not necessary. An antenna emitting signals (FIG. **13C**) will broadcast (CK **1314** of data **1312**) not only its own location information, but also an NTP date/time stamp, which the receiving MS (also having NTP for time synchronization) uses to perform TDOA measurements upon receipt. This will enable a MS to determine how close (e.g. radius **1318** range, radius **1320** range, radius **1322** range, or radius **1316** range) it is located to the location of the antenna by listening for and receiving the broadcast (e.g. of FIG. **13C**). Similarly, in another embodiment, an NTP synchronized MS emits signals (FIG. **13A**) and an NTP synchronized data processing system associated with a receiving antenna can determine a TDOA measurement upon signal receipt. In other embodiments, more than a single unidirectional signal may be used while still preventing the requirement to recognize the MS to locate it. For example, an antenna emitting signals will contain enough information for a MS to respond with correlation for being located. Alternatively, an MS emitting signals will contain enough information for a service to respond with correlation for being located. In any case, there can be multi-directional exchanged signals for determining TDOA. Similarly, a service side data processing system can interact with a MS for AOA information without requiring a known identifier of the MS (use request/response correlation).

FIG. **4A** depicts a locating by GPS triangulation illustration for discussing automatic location of a MS, for example a DLM **200**. A MS, for example DLM **200**, is located through GPS triangulation as is well known in the art. At least three satellites, for example, satellite **134**, satellite **136**, and satellite **138**, are necessary for locating the MS. A fourth satellite would be used if elevation, or altitude, was configured for use by the present disclosure. Ground based stationary references can further enhance whereabouts determination.

FIG. **4B** depicts a flowchart for describing a preferred embodiment of the whereabouts update event of a GPS triangulated MS, for example a DLM **200**. Repeated continuous GPS location processing begins at block **410** and continues to block **412** where the MS initializes to the GPS interface, then to block **414** for performing the conventional locating of the GPS enabled MS, and then to block **416** for calculating location information. In some embodiments, block **412** may only be necessary a first time prior to repeated invocations of FIG. **4B** processing. Block **414** may be an implicit wait for pulses from satellites, or an event driven mechanism when GPS satellite pulses are received for synchronized collection, or a multithreaded implementation concurrently listening for, and processing collaboratively, the signals. Block **414** and block **416** processing is well known in the art. Thereafter, the MS completes a WDR **1100** at block **418**, block **420** prepares parameters for FIG. **2F** invocation, and block **422** invokes, with the WDR, the FIG. **2F** processing (described above). Processing then terminates at block **424**. Parameters prepared at block **420** are: WDRREF=a reference or pointer to the WDR; DELETEQ=FIG. **4B** location queue discard processing; and SUPER=FIG. **4B** supervisory notification processing. GPS location processing is preferably continuous for the MS as long as the MS is enabled.

See FIG. **11A** descriptions. Fields are set to the following upon exit from block **418**:

MS ID field **1100a** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

5 DATE/TIME STAMP field **1100b** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

LOCATION field **1100c** is preferably set with: The GPS location of the MS.

10 CONFIDENCE field **1100d** is preferably set with: Confidence of GPS variety (usually high) which may be set with the same value (e.g. **95** for DGPS, **93** for AGPS, and **90** for GPS). In other embodiments, field **1100d** will be determined (completely, or amending the defaulted value) by the MS for timing measurements, signal strengths, and/or the abundance of particular signals available for processing, similarly to as described above. An MS may not be aware of the variety of GPS, in which case straight GPS is assumed.

15 LOCATION TECHNOLOGY field **1100e** is preferably set with: "GPS", "A-GPS", or "D-GPS", depending on (if known) flavor of GPS. The originator indicator is set to DLM. LOCATION REFERENCE INFO field **1100f** is preferably set with: null (not set) for indicating that data was factored into determining confidence, and none is relevant for a single TDOA or AOA measurement in subsequent processing.

20 COMMUNICATIONS REFERENCE INFO field **1100g** is preferably set with: Parameters referencing MS internals, if desired.

SPEED field **1100h** is preferably set with: Speed determined by the MS using a suitable GPS interface, or historical information (queue **22** and/or history **30**) and artificial intelligence interrogation of MS travels to determine, if reasonable.

25 HEADING field **1100i** is preferably set with: Heading determined by the MS using a suitable GPS interface, or historical information (queue **22** and/or history **30**) and artificial intelligence interrogation of MS travels to determine, if reasonable.

ELEVATION field **1100j** is preferably set with: Elevation/altitude, if available.

30 APPLICATION FIELDS field **1100k** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

CORRELATION FIELD **1100m** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

SENT DATE/TIME STAMP field **1100n** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

45 RECEIVED DATE/TIME STAMP field **1100p** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

FIG. **5A** depicts a locating by stationary antenna triangulation illustration for discussing automatic location of a MS, for example DLM **200**. There may be communication/transmission issues when an MS is taken indoors. Shown is a top view of an indoor floor plan **502**. Antenna stations **504** (shown generally as **504**) are strategically placed over the area so that an MS can be located. Triangulation techniques again apply. At least three antenna stations, for example, station **504f**, station **504h**, and station **504i** are used to locate the MS, for example DLM **200**. In floor plan embodiments where aisles delimit travel, only two antenna stations may be necessary, for example at either end of the particular aisle. While most stations **504** may receive signals from the MS, only the strongest stations are used. FIG. **5A** and associated discussions can also be used for an outside triangulation embodiment using a similar strategic antenna placement scheme. Processing described for FIGS. **3A** to **3C** can also be used for an indoor embodiment as described by FIG. **5A**.

65 FIG. **5B** depicts a flowchart for describing a preferred embodiment of the whereabouts update event of a stationary antenna triangulated MS, for example a DLM **200**. In one

embodiment, indoor location technology of Pinpoint corporation (Pinpoint is a trademark of Pinpoint Corporation) is utilized to locate any MS that moves about the indoor location. The Pinpoint corporation methodology begins at block 510 and continues to block 512. A cell controller drives antenna stations to emit a broadcast signal from every station. Any MS within range (i.e. indoors) will phase modulate its unique identifier onto a return signal it transmits, at block 514. Stations at block 516 receive the transmission and strength of signal. The cell controller that drives stations sorts out and selects the strongest (e.g. 3) signals. The cell controller, at block 518, also extracts the unique MS identifier from the return signal, and TDOA is used to calculate distances from the stations receiving the strongest signals from the MS at block 520. Alternative embodiments can use AOA or MPT to determine locations. The locations of the controller selected stations are registered in an overlay map in an appropriate coordinate system, landmark system, or grid of cells. Block 522 locates the MS using the overlay map, locations of the (e.g. 3) selected stations, and the calculated distances triangulated from the selected stations, using TDOA, AOA, or MPT in various embodiments. Thereafter, block 524 calculates location information of the MS. Processing continues with repeated broadcast at block 512 and subsequent processing for every MS within range.

Thereafter, block 526 accesses historical MS location information, performs housekeeping by pruning location history data for the MS by time, number of entries, or other criteria, and determines a heading (direction) of the MS based on previous location information. Block 526 may perform Artificial Intelligence (AI) to determine where the MS may be going by consulting many or all of the location history data. Thereafter, block 528 completes a service side WDR 1100, block 530 appends the WDR information to location history data and notifies a supervisory service if there is one outside of the service processing of FIG. 5B. Processing continues to block 532 where the service communicates the WDR to the located MS.

Thereafter, the MS completes the WDR at block 534 for adding to WDR queue 22. Thereafter, block 536 prepares parameters passed to FIG. 2F processing for: WDRREF=a reference or pointer to the MS WDR; DELETEQ=FIG. 5B location queue discard processing; and SUPER=FIG. 5B supervisory notification processing (e.g. no supervisory notification processing because it was already handled at block 530, or by being in context of the FIG. 5B service processing). Block 536 continues to block 538 where the MS invokes FIG. 2F processing already described above. After block 538, processing continues back to block 514. Of course, block 532 continues directly to block 514 at the service(s) since there is no need to wait for MS(s) processing in blocks 534 through 538. FIG. 5B processing is continuous for every MS in the wireless network 7 days a week, 24 hours a day.

See FIG. 11A descriptions. Fields are set to the following upon exit from block 534:

MS ID field 1100a is preferably set with: Same as was described for FIG. 2D (block 236) above.

DATE/TIME STAMP field 1100b is preferably set with: Same as was described for FIG. 2D (block 236) above.

LOCATION field 1100c is preferably set with: The triangulated location of the MS as communicated by the service.

CONFIDENCE field 1100d is preferably set with: Confidence of triangulation determined by the service which is passed to the MS at block 532. The confidence value may be set with the same value (e.g. 95 (normally high for triangulation using densely positioned antennas)) regardless of how the MS was triangulated. In other embodiments, field 1100d

will be determined (completely, or adjusting the value of 95) by the service for TDOA measurements used, AOA measurements, signal strengths, wave spectrum involved, and/or the abundance of particular MS signals available for processing. Higher confidences are assigned for smaller TDOA measurements (shorter distances), strong signal strengths, and numerous additional data points beyond what is necessary to locate the MS. Lower confidences are assigned for larger TDOA measurements, weak signal strengths, and minimal data points necessary to locate the MS. A reasonable confidence can be assigned using this information as guidelines where 1 is the lowest confidence and 100 is the highest confidence.

LOCATION TECHNOLOGY field 1100e is preferably set with: "Server Antenna TDOA", "Server Antenna AOA", or "Server Antenna MPT", depending on how the MS was located and what flavor of service was used. The originator indicator is set to DLM.

LOCATION REFERENCE INFO field 1100f is preferably set with: null (not set) for indicating that all triangulation data was factored into determining confidence, and none is relevant for a single TDOA or AOA measurement in subsequent processing (i.e. service did all the work).

COMMUNICATIONS REFERENCE INFO field 1100g is preferably set with: Same as was described for FIG. 2D (block 236) above.

SPEED field 1100h is preferably set with: Service WDR information at block 532, wherein the service used historical information and artificial intelligence interrogation of MS travels to determine, if available.

HEADING field 1100i is preferably set with: Service WDR information at block 532, wherein the service used historical information and artificial intelligence interrogation of MS travels to determine, if available.

ELEVATION field 1100j is preferably set with: Elevation/altitude, if available.

APPLICATION FIELDS field 1100k is preferably set with: Same as was described for FIG. 2D (block 236) above.

CORRELATION FIELD 1100m is preferably set with: Not Applicable (i.e. not maintained to queue 22).

SENT DATE/TIME STAMP field 1100n is preferably set with: Not Applicable (i.e. not maintained to queue 22).

RECEIVED DATE/TIME STAMP field 1100p is preferably set with: Not Applicable (i.e. not maintained to queue 22).

FIG. 6A depicts a flowchart for describing a preferred embodiment of a service whereabouts update event of a physically, or logically, connected MS, for example a DLM 200. A MS may be newly located and physically, or logically, connected, whereby communications between the MS and service is over a physical/logical connection. Physical connections may occur by connecting a conduit for communications to the MS, or from the MS to a connection point. Conduits include ethernet cables, optical fiber, firewire, USB, or any other means for conduit for communications through a physical medium. Conduits also include wireless mediums (air) for transporting communications, such as when an MS comes into physical wireless range eligible for sending and receiving communications. Logical connections may occur, after a physical connection already exists, for example through a successful communication, or authenticated, bind between a MS and other MS, or MS and service. Logical connections also include the result of: successfully logging into an application, successfully authenticated for access to some resource, successfully identified by an application, or any other logical status upon a MS being certified, registered, signed in, authenticated, bound, recognized, affirmed, or the like.

47

Relevant processing begins at block 602 and continues to block 604 where an MS device is physically/logically connected to a network. Thereafter, the MS accesses a service at block 606. Then, at block 608, the service accesses historical MS location history, and block 610 performs housekeeping by pruning the location history data maintained for the MS by time, number of entries, or other criteria. Block 610 may perform Artificial Intelligence (AI) to determine where the MS may be going (e.g. using heading based on previous locations) by consulting much or all of the location history data. Thereafter, service processing at block 612 completes a service side WDR 1100, then the service appends WDR information to location history data at block 614, and may notify a supervisory service if there is one outside of the service processing of FIG. 6A. Processing continues to block 616 where the service communicates WDR information to the newly physically/logically connected MS. There are many embodiments for determining a newly connected MS location using a physical or logical address, for example consulting a database which maps locations to network addresses (e.g. location to logical ip address; location to physical wall jack/port; etc). Then, at block 618 the MS completes its own WDR using some information from block 616, FIG. 2F parameters are prepared at block 620, block 622 invokes FIG. 2F processing already described above, and processing terminates at block 624. Parameters are set at block 620 for: WDRREF=a reference or pointer to the MS WDR; DELETEQ=FIG. 6A location queue discard processing; and SUPER=FIG. 6A supervisory notification processing (e.g. no supervisory notification processing because it was already handled at block 614, or by being in context of the FIG. 6A service processing). Of course, block 616 continues directly to block 624 at the service(s) since there is no need to wait for MS processing in blocks 618 through 622. FIG. 6A processing is available at any appropriate time in accordance with the underlying service.

See FIG. 11A descriptions. Fields are set to the following upon exit from block 618:

MS ID field 1100a is preferably set with: Same as was described for FIG. 2D (block 236) above.

DATE/TIME STAMP field 1100b is preferably set with: Same as was described for FIG. 2D (block 236) above.

LOCATION field 1100c is preferably set with: The location of the MS as communicated by the service.

CONFIDENCE field 1100d is preferably set with: Confidence (determined by the service) according to how the MS was connected, or may be set with the same value (e.g. 100 for physical connect, 77 for logical connect (e.g. short range wireless)) regardless of how the MS was located. In other embodiments, field 1100d will be determined by the service for anticipated physical conduit range, wireless logical connect range, etc. The resulting confidence value can be adjusted based on other parameters analogously to as described above.

LOCATION TECHNOLOGY field 1100e is preferably set with "Service Physical Connect" or "Service Logical Connect", depending on how the MS connected. The originator indicator is set to DLM.

LOCATION REFERENCE INFO field 1100f is preferably set with: null (not set), but if a TDOA measurement can be made (e.g. short range logical connect, and using methodologies described above), then a TDOA measurement, a communications signal strength, if available; and wave spectrum (and/or particular communications interface 70) used, if available. The TDOA measurement may be converted to a

48

distance using wave spectrum information. Possible values populated here should have already been factored into the confidence value.

COMMUNICATIONS REFERENCE INFO field 1100g is preferably set with: Same as was described for FIG. 2D (block 236) above.

SPEED field 1100h is preferably set with: null (not set), but can be set with speed required to arrive to the current location from a previously known location, assuming same time scale is used.

HEADING field 1100i is preferably set with: null (not set), but can be set to heading determined when arriving to the current location from a previously known location.

ELEVATION field 1100j is preferably set with: Elevation/altitude (e.g. of physical connection, or place of logical connection detection), if available.

APPLICATION FIELDS field 1100k is preferably set with: Same as was described for FIG. 2D (block 236) above.

CORRELATION FIELD 1100m is preferably set with: Not Applicable (i.e. not maintained to queue 22).

SENT DATE/TIME STAMP field 1100n is preferably set with: Not Applicable (i.e. not maintained to queue 22).

RECEIVED DATE/TIME STAMP field 1100p is preferably set with: Not Applicable (i.e. not maintained to queue 22).

FIG. 6B depicts a flowchart for describing a preferred embodiment of a MS whereabouts update event of a physically, or logically, connected MS, for example a DLM 200. A MS may be newly located and physically/logically connected, whereby communications between the MS and service is over a physical/logical connection as described in FIG. 6A above. Relevant processing begins at block 640 and continues to block 642 where an MS device is physically/logically connected. Thereafter, at block 644 the MS accesses the connectivity service and waits for an acknowledgement indicating a successful connection. Upon acknowledgement receipt, processing continues to block 646 where the MS requests WDR information via the connectivity service and waits for the data (i.e. connectivity service may be different than the location service, or may be one in the same). As part of connectivity, location service pointer(s) (e.g. ip address for http://112.34.323.18 referencing or a Domain Name Service (DNS) name like http://www.servicename.com) are provided with the connectivity acknowledgement from the connectivity service at block 644, so the MS knows how to proceed at block 646 for retrieving location information. There are various embodiments for the location service determining a MS location as described above for FIG. 6A. In an alternative embodiment, the MS already knows how to locate itself wherein block 644 continues directly to block 648 (no block 646) because the MS maintains information for determining its own whereabouts using the physical or logical address received in the acknowledgement at block 644. Similar mapping of a network address to the MS location can be in MS data, for example data 36, data 8, or data 20. At block 648, the MS completes its WDR 1100. Thereafter, block 650 prepares FIG. 2F parameters, block 652 invokes FIG. 2F processing already described above, and processing terminates at block 654. Parameters set at block 650 are: WDRREF=a reference or pointer to the MS WDR; DELETEQ=FIG. 6B location queue discard processing; and SUPER=FIG. 6B supervisory notification processing. FIG. 6B processing is available at any appropriate time to the MS.

See FIG. 11A descriptions. Fields are set to the following upon exit from block 648:

MS ID field 1100a is preferably set with: Same as was described for FIG. 2D (block 236) above.

DATE/TIME STAMP field **1100b** is preferably set with: Same as was described for FIG. 2D (block **236**) above.

LOCATION field **1100c** is preferably set with: The location determined for the MS.

CONFIDENCE field **1100d** is preferably set with: Confidence (determined by the service) according to how the MS was connected, or may be set with the same value (e.g. 100 for physical connect, 77 for logical connect (e.g. short range wireless)) regardless of how the MS was located. In other embodiments, field **1100d** will be determined by the service for anticipated physical conduit range, wireless logical connect range, etc. The resulting confidence value can be adjusted based on other parameters analogously to as described above.

LOCATION TECHNOLOGY field **1100e** is preferably set with "Client Physical Connect" or "Client Logical Connect", depending on how the MS connected. The originator indicator is set to DLM.

LOCATION REFERENCE INFO field **1100f** is preferably set with: null (not set), but if a TDOA measurement can be made (e.g. short range logical connect, and using methodologies described above), then a TDOA measurement, a communications signal strength, if available; and wave spectrum (and/or particular communications interface **70**) used, if available. The TDOA measurement may be converted to a distance using wave spectrum information. Possible values populated here should have already been factored into the confidence value.

COMMUNICATIONS REFERENCE INFO field **1100g** is preferably set with: Same as was described for FIG. 2D (block **236**) above.

SPEED field **1100h** is preferably set with: null (not set), but can be set with speed required to arrive to the current location from a previously known location using, assuming same time scale is used.

HEADING field **1100i** is preferably set with: null (not set), but can be set to heading determined when arriving to the current location from a previously known location.

ELEVATION field **1100j** is preferably set with: Elevation/altitude (e.g. of physical connection, or place of logical connection detection), if available.

APPLICATION FIELDS field **1100k** is preferably set with: Same as was described for FIG. 2D (block **236**) above.

CORRELATION FIELD **1100m** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

SENT DATE/TIME STAMP field **1100n** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

RECEIVED DATE/TIME STAMP field **1100p** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

FIGS. 7A, 7B and 7C depict a locating by image sensory illustration for discussing automatic location of a MS, for example a DLM **200**. With reference now to FIG. 7A, an image capture device **702** is positioned for monitoring MSs that come into the field of view **704** of device **702**. Device **702** may be a camcorder, video camera, image camera that takes at least one snapshot, timely snapshots, or motion/presence detection snapshots, or any other device capable of producing at least a snapshot image at some point in time containing objects in the field of view **704**. In one preferred embodiment, DLM **200** is sensed within the vicinity of device **702**, perhaps by antenna (or cell tower) **701**, prior to being photographed by device **702**. In another embodiment, DLM **200** is sensed by movement within the vicinity of device **702** with well know motion detection means. In yet another embodiment, device **702** periodically or continually records. Device **702** is connected to a locating service **700** for processing as described by FIG. 7D. Locating service **700** has means for communicating

wirelessly to DLM **200**, for example through a connected antenna (or cell tower) **701**. FIG. 7A illustrates that device **702** participates in pattern recognition for identifying the location of a MS. The MS can have on its exterior a string of characters, serial number, barcode, license plate, graphic symbol(s), textual symbols, combinations thereof, or any other visually perceptible, or graphical, identification **708** that can be recognized optically, or in a photograph. Device **702** is to have graphical/pixel resolution capability matching the requirements for identifying a MS with the sought graphical identification. Graphical identification **708** can be formed on the perceptible exterior of DLM **200**, or can be formed as part of a housing/apparatus **706** which hosts DLM **200**. Graphical identification **708** can be automatically read from an image using well known barcode reader technology, an Optical Character Recognition (OCR) process, a license tag scanner, general pattern recognition software, or the like. Housing **706** is generally shown for representing an automobile (license plate recognition, for example used in prior art toll tag lanes), a shopping cart, a package, or any other hosting article of manufacture which has a DLM **200** as part of it. Upon recognition, DLM **200** is associated with the location of device **702**. Error in locating an MS will depend on the distance within the field of view **704** from device **702**. A distance may be estimated based on the anticipated size of identification **708**, relative its size determined within the field of view **704**.

With reference now to FIG. 7B, image capture device **702** is positioned for monitoring MSs that come into the field of view **704** of device **702**. MSs are preferably distinguishable by appearance (e.g. color, shape, markings, labels, tags, etc), or as attached (e.g. recognized mount to host) or carried (e.g. recognized by its recognized user). Such techniques are well known to those skilled in the art. Device **702** is as described above with connectivity to locating service **700** and antenna (or cell tower) **701**. FIG. 7B illustrates that device **702** uses known measurements within its field of view for determining how large, and where located, are objects that come into the field of view **704**. For example, a well placed and recognizable vertical line **710a** and horizontal line **710b**, which are preferably perpendicular to each other, have known lengths and positions. The objects which come into the field of view are measured based on the known lengths and positions of the lines **710a** and **710b** which may be landscape markings (e.g. parking lot lines) for additional purpose. Field of view **704** may contain many lines and/or objects of known dimensions strategically placed or recognized within the field of view **704** to facilitate image processing by service **700**. Building **714** may serve as a reference point having known dimension and position in measuring objects such as a person **716** or DLM **200**. A moving object such as a shopping cart **712** can have known dimensions, but not a specific position, to facilitate service **700** in locating an MS coming into the field of view **704**. Those skilled in the art recognize that known dimensions and/or locations of anticipated objects in field of view **704** have measurements facilitating discovering positions and measurements of new objects that may travel into the field of view **704**. Using FIG. 7B techniques with FIG. 7A techniques provides additional locating accuracy. A distance may be estimated based on the anticipated sizes of references in the field of view, relative size of the recognized MS.

With reference now to FIG. 7C, image capture device **702** is positioned for monitoring MSs that come into the field of view **704** of device **702**. Device **702** is as described above with connectivity to locating service **700** and antenna (or cell tower) **701**. MSs are preferably distinguishable by appearance (e.g. color, shape, markings, labels, tags, etc), or as

attached (e.g. recognized mount to host) or carried (e.g. recognized by its user), or as identified by FIG. 7A and/or FIG. 7B methodologies. FIG. 7C illustrates that device 702 uses known locations within its field of view for determining how large, and where located, are objects that come into the field of view 704. For example, building 714, tree 720, and traffic sign 722 have its locations known in field of view 704 by service 700. Solving locations of objects that move into the field of view is accomplished with graphical triangulation measurements between known object reference locations (e.g. building 714, tree 720, and sign 722) and the object to be located. Timely snapshots by device 702 provide an ongoing locating of an MS, for example DLM 200. Line segment distances 724 (a, b, c) can be measured using references such as those of FIG. 7B. Whereabouts are determined by providing known coordinates to anticipated objects such as building 714, tree 720, and sign 722. Similarly, graphical AOA measurements (i.e. graphical angle measurements) and graphical MPT measurements can be used in relation to anticipated locations of objects within the field of view 704. There may be many anticipated (known) object locations within field of view 704 to further facilitate locating an MS. Being nearby an object may also be enough to locate the MS by using the object's location for the location of the MS. Using FIG. 7C techniques with FIG. 7A and/or FIG. 7B techniques provides additional locating accuracy.

The system and methodologies illustrated by FIGS. 7A through 7C are preferably used in optimal combination by locating service 700 to provide a best location of an MS. In some embodiments, MS whereabouts is determined as the location of a device 702 by simply being recognized by the device 702. In other embodiments, multiple devices 702 can be strategically placed within a geographic area for being used in combination to a common locating service 700 for providing a most accurate whereabouts of an MS. Multiple field of views 704 from difference angles of different devices 702 enable more precise locating within three dimensional space, including precise elevations.

FIG. 7D depicts a flowchart for describing a preferred embodiment of graphically locating a MS in accordance with locating service 700 described above, for example as illustrated by FIGS. 7A through 7C. Locating service 700 may be a single capable data processing system, or many connected data processing systems for enhanced parallel processing. Locating service 700 may be connected to services involved with any other locating technology described in this application for synergistic services as an MS is mobile. Locating service 700 begins at block 732 and continues to block 734 where the service 700 is initialized in preparation of MS whereabouts analysis. Block 734 initializes its table(s) of sought identifying criteria which can be pattern recognized. In one preferred embodiment, color/shade, shape, appearance and applicable sought information is initialized for each sought identifying criteria. Pattern recognition is well known in the art and initialization is specific for each technology discussed above for FIGS. 7A through 7C. For FIGS. 7B and 7C discussions, positions, measurements, and reference points of known landmarks are additionally accounted. Thereafter, block 736 gets the next snapshot from device(s) 702. If there is none waiting to get, block 736 waits for one. If there is one queued up for processing, then block 736 continues to block 738. FIG. 7D is processing of a service, and is preferably multi-threaded. For example, blocks 736 through 754 can occur concurrently in many threads for processing a common queue of snapshots received from a device 702, or many devices 702. Each thread may process all sought criteria, or may specialize in a subset of sought criteria wherein if

nothing is found, the thread can place the snapshot back on a queue for thread processing for another sought criteria after marking the queue entry as having been processed for one particular subset. So, threads may be specialized and work together in seeking all criteria, or may each work in parallel seeking the same criteria. In preferred embodiments, there is at least one queue of snapshots received by block(s) 736. Block 736 continues to block 738 which attempts to detect an MS having sought criteria using pattern recognition techniques of FIGS. 7A through 7C, in particular, or in combination. In one example embodiment, as device 702 provides service 700 with at least one timely snapshot to block 736, the snapshot graphic is scanned at block 738 for identifying characters/symbols/appearance of sought criteria. Block 738 continues with its search result to block 740. If block 740 determines no MS was detected, then processing continues back to block 736. If block 738 detected at least one MS (as determined at block 740), then block 742 calculates WDR information for the MS(s) detected, block 744 notifies a supervisory service of MS whereabouts if applicable, block 746 communicates the WDR information to MS(s) detected (for example via antenna 701), and processing continues to block 748.

There may be a plurality of MSs in the field of view, so communications at block 746 targets each MS recognized. A MS should not rely on the service to have done its job correctly. At a MS, block 748 checks the MS ID communicated for validation. If block 748 determines the MS ID is incorrect, then processing continues back to block 736 (for the particular MS). If block 748 determines the MS ID is correct, then processing continues to block 750 where the particular MS completes its WDR 1100 received from service 700. Thereafter, MS(s) prepare parameters at block 752, invoke local FIG. 2F processing already described above (at block 754), and processing continues for service 700 back to block 736. Of course, block 746 continues directly to block 736 at the service(s) since there is no need to wait for MS(s) processing in blocks 748 through 754. Parameters set at block 752 are: WDRREF=a reference or pointer to the MS WDR; DELETEQ=FIG. 7D location queue discard processing; and SUPER=FIG. 7D supervisory notification (e.g. no supervisory notification processing because it was already handled at block 744, or by being in context of the FIG. 7D service processing). No snapshots from device 702 are to be missed at block 736.

See FIG. 11A descriptions. Fields are set to the following upon exit from block 750:

MS ID field 1100a is preferably set with: Unique MS identifier of the MS, after validating at the MS that the service 700 has correctly identified it. This field is used to uniquely distinguish this MS WDRs on queue 22 from other originated WDRs. The service 700 may determine a MS ID from a database lookup using above appearance criteria. Field 1100a may also be determined using the transmission methods as described for FIGS. 2A through 2E, for example by way of antenna 701. For example, when the MS comes within range of antenna 701, FIG. 7D processing commences. Another embodiment prevents recognizing more than one MS within the field of view 704 at any time (e.g. a single file entryway), in which case the service can solicit a "who are you" transmission to identify the MS and then send back its whereabouts (in which case the MS sets its own MS ID here). DATE/TIME STAMP field 1100b is preferably set with: Same as was described for FIG. 2D (block 236) above. LOCATION field 1100c is preferably set with: The location determined for the MS by the service.

53

CONFIDENCE field **1100d** is preferably set with: same value (e.g. **76**) regardless of how the MS location was determined. In other embodiments, field **1100d** will be determined by the number of distance measurements and/or the abundance of particular objects used in the field of view **704**. The resulting confidence value can be adjusted based on other graphical parameters involved, analogously to as described above.

LOCATION TECHNOLOGY field **1100e** is preferably set with: "Server Graphic-Patterns" "Server Graphic-Distances", "Server Graphic Triangulate", or a combination field value depending on how the MS was located and what flavor of service was used. The originator indicator is set to DLM. LOCATION REFERENCE INFO field **1100f** is preferably set with: null (not set) for indicating that all whereabouts determination data was factored into the confidence, and none is relevant for a single TDOA or AOA measurement in subsequent processing (i.e. service did all the work).

COMMUNICATIONS REFERENCE INFO field **1100g** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

SPEED field **1100h** is preferably set with: null (not set), but can be set with speed required to arrive to the current location from a previously known time at a location (e.g. using previous snapshots processed), assuming the same time scale is used.

HEADING field **1100i** is preferably set with: null (not set), but can be set to heading determined when arriving to the current location from a previously known location (e.g. using previous snapshots processed).

ELEVATION field **1100j** is preferably set with: Elevation/altitude, if available, if available.

APPLICATION FIELDS field **1100k** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

CORRELATION FIELD **1100m** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

SENT DATE/TIME STAMP field **1100n** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

RECEIVED DATE/TIME STAMP field **1100p** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

In an alternative embodiment, MS **2** may be equipped (e.g. as part of resources **38**) with its own device **702** and field of view **704** for graphically identifying recognizable environmental objects or places to determine its own whereabouts. In this embodiment, the MS would have access to anticipated objects, locations and dimensions much the same way described for FIGS. **7A** through **7D**, either locally maintained or verifiable with a connected service. Upon a successful recognition of an object, place, or other graphically perceptible image which can be mapped to a location, the MS would complete a WDR similarly to above. The MS may recognize addresses, buildings, landmarks, or other pictorial data. Thus, the MS may graphically determine its own location. The MS would then complete a WDR **1100** for FIG. **2F** processing exactly as described for FIG. **7D** with the exceptions of fields that follow:

MS ID field **1100a** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

LOCATION field **1100c** is preferably set with: The location determined for the MS by the MS.

LOCATION TECHNOLOGY field **1100e** is preferably set with: "Client Graphic-Patterns" "Client Graphic-Distances", "Client Graphic Triangulate", or a combination field value depending on how the MS located itself. The originator indicator is set to DLM.

COMMUNICATIONS REFERENCE INFO field **1100g** is preferably set with: null (not set).

54

FIG. **8A** heterogeneously depicts a locating by arbitrary wave spectrum illustration for discussing automatic location of a MS. In the case of acoustics or sound, prior art has shown that a noise emitting animal or object can be located by triangulating the sound received using TDOA by strategically placed microphones. It is known that by figuring out time delay between a few strategically spaced microphones, one can infer the location of the sound. In a preferred embodiment, an MS, for example DLM **200**, emits a pulsed or constant sound (preferably beyond the human hearing range) which can be sensed by microphones **802** through **806**. Data is superimposed on the sound wave spectrum with variations in pitch or tone, or data occurs in patterned breaks in sound transmission. Data may contain a unique identifier of the MS so service(s) attached to microphones **802** through **806** can communicate uniquely to an MS. In some embodiments, sound used by the MS is known to repel certain pests such as unwanted animals, rodents, or bugs in order to prevent the person carrying the MS from encountering such pests during travel, for example during outdoor hiking or mountain climbing. In submarine acoustics, AOA is a method to locate certain objects. The FIGS. **3B** and **3C** flowcharts occur analogously for sound signals received by microphones **802** through **806** which are connected to service processing of FIGS. **3B** and **3C**. The only difference is wave spectrum used.

It has been shown that light can be used to triangulate position or location information (e.g. U.S. Pat. Nos. 6,549,288 (Migdal et al) and 6,549,289 (Ellis)). Optical sensors **802** through **806** detect a light source of, or illumination of, an MS, for example DLM **200**. Data is superimposed on the light wave spectrum with specified frequency/wavelength and/or periodicity, or data occurs in patterned breaks in light transmission. Data may contain a unique identifier of the MS so service(s) attached to sensors **802** through **806** can communicate uniquely to an MS. Mirrors positioned at optical sensors **802** through **806** may be used to determine an AOA of light at the sensor, or alternatively TDOA of recognizable light spectrum is used to position an MS. The FIGS. **3B** and **3C** flowcharts occur analogously for light signals received by sensors **802** through **806** which are connected to service processing of FIGS. **3B** and **3C**. The only difference is wave spectrum used.

Heterogeneously speaking, FIG. **8A** illustrates having strategically placed sensors **802** through **806** for detecting a wave spectrum and using TDOA, AOA, or MPT. Those skilled in the art appreciate that a wave is analogously dealt with by FIGS. **3B** and **3C** regardless of the wave type, albeit with different sensor types **802** through **806** and different sensor interface to service(s) of FIGS. **3B** and **3C**. Wave signal spectrums for triangulation by analogous processing to FIGS. **3B** and **3C** include microwaves, infrared, visible light, ultraviolet light, X-rays, gamma rays, longwaves, magnetic spectrum, or any other invisible, visible, audible, or inaudible wave spectrum. Sensors **802** through **806** are appropriately matched according to the requirements. Alternatively, a MS may be sensing wave spectrums emitted by transmitters **802** through **806**.

Those skilled in the relevant arts appreciate that the point in all this discussion is all the wave forms provide methods for triangulating whereabouts information of an MS. Different types of wave forms that are available for an MS can be used solely, or in conjunction with each other, to determine MS whereabouts. MSs may be informed of their location using the identical wave spectrum used for whereabouts determination, or may use any other spectrum available for communicating WDR information back to the MS. Alternatively, the MS itself can determine WDR information relative applicable

55

sensors/transmitters. In any case, a WDR **1100** is completed analogously to FIGS. **3B** and **3C**.

FIG. **8B** depicts a flowchart for describing a preferred embodiment of locating a MS through physically sensing a MS, for example a DLM **200**. Processing begins at block **810** upon contact with a candidate MS and continues to block **812** where initialization takes place. Initialization includes determining when, where, and how the contact was made. Then, block **814** takes the contact sample and sets it as input containing a unique identifier or handle of the MS which was sensed. There are various known embodiments of how the MS is sensed:

- a) Touching sensors contact the MS (or host/housing having MS) to interpret physical characteristics of the MS in order to uniquely identify it (e.g. Braille, embossed/raised/depressed symbols or markings, shape, temperature, depressions, size, combinations thereof, etc);
- b) Purchase is made with MS while in vicinity of device accepting purchase, and as part of that transaction, the MS is sensed as being at the same location as the device accepting purchase, for example using a cell phone to purchase a soft drink from a soft drink dispensing machine;
- c) Barcode reader is used by person to scan the MS (or host/housing having MS), for example as part of shipping, receiving, or transporting;
- d) The MS, or housing with MS, is sensed by its odor (or host/housing having MS), perhaps an odor indicating where it had been, where it should not be, or where it should be. Various odor detection techniques may be used;
- e) Optical sensing wherein the MS is scanned with optical sensory means, for example to read a serial number; and/or
- f) Any sensing means which can identify the MS through physical contact, or by nearby/close physical contact with some wave spectrum.

Block **814** continues to block **816** where a database is accessed for recognizing the MS identifier (handle) by mapping sensed information with an associated MS handle. If a match is found at block **818**, then block **822** determines WDR **1100** information using the location of where sensing took place. If block **818** determines no match was found, then data is saved at block **820** for an unrecognized entity such as is useful when an MS should have been recognized, but was not. In another embodiment, the MS handle is directly sensed so block **814** continues directly to block **818** (no block **816**). Block **820** continues to block **834** where processing terminates. Block **816** may not use the entire MS identifier for search, but some portion of it to make sure it is a supported MS for being located by sensing. The MS identifier is useful when communicating wirelessly the WDR information to the MS (at block **826**).

Referring now back to block **822**, processing continues to block **824** where a supervisory service may be updated with the MS whereabouts (if applicable), and block **826** communicates the WDR information to the MS. Any available communication method can be used for communicating the WDR information to the MS, as described above. Thereafter, the MS completes the WDR at block **828**, block **830** prepares FIG. **2F** parameters, and block **832** invokes FIG. **2F** processing already described above. Processing terminates thereafter at block **834**. Parameters set at block **830** are: WDRREF=a reference or pointer to the MS WDR; DELETEQ=FIG. **8B** location queue discard processing; and SUPER=FIG. **8B** supervisory notification (e.g. no supervisory notification processing because it was already handled at block **824**, or by

56

being in context of the FIG. **8B** service processing). FIG. **8B** processing is available at any appropriate time for the MS. In an alternate embodiment, the MS senses its environment to determine whereabouts.

See FIG. **11A** descriptions. Fields are set to the following upon exit from block **828**:

MS ID field **1100a** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

DATE/TIME STAMP field **1100b** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

LOCATION field **1100c** is preferably set with: Location of the sensor sensing the MS.

CONFIDENCE field **1100d** is preferably set with: Should be high confidence (e.g. **98**) for indisputable contact sensing and is typically set with the same value.

LOCATION TECHNOLOGY field **1100e** is preferably set with: "Contact", or a specific type of Contact. The originator indicator is set to DLM.

LOCATION REFERENCE INFO field **1100f** is preferably set with: null (not set).

COMMUNICATIONS REFERENCE INFO field **1100g** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

SPEED field **1100h** is preferably set with: null (not set), but can be set with speed required to arrive to the current location from a previously known time at a location, assuming the same time scale is used.

HEADING field **1100i** is preferably set with: null (not set), but can be set to heading determined when arriving to the current location from a previously known location.

ELEVATION field **1100j** is preferably set with: Elevation/altitude, if available.

APPLICATION FIELDS field **1100k** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

CORRELATION FIELD **1100m** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

SENT DATE/TIME STAMP field **1100n** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

RECEIVED DATE/TIME STAMP field **1100p** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

FIG. **8C** depicts a flowchart for describing a preferred embodiment of locating a MS, for example a DLM **200**, through a manually entered location of the MS. MS user interface processing begins at block **850** when a user starts the user interface from code **18** and continues to block **852**. Any of a variety of user interfaces, dependent on the type of MS, is used for manually entering the location of the MS. A user interfaces with the MS at block **852** until one of the monitored actions relevant to this disclosure are detected. Thereafter, if block **854** determines the user has selected to set his location manually, then processing continues to block **860**. If block **854** determines the user did not select to manually set his location, then block **856** determines if the user selected to force the MS to determine its location. If the user did select to force the MS to get its own location, then block **856** continues to block **862**. If the user did not select to force the MS to get its own location as determined by block **856**, then processing continues to block **858**. If block **858** determines the user wanted to exit the user interface, then block **880** terminates the interface and processing terminates at block **882**. If block **858** determines the user did not want to exit the user interface, then block **884** handles any user interface actions which caused exit from block **852** yet were not handled by any action processing relevant to this disclosure.

57

With reference back to block **860**, the user interfaces with the MS user interface to manually specify WDR information. The user can specify:

- 1) An address or any address subset such as a zip code;
- 2) Latitude, longitude, and elevation;
- 3) MAPSCO identifier;
- 4) FEMA map identifier;
- 5) USDA map identifier;
- 6) Direct data entry to a WDR **1100**; or
- 7) Any other method for user specified whereabouts of the MS.

The user can specify a relevant confidence value for the manually entered location, however, processing at block **860** preferably automatically defaults a confidence value for the data entered. For example, a complete address, validated at block **860**, will have a high confidence. A partial address such as city and state, or a zip code will have a low confidence value. The confidence value will reflect how large an area is candidate for where the MS is actually located. To prevent completely relying on the user at block **860** for accurate WDR information, validation embodiments may be deployed. Some examples:

Upon specification (e.g. FEMA), the MS will access connected service(s) to determine accuracy (FEMA conversion tables);

Upon specification (e.g. MAPSCO), the MS will access local resources to help validate the specification (e.g. MAPSCO conversion tables); and/or

Upon specification (e.g. address), the MS can access queue **22** and/or history **30** for evidence proving likelihood of accuracy. The MS may also access services, or local resources, for converting location information for proper comparisons.

In any case, a confidence field **1100d** value can be automatically set based on the validation results, and the confidence may, or may not, be enabled for override by the user.

After WDR information is specified at block **860**, the MS completes the WDR at block **874**, block **876** prepares parameters for FIG. **2F** processing, and (at block **878**) the MS invokes FIG. **2F** processing already described above before returning back to block **852**. Parameters set at block **876** are: WDRREF=a reference or pointer to the MS WDR; DELETEQ=FIG. **8C** location queue discard processing; and SUPER=FIG. **8C** supervisory notification processing. Various embodiments permit override of the confidence floor value by the user, or by FIG. **8C** processing. Block **874** may convert the user specified information into a standardized more usable form in an LN-expanse (e.g. convert to latitude and longitude if possible, truncated precision for more area coverage). WDR **1100** fields (see FIG. **11A**) are set analogously in light of the many variations already described above.

With reference back to block **862**, if it is determined that the MS is equipped with capability (e.g. in range, or in readiness) to locate itself, then processing continues to block **864** where the MS locates itself using MS driven capability described by FIGS. **2E**, **3C**, **4B**, **6B**, and **8A** or MS driven alternative embodiments to FIGS. **2D**, **3B**, **5B**, **6A**, **7D**, **8A**, and **8B**, or any other MS capability for determining its own whereabouts with or without help from other data processing systems or services. Interfacing to locating capability preferably involves a timeout in case there is no, or slow, response, therefore block **864** continues to block **868** where it determined whether or not block **864** timed out prior to determining a location. If block **868** determines a timeout was encountered, then block **872** provides the user with an error to the

58

user interface, and processing continues back to block **852**. Block **872** preferably requires use acknowledgement prior to continuing to block **852**.

If block **868** determines there was no timeout (i.e. whereabouts successfully determined), then block **870** interfaces to the locating interface to get WDR information, block **874** completes a WDR, and blocks **876** and **878** do as described above. If block **862** determines the MS cannot locate itself and needs help, then block **866** emits at least one broadcast request to any listening service which can provide the MS its location. Appropriate correlation is used for an anticipated response. Example services listening are service driven capability described by FIGS. **2D**, **3B**, **5B**, **6A**, **7D**, **8A**, and **8B**, or service side alternative embodiments of FIGS. **2E**, **3C**, **4B**, **6B**, and **8A**, or any other service capability for determining MS whereabouts with or without help from the MS or other data processing systems or services. Block **866** then continues to block **868**.

If block **868** determines a timeout was encountered from the service broadcast request, then block **872** provides the user with an error to the user interface, and processing continues back to block **852**. If block **868** determines there was no timeout (i.e. whereabouts successfully determined), then block **870** receives WDR information from the locating interface of the responding service, block **874** completes a WDR, and blocks **876** and **878** do as already described above.

See FIG. **11A** descriptions. Depending how the MS was located via processing started at block **856** to block **862**, a WDR is completed analogous to as described in Figs. above. If the user manually specified whereabouts at block **860**, fields are set to the following upon exit from block **874**:

MS ID field **1100a** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

DATE/TIME STAMP field **1100b** is preferably set with: Same as was described for FIG. **2D** (block **236**) above.

LOCATION field **1100c** is preferably set with: Location entered by the user, or converted from entry by the user; preferably validated.

CONFIDENCE field **1100d** is preferably set with: User specified confidence value, or a system assigned value per a validated manual specification. Confidence should reflect confidence of location precision (e.g. validated full address high; city and zip code low, etc). Manually specified confidences are preferably lower than other location technologies since users may abuse or set incorrectly, unless validated. Specifying lower confidence values than technologies above, for completely manual WDR specifications (i.e. no validation), ensures that manual specifications are only used by the MS in absence of other technologies. There are many validation embodiments that can be deployed (as described above) for a manually entered address wherein the resulting confidence may be based on validation(s) performed (e.g. compare recent history for plausible current address, use current latitude and longitude for database lookup to compare with address information entered, etc). The system and/or user may or may not be able to override the confidence value determined.

LOCATION TECHNOLOGY field **1100e** is preferably set with: "Manual", or "Manual Validated". Types of validations may further be elaborated. The originator indicator is set to DLM.

LOCATION REFERENCE INFO field **1100f** is preferably set with: null (not set).

COMMUNICATIONS REFERENCE INFO field **1100g** is preferably set with: null (not set).

SPEED field **1100h** is preferably set with: null (not set).

HEADING field **1100i** is preferably set with: null (not set).

ELEVATION field **1100j** is preferably set with: null (not set). APPLICATION FIELDS field **1100k** is preferably set with: Same as was described for FIG. 2D (block **236**) above; or as decided by the user.

CORRELATION FIELD **1100m** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

SENT DATE/TIME STAMP field **1100n** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

RECEIVED DATE/TIME STAMP field **1100p** is preferably set with: Not Applicable (i.e. not maintained to queue **22**).

FIG. 9A depicts a table for illustrating heterogeneously locating a MS, for example a DLM **200**. While many location methods and systems have been exhausted above, there may be other system and methods for locating an MS which apply to the present disclosure. The requirement for LBX is that the MS be located, regardless of how that occurs. MSs disclosed herein can be located by one or many location technologies discussed. As MS prices move lower, and capabilities increase, an affordable MS will contain multiple abilities for being located. GPS, triangulation, in-range detection, and contact sensory may all be used in locating a particular MS as it travels. Equipping the MS with all techniques is straightforward and is compelling when there are competing, or complementary, technologies that the MS should participate in.

The FIG. 9A table has DLM location methods for rows and a single column for the MS (e.g. DLM **200**). Each location technology can be driven by the client (i.e. the MS), or a service (i.e. the location server(s)) as denoted by a row qualifier "C" for client or "S" for service. An MS may be located by many technologies. The table illustrated shows that the MS with unique identifier 0A12:43 EF:985B:012F is able to be heterogeneously located, specifically with local MS GPS capability, service side cell tower in-range detection, service side cell tower TDOA, service side cell tower MPT (combination of TDOA and AOA), service side antenna in-range detection, service side antenna AOA, service side antenna TDOA, service side antenna MPT, service side contact/sensory, and general service side MPT. The unique identifier in this example is a universal product identifier (like Host Bus Adapter (HBA) World Wide Name (WWN) identifiers are generated), but could be in other form as described above (e.g. phone #214-403-4071). An MS can have any subset of technologies used to locate it, or all of the technologies used to locate it at some time during its travels. An MS is heterogeneously located when two or more location technologies are used to locate the MS during MS travels and/or when two or more location technologies with incomplete results are used in conjunction with each other to locate the MS during MS travels, such as MPT. MPT is a heterogeneous location technology because it uses at least two different methods to accomplish a single location determination. Using combinations of different location technologies can be used, for example a TDOA measurement from an in-range antenna with a TDOA measurement relative a cell tower (e.g. as accomplished in MS processing of FIG. 26B), using completely different services that have no knowledge of each other. Another combination is to use a synergy of whereabouts data from one technology with whereabouts data from another technology. For example, in-range detection is used in combination with graphical identification to provide better whereabouts of a MS. In another example, a GPS equipped MS travels to an area where GPS does not work well (e.g. downtown amidst large and tall buildings). The DLM becomes an ILM, and is triangulated relative other MSs. So, an MS is heterogeneously located using two or more tech-

nologies to determine a single whereabouts, or different whereabouts of the MS during travel.

FIG. 9B depicts a flowchart for describing a preferred embodiment of heterogeneously locating a MS, for example DLM **200**. While heterogeneously locating an MS can occur by locating the MS at different times using different location technologies, flowchart **9B** is shown to discuss a generalization of using different location technologies with each other at the same time to locate an MS. Processing begins at block **950** and continues to block **952** where a plurality of parameters from more than one location technology are examined for locating an MS. Processing begins at block **950** by a service (or the MS) when a location technology by itself cannot be used to confidently locate the MS. Data deemed useful at block **952**, when used in conjunction with data from a different location technology to confidently locate the MS, is passed for processing to block **954**. Block **954** heterogeneously locates the MS using data from at least two location technologies to complement each other and to be used in conjunction with each other in order to confidently locate the MS. Once the MS whereabouts are determined at block **954**, WDR information is communicated to the MS for further processing at block **956**. In some embodiments where a service is heterogeneously locating the MS, block **956** communicates WDR information wirelessly to the MS before processing begins at block **958**. In another embodiment where the MS is heterogeneously locating itself, block **956** communicates WDR information internally to WDR completion processing at block **958**. In preferred embodiments, the MS completes its WDR information at block **958**, FIG. 2F parameters are prepared at block **960**, and the MS invokes FIG. 2F processing already described above (at block **962**), before processing terminates at block **964**. Parameters set at block **960** are: WDRREF=a reference or pointer to the MS WDR; DELETEQ=FIG. 9B location queue discard processing; and SUPER=FIG. 9B supervisory notification processing. WDR **1100** fields (see FIG. 11A) are set analogously in light of many variations already described above.

In some embodiments of FIG. 9B processing, Missing Part Triangulation (MPT) is used to heterogeneously locate an MS. For a service side embodiment example, block **950** begins service processing when TDOA information itself cannot be used to confidently locate the MS, or AOA information itself cannot be used to confidently locate the MS, however using angles and distances from each in conjunction with each other enables solving whereabouts confidently. See "Missing Part Triangulation (MPT)" section below with discussions for FIGS. 11A through 11E for MPT processing of blocks **952** and **954**. Data discovered at block **952** and processed by block **954** depends on the embodiment, what stationary reference point locations are known at the time of blocks **952** and **954** processing, and which parts are missing for triangulating the MS. Having three (3) sides (all TDOA) with known stationary vertices location(s) solves the triangle for locating the MS. Three (3) angles (all AOA) with known stationary vertices location(s) solves the triangle for locating the MS. Those skilled in the art appreciate that solving triangulation can make complementary use of different distances (time used to determine length in TDOA) and angles (from AOA) for deducing a MS location confidently (e.g. MPT). Those skilled in the art recognize that having stationary reference locations facilitates requiring less triangular information for deducing a MS location confidently.

While MPT has been discussed by example, flowchart **9B** is not to be interpreted in a limiting sense. Any location technologies, for example as shown in FIG. 9A, can be used in conjunction with each other when not all information

required is available in a single location technology to confidently deduce an MS location. Data available from the different location technologies available will be examined on its own merits, and optionally used in conjunction to deduce a confident location. For example, a TDOA (difference between when signal sent and when received) measurement from “coming within range” technology can be used to distinguish how close, or how far, is an MS in the vicinity. That measurement may be used to more confidently locate the MS using other TDOA measurements from other unrelated “coming within range” whereabouts information.

With the many DLM examples above, it should be clear now to the reader how to set the WDR 1100 for DLM invoked FIG. 2F processing. There can be other location technologies that will set WDR 1100 fields analogously. Locating methodologies of FIGS. 2A through 9B can be used in any combination, for example for more timely or accurate locating. Furthermore, a MS automatically takes on a role of a DLM or ILM depending on what capability is available at the time, regardless of whether or not the MS is equipped for being directly located. As a DLM roams to unsupported areas, it can remain a DLM using different DLM technologies, and it can become an ILM to depend on other MSs (ILMs or DLMs) in the vicinity to locate it.

LBX Indirectly Located Mobile Data Processing Systems (ILMs)

FIGS. 10A and 10B depict an illustration of a Locatable Network Expanse (LN-Expanse) 1002 for describing locating of an ILM with all DLMs. With reference now to FIG. 10A, DLM 200a, DLM 200b, DLM 200c, DLM 200d, and DLM 200e (referred to generally in FIGS. 10A and 10B discussions as DLMs 200) are each automatically and directly located, for example using any of the automatic location technologies heretofore described. ILM 1000b is automatically located using the reference locations of DLM 200b, DLM 200c, and DLM 200e. DLMs 200 can be mobile while providing reference locations for automatically determining the location of ILM 1000b. Timely communications between MSs is all that is required for indirectly locating MSs. In some embodiments, DLMs 200 are used to triangulate the position of ILM 1000b using aforementioned wave spectrum(s) reasonable for the MSs. Different triangulation embodiments can triangulate the location of ILM 1000b using TDOA, AOA, or MPT, preferably by the ILM 1000b seeking to be located. In other embodiments, TDOA information is used to determine how close ILM 1000b is to a DLM for associating the ILM at the same location of a DLM, but with how close nearby. In other embodiments, an ILM is located by simply being in communications range to another MS. DLMs 200 can be referenced for determining elevation of an ILM. The same automatic location technologies used to locate a DLM can be used to automatically locate an ILM, except the DLMs are mobile and serve as the reference points. It is therefore important that DLM locations be timely known when references are needed for locating ILMs. Timely ILM interactions with other MSs, and protocol considerations are discussed in architecture 1900 below. DLMs 200b, 200c, and 200e are preferably selected for locating ILM 1000b by their WDR high confidence values, however any other WDR data may be used whereby wave spectrum, channel signal strength, time information, nearness, surrounded-ness, etc is considered for generating a confidence field 1100d of the WDR 1100 for the located ILM. Preferably, those considerations are factored into a confidence value, so that confidence values can be completely relied upon.

With reference now to FIG. 10B, ILM 1000c has been located relative a plurality of DLMs, namely DLM 200b, DLM 200d, and DLM 200e. ILM 1000c is located analogously to ILM 1000b as described for FIG. 10A, except there are different DLMs involved with doing the locating of ILM 1000c because of a different location of ILM 1000c. FIGS. 10A and 10B illustrate that MSs can be located using other MSs, rather than fixed stationary references described for FIGS. 2A through 9B. ILM 1000b and ILM 1000c are indirectly located using DLMs 200.

FIG. 10C depicts an illustration of a Locatable Network Expanse (LN-Expanse) 1002 for describing locating of an ILM with an ILM and DLM. ILM 1000a is automatically located using the reference locations of DLM 200c, DLM 200b, and ILM 1000b. DLM 200b, DLM 200c and ILM 1000b can be mobile while providing reference locations for automatically determining the location of ILM 1000a. In some embodiments, MSs are used to triangulate the position of ILM 1000a using any of the aforementioned wave spectrum(s) (e.g. WiFi, cellular radio, etc) reasonable for the MSs. Different triangulation embodiments can triangulate the location of ILM 1000a using TDOA, AOA, or MPT, preferably by the ILM 1000a seeking to be located. In other embodiments, TDOA information is used to determine how close ILM 1000a is to a MS (DLM or ILM) for associating the ILM at the same location of a MS, but with how close nearby. In other embodiments, an ILM is located by simply being in communications range to another MS. DLMs or ILMs can be referenced for determining elevation of ILM 1000a. The same automatic location technologies used to locate a MS (DLM or ILM) are used to automatically locate an ILM, except the MSs are mobile and serve as the reference points. It is therefore important that MS (ILM and/or DLM) locations be timely known when references are needed for locating ILMs. Timely ILM interactions with other MSs, and protocol considerations are discussed in architecture 1900 below. DLM 200b, DLM 200c, and ILM 1000b are preferably selected for locating ILM 1000a by their WDR high confidence values, however any other WDR data may be used whereby wave spectrum, channel signal strength, time information, nearness, surrounded-ness, etc is considered for generating a confidence field 1100d of the WDR 1100 for the located ILM. Preferably, those considerations were already factored into a confidence value so that confidence values can be completely relied upon. ILM 1000a is indirectly located using DLM(s) and ILM(s).

FIGS. 10D, 10E, and 10F depict an illustration of a Locatable Network Expanse (LN-Expanse) 1002 describing locating of an ILM with all ILMs. With reference now to FIG. 10D, ILM 1000e is automatically located using the reference locations of ILM 1000a, ILM 1000b, and ILM 1000c. ILM 1000a, ILM 1000b and ILM 1000c can be mobile while providing reference locations for automatically determining the location of ILM 1000e. Timely communications between MSs is all that is required. In some embodiments, MSs are used to triangulate the position of ILM 1000e using any of the aforementioned wave spectrum(s) reasonable for the MSs. Different triangulation embodiments can triangulate the location of ILM 1000e using TDOA, AOA, or MPT processing (relative ILMs 1000a through 1000c), preferably by the ILM 1000e seeking to be located. ILMs can be referenced for determining elevation of ILM 1000e. The same automatic location technologies used to locate a MS (DLM or ILM) are used to automatically locate an ILM, except the MSs are mobile and serve as the reference points. It is therefore important that ILM locations be timely known when references are needed for locating ILMs. Timely ILM interactions with other MSs,

63

and protocol considerations are discussed in architecture **1900** below. ILM **1000a**, ILM **1000b**, and ILM **1000c** are preferably selected for locating ILM **1000e** by their WDR high confidence values, however any other WDR data may be used whereby wave spectrum, channel signal strength, time information, nearness, surrounded-ness, etc is considered for generating a confidence field **1100d** of the WDR **1100** for the located ILM. Preferably, those considerations were already factored into a confidence value so that confidence values can be completely relied upon. ILM **1000e** is indirectly located using ILM **1000a**, ILM **1000b**, and ILM **1000c**.

With reference now to FIG. **10E**, ILM **1000g** is automatically located using the reference locations of ILM **1000a**, ILM **1000c**, and ILM **1000e**. ILM **1000a**, ILM **1000c** and ILM **1000e** can be mobile while providing reference locations for automatically determining the location of ILM **1000g**. ILM **1000g** is located analogously to ILM **1000e** as described for FIG. **10D**, except there are different ILMs involved with doing the locating of ILM **1000g** because of a different location of ILM **1000g**. Note that as ILMs are located in the LN-expanse **1002**, the LN-expanse expands with additionally located MSs.

With reference now to FIG. **10F**, ILM **1000i** is automatically located using the reference locations of ILM **1000f**, ILM **1000g**, and ILM **1000h**. ILM **1000f**, ILM **1000g** and ILM **1000h** can be mobile while providing reference locations for automatically determining the location of ILM **1000i**. ILM **1000i** is located analogously to ILM **1000e** as described for FIG. **10D**, except there are different ILMs involved with doing the locating of ILM **1000i** because of a different location of ILM **1000i**. FIGS. **10D** through **10F** illustrate that an MS can be located using all ILMs, rather than all DLMs (FIGS. **10A** and **10B**), a mixed set of DLMs and ILMs (FIG. **10C**), or fixed stationary references (FIGS. **2A** through **9B**). ILMs **1000e**, **1000g**, and **1000i** are indirectly located using ILMs. Note that in the FIG. **10** illustrations the LN-expanse **1002** has expanded down and to the right from DLMs directly located up and to the left. It should also be noted that locating any MS can be done with at least one other MS. Three are not required as illustrated. It is preferable that triangulation references used surround an MS.

FIGS. **10G** and **10H** depict an illustration for describing the reach of a Locatable Network expanse (LN-Expanse) according to MSs. Location confidence will be dependent on the closest DLMs, how stale an MS location becomes for serving as a reference point, and how timely an MS refreshes itself with a determined location. An MS preferably has highest available processing speed with multithreaded capability in a plurality of hardware processors and/or processor cores. A substantially large number of high speed concurrent threads of processing that can occur within an MS provides for an optimal capability for being located quickly among its peer MSs, and for serving as a reference to its peer MSs. MS processing described in flowcharts herein assumes multiple threads of processing with adequate speed to accomplish an optimal range in expanding the LN-Expanse **1002**.

With reference now to FIG. **10G**, an analysis of an LN-Expanse **1002** will contain at least one DLM region **1022** containing a plurality of DLMs, and at least one DLM indirectly located region **1024** containing at least one ILM that has been located with all DLMs. Depending on the range, or scope, of an LN-Expanse **1002**, there may be a mixed region **1026** containing at least one ILM that has been indirectly located by both an ILM and DLM, and there may be an exclusive ILM region **1028** containing at least one ILM that has been indirectly located by all ILMs. The further in distance the LN-Expanse has expanded from DLM region **1022**

64

with a substantial number of MSs, the more likely there will be an exclusive ILM region **1028**. NTP may be available for use in some regions, or some subset of a region, yet not available for use in others. NTP is preferably used where available to minimize communications between MSs, and an MS and service(s). An MS has the ability to make use of NTP when available.

With reference now to FIG. **10H**, all MSs depicted know their own locations. The upper left-hand portion of the illustration consists of region **1022**. As the reader glances more toward the rightmost bottom portion of the illustration, there can be regions **1024** and regions **1026** in the middle of the illustration. At the very rightmost bottom portion of the illustration, remaining ILMs fall in region **1028**. An ILM is indirectly located relative all DLMs, DLMs and ILMs, or all ILMs. An "Affirmifier" in a LN-expanse confidently knows its own location and can serve as a reference MS for other MSs. An affirmifier is said to "affirmify" when in the act of serving as a reference point to other MSs. A "Pacifier" can contribute to locating other systems, but with a low confidence of its own whereabouts. The LN-Expanse is a network of located/locatable MSs, and is preferably expanded by a substantial number of affirmifiers.

FIG. **10I** depicts an illustration of a Locatable Network expanse (LN-Expanse) for describing a supervisory service, for example supervisory service **1050**. References in flowcharts for communicating information to a supervisory service can refer to communicating information to supervisory service **1050** (e.g. blocks **294** and **296** from parameters passed to block **272** for many processing flows). The only requirement is that supervisory service **1050** be contactable from an MS (DLM or ILM) that reports to it. An MS reporting to service **1050** can communicate directly to it, through another MS (i.e. a single hop), or through a plurality of MSs (i.e. a plurality of hops). Networks of MSs can be preconfigured, or dynamically reconfigured as MSs travel to minimize the number of hops between a reporting MS and service **1050**. A purely peer to peer preferred embodiment includes a peer to peer network of located/locatable MSs that interact with each other as described herein. The purely peer to peer preferred embodiment may have no need to include a service **1050**. Nevertheless, a supervisory service may be warranted to provide certain processing centralization, or for keeping information associated with MSs. In some embodiments, supervisory service **1050** includes at least one database to house data (e.g. data **8**; data **20**; data **36**; data **38**, queue data **22**, **24**, **26**; and/or history **30**) for any subset of MSs which communicate with it, for example to house MS whereabouts information.

FIG. **11A** depicts a preferred embodiment of a Whereabouts Data Record (WDR) **1100** for discussing operations of the present disclosure. A Whereabouts Data Record (WDR) **1100** may also be referred to as a Wireless Data Record (WDR) **1100**. A WDR takes on a variety of formats depending on the context of use. There are several parts to a WDR depending on use. There is an identity section which contains a MS ID field **1100a** for identifying the WDR. Field **1100a** can contain a null value if the WDR is for whereabouts information received from a remote source which has not identified itself. MSs do not require identities of remote data processing systems in order to be located. There is a core section which is required in WDR uses. The core section includes date/time stamp field **1100b**, location field **1100c**, and confidence field **1100d**. There is a transport section of fields wherein any one the fields may be used when communicating WDR information between data processing systems. Transport fields include correlation field **1100m**, sent date/time stamp field **1100n**, and received date/time stamp field **1100p**.

Transport fields may also be communicated to send processing (e.g. queue 24), or received from receive processing (e.g. queue 26). Other fields are of use depending on the MS or applications thereof, however location technology field 1100e and location reference info field 1100f are of particular interest in carrying out additional novel functionality of the present disclosure. Communications reference information field 1100g may be valuable, depending on communications embodiments in the LN-expanse.

Some fields are multi-part fields (i.e. have sub-fields). Whereabouts Data Records (WDRs) 1100 may be fixed length records, varying length records, or a combination with field(s) in one form or the other. Some WDR embodiments will use anticipated fixed length record positions for subfields that can contain useful data, or a null value (e.g. -1). Other WDR embodiments may use varying length fields depending on the number of sub-fields to be populated. Other WDR embodiments will use varying length fields and/or sub-fields which have tags indicating their presence. Other WDR embodiments will define additional fields to prevent putting more than one accessible data item in one field. In any case, processing will have means for knowing whether a value is present or not, and for which field (or sub-field) it is present. Absence in data may be indicated with a null indicator (-1), or indicated with its lack of being there (e.g. varying length record embodiments).

When a WDR is referenced in this disclosure, it is referenced in a general sense so that the contextually reasonable subset of the WDR of FIG. 11A is used. For example, when communicating WDRs (sending/receiving data 1302 or 1312) between data processing systems, a reasonable subset of WDR 1100 is communicated in preferred embodiments as described with flowcharts. When a WDR is maintained to queue 22, preferably most (if not all) fields are set for a complete record, regardless if useful data is found in a particular field (e.g. some fields may be null (e.g. -1)). Most importantly, Whereabouts Data Records (WDRs) are maintained to queue 22 for maintaining whereabouts of the MS which owns queue 22. LBX is most effective the more timely (and continuous) a MS has valid whereabouts locally maintained. WDRs are designed for maintaining whereabouts information independent of any location technology applied. Over time, a MS may encounter a plurality of location technologies used to locate it. WDRs maintained to a first MS queue 22 have the following purpose:

- 1) Maintain timely DLM whereabouts information of the first MS independent of any location technology applied;
- 2) Maintain whereabouts information of nearby MSs independent of any location technology applied;
- 3) Provide DLM whereabouts information to nearby MSs for determining their own locations (e.g. provide whereabouts information to at least a second MS for determining its own location);
- 4) Maintain timely ILM whereabouts information of the first MS independent of any location technology applied; and
- 5) Provide ILM whereabouts information to nearby MSs so they can determine their own locations (e.g. first MS providing whereabouts information to at least a second MS for the second MS determining its own whereabouts).

A MS may go in and out of DLM or ILM roles as it is mobile. Direct location methods are not always available to the MS as it roams, therefore the MS preferably does all of 1 through 5 above. When the WDR 1100 contains a MS ID field 1100a matching the MS which owns queue 22, that WDR

contains the location (location field 1100c) with a specified confidence (field 1100d) at a particular time (date/time stamp field 1100b) for that MS. Preferably the MS ID field 1100a, date/time stamp field 1100b and confidence field 1100d is all that is required for searching from the queue 22 the best possible, and most timely, MS whereabouts at the time of searching queue 22. Other embodiments may consult any other fields to facilitate the best possible MS location at the time of searching and/or processing queue 22. The WDR queue 22 also maintains affirmifier WDRs, and acceptable confidence pacifier WDRs (block 276), which are used to calculate a WDR having matching MS field 1100a so the MS knows its whereabouts via indirect location methods. Affirmifier and pacifier WDRs have MS ID field 1100a values which do not match the MS owning queue 22. This distinguishes WDRs of queue 22 for A) accessing the current MS location; from B) the WDRs from other MSs. All WDR fields of affirmifier and pacifier originated WDRs are of importance for determining a best location of the MS which owns queue 22, and in providing LBX functionality.

MS ID field 1100a is a unique handle to an MS as previously described. Depending on the installation, MS ID field 1100a may be a phone #, physical or logical address, name, machine identifier, serial number, encrypted identifier, concealable derivative of a MS identifier, correlation, pseudo MS ID, or some other unique handle to the MS. An MS must be able to distinguish its own unique handle from other MS handles in field 1100a. For indirect location functionality disclosed herein, affirmifier and pacifier WDRs do not need to have a correct originating MS ID field 1100a. The MS ID may be null, or anything to distinguish WDRs for MS locations. However, to accomplish other LBX features and functionality, MS Identifiers (MS IDs) of nearby MSs (or unique correlations thereof) maintained in queue 22 are to be known for processing by an MS. MS ID field 1100a may contain a group identifier of MSs in some embodiments for distinguishing between types of MSs (e.g. to be treated the same, or targeted with communications, as a group), as long as the MS containing queue 22 can distinguish its own originated WDRs 1100. A defaulted value may also be set for a "do not care" setting (e.g. null).

Date/Time stamp field 1100b contains a date/time stamp of when the WDR record 1100 was completed by an MS for its own whereabouts prior to WDR queue insertion. It is in terms of the date/time scale of the MS inserting the local WDR (NTP derived or not). Date/Time stamp field 1100b may also contain a date/time stamp of when the WDR record 1100 was determined for the whereabouts of an affirmifier or pacifier originating record 1100 to help an MS determine its own whereabouts, but it should still be in terms of the date/time scale of the MS inserting the local WDR (NTP derived or not) to prevent time conversions when needed, and to promote consistent queue 22 searches/sorts/etc. The date/time stamp field 1100b should use the best possible granulation of time, and may be in synch with other MSs and data processing systems according to NTP. A time zone, day/light savings time, and NTP indicator is preferably maintained as part of field 1100b. The NTP indicator (e.g. bit) is for whether or not the date/time stamp is NTP derived (e.g. the NTP use setting is checked for setting this bit when completing the WDR for queue 22 insertion). In some embodiments, date/time stamp field 1100b is measured in the same granulation of time units to an atomic clock available to MSs of an LN-Expanse 1002. When NTP is used in a LN-Expanse, identical time server sources are not a requirement provided NTP derived date/time stamps have similar accuracy and dependability.

67

Location field **1100c** depends on the installation of the present disclosure, but can include a latitude and longitude, cellular network cell identifier, geocentric coordinates, geodetic coordinates, three dimensional space coordinates, area described by GPS coordinates, overlay grid region identifier or coordinates, GPS descriptors, altitude/elevation (e.g. in lieu of using field **1100j**), MAPSCO reference, physical or logical network address (including a wildcard (e.g. ip addresses 145.32.*.*)), particular address, polar coordinates, or any other two/three dimensional location methods/means used in identifying the MS location. Data of field **1100c** is preferably a consistent measure (e.g. all latitude and longitude) for all location technologies that populate WDR queue **22**. Some embodiments will permit using different measures to location field **1100c** (e.g. latitude and longitude for one, address for another; polar coordinates for another, etc) which will be translated to a consistent measure at appropriate processing times.

Confidence field **1100d** contains a value for the confidence that location field **1100c** accurately describes the location of the MS when the WDR is originated by the MS for its own whereabouts. Confidence field **1100d** contains a value for the confidence that location field **1100c** accurately describes the location of an affirmifier or pacifier that originated the WDR. A confidence value can be set according to known timeliness of processing, communications and known mobile variables (e.g. MS speed, heading, yaw, pitch, roll, etc) at the time of transmission. Confidence values should be standardized for all location technologies used to determine which location information is of a higher/lower confidence when using multiple location technologies (as determined by fields **1100e** and **1100f**) for enabling determination of which data is of a higher priority to use in determining whereabouts. Confidence value ranges depend on the implementation. In a preferred embodiment, confidence values range from 1 to 100 (as discussed previously) for denoting a percentage of confidence. 100% confidence indicates the location field **1100c** is guaranteed to describe the MS location. 0% confidence indicates the location field **1100c** is guaranteed to not describe the MS location. Therefore, the lowest conceivable value of a queue **22** for field **1100d** should be 1. Preferably, there is a lowest acceptable confidence floor value configured (by system, administrator, or user) as used at points of queue entry insertion—see block **276** to prevent frivolous data to queue **22**. In most cases, WDRs **1100** contain a confidence field **1100d** up to 100. In confidence value preferred embodiments, pacifiers know their location with a confidence of less than 75, and affirmifiers know their location with a confidence value 75 or greater. The confidence field is skewed to lower values as the LN-expanse **1002** is expanded further from region **1022**. Confidence values are typically lower when ILMs are used to locate a first set of ILMs (i.e. first tier), and are then lower when the first set of ILMs are used to locate a second set of ILMs (second tier), and then lower again when the second set of ILMs are used to locate a third set of ILMs (third tier), and so on. Often, examination of a confidence value in a WDR **1100** can indicate whether the MS is a DLM, or an ILM far away from DLMs, or an MS which has been located using accurate (high confidence) or inaccurate (low confidence) locating techniques.

Location Technology field **1100e** contains the location technology used to determine the location of location field **1100c**. An MS can be located by many technologies. Location Technology field **1100e** can contain a value from a row of FIG. **9A** or any other location technology used to locate a MS. WDRs inserted to queue **22** for MS whereabouts set field **1100e** to the technology used to locate the MS. WDRs

68

inserted to queue **22** for facilitating a MS in determining whereabouts set field **1100e** to the technology used to locate the affirmifier or pacifier. Field **1100e** also contains an originator indicator (e.g. bit) for whether the originator of the WDR **1100** was a DLM or ILM. When received from a service that has not provided confidence, this field may be used by a DLM to determine confidence field **1100d**.

Location Reference Info field **1100f** preferably contains one or more fields useful to locate a MS in processing subsequent of having been inserted to queue **22**. In other embodiments, it contains data that contributed to confidence determination. Location Reference Info field **1100f** may contain information (TDOA measurement and/or AOA measurement—see inserted field **1100f** for FIGS. **2D**, **2E** and **3C**) useful to locate a MS in the future when the WDR originated from the MS for its own whereabouts. Field **1100f** will contain selected triangulation measurements, wave spectrum used and/or particular communications interfaces **70**, signal strength(s), TDOA information, AOA information, or any other data useful for location determination. Field **1100f** can also contain reference whereabouts information (FIG. **3C**) to use relative a TDOA or AOA (otherwise WDR location field assumed as reference). In one embodiment, field **1100f** contains the number of DLMs and ILMs which contributed to calculating the MS location to break a tie between using WDRs with the same confidence values. In another embodiment, a tier of ILMs used to locate the MS is maintained so there is an accounting for the number of ILMs in the LN-expanse between the currently located MS and a DLM. In other embodiments, MS heading, yaw, pitch and roll, or accelerometer values are maintained therein, for example for antenna AOA positioning. When wave spectrum frequencies or other wave characteristics have changed in a transmission used for calculating a TDOA measurement, appropriate information may be carried along, for example to properly convert a time into a distance. Field **1100f** should be used to facilitate correct measurements and uses, if needed conversions have not already taken place.

Communications reference information field **1100g** is a multipart record describing the communications session, channel, and bind criteria between the MS and MSs, or service(s), that helped determine its location. In some embodiments, field **1100g** contains unique MS identifiers, protocol used, logon/access parameters, and useful statistics of the MSs which contributed to data of the location field **1100c**. An MS may use field **1100g** for WDRs originated from affirmifiers and pacifiers for subsequent LBX processing.

Speed field **1100h** contains a value for the MS speed when the WDR is originated by the MS for its own whereabouts. Speed field **1100h** may contain a value for speed of an affirmifier or pacifier when the WDR was originated elsewhere. Speed is maintained in any suitable units.

Heading field **1100i** contains a value for the MS heading when the WDR is originated by the MS for its own whereabouts. Heading field **1100i** may contain a value for heading of an affirmifier or pacifier when the WDR was originated elsewhere. Heading values are preferably maintained in degrees up to 360 from due North, but is maintained in any suitable directional form.

Elevation field **1100j** contains a value for the MS elevation (or altitude) when the WDR is originated by the MS for its own whereabouts. Elevation field **1100j** may contain a value for elevation (altitude) of an affirmifier or pacifier when the WDR was originated elsewhere. Elevation (or altitude) is maintained in any suitable units.

Application fields **1100k** contains one or more fields for describing application(s) at the time of completing, or originating, the WDR **1100**. Application fields **1100k** may include field(s) for:

- a) MS Application(s) in use at time;
- b) MS Application(s) context(s) in use at time;
- c) MS Application(s) data for state information of MS Application(s) in use at time;
- d) MS Application which caused WDR **1100**;
- e) MS Application context which caused WDR **1100**;
- f) MS Application data for state information of MS Application which caused WDR **1100**;
- g) Application(s) in use at time of remote MS(s) involved with WDR;
- h) Application(s) context(s) in use at time of remote MS(s) involved with WDR;
- i) MS Application(s) data for state information of remote MS(s) involved with WDR;
- j) Remote MS(s) criteria which caused WDR **1100**;
- k) Remote MS(s) context criteria which caused WDR **1100**;
- l) Remote MS(s) data criteria which caused WDR **1100**;
- m) Application(s) in use at time of service(s) involved with WDR;
- n) Application(s) context(s) in use at time of service(s) involved with WDR;
- o) MS Application(s) data for state information of service(s) involved with WDR;
- p) Service(s) criteria which caused WDR **1100**;
- q) Service(s) context criteria which caused WDR **1100**;
- r) Service(s) data criteria which caused WDR **1100**;
- s) MS navigation APIs in use;
- t) Web site identifying information;
- u) Physical or logical address identifying information;
- v) Situational location information as described in U.S. Pat. Nos. 6,456,234; 6,731,238; 7,187,997 (Johnson);
- w) Transactions completed at a MS;
- x) User configurations made at a MS;
- y) Environmental conditions of a MS;
- z) Application(s) conditions of a MS;
- aa) Service(s) conditions of a MS;
- bb) Date/time stamps (like field **1100b**) with, or for, any item of a) through aa); and/or
- cc) Any combinations of a) through bb).

Correlation field **1100m** is optionally present in a WDR when the WDR is in a transmission between systems (e.g. wireless communications) such as in data **1302** or **1312**. Field **1100m** provides means for correlating a response to an earlier request, or to correlate a response to an earlier broadcast. Correlation field **1100m** contains a unique handle. In a LN-Expanse which globally uses NTP, there is no need for correlation in data **1302** or **1312**. Correlation field **1100m** may be present in WDRs of queues **24** or **26**. Alternatively, a MS ID is used for correlation.

Sent date/time stamp field **1100n** is optionally present in a WDR when the WDR is in transmission between systems (e.g. wireless communications) such as in data **1302** or **1312**. Field **1100n** contains when the WDR was transmitted. A time zone, day/light savings time, and NTP indicator is preferably maintained as part of field **1100n**. Field **1100n** is preferably not present in WDRs of queue **22** (but can be if TDOA measurement calculation is delayed to a later time). In some embodiments, there is no need for field **1100n**. Whereabouts determined for MSs of an LN-Expanse may be reasonably timely, facilitating simplicity of setting outbound field **1100b** to the transmission date/time stamp at the sending data processing system, rather than when the WDR was originally

completed for whereabouts (e.g. when substantially the same time anyway). Sent date/time field **1100n** may be present in WDRs of queues **24** or **26**.

Received date/time stamp field **1100p** is preferably present in a WDR when inserted to queue **26** by receiving thread(s) upon received data **1302** or **1312**. Field **1100p** contains when the WDR was received by the MS. A time zone, day/light savings time, and NTP indicator is preferably maintained as part of field **1100p**. Field **1100p** is preferably not present in WDRs of queue **22** (but can be if TDOA measurement calculation is delayed to a later time). In some embodiments, there is no need for field **1100p**. For example, thread(s) **1912** may be listening directly on applicable channel(s) and can determine when the data is received. In another embodiment, thread(s) **1912** process fast enough to determine the date/time stamp of when data **1302** or **1312** is received since minimal time has elapsed between receiving the signal and determining when received. In fact, known processing duration between when received and when determined to be received can be used to correctly alter a received date/time stamp. Received date/time stamp field **1100p** is preferably added to records placed to queue **26** by receiving thread(s) feeding queue **26**.

Any fields of WDR **1100** which contain an unpredictable number of subordinate fields of data preferably use a tagged data scheme, for example an X.409 encoding for a Token, Length, and Value (called a TLV encoding). Therefore, a WDR **1100**, or field therein, can be a variable sized record. For example, Location Reference info field **1100f** may contain TTA, 8, 0.1456 where the Token="TTA" for Time Till Arrival (TDOA measurement between when sent and when received), Length=8 for 8 bytes to follow, and Value=0.1456 in time units contained within the 8 bytes; also SS, 4, 50 where Token="Signal Strength", 4=4 for 4 bytes to follow, and Value=50 dBu for the signal strength measurement. This allows on-the-fly parsing of unpredictable, but interpretable, multipart fields. The TLV encoding also enables-on-the-fly configuration for parsing new subordinate fields to any WDR **1100** field in a generic implementation, for example in providing parse rules to a Lex and Yacc implementation, or providing parse rules to a generic top down recursive TLV encoding parser and processor.

Any field of WDR **1100** may be converted: a) prior to insertion to queue **22**; or b) after access to queue **22**; or c) by queue **22** interface processing; for standardized processing. Any field of WDR **1100** may be converted when sending/receiving/broadcasting, or related processing, to ensure a standard format. Other embodiments will store and access values of WDR **1100** field(s) which are already in a standardized format. WDR **1100** fields can be in any order, and a different order when comparing what is in data transmitted versus data maintained to queue **22**.

An alternate embodiment to WDRs maintained to queue **22** preserves transport fields **1100m**, **1100n** and/or **1100p**, for example for use on queue **22**. This would enable **1952** thread(s) to perform TDOA measurements that are otherwise calculated in advance and kept in field **1100f**. However, queue **22** size should be minimized and the preferred embodiment uses transport fields when appropriate to avoid carrying them along to other processing.

FIGS. **11B**, **11C** and **11D** depict an illustration for describing various embodiments for determining the whereabouts of an MS, for example an ILM **1000e**. With reference now to FIG. **11B**, a MS **1000e** location is located by using locations of three (3) other MSs: MS₄, MS₅, and MS₆ (referred to generally as MS_j). MS_j are preferably located with a reasonably high level of confidence. In some embodiments, MS_j are

all DLMs. In some embodiments, MS_j are all ILMs. In some embodiments, MS_j are mixed DLMs and ILMs. Any of the MSs may be mobile during locating of MS 1000e. Wave spectrums in use, rates of data communications and MS processing speed, along with timeliness of processing described below, provide timely calculations for providing whereabouts of ILM 1000e with a high level of confidence. The most confident MSs (MS_j) were used to determine the MS 1000e whereabouts. For example, MS_j were all located using a form of GPS, which in turn was used to triangulate the whereabouts of MS 1000e. In another example, MS₄ was located by a form of triangulation technology, MS₅ was located by a form of "coming into range" technology, and MS₆ was located by either of the previous two, or some other location technology. It is not important how an MS is located. It is important that each MS know its own whereabouts and maintain a reasonable confidence to it, so that other MSs seeking to be located can be located relative highest confidence locations available. The WDR queue 22 should always contain at least one entry indicating the location of the MS 2 which owns WDR queue 22. If there are no entries contained on WDR queue 22, the MS 2 does not know its own location.

With reference now to FIG. 11C, a triangulation of MS 1000e at location 1102 is explained using location (whereabouts) 1106 of MS₄, location (whereabouts) 1110 of MS₅, and location (whereabouts) 1114 of MS₆. Signal transmission distance from MS_j locations are represented by the radiuses, with r₁ the TDOA measurement (time difference between when sent and when received) between MS₄ and MS 1000e, with r₂ the TDOA measurement (time difference between when sent and when received) between MS₅ and MS 1000e, with r₃ the TDOA measurement (time difference between when sent and when received) between MS₆ and MS 1000e. In this example, the known locations of MS_j which are used to determine the location of MS 1000e allow triangulating the MS 1000e whereabouts using the TDOA measurements. In fact, less triangular data in the illustration can be necessary for determining a highly confident whereabouts of MS 1000e.

With reference now to FIG. 11D, a triangulation of MS 1000e at location 1102 is explained using location (whereabouts) 1106 of MS₄, location (whereabouts) 1110 of MS₅, and location (whereabouts) 1114 of MS₆. In some embodiments, AOA measurements taken at a positioned antenna of MS 1000e at location 1102 are used relative the whereabouts 1106, whereabouts 1110, whereabouts 1114 (AOA 1140, AOA 1144 and AOA 1142), wherein AOA measurements are detected for incoming signals during known values for MS heading 1138 with MS yaw, pitch, and roll (or accelerometer readings). AOA triangulation is well known in the art. Line segment 1132 represents the direction of signal arrival to the antenna at whereabouts 1102 from MS₄ at whereabouts 1106. Line segment 1134 represents the direction of signal arrival to the antenna at whereabouts 1102 from MS₅ at whereabouts 1110. Line segment 1136 represents the direction of signal arrival to the antenna at whereabouts 1102 from MS₆ at whereabouts 1114. In this example, the known locations of MS_j which are used to determine the location of MS 1000e allow triangulating the MS 1000e whereabouts using the AOA measurements. In fact, less triangular data in the illustration can be necessary for determining a highly confident whereabouts of MS 1000e. Alternative embodiments will use AOA measurements of outbound signals from the MS at whereabouts 1102 detected at antennas of whereabouts 1106 and/or 1110 and/or 1114.

Missing Part Triangulation (MPT)

FIGS. 11C and 11D illustrations can be used in a complementary manner when only one or two TDOA measurements

are available and/or not all stationary locations, or MS reference locations, are known at the time of calculation. Another example is when only one or two AOA angles is available and/or not all stationary locations, or MS reference locations, are known at the time of calculation. However, using what is available from each technology in conjunction with each other allows solving the MS whereabouts (e.g. blocks 952/954 processing above). MPT is one example of solving for missing parts using more than one location technology. Condition of data known for locating a MS (e.g. whereabouts 1106, 1110 and 1114) may be the following:

- 1) AAS=two angles and a side;
- 2) ASA=two angles and a common side;
- 3) SAS=two sides and the included angle; or
- 4) SSA=two sides and a non-included angle.

TDOA measurements are distances (e.g. time difference between when sent and when received), and AOA measurements are angles. Each of the four conditions are recognized (e.g. block 952 above), and data is passed for each of the four conditions for processing (e.g. block 954 above). For MS (#1) and ASA (#2), processing (e.g. block 954) finds the third angle by subtracting the sum of the two known angles from 180 degrees (i.e. using mathematical law that triangles' interior angles add up to 180 degrees), and uses the mathematical law of Sines (i.e. $a/\sin A=b/\sin B=c/\sin C$) twice to find the second and third sides after plugging in the knowns and solving for the unknowns. For SAS (#3), processing (e.g. block 954) uses the mathematical law of Cosines (i.e. $a^2=b^2+c^2-2bc \cos A$) to find the third side, and uses the mathematical law of Sines ($\sin A/a=\sin B/b=\sin C/c$ (derived from law of Sines above)) to find the second angle. For SSA (#4), processing (e.g. block 954) uses the mathematical law of Sines (i.e. $\sin A/a=\sin B/b=\sin C/c$) twice to get the second angle, and mathematical law of Sines ($a/\sin A=b/\sin B=c/\sin C$) to get the third side. Those skilled in the art recognize other useful trigonometric functions and formulas, and similar uses of the same trigonometric functions, for MPT depending on what data is known. The data discovered and processed depends on an embodiment, what reference locations are available, and which parts are missing for MPT. MPT uses different distances (time used to determine length in TDOA) and/or angles (from AOA or TDOA technologies) for deducing a MS location confidently (e.g. MPT). Even a single AOA measurement from a known reference location (stationary or MS) with a single TDOA measurement relative that reference location can be used to confidently locate a MS, and triangulation measurements used to deduce a MS location need not be from the same location technologies or wave spectrums. Those skilled in the art recognize that having known reference locations facilitates requiring less triangular information for deducing a MS location confidently. MPT embodiments may exist for any aforementioned wave spectrums.

FIG. 11E depicts an illustration for describing various embodiments for automatically determining the location of an MS. An MS can be located relative other MSs which were located using any of a variety of location technologies, for example any of those of FIG. 9A. An MS is heterogeneously located when one of the following conditions are met:

- More than one location technology is used during travel of the MS;
- More than one location technology is used to determine a single whereabouts of the MS;
- MPT is used to locate the MS; and/or
- ADLT is used to locate the MS.

The WDR queue 22 and interactions between MSs as described below cause the MS to be heterogeneously located without special consideration to any particular location tech-