UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/014,509	05/15/2020	9445251	2525.995REX0	1477

172615 7590 05/18/2021
Fabricant LLP
411 Theodore Fremd Road
Suite 206 South
Rye, NY 10580

EXAMINER	
HOTALING, JOHN M	

ART UNIT	PAPER NUMBER
3992	

MAIL DATE	DELIVERY MODE
05/18/2021	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

FABRICANT LLP

230 PARK AVENUE. 3FL W.

NEWYORK, NY 10169

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. *90/014,509* .

PATENT UNDER REEXAMINATION *9445251* .

ART UNIT *3992* .

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

PTOL-465 (Rev.07-04)

Notice of Intent to Issue Ex Parte Reexamination Certificate	**Control No.** 90/014,509	**Patent Under Reexamination** 9445251	
	Examiner JOHN M HOTALING II	**Art Unit** 3992	**AIA Status** No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

1. ☑ Prosecution on the merits is (or remains) closed in this *ex parte* reexamination proceeding. This proceeding is subject to reopening at the initiative of the Office or upon petition. *Cf.* 37 CFR 1.313(a). A Certificate will be issued in view of
   (a) ☑ Patent owner's communication(s) filed: 24 March 2021.
   (b) ☐ Patent owner's failure to file an appropriate timely response to the Office action mailed: _____.
   (c) ☐ Patent owner's failure to timely file an Appeal Brief (37 CFR 41.31).
   (d) ☐ The decision on appeal by the ☐ Board of Patent Appeals and Interferences ☐ Court dated _____
   (e) ☐ Other: _____.
2. The Reexamination Certificate will indicate the following:
   (a) Change in the Specification: ☐ Yes ☑ No
   (b) Change in the Drawing(s): ☐ Yes ☑ No
   (c) Status of the Claim(s):
      (1) Patent claim(s) confirmed: 1-35.
      (2) Patent claim(s) amended (including dependent on amended claim(s)): _____
      (3) Patent claim(s) canceled: _____.
      (4) Newly presented claim(s) patentable: _____.
      (5) Newly presented canceled claims: _____.
      (6) Patent claim(s) ☐ previously ☐ currently disclaimed: _____
      (7) Patent claim(s) not subject to reexamination: _____.
3. ☑ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on 24 March 2021.
4. ☑ Note the attached statement of reasons for patentability and/or confirmation. Any comments considered necessary by patent owner regarding reasons for patentability and/or confirmation must be submitted promptly to avoid processing delays. Such submission(s) should be labeled: "Comments On Statement of Reasons for Patentability and/or Confirmation."
5. ☐ Note attached NOTICE OF REFERENCES CITED (PTO-892).
6. ☐ Note attached LIST OF REFERENCES CITED (PTO/SB/08 or PTO/SB/08 substitute).
7. ☐ The drawing correction request filed on _____ is: ☐approved ☐disapproved.
8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐None of the certified copies have
   ☐been received.
   ☐not been received.
   ☐been filed in Application No. _____.
   ☐been filed in reexamination Control No. _____.
   ☐been received by the International Bureau in PCT Application No. _____.

   * Certified copies not received: _____.
9. ☐ Note attached Examiner's Amendment.
10. ☐ Note attached Interview Summary (PTO-474).
11. ☐ Other: _____.

All correspondence relating to this reexamination proceeding should be directed to the **Central Reexamination Unit** at the mail, FAX, or hand-carry addresses given at the end of this Office action.

/JOHN M HOTALING/
Primary Examiner, Art Unit 3992

cc: Requester (if third party requester)

NOTICE OF INTENT TO ISSUE EX PARTE REEXAMINATION CERTIFICATE

This Notice of Intent to Issue an Ex Parte Reexamination Certificate ("NIRC") is for claims 1-35 of U.S. Pat. No. 9,445,251 to Beyer, Jr. et al. ("the '251 Patent"), which were subject to *ex parte* reexamination.

This NIRC is in response to Patent Owner's ("PO's") Remarks, Declarations and Exhibits filed March 24, 2021.

Claims 1-34 are confirmed.

Prior Art References Cited By Requestor

1. U.S. Patent No. 7,353,034 to Haney ("Haney");
2. U.S. Patent No. 6,868,333 to Melen ("Melen");
3. U.S. Patent No. 6,662,016 to Buckham ("Buckham");
4. U.S. Patent No. 6,040,824 to Maekawa ("Maekawa");
5. U.S. Patent No. 7,630,724 to Beyer, Jr. et al. ("the '724 Patent")

Response to Arguments

PO's arguments filed March 24, 2021 have been fully considered. The following are PO's main arguments:

1) The '724 and '728 Patents are incorporated by reference in the '410 Application and the priority chain remains unbroken from April 17, 2006 to the filing of the '251 Patent (Remarks at 17);

2) Each application in the priority chain of the ’251 Patent including the ’410 Application, provides sufficient written description support for the claimed inventions of the '251 Patent (Remarks at 23); and

3) Haney is not prior art due to the Inventor’s prior invention (Remarks at 65).

<u>The ’724 and ’728 Patents Are Incorporated By Reference In The ’410 Application and The Priority Chain Remains Unbroken from April 17, 2006 to the filing of the '251 Patent (Remarks at 17)</u>

PO argues, “[t]he ’410 application (and every other application in the priority chain) contains an incorporation by reference statement clearly identifying both the ’728 patent and ’724 patent.” Terveen Dec. at ¶20, “In my opinion, the ’410 application incorporates both the ’724 patent, and the ’728 patent (attached as Exhibits C and D, respectively) because the ’410 application states: “The method and operation of communication devices used herein are described in U.S. Pat. No. 7,031,728 which is hereby incorporated by reference and U.S. Pat, No. 7,630,724.” A skilled artisan would understand that there is no ambiguity regarding this incorporation statement. I have reviewed the MPEP which provides that to show clear intent to incorporate by reference both patents, the sentence need only (1) identify the roots words “incorporate” and “reference” and (2) identify the patents. 35 U.S.C, 1.57(c). The sentence incorporating by reference the ”724 patent and ’728 patent includes the required root words and identifies both patents. I have reviewed the ’410 application and there is no language to negate the express identification of the ’724 patent and the ’728 patent in the incorporation-by-reference statement. Given the guidance in the MPEP and the relevant code, a skilled artisan would have understood the incorporation by reference statement to incorporate both the ”724 Patent and the ”728 Patent into the ’410 application.”

Examiner respectfully disagrees. Incorporation by Reference 37 CFR 1.57 (c) says, “Except as provided in paragraph (a) or (b) of this section, an incorporation by reference must be set forth in the specification and must: (1) Express a clear intent to incorporate by reference by using the root words "incorporat(e)" and "reference" (e.g., "incorporate by reference"); and (2) Clearly identify the referenced patent, application, or publication.”

While PO did express a clear intent to incorporate by reference by using the root words "incorporat(e)" and "reference" (e.g., "incorporate by reference"), PO only clearly identified the ‘728 Patent and did not clearly identify the ‘724 Patent as the referenced patent. The ‘410 Application specifically says at 1:60-63, “[t]he method and operation of communication devices used herein are described in U.S. Pat. No. 7,031,728 which is hereby incorporated by reference and U.S. Pat. No. 7,630,724.” The sentence clearly identifies the ‘728 Patent as the patent to be incorporated by reference as the intent to incorporate by reference language immediately follows the identification of the ‘728 Patent and uses the verb “is” to refer back to the <u>single</u> referenced patent[1]. In contrast,

---

[1] Last antecedent rule:
a doctrine of interpretation (construction) of statutes that any qualifying words or phrases refer to the language immediately preceding the qualifier, unless common sense shows that it was meant to apply to something more distant or less obvious. Example: "The commercial vehicular license shall not apply to boats, tractors, and trucks, with only four wheels and under three tons..." then the qualifier "only four wheels and under three tons" applies only to trucks and not boats or tractors.

The People's Law Dictionary. (1981-2005). Gerald N. Hill and Kathleen T. Hill. Retrieved April 21 2021 from https://legal-dictionary.thefreedictionary.com/last+antecedent+rule no intent to incorporate by reference language follows the ‘724 Patent, thereby raising ambiguity as to whether the ‘724 Patent is incorporated by reference.

On page 16 of the PTAB’s Decision to Institute entered November 7, 2018 in IPR2018-00819, the PTAB indicated:

““[A] patent’s claims are not entitled to an earlier priority date merely because the patentee claims priority.” *In re NTP, Inc.*, 654 F.3d 1268, 1276 (Fed. Cir. 2011). “Rather, for a patent’s claims to be entitled to an earlier priority date, the patentee must demonstrate that the claims meet the requirements of 35 U.S.C. § 120.” *Id.* Moreover, to incorporate by reference in order to satisfy this requirement, “the incorporating [document] must use language that is *express* and *clear*, so as to leave no ambiguity about the identity of the document being referenced, nor any reasonable doubt about the fact that the referenced document is being incorporated.” *Northrop Grumman Info. Tech., Inc. v. United States*, 535 F.3d 1339, 1344 (Fed. Cir. 2008).

We are not persuaded by Patent Owner that the ’410 application incorporates the ’724 patent by reference. We agree with Petitioner that a person of ordinary skill in the art would have understood that the phrase, “which is hereby incorporated by reference,” refers only to the immediately preceding ’728 patent and does not include the ’724 patent following it. Patent Owner is responsible for the use of this particular phrasing, and Patent Owner was in the best position to clarify any possible ambiguity. Given the standard that the ’410 application “must use language that is express and clear, so as to leave no ambiguity about the identity of the document being referenced, nor any reasonable doubt about the fact that the referenced document is being incorporated,” we are not persuaded that the ’410 application incorporates the ’724 patent by reference. *Northrop Grumman Info. Tech., Inc.*, 535 F.3d at 1344 (emphasis altered).”

Examiner agrees with the sentiments of the PTAB. Further, Examiner does not find persuasive the argument from the expert testimony that in the incorporation by reference statement, “is” is referring back to the act of incorporation. (Remarks at 21) The clause, “which is incorporated by reference,” is an **adjective clause used to describe the noun, the ‘728 Patent[2].** This issue is an English language issue, not a technical issue and PO’s argument is in opposition to English language grammar.

As such PO’s arguments are not found persuasive. Examiner finds the ‘728 Patent to be incorporated by reference and the ‘724 Patent to not be incorporated by reference.

<u>Each Application in the Priority Chain of the ’251 Patent Including the ’410 Application, Provides Sufficient Written Description Support for the Claimed Inventions of the '251 Patent (Remarks at 23)</u>

PO argues “support for the georeferenced map limitation is set forth (1) in the ’728 Patent, which discloses geo-referenced maps stored in databases, (2) the server-based implementations of these databases and maps as expanded by the ’410 application. See Terveen Dec. at ¶¶21-29.” (Remarks at 25)

[2] Adjective clause:
its role in a noun phrase, like that of adjectives, is as a modifier of the noun.
The Concise Oxford Dictionary of Linguistics (2 ed.). (2007). P. H. Matthews. Retrieved April 21, 2021 from https://www.oxfordreference.com/view/10.1093/acref/9780199202720.001.0001/acref-9780199202720-e-74

The Terveen Dec. points to the '728 Patent at 4:25-33, which says, "By using these soft switches, and hard switches that are part of the cellular phone, the operator can activate different maps, change map scales, select which fixed entities are desired to be displayed, display the information concerning the symbol the operator has touched, initiate phone voice calls, send messages (text, photographs and videos), enter symbols and information representative of other entities, view the locations and statuses of the other communications net participants…" (¶¶24-25) PO also pointed to the '410 application at ¶¶34, 40 and 46 to show support for georeferenced maps and to ¶¶9-13 to show support for a second server.

Examiner finds this argument persuasive in showing written description support for multiple georeferenced maps and a second server.

PO argues '728 Patent has support for "network corresponding to a group" limitation at 4:39-46. Examiner agrees.

Therefore, these arguments have been fully considered and are found persuasive. PO has shown that there is written description support to the '724 Patent and the '251 Patent does benefit from an effective filing date of April 17, 2006.

<u>Haney Is Not Prior Art Due to The Inventor's Prior Invention (Remarks at 65)</u>

PO provides 37 C.F.R. 1.131(a) Declarations from inventors Malcolm L. Beyer, Jr. and Christopher R. Rice to show prior invention of claims 1-35 of the '251 Patent. Specifically, the Declarations and corresponding Exhibits show conception of the invention prior to Haney's effective date of April 4, 2005 coupled with due diligence from prior to April 4, 2005 to reduction to practice in October 2005. (Remarks at 65-84)

Examiner finds the Declarations and corresponding Exhibits effective in swearing behind the Haney reference.

In summary, Examiner finds the '728 Patent to be incorporated by reference and the '724 Patent to not be incorporated by reference in the '410 application. PO has shown that the '251 Patent benefits from an effective filing date of April 17, 2006. PO has also sworn behind the Haney reference.

Statement of Reasons for Confirmation

Claims 1-35 are confirmed.

The following is an examiner's statement of reasons for patentability and/or confirmation of the claims found patentable in this reexamination proceeding:

PO's arguments as discussed above are persuasive in overcoming the previous prior art rejections. PO has shown that the '251 Patent benefits from an effective filing date of April 17, 2006. PO has also sworn behind the Haney reference. Therefore, the claims are allowable over the '724 Patent and the Haney reference because they are no longer considered prior art.

Any comments considered necessary by PATENT OWNER regarding the above statement must be submitted promptly to avoid processing delays. Such submission by the patent owner should be labeled: "Comments on Statement of Reasons for Patentability and/or Confirmation" and will be placed in the reexamination file.

Future Correspondence

If attempts to reach Examiner John Hotaling by telephone at 571-272-4437 are unsuccessful, the examiner's supervisor, Michael Fueling, can be reached on 571-272-1367.

All correspondence relating to this ex parte reexamination proceeding should be directed: By Mail to:

Mail Stop Ex Parte Reexam
Central Reexamination Unit
Commissioner of Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to:

(571) 273-9900
Central Reexamination Unit

By Hand:

Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Registered users of EFS-Web may alternatively submit such correspondence via the electronic filing system EFS-Web, at

https://efs.uspto.gov/efile/myportal/efs-registered

EFS-Web offers the benefit of quick submission to the particular area of the Office that needs to act on the correspondence. Also, EFS-Web submissions are "soft scanned" (i.e., electronically uploaded) directly into the official file for the reexamination proceeding, which offers parties the opportunity to review the content of their submissions after the "soft scanning" process is complete.

Any inquiry concerning this communication should be directed to the Central Reexamination Unit at (571) 272-7705.

Signed:

/JOHN M HOTALING/
Primary Examiner, Art Unit 3992

Conferees:

/CATHERINE M TARAE/
Primary Examiner, Art Unit 3992

/ANDREW J. FISCHER/
Supervisory Patent Reexamination Specialist, Art Unit 3992

CLAIMS

A listing of claims follows:

1. (Original) A computer-implemented method comprising: with a first device, receiving a message from a second device, wherein the message relates to joining a group; based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group; presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial coordinates; sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location; receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates; presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices; and identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on

the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first device does not have access to respective Internet Protocol addresses of the second devices.

2. (Original) The method of claim 1, wherein the data includes a short message service message, a text message, an image, or a video.

3. (Original) The method of claim 1, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC).

4. (Original) The method of claim 1, wherein the second map is a satellite image.

5. (Original) The method of claim 1, further comprising sending, by the first device, updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information comprising a previous location of the first device, displacement of the first device by a predetermined distance relative to a previous location of the first device, or both.

6. (Original) The method of claim 1, further comprising identifying second user interaction with the interactive display selecting at least one of the user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device.

7. (Original) The method of claim 1, wherein the message from the second device is a Short Message Service (SMS) message or a text message.

8. (Original) The method of claim 1, wherein participating in the group further includes sending first status information to the server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second location information comprising a plurality of battery levels of the respective plurality of second devices included in the group, a plurality of signal strengths of wireless signals of the respective plurality of second devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a combination thereof.

9. (Original) The method of claim 1, wherein the first device is a smart phone.

10. (Original) The method of claim 1, further comprising: with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality of second devices; and based on transmitting the group identifier associated with the second group, participating in the second group, wherein participating in the second group includes receiving third location information from the server, the third location information comprising a plurality of locations of the respective second plurality of second devices included in the second group.

11. (Original) The method of claim 1, wherein the data includes a voice recording.

12. (Original) The method of claim 1, further comprising: using a Global Positioning Satellite (GPS) receiver of the first device to obtain data indicative of the location of the first device, wherein sending the first location information to the server comprises using the Internet Protocol (IP) to send the first location information to the server.

13. (Original) The method of claim 1, further comprising identifying, by the first device, user interaction with the display selecting a particular user-selectable symbol positioned on the second georeferenced map and corresponding to a particular second device, wherein identifying the user interaction selecting the particular user-selectable symbol comprises: detecting user selection of a portion of the interactive display corresponding to a position on the second georeferenced map; based at least in part on coordinates of the selected position on the second georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the selected position on the second georeferenced map; and identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position.

14. (Original) The method of claim 13, wherein identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position comprises: searching a database of entities for an entity located nearest to the spatial coordinates represented by the selected position, wherein the entities represented by data in the database include the second devices, wherein the database data include locations of the respective entities, and wherein the database is searchable by location; and based on a result of searching the database, identifying the particular second device as the entity located nearest to the spatial coordinates represented by the selected position, wherein the particular user-selectable symbol corresponds to the particular second device.

15. (Original) The method of claim 14, wherein the entity is a first entity, and wherein the method further comprises performing by the first device: receiving user input via user interaction with the interactive display of the first device, the user input specifying a location and a symbol

corresponding to a second entity other than the first device and the second devices; and based on the user input, adding the user-specified symbol to the interactive display at a position on the second georeferenced map corresponding to the user-specified location of the second entity.

16. (Original) The method of claim 15, further comprising performing by the first device: transmitting the user-specified symbol and location of the second entity to the second devices for addition of the user-specified symbol to respective interactive displays of the second devices at respective positions on respective georeferenced maps corresponding to the user-specified location of the second entity.

17. (Original) The method of claim 16, wherein the user input further specifies information associated with the second entity, and wherein the method further comprises performing, by the first device: transmitting the user-specified information associated with the second entity to the second devices.

18. (Original) The method of claim 17, further comprising performing by the first device: adding data representing the spatial coordinates of the location of the second entity and data representing the information associated with the second entity to the database.

19. (Original) The method of claim 15, wherein the portion of the interactive display is a first portion, wherein the position of the symbol corresponding to the particular second device is a first position, and wherein receiving the user input specifying the location of the second entity comprises: detecting user selection of a second portion of the interactive display corresponding to a second position on the second georeferenced map; and based at least in part on coordinates of the second position on the second georeferenced map and on the data relating positions on the

second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the second position on the second georeferenced map, wherein the location represented by the second position is the location of the second entity.

20. (Original) The method of claim 14, wherein the database is stored on the first device.

21. (Original) The method of claim 14, wherein the database is stored on the server.

22. (Original) The method of claim 1, wherein the spatial coordinates comprise latitude and longitude coordinates.

23. (Original) The method of claim 1, further comprising performing, by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating voice-over-IP (VOIP) communication with the particular second device.

24. (Original) A system comprising: a first device programmed to perform operations comprising: receiving a message from a second device, wherein the message relates to joining a group; based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group; presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on

the first georeferenced map at respective positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial coordinates; sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location; receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates; presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices; and identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first device does not have access to respective Internet Protocol addresses of the second devices.

25. (Original) The system of claim 24, wherein the data includes a short message service message, a text message, an image, or a video.

26. (Original) The system of claim 24, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC).

27. (Original) The system of claim 24, wherein the second map is a satellite image.

28. (Original) The system of claim 24, wherein the operations further comprise sending updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information comprising a previous location of the first device, displacement of the first device by a predetermined distance relative to a previous location of the first device, or both.

29. (Original) The system of claim 24, wherein the operations further comprise identifying second user interaction with the interactive display selecting at least one of the user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device.

30. (Original) The system of claim 24, wherein the message from the second device is a Short Message Service (SMS) message or a text message.

31. (Original) The system of claim 24, wherein participating in the group further includes sending first status information to the server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second location information comprising a plurality of battery levels of the respective plurality of second devices included in the group, a plurality of signal strengths of wireless signals of the respective plurality of second devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a combination thereof.

32. (Original) The system of claim 24, wherein the first device is a smart phone.

33. (Original) The system of claim 24, wherein the operations further include: with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality of second devices; and based on transmitting the group identifier associated with the second group, participating in the second group, wherein participating in the second group includes receiving third location information from the server, the third location information comprising a plurality of locations of the respective second plurality of second devices included in the second group.

34. (Original) The system of claim 24, wherein the data includes a voice recording.

35. (Original) The system of claim 24, wherein the first device includes a Global Positioning Satellite (GPS) receiver, and wherein the operations further include: using the GPS receiver to obtain data indicative of the location of the first device, wherein sending the first location information to the server comprises using the Internet Protocol (IP) to send the first location information to the server.

REMARKS

This paper is in response to the Office Action mailed November 24, 2020. Claims 1-35 (the "Challenged Claims") are subject to reexamination and stand rejected. Patent Owner traverses for the reasons stated herein. No claims are amended, added, or canceled in this paper, and therefore, claims 1-35 are subject to reexamination.

Patent Owner's Summary of Examiner Interview

Patent Owner thanks the Examiners for the courtesy extended during the Examiner interview occurred February 18, 2021 between Examiners, John M. Hotaling II, Catherine Michelle Tarae and Alex J. Kosowski, and Patent Owner's representatives, Jialin Zhong (Reg. No. 62,937) and Vincent Rubino (Reg. No. 68,594).

During the interview, Patent Owner's representatives discussed the priority claims of the present patent to its parent patents (or parent applications) as those presented in the CROSS REFERENCE TO RELATED APPLICATIONS section of the present patent. Patent Owner's representatives also discussed with the Examiners other requirements for responding to the Office Action. No agreement was reached during the interview.

Rejection 1 under 35 U.S.C. § 103

The Office Action states at page 3 that claims 1-35 of U.S. Patent No. 9,445,251 (the "'251 Patent") are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 7,630,724 to Beyer, Jr. et al. (the "'724 Patent") in view of U.S. Patent No. 6,662,016 to Buckham ("Buckham"). For the reasons set forth below, Patent Owner respectfully traverses this rejection because the '724 patent is not prior art to the '251 patent. Because the Office adopted the positions set forth in the Request, this paper responds to the positions set forth in the Request by Requestor. Moreover, while the claims of the '251 patent have been addressed by several decisions denying institution *Inter Partes* Review proceedings, this is the first time that Patent Owner has presented argument including expert and factual testimony in support of the validity of the claims. This response is supported by the expert declaration of Dr. Loren Terveen, Ph.D. and the inventor declarations of Mr. Malcolm K. Beyer, Jr., and Mr. Christopher R. Rice.

The '724 Patent Is Not Prior Art to the '251 Patent because the '251 Patent's Claims are Entitled to Priority to the Filing of the '724 Patent.

It is undisputed that the '251 Patent includes a priority claim to at least April 17, 2006, the filing date of U.S. Application No. 11/308,648, which ultimately issued as U.S. Patent No. 7,630,724. Further, it is undisputed that each of the applications in the priority chain includes a priority claim to at least April 17, 2006 and identify U.S. Application No. 11/308,648. See Terveen Declaration at ¶ 18. Further, it is undisputed that, if the '251 Patent is entitled to priority to April 17, 2006, the rejection based on the '724 Patent is overcome. *Id.*

Patent Owner contends that the claimed inventions of the '251 patent are entitled to a priority date of April 17, 2006. On the face of the '251 patent, the '251 patent properly claims the benefit of priority to earlier applications as follows: the '251 patent is a continuation of U.S. patent application Ser. No. 14/529,978 filed on Oct. 31, 2014, **which is a continuation-in-part of U.S. patent application Ser. No. 14/027,410 (the "'410 application") filed on Sep. 16, 2013, now U.S. Pat. No. 8,880,042**, issued Nov. 4, 2014, which is a continuation of U.S. patent application Ser. No. 13/751,453 filed Jan. 28, 2013, now U.S. Pat. No. 8,538,393 issued Sep. 17, 2013, which is a continuation-in-part of U.S. patent application Ser. No. 12/761,533 filed on Apr. 16, 2010, now U.S. Pat. No. 8,364,129 issued Jan. 29, 2013, which is a continuation-in-part of U.S. patent application Ser. No. 11/615,472 filed on Dec. 22, 2006, now U.S. Pat. No.

8,126,441 issued on Feb. 28, 2012, which is a continuation-in-part of U.S. patent application Ser. No. 11/308,648 filed Apr. 17, 2006, now the ’724 patent issued on Dec. 8, 2009, which is a continuation-in-part of U.S. patent application Ser. No. 10/711,490, filed on Sep. 21, 2004, now U.S. Pat. No. 7,031,728 (the “’728 patent”) issued on Apr. 18, 2006. See Terveen Declaration at ¶ 18. The Requestor’s sole attack on the priority chain that was adopted by the Office concerned only the ’410 application. (bolded above). *Id.* at ¶ 19. The ’251 patent possesses the requisite continuity of disclosure through the entire chain of priority. *Id.* at ¶ 38.

The reexamination request raises two issues with regard to priority: (1) whether the specification of the ’724 Patent was properly incorporated by reference into each application in the priority chain leading up to the ’251 Patent, and (2) whether the inventions claimed in the ’251 Patent can trace priority through continuity of disclosure from the ’724 Patent filed on April 17, 2006 to the filing of the application that resulted in the ’251 Patent. Both of these issues should be resolved in favor of the Patent Owner as set forth below. Resolution of either of these issues in the Patent Owner’s favor is sufficient to traverse the rejection.

In support of the rejection based on the ’724 Patent, the Office relies on “the reasons discussed in the request.” (See Order Granting Request for *Ex Parte* Reexamination mailed July 10, 2020, page 2.) However, the request does not provide an accurate or complete analysis of the priority chain. See Terveen Declaration at ¶ 19. Because the Office has relied on the reasons in the Request, Patent Owner addresses the arguments set forth by the Requestor (referred to as “Requestor” or “Google”) below.

Patent Owner responds to Rejection 1 as follows:

I. Patent Owner explains the incorrect factual and legal bases set forth in the Request, which was adopted by the Office in Rejection 1.

II. Patent Owner demonstrates that the ’724 Patent is incorporated by reference into each application in the priority chain going back to April 17, 2006, the filing date of the ’724 Patent.

III. Patent Owner explains how the priority chain for the subject matter of Claims 1-35 of the ’251 Patent is unbroken even if the ’724 Patent were not incorporated by reference into any subsequent applications. While the specific language of the continuations in the chain of priority changed over time, there is no requirement of ***identity*** of disclosure, only ***continuity*** of disclosure and the disclosure was continuous. See Terveen Declaration at ¶ 20. The Rejection focuses on a specific application in the chain, the ’410 Application and whether the ’410 Application discloses (1) the second geo-referenced map, (2) joining and participating in a group, (3) the receiving a message limitations, and (4) the anonymous communication limitations of the ’251 Patent claims.[1] See Terveen Declaration at ¶ 19. However, the Rejection overlooks the significance of the disclosure of U.S. Patent No. 7,031,728, which is incorporated by reference into the ’410 Application. The Rejection also overlooks other disclosures of the ’410 Application that, when read in context, support claims 1-35 of the ’251 Patent. *Id.* at ¶ 20. When viewed in light of the complete record and further in view of the testimony of Patent Owner’s expert, the Office should reach the conclusion that

[1] Patent Owner refers to these limitations as characterized by Requestor but does not necessarily agree that these phrases properly characterize any claim limitations. Further, it appears that the Office did not adopt Requestor’s arguments regarding (3) and (4), however, for completeness Patent Owner addresses each of these limitations below.

the '410 Application fully supports these limitations and the full scope of the '251 Patent Claims. *Id.* at ¶ 20.

I. Mischaracterizations in the Request

As explained above, the Rejection is based primarily on arguments made in the Request. The Request incorrectly characterizes numerous facts. The mischaracterizations include mischaracterizations of the PTAB record, mischaracterizations of the record regarding the "second georeferenced map" limitation, and mischaracterizations of the record regarding the "joining and participating in a group" limitations.

First, regarding the PTAB, the Request focuses on two Inter Partes Review proceedings: IPR Nos. IPR2018-00817 and IPR2018-01081 which resulted in decisions denying institution. However, as a matter of law, these decisions have no preclusive effect on Patent Owner. Requestor's arguments to the contrary are incorrect. The analysis provided in the institution decisions was based on preliminary responses filed by AGIS without expert or fact declaration testimony. Additionally, the Requestor did not rely on the IPR decisions in their entirety, instead Requestor did not adopt the PTAB's positions that conflicted with its own, such as those with regard to the Haney reference. Request at 11 ("Requestor submits that…the PTAB misapprehended or overlooked critical passages from Haney that contradict the PTAB's argument.") In other words, Requestor's selective identification of favorable excerpts should be viewed with caution, and the Office should consider the entire record placed before it in this proceeding, including the evidence that was not before the PTAB in prior proceedings.

Second, regarding Requestor's arguments concerning whether the '410 Application discloses the "second geo-referenced map," limitation of Claims 1-35, Requestor incorrectly characterizes the '410 Application, stating that the "'410 application includes only one reference to a georeferenced map." Request at 28. This statement is incorrect because the '410 application includes two explicit references to the term "geo-referenced map." '410 Application at ¶¶ 40 and 46; Terveen Declaration at ¶¶ 21-22. Furthermore, it is undisputed that the '728 Patent is incorporated by reference into the disclosure of the '410 Application—the '728 patent includes numerous descriptions of geo-referencing and geo-referenced maps, which are set forth in greater detail in the sections below. See Terveen Declaration at ¶¶ 22-28.

Third, regarding the "joining and participating in a group" limitation, Requestor misrepresented the record regarding the "polling" disclosure. Requestor indicated that the "polling" disclosure supported the "group" limitation, however, Requestor incorrectly stated that the "polling" embodiment of the '724 disclosure was deleted in the '410 Application. Requestor stated that the following passage from the '724 Patent provides support for the alleged "full scope" of the group limitation:

> There are several ways that the network can be established including: A. POLLING—To initialize the communications net, the cellular phone "one" operator selects, from a list, the other users (or all of them), that the operator desires to be part of the communications net. The system then polls the selected phones to activate and become part of the communications net. The selected phones then transmit their GPS positions to all the other phones in the established net.

('724 patent, 14:60-67; '251 File History, 437.) This passage describes a device initializing a

Request at p. 39; Terveen Declaration at ¶ 30. Requestor then states:

> The '410 application lacks any such disclosure. Although the '410 patent also uses the word "polling" several times, it never uses "polling" to "initialize the communication net" or "establish" the network as described in the '724 patent. Most often, the '410 patent uses the term "polling" to poll existing members of the group to obtain their status and location, not to cause their devices to join the group without any further operator action as described in the '724 patent. ('410 application, ¶47 (manually polling the "devices that are used by all of the participants in the communication network" to obtain their location and status), ¶50 ("polling network participants"); Williams, ¶¶101-103.)

Request at p. 39 (emphasis added); Terveen Declaration at ¶ 30. Requestor failed to inform the Office that the '410 application contains ***nearly identical disclosure***. See Terveen Declaration at ¶ 30. The '728 Patent, which is incorporated by reference into the '410 application states:

> ments, vibrate, increase sound levels and other functions. To initialize the communications net, the cellular phone one operator selects, from a list, the other users (or all of them), that the operator desires to be part of the communications net. The system then polls the selected phones to activate and become part of the communications net. The selected phones then transmit their positions to all the other phones in the established net. Through interaction with one or more

'728 Patent at 4:39-46; Terveen Declaration at ¶ 30.

These three issues were misrepresented to the Office by Requestor and serve as the basis for the rejection issued by the Office. See Terveen Declaration at ¶¶ 30-Because the Rejection is based on these incorrect factual and legal underpinnings as set forth in the Request, Patent Owner respectfully requests that the Office withdraw its rejection. Patent Owner sets forth its complete bases for priority below.

II. The '724 and '728 Patents Are Incorporated By Reference In The '410 Application and The Priority Chain Remains Unbroken from April 17, 2006 to the filing of the '251 Patent

Requestor argued, and the Office adopted, the position that the '410 application does not incorporate by reference the '724 patent. This is incorrect and dispositive in favor of Patent Owner. The '410 application (and every other application in the priority chain) contains an incorporation by reference statement clearly identifying both the '728 patent and '724 patent. See Terveen Declaration at ¶ 20. Requestor cited to no authority where an incorporation by reference statement, such as that provided in the '410 application, would be insufficient. The correct standard in determining whether there is an incorporation by reference is whether one reasonably skilled in the art would have understood that the '410 application properly incorporates by reference the '724 patent. *Advanced Display Sys, Inc. v. Kent State Univ.*, 212 F.3d 1272, 1282 (Fed. Cir. 2000). 35 U.S.C. 1.57(c) requires that an incorporation by reference must be set forth in the specification and must (1) "[E]xpress a clear intent to incorporate by reference by using the root words 'incorporat(e)' and 'reference' (*e.g.*, "incorporate by reference");" and (2) "Clearly identify the referenced patent, application, or publication." In the '410 application, one reasonably skilled in the art would have understood that the incorporation statement in the '410 application properly incorporates by reference the '724 patent because the '410 application reads: "The method and operation of communication devices used herein are described in U.S. Pat. No. 7,031,728 which is hereby incorporated by reference and U.S. Pat. No. 7,630,724." The intent of this sentence is clear by use of the root words *incorporate by reference*. The sentence also clearly identifies two patents, the '728 and '724 patents. There is no language negating the clear intent to incorporate or the identification of the '728 and '724 patents. Accordingly, the '410 application properly incorporates the '728 and '724 patents. See Terveen Declaration at ¶ 20.

Furthermore, there is no ambiguity regarding this incorporation statement. See Terveen Declaration at ¶ 20. Requestor contends that the inclusion of the '724 Patent is merely an identification of that patent. Request at 27-28. However, the '724 patent was **already identified in the preceding paragraph**. Thus, under Google's interpretation, the passage would be rendered superfluous and meaningless. Such interpretation is untenable. Further attempts by Requestor to inject ambiguity into the incorporation-by-reference statement should be rejected. First, there is no requirement of any specific clausal order and/or sentence structure to incorporate by reference to a document. To show clear intent to incorporate by reference both patents, the sentence need only (1) identify the roots words "incorporate" and "reference" and (2) identify the patents. 35 U.S.C. 1.57(c). There is no dispute that the sentence includes the required root words and identifies *both* patents. Requestor has identified no language to negate the express identification of the '724 patent in the incorporation-by-reference statement. See Terveen Declaration at ¶ 20.

Because the standard is whether one reasonably skilled in the art would have understood that the '410 application properly incorporates by reference the '724 patent, expert testimony is highly pertinent to the issue of incorporation by reference. *Innovation Scis., LLC v. Amazon.com, Inc.*, No. 4:18-CV-474, 2020 WL 3453728, at *2 (E.D. Tex. June 24, 2020) (excusing typographic error in incorporation by reference statement) (*citing Advanced Display Sys,* 212 F.3d at 1282); *see also Apple, Inc. v. Samsung Elecs. Co.*, No. 5:12-cv-00630-LHK, 2014 WL 252045, at *23 (N.D. Cal. Jan. 21, 2014) ("expert testimony may in some circumstances help a court determine whether a host document incorporates another document by reference."). On this issue, three qualified experts, who are opining from the perspective of skilled artisans, have given sworn expert testimony that the '410 application incorporates by reference both the '724 and '728 patents.

Dr. Terveen testifies that an incorporation by reference statement is expressly recited in the '410 application, the statement uses the words 'incorporation by reference,' and that the same statement clearly identifies two patents: the '724 and '728 patents. Dr. Terveen testifies that there is no language or evidence to negate the incorporation by reference of both patents. Dr. Terveen testifies that a skilled artisan would have understood the incorporate by reference statement to incorporate both the ’724 patent and the ’728 patent into the ’410 application. See Terveen Declaration at ¶ 20.

Further, in the pending district court cases, two experts have given sworn testimony that a skilled artisan would have understood the incorporation statement of the ’410 application to incorporate both the ’724 patent and the ’728 patent. Ex. K, McAlexander Rebuttal Expert Report e at 45-49; Ex. L, Kia Rebuttal Expert Report at 39-44. Mr. McAlexander has opined that, from the perspective of skilled artisans, “the identity of the ’724 patent within this incorporation by reference statement is clear” and that “[t]his statement uses the words ‘incorporation by reference’ and it specifically identifies the ’724 patent.” *Id.* at 47. Dr. Kia has confirmed his opinion that the incorporation sentence incorporates by reference both the ’728 and ’724 patents. Ex. M, Kia Dep. 214:15-218:25, 220:14-223:7, Dec. 17, 2020. Mr. McAlexander further compared the similar contents of the ’724 patent and the ’728 patent after reviewing the incorporation statement and opined that “the similarity of disclosure in both field and content would have indicated to a person of ordinary skill in the art that the incorporate by reference statement was intended to include both the ’728 patent and ’724 patent.” Ex. K at47. Requestor’s misreading of the sentence’s use of the verb “is” should also be rejected. During their respective depositions in the district court, Dr. Kia and Mr. McAlexander were confronted with questions regarding the incorporation by reference sentence. Mr. McAlexander confirmed that he has experience testifying regarding incorporation statements. Ex. N, McAlexander Dep. 230:3-6 (Q: “Other than in this case, have you ever opined on whether a patent properly incorporates another patent by reference?” A: Yes, a number of times.). With respect to the grammar used by the patentee’s draftsperson, Mr. McAlexander confirmed that the language includes a singular verb associated with the act of incorporation for two documents. *Id.* at 240:18-241:9 (Q: You read the sentence -- phrase, "U.S. Patent 7,031,728 which is hereby incorporated by reference and U.S. Patent No. 7,630,724" as the "which is hereby" means both the '728 and '724? That's your testimony? A: I do and that's what I represented in my report. Q: You understand that "is" is a singular verb and not a plural verb; correct? A: It is a singular verb pointing back to the act of incorporation by reference. So, yes. And you can still have that singular verb to that act refer to multiple different – point to multiple different things and still be a singular verb because it refers to the act of incorporation.) Dr. Kia similarly confirmed that a skilled artisan would understand that the verb "is" refers to a single entity of inclusion for both the '724 and '728 patents. Ex. M, Kia Dep. 222:9-223:7.

Further, Mr. McAlexander confirmed that the absence of commas to separate the phrase "which is hereby incorporated by reference" from the patents identified in the incorporation statement means that the phrase "which is hereby incorporated by reference" does not refer only to the '728 patent. Ex. N, McAlexander Dep. 242:15-243:6 ("Would your answer change if there was a comma after 7,031,728 and after the word 'reference,' or would your opinion be the same if the phrase, 'which is incorporated by reference' was not separated by commas? A: If that phrase, 'which is hereby incorporated by reference' is [] set off by commas, my statement would be different. Q: Why is that? A: Because that [] phrase set off by commas would refer back to its predecessor, which is the '728, but it's not set off by commas. So grammatically [] I think it's inappropriate to make it as restrictive as what you're trying to propose.") Accordingly, both the '724 and '728 patents are incorporated by reference in the '410 application.

On the other hand, Requestor's reliance on certain PTAB denials of institution should not be credited any weight because AGIS's papers in those proceedings were merely preliminary.

Indeed, there is no caselaw to suggest that a PTAB denial of institution should be preclusive. In those pre-institution denials, Requestor filed petitions for inter partes review, and AGIS filed preliminary responses addressing select arguments limited to the issue of whether the petition should be instituted. 37 C.F.R. § 42.107(a) (In response to an inter partes petition, "[a] patent owner ***may file*** a preliminary response to the petition . . . ***limited to setting forth the reasons why no inter partes review should be instituted***.") (emphasis added.) AGIS did not present any argument or evidence in response to priority attacks because no review had been instituted and no response was due. In other words, procedurally, AGIS was not required respond to the merits of any issue (including priority attacks) raised in the petition until after institution (when the Board issues a scheduling order setting a deadline for full responses from patent owner). Because the IPR petitions were denied before AGIS had an opportunity to fully respond with evidence and because the PTAB's "findings" were based on a preliminary one-sided record, the PTAB institution denials carry no weight on the issue of priority. Accordingly, the Office should confirm that both the '724 and '728 patents are incorporated by reference in the '410 application and thus consider both the '724 and '728 patents as part of the original disclosure of the '410 application in its determination of whether the '410 application contains sufficient written description support for certain claim limitations raised by Requestor.

A finding that the '724 Patent was properly incorporated by reference should terminate the inquiry and result in confirmation of AGIS's claims. However, even setting aside whether the '724 Patent was incorporated by reference, the claims are still entitled to priority to the '724 application because the subject matter of the claims of the '251 Patent was continuous in each application in the priority chain including in the '410 application.

III. Each Application in the Priority Chain of the ’251 Patent, Including the ’410 Application, Provides Sufficient Written Description Support for the Claimed Inventions of the ’251 Patent

Contrary to Requestor’s arguments, which were adopted by the Office, the ’410 application provides sufficient written description support for 1) requesting and receiving a second georeferenced map from a server and displaying it with a second set of symbols, 2) the full scope of the features of joining and participating in a group, 3) joining and participating in the group based on receiving a message from a second device, and 4) anonymous communication. See Terveen Declaration at ¶¶ 20-38. Furthermore, as set forth in the conclusion of this response, each application in the priority chain of the ’251 Patent includes sufficient written description to support each of the claims of the ’251 Patent. These applications are:

U.S. Application No. 14/529,978 (“the ’978 Application) filed on Oct. 31, 2014 (including the incorporated by reference U.S. Patent No. 7,031,728).

U.S. Patent Application No. 14/027,410 (the “’410 Application”) filed on Sep. 16, 2013 (including the incorporated by reference U.S. Patent No. 7,031,728).

U.S. Patent Application No. 13/751,453 (the “’453 Application”) (including the incorporated by reference U.S. Patent No. 7,031,728).

U.S. Patent Application No. 12/761,533 filed on Apr. 16, 2010 (the “’533 Application”) (including the incorporated by reference U.S. Patent No. 7,031,728).

U.S. Patent Application No. 11/615,472 filed on Dec. 22, 2006 (the “’472 Application”) (including the incorporated by reference U.S. Patent No. 7,031,728).

U.S. Patent Application No. 11/308,648 filed Apr. 17, 2006 ("the '724 patent) (including the incorporated by reference U.S. Patent No. 7,031,728).

Because the Office Action adopted the Requestor's arguments, which relate only to the '410 Application, Patent Owner's arguments below focus primarily on the '410 Application.

Furthermore, during prosecution, the Patent Owner cited to portions of the '724 Patent specification for written description support. This support was exemplary and does not preclude Patent Owner from identifying support in the earlier '728 Patent or in the continuous disclosure of each continuation in the priority chain.

"To obtain the benefit of the filing date of a parent application, the claims of the later-filed application must be supported by the written description in the parent." *Anascape, Ltd. v. Nintendo of Am. Inc.*, 601 F.3d 1333, 1335 (Fed. Cir. 2010). The test for sufficiency of a patent's written description "is whether the disclosure of the application relied upon reasonably conveys to those skilled in the art that the inventor had possession of the claimed subject matter as of the filing date." *Centrak, Inc. v. Sonitor Techs., Inc.*, 915 F.3d 1360, 1365 (Fed. Cir. 2019) (quoting *Ariad Pharms., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2019)). The Federal Circuit has explained that "the test requires an objective inquiry into the four corners of the specification from the perspective of a person of ordinary skill in the art. Based on that inquiry, the specification must describe an invention understandable to that skilled artisan and show that the inventor actually invented the invention claimed." *Ariad*, 598 F.3d at 1351. The Federal Circuit has further explained that "determining whether a patent complies with the written description requirement will necessarily vary depending on the context. Specifically, the level of detail required to satisfy the written description requirement varies depending on the nature and scope of the claims and on the complexity and predictability of the relevant technology." *Id.* (citations omitted).

The '251 patent is entitled to a priority date of April 17, 2006. Because the '410 application reasonably conveys to those skilled in the art that the inventor had possession of the claimed subject matter of the '251 patent claims as of April 17, 2006, the '724 patent is not prior art to the '251 patent and the rejections should be withdrawn.

A. The '410 Application Provides Sufficient Written Description Support for the Georeferenced Map Limitations of the '251 Patent

The '410 application (and all other applications in the priority chain) includes sufficient written description to support the geo-referenced map limitation. The support for the georeferenced map limitation is set forth (1) in the '728 Patent, which discloses geo-referenced maps stored in databases, (2) the server-based implementations of these databases and maps as expanded by the '410 application. See Terveen Declaration at ¶¶ 21-28. The specification of each application in the priority chain thus provides sufficient written description support for what applicant characterizes as "requesting, retrieving, and using a second georeferenced map and georeferencing data," i.e. for the limitations sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location; receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates; presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices; and identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, as recited in the ’251 patent claims. *Id.* at ¶¶ 21-22.

As a preliminary matter, the terms Geo-Referenced Map and Geo-Referenced Map Data are not defined in the specification. As set forth in the declaration of Dr. Terveen, Geo-Referenced Maps mean that the internal coordinate system of a digital map or aerial photo has been related to a ground system of geographic coordinates.[2] See Terveen Declaration at ¶ 24.

i. Support for Geo-Referenced Maps in the ’728 Patent

The ’728 Patent, which is incorporated by reference, describes the interplay between maps, which must be geo-referenced, and databases on the device. See Terveen Declaration at ¶ 23. This evidence was not identified in the Request, the Office Action, or in any IPR proceeding.[3] In the first paragraph of the summary of the invention, the ’728 patent describes the software including “a map database and a database of geographically referenced fixed locations” and “another database with the constantly updated GPS location and status of all the

[2] This definition is consistent with the meaning of georeferenced map set forth by the U.S. Government, U.S.G.S. at https://www.usgs.gov/faqs/what-does-georeferenced-mean?qt-news_science_products=0#qt-news_science_products

[3] Requestor misled the Office when it states that the PTAB previously “held” that the ’410 application does not provide support for this limitation. The PTAB did not consider the disclosure from the perspective of the ’728 patent, which is incorporated by reference. Additionally, the PTAB’s analysis of the written description support was not intended to be a “holding” or “finding” for any issue other than institution (similar to the whether a request is ordered in reexamination proceedings). This reexamination proceeding is the first review in which AGIS has had a full and fair opportunity to fully respond to any priority attack on the ’251 patent and the first time that AGIS has provided expert testimony to the Office.

software equipped cellular phone/PDA/GPS systems that are part of the communications net." '410 appl. At '728 Pat. Col. 3 ll. 35-43; Terveen Declaration at ¶ 23. The '728 patent further describes accessing multiple geo-referenced maps at 4:25-33:

> remain. The operator can also use the phone's hardware pointing device (Navigation Pad) to control the soft switches. By using these soft switches, and hard switches that are part of the cellular phone, the operator can activate different maps, change map scales, select which fixed entities are desired to be displayed, display the information concerning the symbol the operator has touched, initiate phone voice calls, send messages (text, photographs and videos), enter symbols and information representative of other entities, view the locations and statuses of the other communications net participants, establish conference calls, pre-establish conference sub-nets that, when activated, cause all the phone numbers that are specified to be conferenced for voice, text and photograph and video communications, and transmit messages to remote phones which cause the remote phones to make calls, verbal announcements, vibrate, increase sound levels and other functions. To initialize the communications net, the cellular phone one operator selects, from a list, the other users (or all of them), that the operator desires to be part of the communications net. The system then polls the selected phones to activate

(emphasis added). Here, the specification describes using buttons on the device to activate different maps (which in the context of the '728 Patent are stored in local databases), those maps which can display the locations and statuses of other communications net participants. See Terveen Declaration at ¶ 23. As per this disclosure, the maps stored in the databases of the '728 Patent are georeferenced maps. *Id.*

The '728 Patent also describes the geo-referenced map interface: "the touch screen provided with the LCD display in the cellular phone includes x, y coordinates that are **correlated** with the map on the cellular phone display and the **geographic location** of the fixed sites and the cellular phones in the communications net. '728 patent at 4:57-64. (emphasis added). The '728

Patent further describes that the disclosed software on the device relates map locations to actual locations: "because each of the receiving telephone units has software that **automatically converts the received data to the correct map location**, the transmitted symbols appear at the correct location without operator intervention and their category information is available by touching the symbol on the display screen." '728 Patent at 5:3-8. One of skill in the art would have understood that the inventor was in possession of embodiments where the databases must store geo-referenced maps including geo-referenced map data, otherwise the software woud not be able to convert data to correct map locations. See Terveen Declaration at ¶ 24.

Further support for geo-referenced maps in the '728 Patent can be found in the object of the invention:

> And yet another object of this invention is to enable each participant in the communications net to be able to easily transmit entities of interest to the other participants of the net by touching the display at the entities location on the map and causing a symbol to be entered and then entering the entities category information.

'728 Patent at 6:41-46. According to this object of the invention, an operator could touch on a map in order to send location information to other participants. In order for this embodiment to work, the maps described ***must*** include georeferenced map data. See Terveen Declaration at ¶ 24. If the maps were not geo-referenced, it would not be possible to touch on a location on the map and convey a location to another user and a user would not be able to relate latitude and longitude positions to maps. *Id.* Further indication that all maps displayed by the invention are georeferenced is that the main map area display is referred to by the '728 Patent as the "main geographical screen." '728 Patent at 8:32.

Accordingly, one of skill in the art would understand that the types of maps described by the '728 patent, and incorporated into the '410 application must be geo-referenced. See Terveen Declaration at ¶ 24.

ii. Support for Geo-Referenced Maps in the Remainder of the '410 Disclosure

The '410 Application fully incorporates these disclosures and makes clear that the databases of the '728 patent, i.e. the maps database, which includes multiple maps, and the GPS position database would have been implemented at central servers to distribute maps and location information to each of the mobile devices. See Terveen Declaration at ¶¶ 25-28. For example, the ''410 Application states that the system includes a server "from which data can be requested by network participants (i.e. maps, **satellite images**, and the like)". '410 Application at 0013.

The '410 application makes more than one reference to a georeferenced map, and the maps of the '410 application are the same as the geo-referenced maps disclosed stored by the '728 Patent's databases, with the additional disclosure of those databases residing on servers. See Terveen Declaration at ¶¶ 25-26. The '410 application, similar to the incorporated '728 patent, describes that the "heart of the invention" as the "ACS application programs" which "provide for a **geographical map and georeferenced entities** that are shown as display portion 16b that includes as part of the display various areas of interest in the particular local map section." '410 application at para. 34 and Fig. 1; Terveen Declaration at ¶ 22. The '410 application describes that the "map display" is a "geo-referenced map display" which includes "symbols" which "represent[] an entity on the geo-referenced map display." '410 application at para. 40 and Fig. 1. Terveen Declaration at ¶ 22

Given this description and the additional disclosure of "geo-referenced symbols" representing entities on the "geo-referenced map display," a person of ordinary skill in the art would understand that any disclosure of a "map" or "map display" in the '410 application must be a "geo-referenced map," and must be interchangeable with the maps disclosed by the '728 Patent. See Terveen Declaration at ¶¶ 22. The '410 application proceeds to describe the map display as "a touch screen, provid[ing] x and y coordinates of the screen 16b to the CPU's software from a map in a geographical database" and "an algorithm that relates the x and y coordinates to latitude and longitude." '410 application at para. 36; Terveen Declaration at ¶ 24. The '410 application describes that each device "is identified on the map display of the other network participant's user's phone devices by a display symbol that is generated on each user phone display to indicate each user's own location and identity" and "each symbol is placed at the correct geographical location on the user display and is correlated with the map on the display and is transmitted and automatically displayed" on the other devices. '410 application at para. 39; Terveen Declaration at ¶ 22.

As another example of geo-referencing in the '410 application, the specification of the '410 application describes "tracks" that reflect geo-referenced propitiations overlaid on maps to relate positions on a globe to the graphical user interface. See Terveen Declaration at ¶ 22. The '410 application explains that the software can "track" entities by "hooking" them on the "geo-referenced map" and that such tracked entities are represented as "georeferenced symbols." '410 application at para. 40 and Fig. 1; Terveen Declaration at ¶ 22. As disclosed in the '410 application, the "tracked" entities are displayed on the geo-referenced map at their correct geo-referenced locations and the display is updated to present an updated map as the "tracked" entities changes their respective locations. '410 application at para. 10, 30, 40, and 53, and Fig.

1; Terveen Declaration at ¶ 22. The ’410 application provide sufficient written description support for updating the display to present an updated map automatically or upon manual request. See Terveen Declaration at ¶ 22. For example, a user can request an updated map containing another user’s updated location by sending a “polling command” other users for location updates which would result in presenting the updated map or phones can be set to report “automatically.” ’410 application at para. 47; Terveen Declaration at ¶ 22. In another example, a user can manually “hook” a symbol to start tracking the device corresponding to the symbol and the action of “hooking” that symbol is a manual request to obtain updated locations for that symbol in a displayed map, i.e. a second, updated map. ’410 application at para. 40; Terveen Declaration at ¶ 22. A person of ordinary skill in the art would have understood that this is one reason why the claimed invention uses a second map, i.e. an updated map. In the same disclosure, the ’410 application explains that an operator can perform “zoom in,” “zoom out,” and “pan” functions. ’410 application at para. 35 and Fig. 1; Terveen Declaration at ¶ 22. A person of ordinary skill in the art would have understood that the zoom in, zoom out, and pan features would result in the request and reception of a second map, i.e., an updated map based on the zoom or pan function. See Terveen Declaration at ¶ 22. This is another reason why the claimed invention uses a second map. *Id.* Finally, a third reason why the claimed invention uses a second map is that it provides for selecting different map types. *Id.* For example, the ’410 application explains that an operator can select from different “map types” on the map display. ’410 application at para. 50 and Fig. 1; Terveen Declaration at ¶ 22.

As Dr. Terveen opines, a person of ordinary skill in the art would have understood that the georeferenced map is the map display provided in Fig. 1 and that the data relating positions on the second georeferenced map to spatial coordinates can be any information related to the map display or the contents of the map display such as locations for symbols. See Terveen Declaration at ¶ 26. Further, a person of ordinary skill in the art would have also understood that there can be multiple servers for serving maps. *Id.* For example, the '410 application describes "servers" at multiple locations. '410 application at para. 8, 38; Terveen Declaration at ¶ 26. The '410 application describes that the ACS application programs and databases provides a communication network in conjunction with a remote server" and that the server "also fills another role of being a database from which data can be requested by network participants (i.e. maps, satellite images, and the like) or can be pushed to network participants." '410 application at para. 9-13; Terveen Declaration at ¶ 26. The '410 application describes that the server "acts as a forwarder of IP communications between any combination of cell phone/PDA users" and that the forwarded data can include "Network participant location, identity, and status messages…Network participant entered tracks" and data such as "free text, preformatted messages, photographs, video, email and URL data." '410 application at para. 30, 32; Terveen Declaration at ¶ 26. The '410 application describes software "capable of initiating a cellular phone call by touch only and initiating conference calls by touching the geo-referenced map symbols" and "by using a similar symbol touching technique, a cellular phone can send user selected messages to cause a remote cellular phone to display and optionally announce emergency and other messages and to optionally elicit a response from the remote cellular phone." '410 application at para. 46; Terveen Declaration at ¶ 26. Moreover, the '410 application describes that "[w]hen the PDA/PC user wants to address particular data (a text message, photograph, video clip, voice recording, white board, or chat), the user…touch[es] his or her symbol" and "[s]ince the Server knows the IP address of the name or symbol, the Server forwards the data appropriately to that network participant." '410 application at para. 54; Terveen Declaration at ¶ 26.

iii. The Request, which was Adopted by the Office, Conceded that the Specification Discloses Multiple Georeferenced Maps.

There should not be any dispute that the '410 application or '728 patent disclose ***multiple*** georeferenced maps as Requestor had conceded that point, and Requestor's arguments were incorporated by the Office Action. See Terveen Declaration at ¶ 27. It is undisputed that the specification of the '410 application includes a server that fills the role of a database "from which data can be requested by network participants (i.e. maps, **satellite images**, and the like)". '410 Application at 0013; Terveen Declaration at ¶ 27. In addressing dependent claim 4, Google's expert, Mr. Williams conceded:

> As explained for claim element 1[d], the map requested from the server corresponds to the claimed "*second georeferenced map*." The '724 patent thus discloses that the second georeferenced map can be a "chart, aerial photograph or satellite image."

> 236. Further, AGIS admitted during prosecution that this claim element is taught in the '724 patent at 18:57-19:7 (satellite and aerial imagery). ('251 File History, 505; '838 File History, 771.)

Williams Decl. at 235-236; Terveen Declaration at ¶ 27. Accordingly, the Requestor conceded that the disclosure of satellite imagery as served by a server can be a disclosure of the second georeferenced map. See Terveen Declaration at ¶ 27. This is yet another example of support for the second geo-referenced map limitation.

Thus, for at least these reasons, a person of ordinary skill in the art would have understood that the '410 application provides sufficient written description support what applicant characterizes as "requesting, retrieving, and using a second georeferenced map and georeferencing data," i.e. for the limitations sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location; receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates; presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices; and identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, as recited in the '251 patent claims. *Id.* at ¶ 28.

B. The '410 Application Provides Sufficient Written Description Support for the Group Limitations

The Office Action does not explain which portion of this limitation is allegedly missing from the priority documents. Patent Owner therefore responds to the arguments set forth by the Requestor, which were incorporated by the Office. The '410 application provides sufficient written description support for what Requestor characterizes as the "full scope of the term group" and "participating in a group based on receiving a message,"[4] i.e. for the limitations with a first device, receiving a message from a second device, wherein the message relates to joining a group; based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group, as recited in the '251 patent claims. See Terveen Declaration at ¶ 29.

Regarding the full scope of the group limitation, Requestor disregards the contents of the '728 patent, which it admits is incorporated by reference in the '410 application. *Id.* at ¶ 30. Specifically, Google admits that the following passage from the '724 Patent provides support for the alleged "full scope" of the group limitation:

[4] Patent Owner addresses both of these questions in this section.

> There are several ways that the network can be established including: A. POLLING—To initialize the communications net, the cellular phone "one" operator selects, from a list, the other users (or all of them), that the operator desires to be part of the communications net. The system then polls the selected phones to activate and become part of the communications net. The selected phones then transmit their GPS positions to all the other phones in the established net.
>
> ('724 patent, 14:60-67; '251 File History, 437.) This passage describes a device initializing a

Request at p. 39; Terveen Declaration at ¶ 30. Requestor then states:

> The '410 application lacks any such disclosure. Although the '410 patent also uses the word "polling" several times, it never uses "polling" to "initialize the communication net" or "establish" the network as described in the '724 patent. Most often, the '410 patent uses the term "polling" to poll existing members of the group to obtain their status and location, not to cause their devices to join the group without any further operator action as described in the '724 patent. ('410 application, ¶47 (manually polling the "devices that are used by all of the participants in the communication network" to obtain their location and status), ¶50 ("polling network participants"); Williams, ¶¶101-103.)

Request at p. 39; Terveen Declaration at ¶ 30. However, Requestor is incorrect because the '410 application contains ***nearly identical disclosure***. See Terveen Declaration at ¶ 30. The '728 Patent, which is admittedly incorporated by reference into the '410 application states:

> ments, vibrate, increase sound levels and other functions. To initialize the communications net, the cellular phone one operator selects, from a list, the other users (or all of them), that the operator desires to be part of the communications net. The system then polls the selected phones to activate and become part of the communications net. The selected phones then transmit their positions to all the other phones in the established net. Through interaction with one or more

728 Patent at 4:39-46; Terveen Declaration at ¶ 30. This identical disclosure is ignored by Requestor, and is sufficient to show possession of the full scope of the group limitation. See Terveen Declaration at ¶ 30.

Requestor incorrectly contends that the claimed term group must be supported for both "closed" and "open" groups. This is not the case. See Terveen Declaration at ¶ 31. During prosecution of the '838 patent, the patentee identified exemplary, non-limiting written description support for the purposes of assisting the examiner. Patentee did not use the terms "closed" or "open" to characterize the types of groups contemplated by the claimed invention and none of the terms in '251 patent or the '838 patent application are subject to 35 USC 112 6th paragraph. *Id.* A person of ordinary skill in the art would not have understood the specification to be limited to the types of groups specifically enumerated in the specification or that the types of groups enumerated in the specification to be characterized simply as "closed" or "open." *Id.* Rather, a person of ordinary skill in the art would have understood that groups could be any type of group within the scope of the claimed invention and the skilled artisan would not have understood that certain "closed" or "open" groups were being precluded in accordance with the non-infringement arguments advanced by Requestor in its district court case. *Id.*

Regardless of how Requestor arbitrarily chooses to characterize groups, the '410 application provides sufficient written description support for the purported characterizations of "open" and "closed" types of groups. *Id.* at ¶ 32. The '410 application describes groups in a manner covering both "open" and "closed" or invitation-only groups, because the specification describes all of "public and private emergency groups," "private networks," and "intended participants," demonstrating that there is sufficient written description for groups. '410 application at para. 6, 7, 9, 13, 15, 17, 51, 63, 64; Terveen Declaration at ¶ 32. The '410 application describes providing a communication network in conjunction with a remote Server that provides the ability to establish a network of devices so that the devices can either broadcast to a group or selectively transmit to each of the other whereby "each PDA/GPS phone starts by requesting access to the Server...and once granted, reports its GPS position and status; the Server then routes the data to all signed on network participants so that each of the devices exchange location, status, and other information." '410 application at para. 9; Terveen Declaration at ¶ 32. Certain networks or groups can be established by "event name and password." '410 application at para. 13; Terveen Declaration at ¶ 32. In other embodiments, the networks or groups could be formed when a user can "specify a group of the network participants by assigning their symbol or unit name to a list of network participants." '410 application at para. 63; Terveen Declaration at ¶ 32. Second, it is undisputed that the '728 patent is incorporated by reference in the '410 application as Requestor admits that the '728 patent is incorporated by reference through the statement "[t]he method and operation of communication devices used herein are described in U.S. Patent 7,031,728 which is hereby incorporated by reference..." See Terveen Declaration at ¶ 32. The text of the '410 application expressly incorporates the issued '728 patent. *Id.* The '728 patent describes various types of groups and networks, such as "nets" and "sub-nets" which are not limited to "open" or "closed" groups as Requestor contends. '728 patent at 1:5-15, 2:18-54, 3:24-43, 3:44-56, 3:62-5:8, 5:21-54, 5:55-6:4, 6:5-28, 6:29-33, 6:34-36, 6:37-40, 6:41-46, 6:47-51, 6:52-55, 6:56-60, 8:35-9:11, 9:12-35, 10:12-31, 10:40-61, 11:20-37, 11:43-55; Terveen Declaration at ¶ 32. For at least the reasons provided herein, the '410 application provides sufficient written description support for what Requestor characterizes as the full scope of the term group and "participating in a group based on receiving a message," i.e. for the limitations with a first device, receiving a message from a second device, wherein the message relates to joining a group; based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group, as recited in the ’251 patent claims. See Terveen Declaration at ¶ 33.

D. The ’410 Application Provides Sufficient Written Description Support for the Limitation “wherein the first device does not have access to respective Internet Protocol addresses of the second devices” As Recited in the ’251 Patent

The ’410 application provides sufficient written description support for what applicant characterizes as “anonymous communications between devices,” i.e. for the limitations identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server **wherein the first device does not have access to respective Internet Protocol addresses of the second devices**, as recited in the ’251 patent claims. See Terveen Declaration at ¶ 34.

The ’410 applications describes “users need to be able to rapidly coordinate their activities eliminating the need for pre-entry of data into a web or identifying others…” ’410 application at para. 6. In one example, the ’410 application describes that it permits communication between devices without devices knowing each others’ IP addresses and it does this by using the server as a “forwarder” of data addressed from one participant to one or more participants. ’410 application at para. 11; Terveen Declaration at ¶ 35. The ’410 application describes that the application, not the user, uses an IP address associated with a unit to send data. ’410 application at para. 44; Terveen Declaration at ¶ 35. The ’410 application describes “[s]ince the server has all the parties’ IP addresses, the server is able to pass location and status information automatically” even when “participants have not entered…or do not have the other network participants’ IP addresses.” ’410 application at para. 53, 14; Terveen Declaration at ¶ 35. The ’410 application describes that the application uses this connection to send data from the server to other participants. ’410 application at para. 44, 53, 54; Terveen Declaration at ¶ 35. The ’410 application describes an example in which the user touches another user’s symbol to send data and “[s]ince the Server knows the IP address of the name or symbol, the Server forwards the data appropriately to that network participant.” ’410 application at para. 54; Terveen Declaration at ¶ 36. A person of ordinary skill in the art would understand that the ’410 application describes sending data, such as IP communications, over Internet Protocol. ’410 application at para. 9, 10, 30, 33; Terveen Declaration at ¶ 36. The ’728 patent, which Requestor admits is incorporated by reference, also provides similar descriptions for this limitation. See Terveen Declaration at ¶ 36.

Accordingly, for at least the reasons provided herein, the ’410 application provides sufficient written description support for what applicant characterizes as “anonymous communications between devices,” i.e. for the limitations identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server wherein the first device does not have access to respective Internet Protocol addresses of the second devices, as recited in the '251 patent claims. See Terveen Declaration at ¶ 37.

IV. There Exists Continuity of Disclosure Each of the Claim Inventions Throughout the Priority Chain of the '251 Patent

As confirmed by AGIS's expert, Dr. Terveen, the '251 patent possesses the requisite continuity of disclosure through the entire chain of priority. See Terveen Declaration at ¶ 38.

In addition to the written description support identified above as to certain claimed features at issue (i.e., 1) requesting, retrieving, and using second georeferenced map and its georeferencing data; 2) the full scope of the "group" feature; 3) participating in the group "based on receiving the message from the second device;" and 4) anonymous communications), the following chart identifies paragraphs providing written description support for each limitation of the '251 patent claims. See Terveen Declaration at ¶ 38. Accordingly, Requestor's attack on the chain of priority is unsupportable and should be rejected by the Office. *Id.* at ¶ 39.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
1[P] A computer-implemented method comprising:	Abstract, Paras. 1-73, Claims 1-4, Figs. 1-10	Abstract, Paras. 1-73, Claims 1-4, Figs. 1-10	Abstract, Paras. 1-73, Claims 1-4, Figs. 1-10	Abstract, Specification Pages 1-30, Figs. 1-8	Abstract, Specification Pages 1-29, Claims 1-5, Figs. 1-2d	Abstract, Paras. 1-75, Claims 1-20, Figs. 1-6
[1A] with a first device, receiving a message from a second device, wherein the message relates to joining a group;	Abstract, Paras. 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 56, 58, 59, 60, 61, 62, 63, 65,	Abstract, Paras. 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 56, 58, 59, 60, 61, 62, 63, 65,	Abstract, Paras. 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 56, 58, 59, 60, 61, 62, 63, 65,	Abstract, 1:11-19, 1:20-22, 2:1-11, 2:12-2:13, 2:14-2:21, 4:5-20, 5:1-8, 5:17-21, 6:1-7:2, 7:3-9, 7:16-18, 8:17-20, 9:1-7, 9:8-9,	Abstract, 1:10-16, 1:18-2:4, 2:5-7, 2:8-11, 2:18-3:8, 3:9-15, 4:6-8, 4:9-13, 5:16-20, 6:1-7, 15:8-17, 16:12-17:2, 17:3-5, 17:6-	Abstract, 1, 5, 12, 23, 24, 25, 27, 28, 29, 35, 40, 42, 54, 58, 59, Claims 1-20, Figs. 1-6

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
	66, 73, Claims 1-4, Figs. 1-10	66, 73, Claims 1-4, Figs. 1-10	66, 73, Claims 1-4, Figs. 1-10	17:9-16, 17:17-:18:4, 19:13-20:2, 20:3-9, 20:10-19, 22:8-21, 23:3-5, 23:6-12, 23:13-15, 23:16-24:7, 24:8-16, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	16, 17:17-18:10, 18:11-12, 18:13-19:2, 19:3-14, 19:15-20:8, Claims 1-5, Figs. 1-2d	
[1B] based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group;	Abstract, Paras. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 53, 54, 56, 58, 59, 60, 61, 62, 63, 65, 66, 70, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 53, 54, 56, 58, 59, 60, 61, 62, 63, 65, 66, 70, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 53, 54, 56, 58, 59, 60, 61, 62, 63, 65, 66, 70, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, 1:11-19, 1:20-22, 2:1-11, 2:12-2:13, 2:14-2:21, 3:1-19, 4:2-4, 4:5-20, 5:1-8, 5:9-12, 5:13-16, 5:17-21, 6:1-7:2, 7:3-9, 7:16-18, 8:17-20, 9:1-7, 9:1-7, 9:8-9, 9:10-13, 9:14-11:4, 12:13-13:2, 14:10-21, 17:9-16, 17:17-:18:4, 19:13-20:2, 20:3-9, 20:10-19, 20:20-21:14, 21:15-22:3, 22:8-21, 23:3-5, 23:6-12, 23:13-15, 23:16-24:7, 24:8-16, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:10-16, 27:17-20,	Abstract, 1:10-16, 2:5-7, 2:18-3:8, 3:9-15, 3:16-19, 3:20-4:2, 4:3-5, 6:1-7, 6:8-11, 6:12-8:2, 8:9-9:10, 9:11-21, 11:8-18, 11:19-12:20, 13:3-16,14:7-14, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, Claims 1-5, Figs. 1-2d	Abstract, 1, 5, 21, 23, 24, 25, 27, 28, 30, 32, 40, 42, 48, 49, 50, 54, 62, 63, 66, 68, 73, 74, 75, Claims 1-20, Figs. 1-6

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
				Claims 1-6, Figs. 1-8		
[1C] presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial coordinates;	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, 4:5-20, 5:17-21, 7:3-9, 7:16-18, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9, 19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	Abstract, 2:5-7, 2:18-3:8, 4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	Abstract, 1, 5, 9, 10, 11, 13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75, Claims 1-20, Figs. 1-6
[1D] sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location;	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, 4:5-20, 5:17-21, 7:3-9, 7:16-18, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9,	Abstract, 2:5-7, 2:18-3:8, 4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-	Abstract, 1, 5, 9, 10, 11, 13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75,

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
				19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	Claims 1-20, Figs. 1-6
[1E] receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates;	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, 4:5-20, 5:17-21, 7:3-9, 7:16-18, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9, 19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	Abstract, 2:5-7, 2:18-3:8, 4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	Abstract, 1, 5, 9, 10, 11, 13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75, Claims 1-20, Figs. 1-6
[1F] presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, 4:5-20, 5:17-21, 7:3-9, 7:16-18, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9,	Abstract, 2:5-7, 2:18-3:8, 4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-	Abstract, 1, 5, 9, 10, 11, 13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75,

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices; and				19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	Claims 1-20, Figs. 1-6
[1G] and identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first device does not have access to respective Internet Protocol addresses of the second devices.	Abstract, Paras. 5, 9, 11, 15, 16, 17, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 50, 53, 54, 55, 56, 59, 60, 61, 63, 64, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 11, 15, 16, 17, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 50, 53, 54, 55, 56, 59, 60, 61, 63, 64, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 11, 15, 16, 17, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 50, 53, 54, 55, 56, 59, 60, 61, 63, 64, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, 1:12-19, 2:1-11, 2:12-13, 3:1-19, 4:5-20, 5:1-8, 5:9-12, 5:13-16, 5:17-21, 6:1-7:2, 7:3-9, 7:9-15, 7:16-18, 9:1-7, 9:10-13, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9, 19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	Abstract, 2:5-7, 2:18-3:8, 4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	Abstract, 1, 5, 9, 10, 11, 13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75, Claims 1-20, Figs. 1-6
2. The method of claim 1, wherein the data includes a short message service message, a text message, an image, or a video.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
3. The method of claim 1, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC).	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.
4. The method of claim 1, wherein the second map is a satellite image.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.
5. The method of claim 1, further comprising sending, by the first device, updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information comprising a previous location of the first device, displacement of the first device by a predetermined distance relative to a previous location of the first device, or both.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.
6. The method of claim 1, further comprising identifying second user interaction with	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
the interactive display selecting at least one of the user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device.						
7. The method of claim 1, wherein the message from the second device is a Short Message Service (SMS) message or a text message.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.
8. The method of claim 1, wherein participating in the group further includes sending first status information to the server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second location information comprising a plurality of battery levels of the respective plurality of second devices included in the group, a plurality of signal strengths of wireless signals of the respective plurality of second devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a combination thereof.						
9. The method of claim 1, wherein the first device is a smart phone.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.
10[P]. The method of claim 1, further comprising: with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
of second devices;						
[10A] and based on transmitting the group identifier associated with the second group, participating in the second group, wherein participating in the second group includes receiving third location information from the server, the third location information comprising a plurality of locations of the respective second plurality of second devices included in the second group.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.
11. The method of claim 1, wherein the data includes a voice recording.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.
12. The method of claim 1, further comprising: using a Global Positioning Satellite (GPS) receiver of the first device to obtain data indicative of the location of the first device, wherein sending the first location information to the server	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
comprises using the Internet Protocol (IP) to send the first location information to the server.						
13[P]. The method of claim 1, further comprising identifying, by the first device, user interaction with the display selecting a particular user-selectable symbol positioned on the second georeferenced map and corresponding to a particular second device, wherein identifying the user interaction selecting the particular user-selectable symbol comprises:	See 1P above.	See 1P above.	See 1P above.	See 1P above.	See 1P above.	See 1P above.
[13A] detecting user selection of a portion of the interactive display corresponding to a position on the second georeferenced map;	See 1A above.	See 1A above.	See 1A above.	See 1A above.	See 1A above.	See 1A above.
[13B] based at least in part on coordinates of the selected position on the second georeferenced map and on the	See 1B above.	See 1B above.	See 1B above.	See 1B above.	See 1B above.	See 1B above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the selected position on the second georeferenced map; and						
[13C] identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position.	See 1C above.	See 1C above.	See 1C above.	See 1C above.	See 1C above.	See 1C above.
14[P]. The method of claim 13, wherein identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position comprises: searching a database of entities for an entity located nearest to the spatial coordinates represented by the selected position, wherein the entities	See 1P-1G, 13 above.	See 1P-1G, 13 above.	See 1P-1G, 13 above.	See 1P-1G, 13 above.	See 1P-1G, 13 above.	See 1P-1G, 13 above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
represented by data in the database include the second devices, wherein the database data include locations of the respective entities, and wherein the database is searchable by location;						
[14A] and based on a result of searching the database, identifying the particular second device as the entity located nearest to the spatial coordinates represented by the selected position, wherein the particular user-selectable symbol corresponds to the particular second device.	See 1P-1G, 13 above.	See 1P-1G, 13 above.	See 1P-1G, 13 above.	See 1P-1G, 13 above.	See 1P-1G, 13 above.	See 1P-1G, 13 above.
15[P]. The method of claim 14, wherein the entity is a first entity, and wherein the method further comprises performing by the first device: receiving user input via user interaction with the interactive display of the first device, the user input specifying a location and a	See 1P-1G, 14 above.	See 1P-1G, 14 above.	See 1P-1G, 14 above.	See 1P-1G, 14 above.	See 1P-1G, 14 above.	See 1P-1G, 14 above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
symbol corresponding to a second entity other than the first device and the second devices;						
[15A] and based on the user input, adding the user-specified symbol to the interactive display at a position on the second georeferenced map corresponding to the user-specified location of the second entity.	See 1P-1G, 14 above.	See 1P-1G, 14 above.	See 1P-1G, 14 above.	See 1P-1G, 14 above.	See 1P-1G, 14 above.	See 1P-1G, 14 above.
16. The method of claim 15, further comprising performing by the first device: transmitting the user-specified symbol and location of the second entity to the second devices for addition of the user-specified symbol to respective interactive displays of the second devices at respective positions on respective georeferenced maps corresponding to the user-specified location of the second entity.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
17. The method of claim 16, wherein the user input further specifies information associated with the second entity, and wherein the method further comprises performing, by the first device: transmitting the user-specified information associated with the second entity to the second devices.	See 1P-1G, 16 above.	See 1P-1G, 16 above.	See 1P-1G, 16 above.	See 1P-1G, 16 above.	See 1P-1G, 16 above.	See 1P-1G, 16 above.
18. The method of claim 17, further comprising performing by the first device: adding data representing the spatial coordinates of the location of the second entity and data representing the information associated with the second entity to the database.	See 1P-1G, 17 above.	See 1P-1G, 17 above.	See 1P-1G, 17 above.	See 1P-1G, 17 above.	See 1P-1G, 17 above.	See 1P-1G, 17 above.
19[P]. The method of claim 15, wherein the portion of the interactive display is a first portion, wherein the position of the symbol corresponding to the particular second device is a first position,	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
and wherein receiving the user input specifying the location of the second entity comprises: detecting user selection of a second portion of the interactive display corresponding to a second position on the second georeferenced map;						
[19A] and based at least in part on coordinates of the second position on the second georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the second position on the second georeferenced map, wherein the location represented by the second position is the location of the second entity.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.	See 1P-1G, 15 above.
20. The method of claim 14, wherein the database is	See 14P-14A above.	See 14P-14A above.	See 14P-14A above.	See 14P-14A above.	See 14P-14A above.	See 14P-14A above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
stored on the first device.						
21. The method of claim 14, wherein the database is stored on the server.	See 14P-14A above.	See 14P-14A above.	See 14P-14A above.	See 14P-14A above.	See 14P-14A above.	See 14P-14A above.
22. The method of claim 1, wherein the spatial coordinates comprise latitude and longitude coordinates.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.
23. The method of claim 1, further comprising performing, by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating voice-over-IP (VOIP) communication with the particular second device.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.	See 1P-1G above.
24[P]. A system comprising:	Abstract, Paras. 1-73, Claims 1-4, Figs. 1-10	Abstract, Paras. 1-73, Claims 1-4, Figs. 1-10	Abstract, Paras. 1-73, Claims 1-4, Figs. 1-10	Abstract, Specification Pages 1-30, Figs. 1-8	Abstract, Specification Pages 1-29, Claims 1-5, Figs. 1-2d	Abstract, Paras. 1-75, Claims 1-20, Figs. 1-6
a first device programmed to perform	Abstract, Paras. 1-73,	Abstract, Paras. 1-73,	Abstract, Paras. 1-73,	Abstract, Specification	Abstract, Specification Pages 1-29,	Abstract, Paras. 1-75,

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
operations comprising:	Claims 1-4, Figs. 1-10	Claims 1-4, Figs. 1-10	Claims 1-4, Figs. 1-10	Pages 1-30, Figs. 1-8	Claims 1-5, Figs. 1-2d	Claims 1-20, Figs. 1-6
[24A] receiving a message from a second device, wherein the message relates to joining a group;	Abstract, Paras. 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 56, 58, 59, 60, 61, 62, 63, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 56, 58, 59, 60, 61, 62, 63, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 2, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 56, 58, 59, 60, 61, 62, 63, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, 1:11-19, 1:20-22, 2:1-11, 2:12-2:13, 2:14-2:21, 4:5-20, 5:1-8, 5:17-21, 6:1-7:2, 7:3-9, 7:16-18, 8:17-20, 9:1-7, 9:8-9, 17:9-16, 17:17-:18:4, 19:13-20:2, 20:3-9, 20:10-19, 22:8-21, 23:3-5, 23:6-12, 23:13-15, 23:16-24:7, 24:8-16, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	Abstract, 1:10-16, 1:18-2:4, 2:5-7, 2:8-11, 2:18-3:8, 3:9-15, 4:6-8, 4:9-13, 5:16-20, 6:1-7, 15:8-17, 16:12-17:2, 17:3-5, 17:6-16, 17:17-18:10, 18:11-12, 18:13-19:2, 19:3-14, 19:15-20:8, Claims 1-5, Figs. 1-2d	Abstract, 1, 5, 12, 23, 24, 25, 27, 28, 29, 35, 40, 42, 54, 58, 59, Claims 1-20, Figs. 1-6
[24B] based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the	Abstract, Paras. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 53, 54, 56, 58, 59, 60, 61, 62, 63, 65, 66, 70, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 53, 54, 56, 58, 59, 60, 61, 62, 63, 65, 66, 70, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 29, 30, 31, 32, 33, 42, 50, 51, 52, 53, 54, 56, 58, 59, 60, 61, 62, 63, 65, 66, 70, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, 1:11-19, 1:20-22, 2:1-11, 2:12-2:13, 2:14-2:21, 3:1-19, 4:2-4, 4:5-20, 5:1-8, 5:9-12, 5:13-16, 5:17-21, 6:1-7:2, 7:3-9, 7:16-18, 8:17-20, 9:1-7, 9:1-7, 9:8-9, 9:10-13, 9:14-11:4, 12:13-13:2, 14:10-21, 17:9-16, 17:17-:18:4, 19:13-20:2, 20:3-9, 20:10-19,	Abstract, 1:10-16, 2:5-7, 2:18-3:8, 3:9-15, 3:16-19, 3:20-4:2, 4:3-5, 6:1-7, 6:8-11, 6:12-8:2, 8:9-9:10, 9:11-21, 11:8-18, 11:19-12:20, 13:3-16,14:7-14, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, Claims 1-5, Figs. 1-2d	Abstract, 1, 5, 21, 23, 24, 25, 27, 28, 30, 32, 40, 42, 48, 49, 50, 54, 62, 63, 66, 68, 73, 74, 75, Claims 1-20, Figs. 1-6

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group;				20:20-21:14, 21:15-22:3, 22:8-21, 23:3-5, 23:6-12, 23:13-15, 23:16-24:7, 24:8-16, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:10-16, 27:17-20, Claims 1-6, Figs. 1-8		
[24C] presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial coordinates;	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, 4:5-20, 5:17-21, 7:3-9, 7:16-18, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9, 19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	Abstract, 2:5-7, 2:18-3:8, 4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	Abstract, 1, 5, 9, 10, 11, 13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75, Claims 1-20, Figs. 1-6
[24D] sending, from the first	Abstract, Paras. 5, 9,	Abstract, Paras. 5, 9,	Abstract, Paras. 5, 9,	Abstract, 4:5-20, 5:17-21,	Abstract, 2:5-7, 2:18-3:8,	Abstract, 1, 5, 9, 10, 11,

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location;	10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	7:3-9, 7:16-18, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9, 19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75, Claims 1-20, Figs. 1-6
[24E] receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates;	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	Abstract, 4:5-20, 5:17-21, 7:3-9, 7:16-18, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9, 19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	Abstract, 2:5-7, 2:18-3:8, 4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	Abstract, 1, 5, 9, 10, 11, 13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75, Claims 1-20, Figs. 1-6
[24F] presenting, via the	Abstract, Paras. 5, 9,	Abstract, Paras. 5, 9,	Abstract, Paras. 5, 9,	Abstract, 4:5-20, 5:17-21,	Abstract, 2:5-7, 2:18-3:8,	Abstract, 1, 5, 9, 10, 11,

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices;	10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	10, 13, 15, 16, 33, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 50, 53, 54, 63, 64, 65, 66, 73, Claims 1-4, Figs. 1-10	7:3-9, 7:16-18, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9, 19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2, 25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8	4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75, Claims 1-20, Figs. 1-6
[24G] and identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first	Abstract, Paras. 5, 9, 11, 15, 16, 17, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 50, 53, 54, 55, 56, 59, 60, 61, 63, 64, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 11, 15, 16, 17, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 50, 53, 54, 55, 56, 59, 60, 61, 63, 64, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, Paras. 5, 9, 11, 15, 16, 17, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 50, 53, 54, 55, 56, 59, 60, 61, 63, 64, 71, 72, 73, Claims 1-4, Figs. 1-10	Abstract, 1:12-19, 2:1-11, 2:12-13, 3:1-19, 4:5-20, 5:1-8, 5:9-12, 5:13-16, 5:17-21, 6:1-7:2, 7:3-9, 7:9-15, 7:16-18, 9:1-7, 9:10-13, 9:14-11:4, 11:5-10, 11:11-12:12, 12:13-13:2, 13:3-17, 13:18-14:9, 14:10-21, 15:1-16:2, 16:5-18, 16:19-17:8, 17:9-16, 17:17-:18:4, 18:5-9, 19:13-20:2, 20:20-21:14, 21:15-22:3, 24:17-25:2,	Abstract, 2:5-7, 2:18-3:8, 4:3-5, 4:9-14, 6:12-8:2, 8:3-8, 8:9-9:10, 9:11-21, 10:1-15, 10:16-11:7, 11:8-18, 11:19-12:20, 13:3-16, 13:17-14:6, 14:7-14, 14:15-15:2, 15: 3-7, 15:8-17, 16:12-17:2, 17:6-16, 17:17-18:10, 18:13-19:2, 19:3-14, 19:15-20:8, 25:1-4, Claims 1-5, Figs. 1-2d.	Abstract, 1, 5, 9, 10, 11, 13, 21, 22, 23, 24, 25, 26, 27, 28, 30, 31, 32, 33, 34, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 50, 51, 53, 54, 59, 60, 61, 62, 63, 67, 68, 69, 70, 72, 73, 74, 75, Claims 1-20, Figs. 1-6

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
device does not have access to respective Internet Protocol addresses of the second devices.				25:4-10, 25:11-18, 25:19-26:13, 27:17-20, Claims 1-6, Figs. 1-8		
25. The system of claim 24, wherein the data includes a short message service message, a text message, an image, or a video.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.
26. The system of claim 24, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC).	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.
27. The system of claim 24, wherein the second map is a satellite image.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.
28. The system of claim 24, wherein the operations further comprise sending updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information comprising a previous location of the first	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
device, displacement of the first device by a predetermined distance relative to a previous location of the first device, or both.						
29. The system of claim 24, wherein the operations further comprise identifying second user interaction with the interactive display selecting at least one of the user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.
30. The system of claim 24, wherein the message from the second device is a Short Message Service (SMS) message or a text message.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.
31. The system of claim 24, wherein	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
participating in the group further includes sending first status information to the server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second location information comprising a plurality of battery levels of the respective plurality of second devices included in the group, a plurality of signal strengths of wireless signals of the respective plurality of second devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a						

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
combination thereof.						
32. The system of claim 24, wherein the first device is a smart phone.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.
33[P]. The system of claim 24, wherein the operations further include: with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality of second devices;	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.
[33A] and based on transmitting the group identifier associated with the second group, participating in the second group, wherein participating in the second group includes receiving third location information from the server, the third location information comprising a plurality of locations of the respective second plurality of second devices included in the second group.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.

U.S. Patent No. 9,445,251	14/529,978	14/027,410	13/751,453	12/761,533	11/615,472	11/308,648
34. The system of claim 24, wherein the data includes a voice recording.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.
35. The system of claim 24, wherein the first device includes a Global Positioning Satellite (GPS) receiver, and wherein the operations further include: using the GPS receiver to obtain data indicative of the location of the first device, wherein sending the first location information to the server comprises using the Internet Protocol (IP) to send the first location information to the server.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.	See 24P-24G above.

Rejections 2-5 under 35 U.S.C. § 103

The Office Action, starting at page 46, identifies Rejections 2-5 which are based on U.S. Patent No. 7,353,034 to Haney ("Haney"). Specifically, the Office provides that [REJECTION 2] Claim 1, 2, 4, 5, 6, 8, 10, 12, 22, 23, 24, 27, 29, 31, 32, and 35are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 7,353,034 to Haney ("Haney") in view of U.S. Patent No. 6,868,333 to Melen ("Melen"); [REJECTION 3] Claim 1, 2, 4, 5, 6, 8, 10, 12, 22, 23, 24, 27, 29, 31, 32, and 35 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 7,353,034 to Haney ("Haney") in views of U .S. Patent No. 6,868,333 to Melen ("Melen") further in view of U.S. Patent No. 6,662,016 to Buckham ("Buckham"); [REJECTION 4] Claim 13-19 and 21 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 7,353,034 to Haney ("Haney") in view of U.S. Patent No. 6,868,333 to Melen ("Melen") in further view of U.S. Patent No. 6,040,824 to Maekawa ("Maekawa"); [Rejection 5] Claim 13-19 and 21 are rejected under pre-AIA 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 7,353,034 to Haney ("Haney") in view of U.S. Patent No. 6,868,333 to Melen ("Melen") in view of U.S. Patent No. 6,662,016 to Buckham ("Buckham") in further view of U.S. Patent No. 6,040,824 to Maekawa ("Maekawa").

For the reasons set forth below, Patent Owner respectfully traverses this rejection because the Haney reference is not prior art.

<u>Haney Is Not Prior Art Due to The Inventors' Prior Invention</u>

Haney on its face has a filing date of April 4, 2005 with no claim of priority to other applications. Therefore, the priority date of Haney is April 4, 2005. See Terveen Declaration at ¶ 40. Patent Owner respectfully submits that Haney is not a prior reference based on the following reasons.

Attached hereto, Patent Owner submits an affidavit under 37 C.F.R. 1.131(a) by Mr. Malcolm K. Beyer, Jr. showing prior invention of claims 1-35 of the '251 patent.

As Mr. Beyer states, he is the CEO of AGIS and AGIS, Inc. and the first-named inventor on the '251 patent. See Beyer Declaration at ¶¶ 1, 6. His company employs approximately 15 people and its primary business has revolved around offering and selling the AGIS LifeRing product and solutions since 2004. See Beyer Declaration at ¶¶ 5, 7. To date, LifeRing has been successfully deployed and tested in operational environments, including:

- New York Emergency Operations Center Test
- National Incident Management System (NIMS)Test
- Coalition Warrior Interoperability Demonstration (CWID)
- Army Network Integration Evaluation (NIE 12.1, 12.2 and13.1)
- Numerous US Joint Chiefs of Staff Exercises
- The Defense Intelligence Agency (LifeRing won 4 Stars in the DIA's PLUGFEST)
- SOCOM TNT Exercises (2012 &2013)
- US NATO Bold Quest (2012, 2013)
- Joint-Interagency Field Experimentation (JIFX) Exercises
- Army Expeditionary Warrior Experiment (AEWE 2013)
- Jolted Tactics
- US Navy
- International Partners (e.g., Australian, Dutch Defense Forces)

See Beyer Declaration at ¶ 11.

Mr. Christopher R. Rice is identified as a co-inventor on the '251 patent. *Id.* at ¶ 13. With respect to the '251 patent, Mr. Rice's contributions included the client-server communications. See *id.* at ¶ 13; Rice Declaration at ¶ 7. For example, Mr. Rice contributed to the portion of the claimed invention related to the following recited limitations: "sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location" and "receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates." Beyer Declaration at ¶ 13; Rice Declaration at ¶ 7.

As Mr. Beyer states, the claimed inventions of the '251 patent were conceived of by at least January 19, 2005 and were reduced to practice by October 22, 2005 with the server-based implementation of the AGIS LifeRing product. See Beyer Declaration at ¶ 15; Rice Declaration at ¶ 8. The October 22, 2005 reduction to practice of the AGIS LifeRing product was the product of diligent work on the part of a team of engineers and programmers under his direction and supervision. See Beyer Declaration at ¶ 15; Rice Declaration at ¶ 21.

<u>The Inventors Conceived of the Claimed Inventions of the '251 Patent by January 19, 2005</u>

As Mr. Beyer states, the claimed inventions recited in the '251 patent were conceived of by at least January 19, 2005. See Beyer Declaration at ¶¶ 15-16. In August 2004, AGIS and its customer Raytheon engaged in discussions to develop a homeland security solution based on AGIS's LifeRing solution. *Id.* at ¶ 16. Exhibit 1 to the Beyer Declaration is an email summarizing an outline of the planned phases of development. Beyer Declaration at ¶ 16. Pursuant to AGIS's commitment to move forward, AGIS subsequently submitted a proposal for a system of PDA/cell phone devices for use in a homeland defense system in the form of a statement of work (SOW) on or around August 29, 2004. *Id.* at ¶ 16. Exhibit 2 to the Beyer Declaration is an August 29, 2004 SOW outlining the system using PDA cell phone devices for transmitting and displaying tracked participants on maps, i.e., "correctly geographically superimposed on a map" and for interacting with the map to exchange data. Beyer Declaration at ¶ 16.

On January 19, 2005 Mr. Beyer prepared a presentation for Raytheon, attached as Exhibit 3 to the Beyer Declaration. Beyer Declaration at ¶ 17. As Mr. Beyer states, the January 19, 2005 presentation was presented to Raytheon personnel, including Kurt Winckler, on January 21, 2005. Beyer Declaration at ¶ 17.

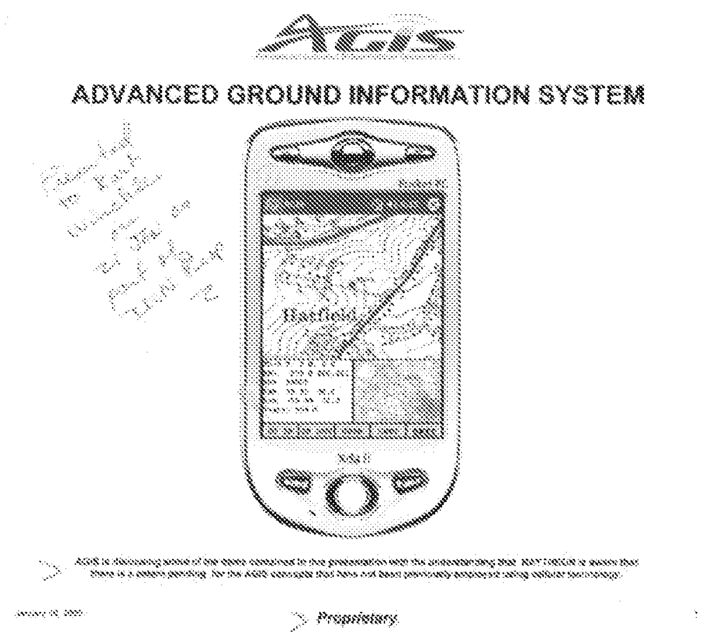


Exhibit 3 to Beyer Declaration, at 1; Beyer Declaration at ¶ 17.

The January 19, 2005 presentation confirms that Mr. Beyer and Mr. Rice had conceived of the claimed inventions as early as January 19, 2005. See Beyer Declaration at ¶¶ 17-18; Rice Declaration at ¶ 8. It describes that the AGIS LifeRing solution had cell phone/PDA devices programmed with software to permit users to display different maps and overlay maps, zoom and offset maps, enter other entities (track symbols), assign information associated with entities, obtain location and information of other AGIS units and tracking them, displaying tracking histories, and selecting different maps (e.g., fixed versus moving). Beyer Declaration at ¶ 18.

AGIS OPERATION
Display

The AGIS operator can:
- Display different maps and overlay maps
- Zoom and Offset maps
- Enter other entities (track symbols) such as accidents, fires, trucks, military units, etc., and assign information associated with them
- Obtain range and bearing readings to other locations
- Obtain location and information of other AGIS units and the tracks that they have entered
- Display the track histories
- Select a fixed map (his symbol moves across the map)
- Select a moving map (his symbol stays at the center of the display and the map moves underneath)
- Specify that the map and the symbols occupy most of the display

Proprietary

Exhibit 3 to Beyer Declaration, at 3; Beyer Declaration at ¶ 18. The January 21, 2005 presentation describes that the AGIS software provides each user with the following features: display user's phone's and other users' phones' location, status, heading, speed, altitude, phone signal strength, and GPS status; displaying phones' locations, identities, and status information on different maps; establishing voice and data communications with other participants by interacting with the symbols; and transmitting messages and multimedia between participants. Beyer Declaration at ¶ 19.

AGIS OPERATION
Display

The AGIS operator can:
- Display different maps and overlay maps
- Zoom and Offset maps
- Enter other entities (track symbols) such as accidents, fires, trucks, military units, etc., and assign information associated with them
- Obtain range and bearing readings to other locations
- Obtain location and information of other AGIS units and the tracks that they have entered
- Display the track histories
- Select a fixed map (his symbol moves across the map)
- Select a moving map (his symbol stays at the center of the display and the map moves underneath)
- Specify that the map and the symbols occupy most of the display

Proprietary

Exhibit 3 at 3 to Beyer Declaration; Beyer Declaration at ¶ 19.

AGIS SOFTWARE

PROVIDES EACH USER.

- Display of his and other phones' location, status, heading, speed, altitude, phone signal strength and GPS status (on, off, 2D or 3D).
- His and other AGIS participants' location, identity and status information superimposed on a variety of maps, aerial photographs, and satellite images.
- Multiple, easily accessible, layered *SoftSwitches* organized by category to enable the AGIS operator to quickly operate the system.
- The ability quickly and easily to establish voice and data communications (*point to talk*) with any other AGIS net participant without losing the tactical picture by simply hooking other site's symbols.
- The ability quickly to transmit free text, fixed formatted messages, photographs and video clips between each user.

Proprietary

Exhibit 3 to Beyer Declaration, at 5; Beyer Declaration at ¶ 19.

AGIS SOFTWARE
Continued

AGIS FURTHER PROVIDES EACH USER:

- The ability to enter friendly and hostile forces (or other information such as fires, emergencies, accidents, pilot boats, fire boats, etc.*) locations on a map and to associate data with their symbol. This information is then transmitted along with the AGIS participants location on the AGIS cellular net to all other AGIS equipped users.
- The ability to receive and transmit free text, fixed formatted, photograph**, and video clip** messages by pointing at the AGIS user to whom they are to be sent.
- The ability to establish range and bearing data from the user to any point.
- The ability for each AGIS user to view his own and other tracks' history trail and last heading data.
- The ability to also operate from a PC and a Tablet*.

Proprietary

Exhibit 3 to Beyer Declaration, at 6; Beyer Declaration at ¶ 19.

The January 21, 2005 presentation describes the formation and use of groups or "nets" of participants. Beyer Declaration at ¶ 20.

AGIS OPERATION

The operator controls the AGIS with either a Stylus or using Finger On Glass

The operator interacts with the AGIS through the use of layered *SoftSwitches* which are located at the bottom of the display.

When initializing the system, the AGIS operator specifies from a list those AGIS users with whom he desires to net location data and with whom he desires to be able to conduct rapid voice and data communications.

When the AGIS is prompted to respond with its location, it sends its location and status to all that are part of the net.

MilStd 2525 symbols (or Homeland Defense defined symbols*) are associated with each user and appear at the correct location on each of the displays.

The AGIS operator obtains information concerning other symbols appearing on the display by touching the screen at their location thereby "hooking" them. Data concerning the other symbol then appears on the lower part of the AGIS display.

Proprietary

Exhibit 3 to Beyer Declaration, at 8; Beyer Declaration at ¶ 20.

AGIS OPERATION
Cellular phone calls

- When the AGIS operator decides to call another AGIS user, he simply hooks the other user's symbol and selects the Call SoftSwitch. The phone of the AGIS receiving call then hears a ring and sees a box appear around the sender's symbol.
- The AGIS operator can setup pre-established nets of up to six AGIS net participants. When he desires to talk to that group of users he selects the appropriate net number and a conference call is automatically made.
- The AGIS operator can conference an almost unlimited number of AGIS participants by using the AGIS's 800 conferencing capability. Again the AGIS operator assigns AGIS participants to a net number. When he desires to make the conference call, he selects the net number. This action causes a message to be sent to all of the AGIS participants, causing their phones to automatically call an 800 number and automatically entering their participant code.

Proprietary

Exhibit 3 to Beyer Declaration, at 9; Beyer Declaration at ¶ 20.

As Mr. Beyer states, AGIS's LifeRing solution was under development throughout 2004 and 2005. Beyer Declaration at ¶ 21. During this period, Mr. Beyer conceived and reduced to practice both non-server and server-based implementations. *Id.* Mr. Rice's contributions included the client-server communications, including "sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location" and "receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates." See Beyer Declaration at ¶ 21; Rice Declaration at ¶¶ 7-9. In the non-server implementations, AGIS's maps were uploaded to and called from databases and the transmissions occurred over SMS only. Beyer Declaration at ¶ 21; Rice Declaration at ¶¶ 9-10. However, in the server-based implementations, which were conceived of by January 19, 2005, the devices communicated with intermediary servers and the maps requested and received from servers. Beyer Declaration at ¶ 21; Rice Declaration at ¶¶ 9-10. The January 21, 2005 presentation describes an AGIS server implementation and the development and testing of preproduction server-based implementations were underway. Beyer Declaration at ¶ 21.

AGIS SERVER

- When AGIS is operating in a SMS (slow speed) mode of operation, communications are from the AGIS phone to the telephone company and then from the phone company to all other AGIS phones.
- When the AGIS is operating in a high speed GRPS / CDMA2000 1xEVDO mode, communications are from the AGIS phone to the telephone phone company, then to the TCP/IP Server and from the TCP/IP Server to all the other AGIS phones.
- The TCP/IP Server pushes the received data to the AGIS phones so that information is received within approximately 5 seconds.
- The TCP/IP Server provides interoperability between the GRPS / CDMA2000 1xEV-DO communications.
- The AGIS accounts for when the phones go into Vcice mode and then transmits data in SMS until high speed communications are available.
- The TCP/IP Server is operating on a 24 hour basis. We are still in the process of resolving Server Trouble Reports.

Proprietary

Exhibit 3 to the Beyer Declaration, at 14; Beyer Declaration at ¶ 21.

AGIS STATUS

THE PROTOTYPE AGIS HAS UNDERGONE A WEEK OF TESTING AT THE NAVAL RESEARCH LABORATORY'S (NRL) ANTITERRIORISM GROUP. DURING THE TEST PERIOD, THE AGIS OPERATED WITHOUT FAILURE. THE RESULTS OF THIS TESTING WAS POSITIVE. IT WAS REQUESTED THAT:

AGIS FUNCTION SWITCHES BE MADE EASIER TO USE AND REMEMBER*

MULTIPLE MAP SETS CAPABILITY BE PROVIDED*

DIRECTION UP MAPS AND MULTIPLE LOCATION MAPS BE PROVIDED**

WE ARE UNDER CONTRACT TO ADD THE OTHER FEATURES (OTHER THAN SECURITY) DISCUSSED IN THIS PRESENTATION.

THE PREPRODUCTION AGIS IS DUE TO BE TESTED AT NRL IN FEB 2005.

ALL THE FEATURES (EXCEPT SECURITY) WILL BE INCORPORATED INTO THE SEATTLE DELIVERY

* Completed. ** Will be available in January or early February

[illegible] *Proprietary*

Exhibit 3 to the Beyer Declaration, at 15; Beyer Declaration at ¶ 21. This is consistent with AGIS's plans to provide "over the air" georeferenced maps via servers. Exhibit 4 at 8-9; Beyer Declaration at ¶ 22; Rice Declaration at ¶¶ 9-10, 22.

and Figure 19 depicts selection of coastlines and geopolitical boundaries. As part of a

Proprietary

Advanced Ground Information Systems, Inc. Topic Number N05-069 Proposal Number N051-069-0725 Proprietary 9

separate non government funded effort, AGIS plans to incorporate "over the air" downloading of georeferenced maps.

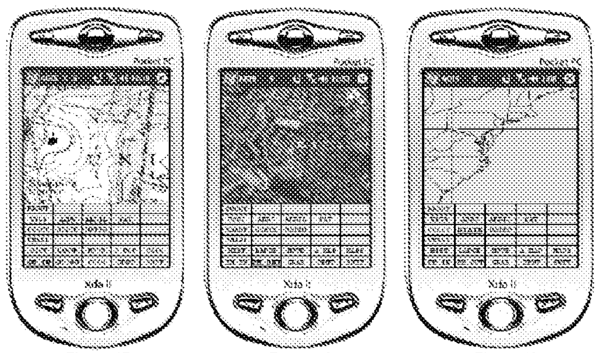


Figure 17 Figure 18 Figure 19

Exhibit 4 at 8-9; Beyer Declaration at ¶ 22.

<u>The Inventors Were Diligent In Reducing To Practice The Claimed Inventions of the '251 Patent</u>

There is sufficient evidence to demonstrate that Mr. Beyer and Mr. Rice were diligent in reducing to practice the claimed inventions of the '251 patent from the conception date of at least January 19, 2005 and from prior to the filing date of Haney (April 4, 2005) to October 2005. As Mr. Beyer states, he assembled a team to begin development and testing of various implementations immediately after conception. See Beyer Declaration at ¶ 23; Rice Declaration at ¶¶ 21. While the AGIS project timeline consisted of multiple phases and the project timeline was updated continuously over the course of development (Exhibits 5-8 show multiple timeline updates based on changes to the server-based implementation), the AGIS team worked tirelessly and continuously to reduce to practice the claimed inventions of the '251 patent. See Beyer Declaration at ¶ 23; Rice Declaration at ¶ 21.

From April 3 2005 (the Haney filing date is April 4, 2005) to October 2005, AGIS dedicated hundreds of billable hours to the development and testing of the server-based implementations of the AGIS LifeRing solution which were conceived of by January 19, 2005. See Beyer Declaration at ¶ 23. Exhibits 9-14 to the Beyer Declaration include timesheets from the individuals involved in the development and evaluation of the AGIS LifeRing solution during the period of April 2005 through October 2005, including Christopher R. Rice, Sandel Blackwell, Dennis Hoff, Scott Brown, and Jason Cardamone. See Beyer Declaration at ¶ 23; Rice Declaration at ¶ 21. This demonstrates that Mr. Beyer and Mr. Rice dedicated significant effort and resources in a continuous and diligent manner during the relevant period. See Beyer Declaration at ¶ 23; Rice Declaration at ¶¶ 9-12.

Through April 2005, Mr. Beyer's records show that AGIS was diligent in reducing to practice the claimed inventions of the '251 patent. See Beyer Declaration at ¶ 23. In April 2005, Mr. Rice dedicated at least 39 hours to fixing the SMS transmission and TCP protocol portions of LifeRing. See Beyer Declaration at ¶ 24; Rice Declaration at ¶ 13. In April 2005, Mr. Blackwell billed 89 hours for work including, for example, updating the change rate of the displayed information, grouping and selecting participant groups, and user location updates. See Beyer Declaration at ¶ 24. Mr. Blackwell was also involved in testing TCP and SMS communications issues and sever issues including fixing database scroll issues on the server-based implementation. *Id.* Mr. Blackwell also worked on addressing memory leakage issues. *Id.* Mr. Blackwell also worked on integrating and developing global mapper features and other different maps used in LifeRing. *Id.* Mr. Blackwell communicated with and worked closely with Mr. Rice to address these problems. Beyer Declaration at ¶ 24; Rice Declaration at ¶ 20. Mr. Blackwell also worked on finalizing several versions of code. See Beyer Declaration at ¶ 24. In

April 2005, Mr. Hoff billed 17 hours for work on testing portions of LifeRing and developing code for the client side of LifeRing. See Beyer Declaration at ¶ 24. Mr. Hoff was involved in testing location tracking and the loading of the client side application. *Id.* Prior to April, Mr. Hoff was involved in development and testing for the location tracking, mapping data, interaction with maps and symbols, displaying of locations, maps and symbols, data and multimedia transfers, memory issues, and other issues related to both client and server side issues and had billed over 190 hours for such work between January 2005's conception date and April 2005. *Id.* In April 2005, Mr. Brown billed 160 hours for work on testing the LifeRing system. *Id.* Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. *Id.* Mr. Brown also assisted Mr. Blackwell, Mr. Rice and Mr. Beyer on documentation of the LifeRing reports, manuals, and versions. *Id.*

Mr. Beyer's May 2005 records show that AGIS was diligent in reducing to practice the claimed inventions of the '251 patent. See Beyer Declaration at ¶ 25. In May 2005, Mr. Rice billed at least 60 hours for work related to fixing problems related to SMS and voice transitions, photo transmissions, and SMS transmissions. See Beyer Declaration at ¶ 25; Rice Declaration at ¶ 14. Mr. Rice also spent considerable time creating and testing UDP communications protocol for establishing communications between applications and servers, including testing the new implementation of the protocol with a device. See Beyer Declaration at ¶ 25; Rice Declaration at ¶ 14. Mr. Rice also worked on the development of the CSIF code. See Beyer Declaration at ¶ 25; Rice Declaration at ¶ 14. During this time, Mr. Rice communicated frequently with Mr. Blackwell, who was also working on updates to the code. See Beyer Declaration at ¶ 25; Rice Declaration at ¶ 14. In May 2005, Mr. Hoff billed 11 hours for work on testing portions of LifeRing and working with Mr. Rice and Mr. Blackwell on testing features. See Beyer

Declaration at ¶ 25. In May 2005, Mr. Cardamone billed 80 hours for work related to the code and configuration of an SVN code repository. *Id.* Mr. Cardamone also worked on the communications and display issues for LifeRing. *Id.* Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. *Id.* In May 2005, Mr. Brown billed 80 hours for work on testing the LifeRing system. *Id.* Mr. Brown also assisted Mr. Blackwell, Mr. Rice and Mr. Beyer on documentation of the LifeRing reports, manuals, and versions. *Id.*

Mr. Beyer's June 2005 records show that AGIS was diligent in reducing to practice the claimed inventions of the '251 patent. See Beyer Declaration at ¶ 26. In June 2005, Mr. Rice billed at least 60 hours for work related to transmission issues with SMS and phone/conference calls. See Beyer Declaration at ¶ 26; Rice Declaration at ¶ 15. He worked on enhancing the stability of the SMS transmissions and supported efforts on data reductions, UDP communications protocol, and CSIF code. See Beyer Declaration at ¶ 26; Rice Declaration at ¶ 15. Mr. Rice also made changes to the code for location tracking and message handling. See Beyer Declaration at ¶ 26; Rice Declaration at ¶ 15. Mr. Rice also worked on network configuration for the development server. See Beyer Declaration at ¶ 26; Rice Declaration at ¶ 15. During this time, Mr. Rice communicated frequently with Mr. Blackwell, who was also working on updates to the code. See Beyer Declaration at ¶ 26; Rice Declaration at ¶ 20. In June 2005, Mr. Hoff billed 10 hours for work on testing new versions of LifeRing. See Beyer Declaration at ¶ 26. Mr. Hoff was involved in testing SMS features of LifeRing. *Id.* In June 2005, Mr. Cardamone billed 160 hours for work related to the LifeRing code repository and data reductions/transmissions related to LifeRing data. *Id.* Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. *Id.* In June 2005, Mr. Brown billed over 160 hours for work on testing the LifeRing system. *Id.* Mr. Brown also assisted Mr. Blackwell, Mr. Rice and Mr. Beyer on documentation of the LifeRing reports, manuals, and versions. *Id.*

Mr. Beyer's July 2005 records show that AGIS was diligent in reducing to practice the claimed inventions of the '251 patent. See Beyer Declaration at ¶ 27. In July 2005, Mr. Rice billed over 40 billable hours on communications infrastructure issues and network configuration and server migration issues. See Beyer Declaration at ¶ 27; Rice Declaration at ¶ 16. In July 2005, Mr. Blackwell billed over 80 billable hours on various issues related to LifeRing including communications issues, display issues, and location tracking issues. See Beyer Declaration at ¶ 27. Mr. Blackwell performed research to solve some issues that AGIS had with the server-based implementation and he worked on developing the new maps that had been integrated and tested in April and May 2005. *Id.* Again, during this period, Mr. Blackwell remained in constant communication with Mr. Rice in order to resolve issues with the development of LifeRing. See Beyer Declaration at ¶ 27; Rice Declaration at ¶ 20. In July 2005, Mr. Cardamone billed 120 hours for work on data processing and data reduction relating to the georeferenced maps and georeferenced map data for presentation and use in LifeRing. Beyer Declaration at ¶ 27. In April 2005, Mr. Brown billed 80 hours for work on testing the LifeRing system. *Id.* Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. *Id.* Mr. Brown also assisted Mr. Blackwell, Mr. Rice and Mr. Beyer on documentation of the LifeRing reports, manuals, and versions. *Id.*

Mr. Beyer's August 2005 records show that AGIS was diligent in reducing to practice the claimed inventions of the '251 patent. See Beyer Declaration at ¶ 28. In August 2005, Mr. Rice billed over 80 hours for work on the LifeRing database systems, including databases used for location tracking data. See Beyer Declaration at ¶ 28; Rice Declaration at ¶ 17. Mr. Rice's August 2005 work also included work on the processing output (maps) for LifeRing. See Beyer Declaration at ¶ 28; Rice Declaration at ¶ 17. During August 2005, AGIS ran into problems with the startup and shutdown of the LifeRing application. See Beyer Declaration at ¶ 28; Rice Declaration at ¶ 17. Mr. Rice worked on researching ways to reduce delays in starting the program and obtaining new maps and location data immediately upon startup. See Beyer Declaration at ¶ 28; Rice Declaration at ¶ 17. This included the retrieval of maps in both non-server and server-based implementations. See Beyer Declaration at ¶ 28; Rice Declaration at ¶ 17. In August 2005, Mr. Blackwell billed over 150 hours for work on the development of LifeRing. See Beyer Declaration at ¶ 28. During this time, Mr. Blackwell was intimately involved in researching and development of the server-based implementation. *Id.* He also spent considerable time on testing communications issues and updating the display of information on the client. *Id.* In August 2005, Mr. Cardamone billed 200 hours for work on data processing and data reduction relating to the georeferenced maps and georeferenced map data for presentation and use in LifeRing. *Id.* During this time, Mr. Cardamone also worked on code processing output for display (e.g., georeferenced maps) and various display filter issues in the LifeRing code. *Id.* In August 2005, Mr. Brown billed 160 hours for work on testing the LifeRing system. *Id.* Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. *Id.* Mr. Brown also assisted Mr. Blackwell, Mr. Rice and Mr. Beyer on documentation of the LifeRing reports, manuals, and versions. *Id.* Mr. Brown also assisted on preparing the LifeRing product for presentation to customers in Seattle. *Id.*

Mr. Beyer's September 2005 records show that AGIS was diligent in reducing to practice the claimed inventions of the '251 patent. See Beyer Declaration at ¶ 29. In September 2005, Mr. Rice billed over 70 hours for work on the LifeRing location tracking database and location history issues. See Beyer Declaration at ¶ 29; Rice Declaration at ¶ 18. Mr. Rice also began to develop code to more efficiently manage and distribute processing tasks in order to improve the delay issues in retrieving and processing maps and location information while also maintaining a reliable data communications flow. See Beyer Declaration at ¶ 29; Rice Declaration at ¶ 18. These improvements were being developed to support the non-server and server-based implementations. See Beyer Declaration at ¶ 29; Rice Declaration at ¶ 18. In September 2005, Mr. Blackwell billed over 150 hours for work on the development of LifeRing. Beyer Declaration at ¶ 29. Considerable time was spent on the server-based implementation, including on display, mapping, and communications tests and integrating the former with the existing non-server implementation. *Id.* Mr. Blackwell spent a considerable amount of time researching whether and how to migrate the existing data and programs to the server and the impact of "over the air" transfer on the performance of the client. *Id.* Mr. Blackwell worked closely with Mr. Rice to press for an October 2005 release of the server-based implementation. See Beyer Declaration at ¶ 29; Rice Declaration at ¶ 20. In September 2005, Mr. Brown billed over 190 hours for work on testing the LifeRing system. See Beyer Declaration at ¶ 29. Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. *Id.* Mr. Brown also assisted Mr. Blackwell, Mr. Rice and Mr. Beyer on documentation of the LifeRing reports, manuals, and versions. *Id.*

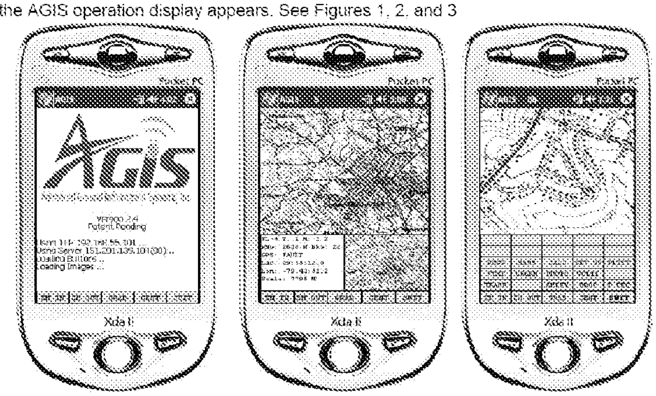
Mr. Beyer's October 2005 records show that AGIS was diligent in reducing to practice the claimed inventions of the '251 patent. See Beyer Declaration at ¶ 30. In October 2005, Mr.

Rice billed over 85 hours for work on the LifeRing development. See Beyer Declaration at ¶ 30; Rice Declaration at ¶ 19. In addition to working on the location tracking database issues identified in bug reports, Mr. Rice worked on finalizing the development of the image server implementation, which was the project for requesting/receiving maps from servers. See Beyer Declaration at ¶ 30; Rice Declaration at ¶ 19. Mr. Rice's work on this included writing and testing code for communicating with different servers for providing georeferenced maps to the LifeRing client applications on devices. See Beyer Declaration at ¶ 30; Rice Declaration at ¶ 19. In October 2005, Mr. Blackwell billed over 140 hours for work on the development of LifeRing. See Beyer Declaration at ¶ 30. Again, Mr. Blackwell spent considerable time on the server-based implementation and working to finalize the new release with Mr. Rice. See Beyer Declaration at ¶ 30; Rice Declaration at ¶ 20. This was the culmination of significant effort as the requested maps come in various formats and protocols from different servers. Beyer Declaration at ¶ 30. Mr. Rice and Mr. Blackwell were deeply involved in finalizing the October 22, 2005 code base for the server-based implementation and addressing issues found in bug reports and testing services. See Beyer Declaration at ¶ 30; Rice Declaration at ¶ 20. In October 2005, Mr. Brown billed 180 hours for work on testing the LifeRing system. See Beyer Declaration at ¶ 30. Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. *Id.* Mr. Brown also assisted Mr. Blackwell, Mr. Rice and Mr. Beyer on documentation of the LifeRing reports, manuals, and versions. *Id.* Mr. Brown spent considerable amount of time testing the October 22, 2005 server-based implementation. *Id.*

On October 22, 2005, The Inventors Reduced to Practice the Claimed Inventions Conceived by January 19, 2005

The claimed inventions of the ’251 patent were reduced to practice by at least October 22, 2005. See Beyer Declaration at ¶ 31. As Mr. Beyer declares, Exhibit 15 consists of code files that were checked into and present in AGIS’s code repository by October 22, 2005. See Beyer Declaration at ¶¶ 31-32; Rice Declaration at ¶ 22. As Dr. Terveen confirms in his expert declaration, the October 22, 2005 server-based implementation of AGIS LifeRing practices each and every claim of the ’251 patent. See Terveen Declaration at ¶¶ 41-42. Dr. Terveen has submitted the below claim chart mapping the October 22, 2005 version of LifeRing to the claims. See Terveen Declaration at ¶¶ 41-42; Beyer Declaration at ¶ 32; Rice Declaration at ¶ 23.

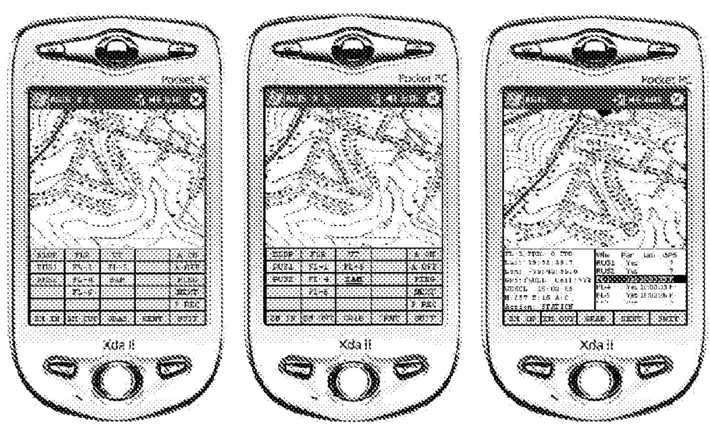
U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
1. A computer-implemented method comprising:	The AGIS LifeRing product practices the computer implemented method of claim 1. The AGIS LifeRing product includes systems and methods for executing an application on PDA/cell phones, now referred to as a smartphone devices. See, e.g., “call.cpp” references to “phone” and “other phones.”

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	The operator can then send voice conference or digital data to those AGIS equipped units by simply selecting their assigned NET SoftSwitch. The AGIS application software does not affect the operation of the PDA, the Cell phone or the GPS until it is activated. When the AGIS application software is selected, the AGIS PDA Cell phone is turned on, the GPS is automatically connected to the PDA / Cell phone through a Bluetooth interface and the AGIS Logo appears. Shortly thereafter the AGIS operation display appears. See Figures 1, 2, and 3  Figure 1 Figure 2 Figure 3 Exhibit 4 to the Beyer Declaration, depicting AGIS PDA/cell phones.
with a first device, receiving a message from a second device, wherein the message relates to joining a group;	The AGIS LifeRing product practices "with a first device, receiving a message from a second device, wherein the message relates to joining a group." The first device is programmed to receive a message from a second device related to joining a group. The process of joining a group is performed by sending messages to server and other users to join a group. The file "display/msgproc.cpp" implements the function "process_net_msg()"[5] where the logging function indicates "rcvd net msg" indicative of receiving a network message. The "nmsg" variable of type "NETUSERS" which is input to this function is tested against its member variable requestType which is specified in the file "csiftranscode.h"[6] and states to have "J)oining or R)equest current participants". The "nmsg.requestType" is then branched for cases of "I" for initial joining, "J" for joining, "R" for requesting and "X" for exiting the network. For the case of "J" for joining, the function makes a call to the "update_address_book()" function with sender information as the input parameters.

[5] *See, e.g.*, display/msgproc.cpp at L1234-1292..
[6] *See, e.g.*, csiftranscode/csiftranscode.h at L254-263.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	The file "initfiles.cpp" implements the "update_address_book()"[7] where the sender information is used to find its index in the address book by a call to "find_addr_number()". The return from this call is used to update the array variable "csif_addr[]" by either adding the address if it does not exist indicative of a return value of -1 or by copying the "ipaddr" into a struct member of the indexed variable. The "csif_addr[]" array is an array of type "ADDRESS_ENTRY"[8] which is specified in the file "csiftranscode.h"[9] The members of this struct includes "name", "number", "group", "type", "level" and "ipaddr". The marking of the "csif_addr[]" variable array constitutes the joining of a group. The server further provides additional functionality in file "server/msgproc.cpp" implements the function "process_net_msg()"[10] where the logging function indicates "rcvd net msg" indicative of receiving a network message. The variables "from" of type character string and "nmsg" of type "NETUSERS" as described above is input to this function. Similar to described above, the "nmsg.requestType" is then branched for cases of "J" and not "J" described to mean of type "R" are handled. For the case of joining a group where "nmsg.requestType" equals "J", the function "update_address_book()" is called with the sender and receiver information and returns with an index into the address book array "csif_addr" as discussed above. Then the logging command indicates that the "address book [address book index] updated with ipaddr [address book ip address]" and writes "user [name] joined network" to the display. Followed by two calls to "send_net_msg_participants()" using a wildcard designator "-1" indicating everything in the respective field is to be sent. The first call sends everything in the table to the currently added user. The second call sends to everyone the currently added user. In the same file the function "send_net_msg_participants()"[11] documents that the user of "-1" in the first field indicates a "send to all" and the use of "-1" in the second field indicates a "send all net participants" condition.

[7] *See, e.g.*, display/initfiles.cpp at L701-721.
[8] *See, e.g.*, display/initfiles.h at L 35
[9] *See, e.g.*, csiftranscode/csiftranscode.h at L45-54.
[10] *See, e.g.*, server/msgproc.cpp at L235-293.
[11] *See, e.g.*, server/msgproc.cpp at L108-191.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Thus, one of the objectives of the server function "send_net_msg_participants" is to process messages that are received from the second device related to joining a group and to send messages to the first device related to joining a group. For example, the source code reflects the parameter "nmsg.requetType='R'" that reflects a request to join a group being transmitted from a second device to a first device via the server. 2.5 AGIS Net Assignment and Ping The AGIS permits the person in charge of the network to specify which AGIS cell phones are to be part of the network. This is accomplished by the AGIS operator selecting the P REQ SoftSwitch which causes a matrix of the AGIS phones to appear. Note that this is a multilayered matrix and, if there are more than 28 phones, the operator can select the NEXT SoftSwitch which will cause yet another group of participants to appear. The operator can drop highlighted AGIS units from the net by selecting their SoftSwitch or add new ones to the net by selecting them. In Figure 22 note that AGIS Participant RUS 2 is not part of the AGIS communications net. In Figure 23 note that RUS 2 has been made part of the AGIS communications net and FL 5 and SAM have been dropped from the AGIS communications net. The AGIS operator can also select to immediately transmit his location and all his tracks to another AGIS unit and cause that unit to immediately report its location and all its tracks by selecting the PING SoftSwitch To determine the status of the various AGISs in the communications net, the AGIS operator selects the PLIST SoftSwitch which causes display of the current AGIS communications net participants to be displayed in the lower left of the AGIS display. See Figure 24.  Figure 22 Figure 23 Figure 24 Exhibit 4 to the Beyer Declaration, at 10-11. Similarly, the server further provides functionality to support the server side function similar to the client side in the file "server/initfiles.cpp" by implementing the function "update_address_book()"[12] where similar

[12] *See, e.g.,* server/initfiles.cpp at L194-211.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	to the description above the call to "find_addr_number()" function retrieves an address index. If the address does not exist indicative of a return index value of "-1" calls the function "add_address_book()" otherwise updates the "csif_addr" array with the ip address input to the function. In the same file the function "add_address_book()"[13] is implemented where the "csif_addr" array is populated with the "name", "number", "group", "ipaddr" and a "isValid" member variable.
based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group;	The AGIS LifeRing product practices "based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group." Once the devices have formed a group they share their location with each other. The AGIS software captures location information on the device through a serial communication link configured to continuously process GPS information messages. Messages such as location, precision and constellation are captured on an independent thread and when a valid message is received the thread triggers an event indicating the presence of a new message. The software then sends the new location to others through either a server or direct connection. The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function "initGPS()"[14] is implemented where the variable "comport" is the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[15] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[16] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[17] and once a message is detected that it calls the

[13] *See, e.g.*, server/initfiles.cpp at L163-191.
[14] *See, e.g.*, csif/csif.cpp at L126-148.
[15] *See, e.g.*, csif/gpslistener.cpp at L36-51.
[16] *See, e.g.*, csif/gpslistener.cpp at L101-235.
[17] *See, e.g.*, csif/gpslistener.cpp at L158.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[18] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[19] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[20] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[21] is implemented where the queued GPS messages are processed incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[22] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[23] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[24] is an example of communication with transmission of location information. This function is conditioned to use certain code for the

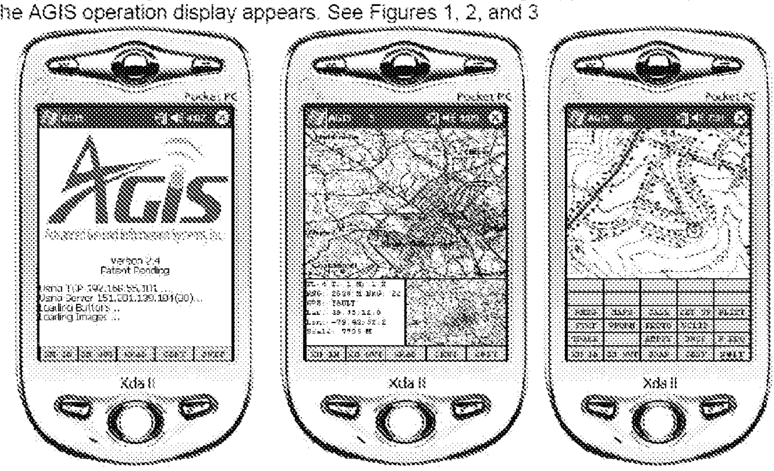
[18] *See, e.g.*, csif/gpslistener.cpp at L76-99.
[19] *See, e.g.*, display/display.cpp at L274-581.
[20] *See, e.g.*, display/display.cpp at L584-969.
[21] *See, e.g.*, display/gps.cpp at L346-365.
[22] *See, e.g.*, display/gps.cpp at L292-336.
[23] *See, e.g.*, display/gps.cpp at L77-126.
[24] *See, e.g.*, display/display.cpp at L1015-1037 and L1076-1115.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[25] and are implemented in the function "button_actions()"[26] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[27] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[28] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. As a further example of receiving messages from the second devices that include location information, each time a message is received by the server from a device, the location information from that device is updated at the server. The location information for each device in the group (e.g. the second devices) is then transmitted as part of the update

[25] *See, e.g.*, display/buttons.cpp at L594-598 and L1344-1346.
[26] *See, e.g.*, display/buttons.cpp at L1344-2365.
[27] *See, e.g.*, display/msgproc.cpp at L170-325.
[28] *See, e.g.*, display/msgproc.cpp at L615-682.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	message to the first device, as depicted in nmsg.numMapEntries. msgproc.cpp at ll. 862-892. The "send_track_msg()"function is used under the condition where "add_track()"[29] and "update_track_id()"[30] functions in the "display/track.cpp" is invoked. The "update_track_id()" is used in user interface as views in the "buttons.cpp" file in the main routine where button actions are implemented "button_actions()" as discussed above where the case for "BUTTON_TRACK_ID_UNKN", "BUTTON_TRACK_ID_FRND" and "BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[31] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent the "DDL_MSG_OWN_POSITION" case where the message is passed through by a call to "pass_thru_msg()" function. The file "server/server.cpp" implements the main function "WndProc()"[32] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through.
presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second	The AGIS LifeRing product practices "presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial coordinates." The devices running the software display an interactive display with locations of group devices, georeferenced and

[29] *See, e.g.*, display/track.cpp at L373-477.
[30] *See, e.g.*, display/track.cpp at L482-510.
[31] *See, e.g.*, server/msgproc.cpp at L377-474.
[32] *See, e.g.*, server/server.cpp at L228-386.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
devices, wherein the symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial coordinates;	shown in relation to others. The display further draws tracks and other features related to the display of devices. An exemplary interface from the AGIS client running on a mobile device is depicted below: The AGIS application software does not affect the operation of the PDA, the Cell phone or the GPS until it is activated. When the AGIS application software is selected, the AGIS PDA Cell phone is turned on, the GPS is automatically connected to the PDA / Cell phone through a Bluetooth interface and the AGIS Logo appears. Shortly thereafter the AGIS operation display appears. See Figures 1, 2, and 3  Figure 1 Figure 2 Figure 3 Exhibit 4 to Beyer Declaration, at 4. The code to render this UI, which includes the capability to display each of the devices on a map with data relating the positions to spatial coordinates is found primarily in the display/display.cpp code file. In the file “display/display.cpp” the main method “WndProc()”[33] where display events are captured and processed, the case for “WM_PAINT” where the display is drawn is implemented. When a “WM_PAINT” event occurs calls are made to “refresh_tact()”, “refresh_stat()” and “refresh_inset()” passing the display window handle where the map, locations and other information gets displayed. In the file “tact.cpp” the function “refresh_tact()”[34] is implemented. There are three types of this function, two with an input handle and one

[33] *See, e.g.*, display/display.cpp at L584-969.
[34] *See, e.g.*, display/tact.cpp at L200-306.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
	without. The one without the handle first retrieves the current handle and then calls the "refresh_tact()" function version with the handle. In this function, among other features, the system draws the map by a call to "ww_draw_map()" and draws locations by a call to "track_draw()". A special "refresh_tact()" implementation only draws the tactical area which is documented to correspond to the display's WM_PAINT event and only accepts a window handle of type HWND. The function "refresh_tact()" is used throughout the code to signal an update to the view within "buttons.cpp", "display.cpp", "images.cpp", "inset.cpp", "maps.cpp", "msgproc.cpp", "track.cpp" and others to signal the update of the graphical window. In the file "maps.cpp" the function "ww_draw_map()"[35] is implemented. The function utilizes georeferenced latitude and longitude information to position the map using variables "urlat", "urlon", "lllat" and "lllon" for the upper right and lower left latitude and longitude. The variable array "ww_map" is used to hold the map data and is read in the same file using the function "ww_read_map()"[36] from local files. The function "ww_read_maps()"[37] makes individual calls to "ww_read_map()" to read the "world.pnt", "plit.pnt", "water.pnt" and a local map through invocation of the "local_read_map()" function pointed to by the "test user.ump" file. The "ww_read_maps()" function is called in the file "display/display.cpp" by the function "InitInstance()"[38] when the system has started. In the file "stat.cpp" the function "refresh_stat()"[39] is implemented. There are two types of this function, one with an input handle and one without. The one without the handle first retrieves the current handle and then calls the "refresh_stat()" function version with the handle. In this function, among other features, the system "refresh status window" by a call to "stat_template_refresh()". There are other methods that the system implements such as "show_status_window()"[40] and the main window process in the "WndStatProc()"[41] callback function "WM_PAINT" event which calls the "refresh_stat()" and in turn

[35] *See, e.g.*, maps.cpp at L213-276.
[36] *See, e.g.*, maps.cpp at L69-153.
[37] *See, e.g.*, maps.cpp at L726-732.
[38] *See, e.g.*, display/display.cpp at L268-581.
[39] *See, e.g.*, stat.cpp at L321-371.
[40] *See, e.g.*, stat.cpp at L387-403.
[41] *See, e.g.*, stat.cpp at L39-115.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"stat_template_refresh()". Upon creation of the stat view by use of the function "create_stat()"[42] to create the graphical interface followed by a call to the "stat_template_refresh()". Furthermore, the function "refresh_stat()" is used throughout the code within "buttons.cpp", "display.cpp", "gps.cpp", "msgproc.cpp", "partlist.cpp" to signal the update of the stat window. The function "stat_template_refresh()"[43] is implemented with a call to "status_hook()". The function "status_hook()"[44] is implemented where various information gets presented such as track information, current latitude, current longitude, GPS status amongst others. In the file "inset.cpp" the function "refresh_inset()"[45] is implemented. There are two types of this function, one with an input handle and one without. Both versions retrieve the Inset window handle through either a global handle or passed in handle and make a call to the "refresh_inset_window()" function. The function "refresh_inset_window()"[46] is implemented in the same file with calls made to "refresh_inset_map()", "refresh_inset_gps()" and "refresh_inset_trackamp()" based on the value held in the "inset_mode" variable set to "INSETWIN_MAP_USE", "INSETWIN_GPS_USE" or "INSETWIN_TRACKAMP_USE". The function "refresh_inset_map()"[47] is implemented in the same file and the georeferenced latitude and longitude values such as "inset_maplllon" or "inset_mapurlat" that are used to draw the map with the track information used from the "track_file" variable. The function draws the map using the system "Bit_Blt()" or call to "ww_draw_image_inset() and then calls the "inset_track_draw()" function which then draws all tracks. The function "ww_draw_image_inset()"[48] is implemented in the "display/images.cpp" and utilizes georeferenced latitude and longitude values. The function "inset_track_draw()"[49] is documented that to "plot all tracks as dots in inset window" and utilizes a loop across all tracks

[42] *See, e.g.*, stat.cpp at L119-166.
[43] *See, e.g.*, stat.cpp at L296-317.
[44] *See, e.g.*, stat.cpp at L207-292
[45] *See, e.g.*, inset.cpp at L708-737.
[46] *See, e.g.*, inset.cpp at L684-705.
[47] *See, e.g.*, inset.cpp at L285-393.
[48] *See, e.g.*, display/images.cpp at L534-596.
[49] *See, e.g.*, inset.cpp at L220-282.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	to draw all tracks. The function "refresh_inset_gps()"[50] is implemented in the same file and primarily provides information about the gps satelites and their respective signal strengths. The function "refresh_inset_trackamp()"[51] updates the amplified track mode equipped with its own series of functions including messaging, location and amplified views.
sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location;	The AGIS LifeRing Product practices "sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location." The first device makes a request for a georeferenced map by interacting with the device's user interface. In the file "buttons.cpp" the function "button_actions()"[52] implements the main routine for capturing the button press event and processing based on the button pressed. For the case of the "BUTTON_IMAGE_REQUEST" a call is made to the "image_server_request()" function. In the file "display/images.cpp" the function "image_server_request()"[53] is implemented. In this function the coordinate boundaries are retrieved by a call to "coord_bounds()" and sent to the function "SendImageRequest()". In the file "coords.cpp" the function "coord_bounds()"[54] is implemented where the latitude and longitude are used to generate the values required. This code is at least an example of a request made to a server for a georeferenced map where the request specifies a map location, e.g., the latitude and longitude of the boundaries for the requested map. In the file "display/msgproc.cpp" the function "SendImageRequest()"[55] is implemented where a message is composed using the "DDL_MSG_MAP_REQUEST" type. The message is then processed by the "put_ddl_msg()" routine as described above and sent to the server with the call to "send_message_bulk()" as described above using the "server_ip" variable as the target for the server.

[50] *See, e.g.*, inset.cpp at L395-512.
[51] *See, e.g.*, inset.cpp at L535-657.
[52] *See, e.g.*, buttons.cpp at L1344-2365.
[53] *See, e.g.*, display/images.cpp at L752-761.
[54] *See, e.g.*, coords.cpp at L88-98.
[55] *See, e.g.*, display/msgproc.cpp at L963-996.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[56] is implemented to process received messages. The condition "DDL_MSG_MAP_REQUEST" is captured and a call is made to "process_map_request()". The function "process_map_request()"[57] is implemented in the same file where a message of type "DDL_MSG_MAP_RESPONSE" is formed and returns a map by calls to "put_ddl_msg()" and "send_message()". In the file "display/msgproc.cpp" the function "process_csif_msg()"[58] is implemented where the condition "DDL_MSG_MAP_RESPONSE" is processed by a call to "process_image_server_response()". In the file "display/images.cpp" the function "process_image_server_response()"[59] is implemented where it is logged that with the "response from image server".

[56] *See, e.g.*, server/msgproc.cpp at L377-474.
[57] *See, e.g.*, server/msgproc.cpp at L312-332.
[58] *See, e.g.*, display/msgproc.cpp at L1391-1472.
[59] *See, e.g.*, display/images.cpp at L763-768.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING

MAP REQUEST

RQST

Selecting the Map SoftSwitch from The AGIS handset's Maps Function SoftSwitch Matrix will cause the AGIS Maps Action SoftSwitch matrix to appear. The AGIS User can access more detailed map media by zooming to a specific area and selecting the RQST (REQUEST) SoftSwitch.
See Figure 1

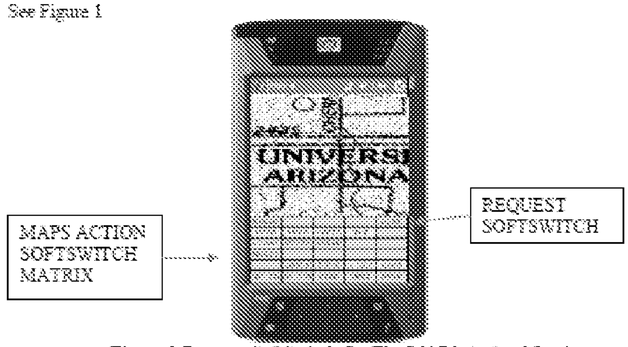


Figure 1 Request SoftSwitch On The MAPS Action Matrix

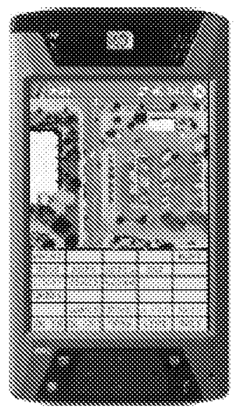
Once selected, the AGIS Handset sends a map request via the internet to the AGIS SERVER requesting the best available map image to illustrate the center point of the Handset's tactical map area.
Using the Handsets GPS coordinates, the AGIS Server searches the MAPSERVER directory and returns to the AGIS Handset four (4) appropriate aerial photographs for the correct zoom level. The AGIS SERVER's online DATA REDUCTION FILE illustrates this process. See figure 2

```
10:09:40 msg received from 192.168.0.101 type MapReq routing "*,*,*"
10:09:40 map request found 4 maps
10:09:40 send message MapRsp to "PICST [@ 192.168.0.101]
```

HANDSET REQUESTS MAPS
SERVER FINDS MAPS
SERVER SENDS MAPS

Figure 2 AGIS SERVER Online Data Reduction
(The AGIS SERVER receives the request from the AGIS hand set and returns four map files.)

Maps are transferred as data, exactly like photographs and video clips. Unlike that data, however, the Transferred Map media is automatically received and displayed on the Handsets Tactical Map Area, upon receipt. See Figure 3.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	 Figure 3 The AGIS Handset Has Displayed the Aerial Photograph on the TACTICAL MAP AREA Exhibit 16 to Beyer Declaration.
receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates;	The AGIS LifeRing product practices "receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates." The first device makes a request for a georeferenced map by interacting with the device's user interface. In the file "buttons.cpp" the function "button_actions()"[60] implements the main routine for capturing the button press event and processing based on the button pressed. For the case of the "BUTTON_IMAGE_REQUEST" a call is made to the "image_server_request()" function. In the file "display/images.cpp" the function "image_server_request()"[61] is implemented. In this function the coordinate boundaries are retrieved by a call to "coord_bounds()" and sent to the function "SendImageRequest()". In the file "coords.cpp" the function "coord_bounds()"[62] is implemented where the latitude and longitude are used to generate the values required. In the file "display/msgproc.cpp" the function "SendImageRequest()"[63] is

[60] *See, e.g.*, buttons.cpp at L1344-2365.
[61] *See, e.g.*, display/images.cpp at L752-761.
[62] *See, e.g.*, coords.cpp at L88-98.
[63] *See, e.g.*, display/msgproc.cpp at L963-996.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	implemented where a message is composed using the "DDL_MSG_MAP_REQUEST" type. The message is then processed by the "put_ddl_msg()" routine as described above and sent to the server with the call to "send_message_bulk()" as described above using the "server_ip" variable as the target for the server. The response from the server includes the map at the specified location. Furthermore, the maps utilized by AGIS were each georeferenced and represent locations in real world locations. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[64] is implemented to process received messages. The condition "DDL_MSG_MAP_REQUEST" is captured and a call is made to "process_map_request()". The function "process_map_request()"[65] is implemented in the same file where a message of type "DDL_MSG_MAP_RESPONSE" is formed and returns a map by calls to "put_ddl_msg()" and "send_message()". In the file "display/msgproc.cpp" the function "process_csif_msg()"[66] is implemented where the condition "DDL_MSG_MAP_RESPONSE" is processed by a call to "process_image_server_response()". In the file "display/images.cpp" the function "process_image_server_response()"[67] is implemented where it is logged that with the "response from image server".

[64] *See, e.g.*, server/msgproc.cpp at L377-474.
[65] *See, e.g.*, server/msgproc.cpp at L312-332.
[66] *See, e.g.*, display/msgproc.cpp at L1391-1472.
[67] *See, e.g.*, display/images.cpp at L763-768.

<table>
<tr><th>U.S. Patent No. 9,445,251</th><th>October 22, 2005 AGIS LIFERING</th></tr>
<tr><td></td><td>

MAP REQUEST

RQST

Selecting the Map SoftSwitch from The AGIS handset's Maps Function SoftSwitch Matrix will cause the AGIS Maps Action SoftSwitch matrix to appear. The AGIS User can access more detailed map media by zooming to a specific area and selecting the RQST (REQUEST) SoftSwitch.
See Figure 1

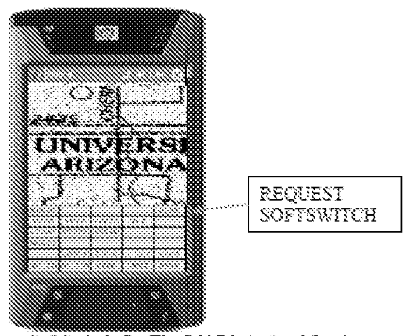


Figure 1 Request SoftSwitch On The MAPS Action Matrix

Once selected, the AGIS Handset sends a map request via the internet to the AGIS SERVER requesting the best available map image to illustrate the center point of the Handset's tactical map area.
Using the Handsets GPS coordinates, the AGIS Server searches the MAPSERVER directory and returns to the AGIS Handset four (4) appropriate aerial photographs for the correct zoom level. The AGIS SERVER's online DATA REDUCTION FILE illustrates this process. See figure 2

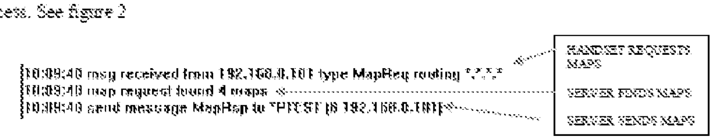
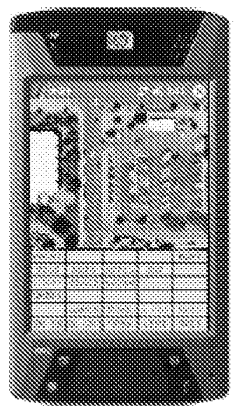
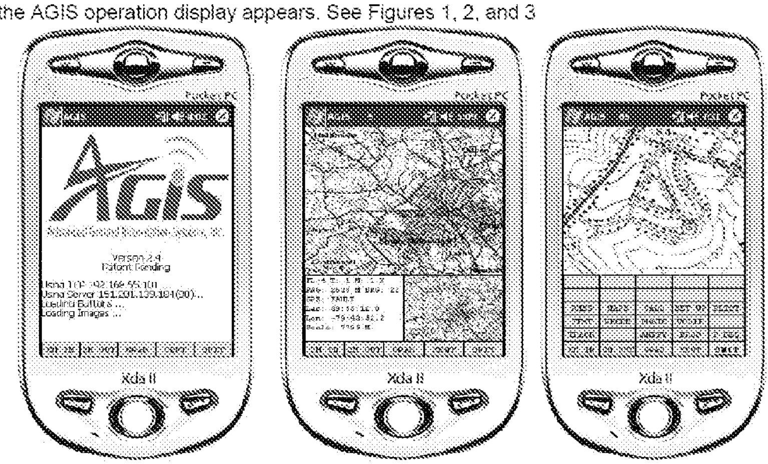


Figure 2 AGIS SERVER Online Data Reduction
(The AGIS SERVER receives the request from the AGIS hand set and returns four map files.)

Maps are transferred as data, exactly like photographs and video clips. Unlike that data, however, the Transferred Map media is automatically received and displayed on the Handsets Tactical Map Area, upon receipt. See Figure 3.

</td></tr>
</table>

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	 Figure 3 The AGIS Handset Has Displayed the Aerial Photograph on the TACTICAL MAP AREA Exhibit 16 to Beyer Declaration.
presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices; and	The AGIS LifeRing product practices "presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices." The device has an interactive display showing the locations of other devices overlayed on a georeferenced map along with its own. The display of locations and tracks are described above. As depicted above, the user client of the AGIS software includes a user interface display with a map overlayed with symbols representing other members of a group at their geographic locations on the georeferenced map.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	The AGIS application software does not affect the operation of the PDA, the Cell phone or the GPS until it is activated. When the AGIS application software is selected, the AGIS PDA Cell phone is turned on, the GPS is automatically connected to the PDA / Cell phone through a Bluetooth interface and the AGIS Logo appears. Shortly thereafter the AGIS operation display appears. See Figures 1, 2, and 3  Figure 1 Figure 2 Figure 3 Exhibit 4 to Beyer Declaration, at 4.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING

MAP REQUEST

RQST

Selecting the Map SoftSwitch from The AGIS handset's Maps Function SoftSwitch Matrix will cause the AGIS Maps Action SoftSwitch matrix to appear. The AGIS User can access more detailed map media by zooming to a specific area and selecting the RQST (REQUEST) SoftSwitch.
See Figure 1

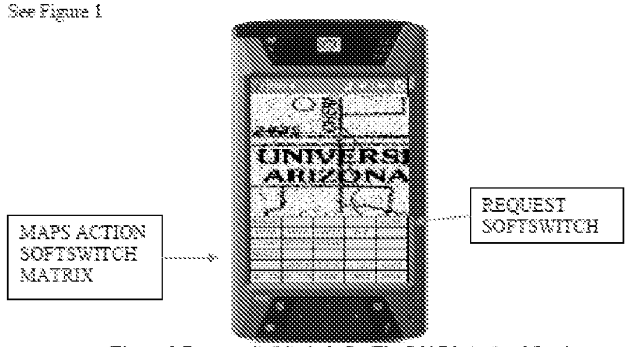


Figure 1 Request SoftSwitch On The MAPS Action Matrix

Once selected, the AGIS Handset sends a map request via the internet to the AGIS SERVER requesting the best available map image to illustrate the center point of the Handset's tactical map area.
Using the Handsets GPS coordinates, the AGIS Server searches the MAPSERVER directory and returns to the AGIS Handset four (4) appropriate aerial photographs for the correct zoom level. The AGIS SERVER's online DATA REDUCTION FILE illustrates this process. See figure 2

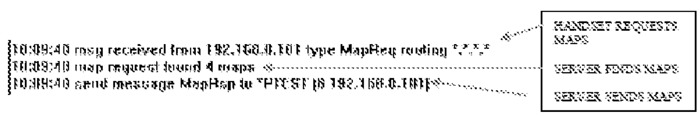
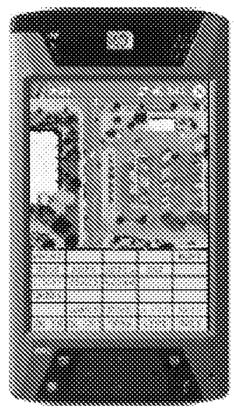


Figure 2 AGIS SERVER Online Data Reduction
(The AGIS SERVER receives the request from the AGIS hand set and returns four map files.)

Maps are transferred as data, exactly like photographs and video clips. Unlike that data, however, the Transferred Map media is automatically received and displayed on the Handsets Tactical Map Area, upon receipt. See Figure 3.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
	 Figure 3 The AGIS Handset Has Displayed the Aerial Photograph on the TACTICAL MAP AREA Exhibit 16 to Beyer Declaration. In the file "display/display.cpp" the main method "WndProc()"[68] where display events are captured and processed, the case for "WM_PAINT" where the display is drawn is implemented. When a "WM_PAINT" event occurs calls are made to "refresh_tact()", "refresh_stat()" and "refresh_inset()" passing the display window handle where the map, locations and other information gets displayed. In the file "tact.cpp" the function "refresh_tact()"[69] is implemented. There are three types of this function, two with an input handle and one without. The one without the handle first retrieves the current handle and then calls the "refresh_tact()" function version with the handle. In this function, among other features, the system draws the map by a call to "ww_draw_map()" and draws locations by a call to "track_draw()". A special "refresh_tact()" implementation only draws the tactical area which is documented to correspond to the display's WM_PAINT event and only accepts a window handle of type HWND. The function "refresh_tact()" is used throughout the code to signal an update to the view within "buttons.cpp", "display.cpp", "images.cpp", "inset.cpp",

[68] *See, e.g.*, display/display.cpp at L584-969.
[69] *See, e.g.*, display/tact.cpp at L200-306.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"maps.cpp", "msgproc.cpp", "track.cpp" and others to signal the update of the graphical window. In the file "maps.cpp" the function "ww_draw_map()"[70] is implemented. The function utilizes georeferenced latitude and longitude information to position the map using variables "urlat", "urlon", "lllat" and "lllon" for the upper right and lower left latitude and longitude. The variable array "ww_map" is used to hold the map data and is read in the same file using the function "ww_read_map()"[71] from local files. The function "ww_read_maps()"[72] makes individual calls to "ww_read_map()" to read the "world.pnt", "plit.pnt", "water.pnt" and a local map through invocation of the "local_read_map()" function pointed to by the "test user.ump" file. The "ww_read_maps()" function is called in the file "display/display.cpp" by the function "InitInstance()"[73] when the system has started. In the file "stat.cpp" the function "refresh_stat()"[74] is implemented. There are two types of this function, one with an input handle and one without. The one without the handle first retrieves the current handle and then calls the "refresh_stat()" function version with the handle. In this function, among other features, the system "refresh status window" by a call to "stat_template_refresh()". There are other methods that the system implements such as "show_status_window()"[75] and the main window process in the "WndStatProc()"[76] callback function "WM_PAINT" event which calls the "refresh_stat()" and in turn "stat_template_refresh()". Upon creation of the stat view by use of the function "create_stat()"[77] to create the graphical interface followed by a call to the "stat_template_refresh()". Furthermore, the function "refresh_stat()" is used throughout the code within "buttons.cpp", "display.cpp", "gps.cpp", "msgproc.cpp", "partlist.cpp" to signal the update of the stat window. The function "stat_template_refresh()"[78] is implemented with a call to "status_hook()". The function

[70] *See, e.g.*, maps.cpp at L213-276.
[71] *See, e.g.*, maps.cpp at L69-153.
[72] *See, e.g.*, maps.cpp at L726-732.
[73] *See, e.g.*, display/display.cpp at L268-581.
[74] *See, e.g.*, stat.cpp at L321-371.
[75] *See, e.g.*, stat.cpp at L387-403.
[76] *See, e.g.*, stat.cpp at L39-115.
[77] *See, e.g.*, stat.cpp at L119-166.
[78] *See, e.g.*, stat.cpp at L296-317.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"status_hook()"[79] is implemented where various information gets presented such as track information, current latitude, current longitude, GPS status amongst others. In the file "inset.cpp" the function "refresh_inset()"[80] is implemented. There are two types of this function, one with an input handle and one without. Both versions retrieve the Inset window handle through either a global handle or passed in handle and make a call to the "refresh_inset_window()" function. The function "refresh_inset_window()"[81] is implemented in the same file with calls made to "refresh_inset_map()", "refresh_inset_gps()" and "refresh_inset_trackamp()" based on the value held in the "inset_mode" variable set to "INSETWIN_MAP_USE", "INSETWIN_GPS_USE" or "INSETWIN_TRACKAMP_USE". The function "refresh_inset_map()"[82] is implemented in the same file and the georeferenced latitude and longitude values such as "inset_maplllon" or "inset_mapurlat" that are used to draw the map with the track information used from the "track_file" variable. The function draws the map using the system "Bit_Blt()" or call to "ww_draw_image_inset() and then calls the "inset_track_draw()" function which then draws all tracks. The function "ww_draw_image_inset()"[83] is implemented in the "display/images.cpp" and utilizes georeferenced latitude and longitude values. The function "inset_track_draw()"[84] is documented that to "plot all tracks as dots in inset window" and utilizes a loop across all tracks to draw all tracks. The function "refresh_inset_gps()"[85] is implemented in the same file and primarily provides information about the gps satelites and their respective signal strengths. The function "refresh_inset_trackamp()"[86] updates the amplified track mode equipped with its own series of functions including messaging, location and amplified views.

[79] *See, e.g.*, stat.cpp at L207-292
[80] *See, e.g.*, inset.cpp at L708-737.
[81] *See, e.g.*, inset.cpp at L684-705.
[82] *See, e.g.*, inset.cpp at L285-393.
[83] *See, e.g.*, display/images.cpp at L534-596.
[84] *See, e.g.*, inset.cpp at L220-282.
[85] *See, e.g.*, inset.cpp at L395-512.
[86] *See, e.g.*, inset.cpp at L535-657.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
and identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first device does not have access to respective Internet Protocol addresses of the second devices.	The AGIS LifeRing product practices "identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first device does not have access to respective Internet Protocol addresses of the second devices." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[87] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[88] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[89] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[90] and "num_on_que_list()"[91] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file

[87] *See, e.g.*, buttons.cpp at L1344-2365.
[88] *See, e.g.*, display/ftext.cpp at L176-244.
[89] *See, e.g.*, netselect.cpp at L152-221.
[90] *See, e.g.*, quelist.cpp at L16-22.
[91] *See, e.g.*, quelist.cpp at L60-71.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"display/msgproc.cpp" the function "send_ftext_msg()"[92] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[93] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[94] and subsequently to "form_net_line()"[95] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[96] and subsequently "send_net_msg_participants()"[97] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[98] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[99] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[100] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the

[92] *See, e.g.*, display/msgproc.cpp at L328-399.
[93] *See, e.g.*, netselect.cpp at L391-492.
[94] *See, e.g.*, netselect.cpp at L373-384.
[95] *See, e.g.*, netselect.cpp at L352-370.
[96] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[97] *See, e.g.*, display/msgproc.cpp at L828-922.
[98] *See, e.g.*, photo.cpp at L189-292.
[99] *See, e.g.*, server/msgproc.cpp at L377-474.
[100] *See, e.g.*, server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
2. The method of claim 1, wherein the data includes a short message service message, a text message, an image, or a video.	The AGIS LifeRing product practices "wherein the data includes a short message service message, a text message, an image, or a video." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[101] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[102] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[103] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[104] and "num_on_que_list()"[105] functions where the "que_list" array is maintained. Once the message is compiled the

[101] *See, e.g.*, buttons.cpp at L1344-2365.
[102] *See, e.g.*, display/ftext.cpp at L176-244.
[103] *See, e.g.*, netselect.cpp at L152-221.
[104] *See, e.g.*, quelist.cpp at L16-22.
[105] *See, e.g.*, quelist.cpp at L60-71.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	message is send via “send_ftext_msg()” function. In the file “display/msgproc.cpp” the function “send_ftext_msg()”[106] is implemented where the free text message is transmitted directly or through a server as described above through the use of the “send_message_direct()” and “send_message_bulk()” using the “server_ip” variable. Similarly for the case of “BUTTON_NET” a call is made to “create_netlist()”. In the file netselect.cpp” the function “create_netlist()”[107] is implemented where the display is configured for net communication. The call to “send_to_display_net_list()”[108] and subsequently to “form_net_line()”[109] forms the list of participants to receive the communication. In the file “display/msgproc.cpp” the function “process_net_msg()”[110] and subsequently “send_net_msg_participants()”[111] the message is transmitted directly or through a server as described above through the use of the “send_message_direct()” and “send_message_bulk()” using the “server_ip” variable. For the case of “BUTTON_MSG_PHOTO” and “BUTTON_ MSG_VIDEO” the call to “create_photo()” is made. In the file “photo.cpp” the function “create_photo()”[112] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function “build_sending_text()” similar to the free text mode above. In the file “server/msgproc.cpp” the function “process_csif_ddl_msg()”[113] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the “pass_thru_msg()” function. In the file “server/msgproc.cpp” the function “pass_thru_msg()”[114] is implemented where the “trouting”

[106] *See, e.g.*, display/msgproc.cpp at L328-399.
[107] *See, e.g.*, netselect.cpp at L391-492.
[108] *See, e.g.*, netselect.cpp at L373-384.
[109] *See, e.g.*, netselect.cpp at L352-370.
[110] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[111] *See, e.g.*, display/msgproc.cpp at L828-922.
[112] *See, e.g.*, photo.cpp at L189-292.
[113] *See, e.g.*, server/msgproc.cpp at L377-474.
[114] *See, e.g.*, server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
3. The method of claim 1, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC).	The AGIS LifeRing product practices "wherein the first device is a personal digital assistant (PDA) or a personal computer (PC)." The AGIS LifeRing product includes systems and methods for executing an application on PDA/cell phones, now referred to as smartphone devices. See, e.g., "call.cpp" references to "phone" and "other phones."
4. The method of claim 1, wherein the second map is a satellite image.	The AGIS LifeRing product practices "wherein the second map is a satellite image." See, e.g., "satelliteID" in "csiftranscode.h" and "display\gps.cpp."
5. The method of claim 1, further comprising sending, by the first device, updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information comprising a previous location of the first device,	The AGIS LifeRing product practices "sending, by the first device, updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information comprising a previous location of the first device, displacement of the first device by a predetermined distance relative to a previous location of the first device, or both." The AGIS software captures location information on the device through a serial communication link configured to continuously process GPS information messages. Messages such as location, precision and constellation are captured on an independent thread and when a valid message is received the thread triggers an event indicating the presence of a new message. The software then sends the new location to others through either a server or direct connection. The AGIS operator can also assign AGIS reporting intervals by time period and distance traveled since the last report. In Figure 26 the time interval for reporting has been set to every 5 minutes and distance traveled that will cause a report (even if the 5 minute interval has not lapsed) is set to 1 mile. Exhibit 4 at 11.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
displacement of the first device by a predetermined distance relative to a previous location of the first device, or both.	The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function "initGPS()"[115] is implemented where the variable comport is as the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[116] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[117] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[118] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[119] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[120] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[121] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[122] is implemented where the queued GPS messages are process incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[123] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three

[115] *See, e.g.*, csif/csif.cpp at L126-148.
[116] *See, e.g.*, csif/gpslistener.cpp at L36-51.
[117] *See, e.g.*, csif/gpslistener.cpp at L101-235.
[118] *See, e.g.*, csif/gpslistener.cpp at L158.
[119] *See, e.g.*, csif/gpslistener.cpp at L76-99.
[120] *See, e.g.*, display/display.cpp at L274-581.
[121] *See, e.g.*, display/display.cpp at L584-969.
[122] *See, e.g.*, display/gps.cpp at L346-365.
[123] *See, e.g.*, display/gps.cpp at L292-336.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[124] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" variables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[125] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[126] and are implemented in the function "button_actions()"[127] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[128] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a

[124] *See, e.g.*, display/gps.cpp at L77-126.
[125] *See, e.g.*, display/display.cpp at L1015-1037 and L1076-1115.
[126] *See, e.g.*, display/buttons.cpp at L594-598 and L1344-1346.
[127] *See, e.g.*, display/buttons.cpp at L1344-2365.
[128] *See, e.g.*, display/msgproc.cpp at L170-325.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[129] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. The "send_track_msg()"function is used under the condition where "add_track()"[130] and "update_track_id()"[131] functions in the "display/track.cpp" is invoked. The "update_track_id()" is used in user interface as views in the "buttons.cpp" file in the main routine where button actions are implemented "button_actions()" as discussed above where the case for "BUTTON_TRACK_ID_UNKN", "BUTTON_TRACK_ID_FRND" and "BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[132] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent the "DDL_MSG_OWN_POISITION" case where the message is passed through by a call to "pass_thru_msg()" function. The file "server/server.cpp" implements the main function "WndProc()"[133] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through.
6. The method of claim 1, further	The AGIS LifeRing product practices "identifying second user interaction with the interactive display selecting at least one of the user-

[129] *See, e.g.*, display/msgproc.cpp at L615-682.
[130] *See, e.g.*, display/track.cpp at L373-477.
[131] *See, e.g.*, display/track.cpp at L482-510.
[132] *See, e.g.*, server/msgproc.cpp at L377-474.
[133] *See, e.g.*, server/server.cpp at L228-386.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
comprising identifying second user interaction with the interactive display selecting at least one of the user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device.	selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[134] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[135] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[136] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[137] and "num_on_que_list()"[138] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[139] is implemented where the free text message is transmitted directly or through a server as described above through the use of the

[134] *See, e.g.*, buttons.cpp at L1344-2365.
[135] *See, e.g.*, display/ftext.cpp at L176-244.
[136] *See, e.g.*, netselect.cpp at L152-221.
[137] *See, e.g.*, quelist.cpp at L16-22.
[138] *See, e.g.*, quelist.cpp at L60-71.
[139] *See, e.g.*, display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[140] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[141] and subsequently to "form_net_line()"[142] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[143] and subsequently "send_net_msg_participants()"[144] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[145] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. See also BUTTON_CALL and BUTTON_CCALL in buttons.h. and buttons.cpp for phone calls and conferences. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[146] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[147] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the

[140] *See, e.g.*, netselect.cpp at L391-492.
[141] *See, e.g.*, netselect.cpp at L373-384.
[142] *See, e.g.*, netselect.cpp at L352-370.
[143] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[144] *See, e.g.*, display/msgproc.cpp at L828-922.
[145] *See, e.g.*, photo.cpp at L189-292.
[146] *See, e.g.*, server/msgproc.cpp at L377-474.
[147] *See, e.g.*, server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
7. The method of claim 1, wherein the message from the second device is a Short Message Service (SMS) message or a text message.	The AGIS LifeRing product practices "wherein the message from the second device is a Short Message Service (SMS) message or a text message." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[148] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[149] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[150] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the

[148] *See, e.g.*, buttons.cpp at L1344-2365.
[149] *See, e.g.*, display/ftext.cpp at L176-244.
[150] *See, e.g.*, netselect.cpp at L152-221.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"on_que_list()"[151] and "num_on_que_list()"[152] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[153] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[154] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[155] and subsequently to "form_net_line()"[156] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[157] and subsequently "send_net_msg_participants()"[158] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[159] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[160] is implemented as described above with various conditions. For the default condition catching messages for

[151] *See, e.g.*, quelist.cpp at L16-22.
[152] *See, e.g.*, quelist.cpp at L60-71.
[153] *See, e.g.*, display/msgproc.cpp at L328-399.
[154] *See, e.g.*, netselect.cpp at L391-492.
[155] *See, e.g.*, netselect.cpp at L373-384.
[156] *See, e.g.*, netselect.cpp at L352-370.
[157] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[158] *See, e.g.*, display/msgproc.cpp at L828-922.
[159] *See, e.g.*, photo.cpp at L189-292.
[160] *See, e.g.*, server/msgproc.cpp at L377-474.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[161] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
8. The method of claim 1, wherein participating in the group further includes sending first status information to the server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second	The AGIS LifeRing product practices "wherein participating in the group further includes sending first status information to the server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second location information comprising a plurality of battery levels of the respective plurality of second devices included in the group, a plurality of signal strengths of wireless signals of the respective plurality of second devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a combination thereof." The device utilizes a GPS receiver to locate a first device and the first device sends its location to the server using IP. In the file "gpslistener.cpp" the "start()"[162] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[163] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[164] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[165] calls the

[161] *See, e.g.*, server/msgproc.cpp at L 51-61.
[162] *See, e.g.*, csif/gpslistener.cpp at L36-51.
[163] *See, e.g.*, csif/gpslistener.cpp at L101-235.
[164] *See, e.g.*, csif/gpslistener.cpp at L158.
[165] *See, e.g.*, csif/gpslistener.cpp at L76-99.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
location information comprising a plurality of battery levels of the respective plurality of second devices included in the group, a plurality of signal strengths of wireless signals of the respective plurality of second devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a combination thereof.	"enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[166] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[167] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[168] is implemented where the queued GPS messages are process incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[169] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[170] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[171] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group

[166] *See, e.g.*, display/display.cpp at L274-581.
[167] *See, e.g.*, display/display.cpp at L584-969.
[168] *See, e.g.*, display/gps.cpp at L346-365.
[169] *See, e.g.*, display/gps.cpp at L292-336.
[170] *See, e.g.*, display/gps.cpp at L77-126.
[171] *See, e.g.*, display/display.cpp at L1015-1037 and L1076-1115.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[172] and are implemented in the function "button_actions()"[173] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[174] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[175] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. In the file "display/msgproc.cpp" the functions "send_message_bulk()" and "send_message_direct()"[176] are implemented where both have IP transmit capability with the addition of "send_message_direct()" also having SMS transmission capability. Within the IP transmit capability, both functions make calls to the "sendTCPMessage()" function. In the

[172] *See, e.g.*, display/buttons.cpp at L594-598 and L1344-1346.
[173] *See, e.g.*, display/buttons.cpp at L1344-2365.
[174] *See, e.g.*, display/msgproc.cpp at L170-325.
[175] *See, e.g.*, display/msgproc.cpp at L615-682.
[176] *See, e.g.*, display/msgproc.cpp at L47-127.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	file “csif.cpp” the “sendTCPMessage()”[177] function is implemented where it utilizes the “tcpConnection” variable to call the “enqueueForSending()” function. The “tcpConnection” variable holds the necessary information to complete an IP based transmission. In the file “csif.cpp” the function “initTCP()”[178] is implemented where the “tcpConnection” variable is instantiated. In this function a network management object “nm” from the “NetworkMgmt” class is used to call the “connectIPNetwork()” function and implement an “ipMonitor” variable to monitor the IP network. Also the “tcpConnection” variable is set to a new object by a call to “TCPServerConnection()” function. See “getGPSStatus” and getDDLStatus The devices process battery levels to populate a navbar display that includes the display of the battery levels that are transmitted to the server for display and sharing with other devices. In the file “display.cpp” the function “InitInstance()”[179] and the function “WndProc()”[180] are implemented for display initialization and run time processing where calls to “display_nav_bar()” and “timed_display_nav_bar()” are made respectively. Also, there are calls to “check_ip_address()” in both initialization and run time processing where subsequent uses of navbar data is present. In the file “navbar.cpp” the function “timed_display_nav_bar()”[181] is implemented where it calls the function “display_nav_bar()” . The function “display_nav_bar()”[182] is also implemented where it is documented to “display all indicators from previous update”. The function stores the result of the function “battery_check()” in the variable “bat_ret” and builds the “str” array at location [10] to empty for full battery, “b” for very low level and “B” for low level battery. The “str” variable is then copied to the “outstr” array and is used to display the result with calls to either “SHSetNavBarText()” or “system_readout()” based on the type of processor running the code where it stores the navbar data in the system memory.

[177] *See, e.g.*, csif.cpp at L378-386.
[178] *See, e.g.*, csif.cpp at L150-204.
[179] *See, e.g.*, display.cpp at L268-581.
[180] *See, e.g.*, display.cpp at L584-969.
[181] *See, e.g.*, navbar.cpp at L163-173.
[182] *See, e.g.*, navbar.cpp at L69-161.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "display.cpp" the function "check_ip_address()"[183] is implemented for two different cases of "PPC2003" or else. The case of "PPC2003" indicative of a mobile handset having battery levels the call to "display_nav_bar()" is made followed by "SendOwnPosString()" which is described above. The code is also conditioned on the use of "SMS" or not, indicative of a server. The position and battery level data are used as part of a message to be sent to either the server or other devices directly.
9. The method of claim 1, wherein the first device is a smart phone.	The AGIS LifeRing product practices "wherein the first device is a smart phone." The AGIS LifeRing product includes systems and methods for executing an application on PDA/cell phones, now referred to as smartphone devices. See, e.g., "call.cpp" references to "phone" and "other phones."
10. The method of claim 1, further comprising: with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality of second devices; and based on transmitting the group identifier associated with the second group, participating in the second group, wherein	The AGIS LifeRing product practices "with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality of second devices; and based on transmitting the group identifier associated with the second group, participating in the second group, wherein participating in the second group includes receiving third location information from the server, the third location information comprising a plurality of locations of the respective second plurality of second devices included in the second group." The file "display/msgproc.cpp" implements the function "process_net_msg()"[184] where the logging function indicates "rcvd net msg" indicative of receiving a network message. The "nmsg" variable of type "NETUSERS" which is input to this function is tested against its member variable requestType which is specified in the file "csiftranscode.h"[185] and states to have "J)oining or R)equest current participants". The "nmsg.requestType" is then branched for cases of "I" for initial joining, "J" for joining, "R" for requesting and "X" for exiting the network. For the case of "J" for joining, the function makes

[183] *See, e.g.*, display.cpp at L1014-1117.
[184] *See, e.g.*, display/msgproc.cpp at L1234-1292..
[185] *See, e.g.*, csiftranscode/csiftranscode.h at L254-263.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
participating in the second group includes receiving third location information from the server, the third location information comprising a plurality of locations of the respective second plurality of second devices included in the second group.	a call to the "update_address_book()" function with sender information as the input parameters. The file "initfiles.cpp" implements the "update_address_book()"[186] where the sender information is used to find its index in the address book by a call to "find_addr_number()". The return from this call is used to update the array variable "csif_addr[]" by either adding the address if it does not exist indicative of a return value of -1 or by copying the "ipaddr" into a struct member of the indexed variable. The "csif_addr[]" array is an array of type "ADDRESS_ENTRY"[187] which is specified in the file "csiftranscode.h"[188] The members of this struct includes "name", "number", "group", "type", "level" and "ipaddr". The marking of the "csif_addr[]" variable array constitutes the joining of a group. The server further provides additional functionality in file "server/msgproc.cpp" implements the function "process_net_msg()"[189] where the logging function indicates "rcvd net msg" indicative of receiving a network message. The variables "from" of type character string and "nmsg" of type "NETUSERS" as described above is input to this function. Similar to described above, the "nmsg.requestType" is then branched for cases of "J" and not "J" described to mean of type "R" are handled. For the case of joining a group where "nmsg.requestType" equals "J", the function "update_address_book()" is called with the sender and receiver information and returns with an index into the address book array "csif_addr" as discussed above. Then the logging command indicates that the "address book [address book index] updated with ipaddr [address book ip address]" and writes "user [name] joined network" to the display. Followed by two calls to "send_net_msg_participants()" using a wildcard designator "-1" indicating everything in the respective field is to be sent. The first call sends everything in the table to the currently added user. The second call sends to everyone the currently added user. In the same file the function "send_net_msg_participants()"[190] documents that the user of

[186] *See, e.g.*, display/initfiles.cpp at L701-721.
[187] *See, e.g.*, display/initfiles.h at L 35
[188] *See, e.g.*, csiftranscode/csiftranscode.h at L45-54.
[189] *See, e.g.*, server/msgproc.cpp at L235-293.
[190] *See, e.g.*, server/msgproc.cpp at L108-191.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"-1" in the first field indicates a "send to all" and the use of "-1" in the second field indicates a "send all net participants" condition. Similarly, the server further provides functionality to support the server side function similar to the client side in the file "server/initfiles.cpp" by implementing the function "update_address_book()"[191] where similar to the description above the call to "find_addr_number()" function retrieves an address index. If the address does not exist indicative of a return index value of "-1" calls the function "add_address_book()" otherwise updates the "csif_addr" array with the ip address input to the function. In the same file the function "add_address_book()"[192] is implemented where the "csif_addr" array is populated with the "name", "number", "group", "ipaddr" and a "isValid" member variable.
11. The method of claim 1, wherein the data includes a voice recording.	The AGIS LifeRing product practices "wherein the data includes a voice recording." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[193] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[194] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the

[191] *See, e.g.*, server/initfiles.cpp at L194-211.
[192] *See, e.g.*, server/initfiles.cpp at L163-191.
[193] *See, e.g.*, buttons.cpp at L1344-2365.
[194] *See, e.g.*, display/ftext.cpp at L176-244.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"build_sending_text()"[195] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[196] and "num_on_que_list()"[197] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[198] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[199] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[200] and subsequently to "form_net_line()"[201] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[202] and subsequently "send_net_msg_participants()"[203] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[204] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above.

[195] *See, e.g.*, netselect.cpp at L152-221.
[196] *See, e.g.*, quelist.cpp at L16-22.
[197] *See, e.g.*, quelist.cpp at L60-71.
[198] *See, e.g.*, display/msgproc.cpp at L328-399.
[199] *See, e.g.*, netselect.cpp at L391-492.
[200] *See, e.g.*, netselect.cpp at L373-384.
[201] *See, e.g.*, netselect.cpp at L352-370.
[202] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[203] *See, e.g.*, display/msgproc.cpp at L828-922.
[204] *See, e.g.*, photo.cpp at L189-292.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[205] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[206] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file. Once the devices have formed a group they share their location with each others. The AGIS software captures location information on the device through a serial communication link configured to continuously process GPS information messages. Messages such as location, precision and constellation are captured on an independent thread and when a valid message is received the thread triggers an event indicating the presence of a new message. The software then sends the new location to others through either a server or direct connection. The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function "initGPS()"[207] is implemented where the variable comport is as the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[208] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[209] and uses a loop to scan the comport documented as "read in a message from serial port, will only

[205] *See, e.g.*, server/msgproc.cpp at L377-474.
[206] *See, e.g.*, server/msgproc.cpp at L 51-61.
[207] *See, e.g.*, csif/csif.cpp at L126-148.
[208] *See, e.g.*, csif/gpslistener.cpp at L36-51.
[209] *See, e.g.*, csif/gpslistener.cpp at L101-235.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
	wait for 1 second for data"[210] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[211] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[212] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[213] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[214] is implemented where the queued GPS messages are process incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[215] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[216] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct".

[210] *See, e.g.*, csif/gpslistener.cpp at L158.
[211] *See, e.g.*, csif/gpslistener.cpp at L76-99.
[212] *See, e.g.*, display/display.cpp at L274-581.
[213] *See, e.g.*, display/display.cpp at L584-969.
[214] *See, e.g.*, display/gps.cpp at L346-365.
[215] *See, e.g.*, display/gps.cpp at L292-336.
[216] *See, e.g.*, display/gps.cpp at L77-126.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "display/display.cpp" the function "check_ip_address()"[217] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[218] and are implemented in the function "button_actions()"[219] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[220] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[221] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message.

[217] *See, e.g.*, display/display.cpp at L1015-1037 and L1076-1115.
[218] *See, e.g.*, display/buttons.cpp at L594-598 and L1344-1346.
[219] *See, e.g.*, display/buttons.cpp at L1344-2365.
[220] *See, e.g.*, display/msgproc.cpp at L170-325.
[221] *See, e.g.*, display/msgproc.cpp at L615-682.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	The "send_track_msg()"function is used under the condition where "add_track()"[222] and "update_track_id()"[223] functions in the "display/track.cpp" is invoked. The "update_track_id()" is used in user interface as views in the "buttons.cpp" file in the main routine where button actions are implemented "button_actions()" as discussed above where the case for "BUTTON_TRACK_ID_UNKN", "BUTTON_TRACK_ID_FRND" and "BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[224] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent the "DDL_MSG_OWN_POISITION" case where the message is passed through by a call to "pass_thru_msg()" function. The file "server/server.cpp" implements the main function "WndProc()"[225] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through.
12. The method of claim 1, further comprising: using a Global Positioning Satellite (GPS) receiver of the first device to obtain data indicative of the location of the first device, wherein sending the first location information	The AGIS LifeRing product practices "using a Global Positioning Satellite (GPS) receiver of the first device to obtain data indicative of the location of the first device, wherein sending the first location information to the server comprises using the Internet Protocol (IP) to send the first location information to the server." The device utilizes a GPS receiver to locate a first device and the first device sends it's location to the server using IP. In the file "gpslistener.cpp" the "start()"[226] method is implemented where a thread is created for processing in the background. The thread

[222] *See, e.g.*, display/track.cpp at L373-477.
[223] *See, e.g.*, display/track.cpp at L482-510.
[224] *See, e.g.*, server/msgproc.cpp at L377-474.
[225] *See, e.g.*, server/server.cpp at L228-386.
[226] *See, e.g.*, csif/gpslistener.cpp at L36-51.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
to the server comprises using the Internet Protocol (IP) to send the first location information to the server.	performs the function "process()"[227] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[228] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[229] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[230] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[231] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[232] is implemented where the queued GPS messages are process incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[233] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[234] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct".

[227] *See, e.g.*, csif/gpslistener.cpp at L101-235.
[228] *See, e.g.*, csif/gpslistener.cpp at L158.
[229] *See, e.g.*, csif/gpslistener.cpp at L76-99.
[230] *See, e.g.*, display/display.cpp at L274-581.
[231] *See, e.g.*, display/display.cpp at L584-969.
[232] *See, e.g.*, display/gps.cpp at L346-365.
[233] *See, e.g.*, display/gps.cpp at L292-336.
[234] *See, e.g.*, display/gps.cpp at L77-126.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "display/display.cpp" the function "check_ip_address()"[235] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[236] and are implemented in the function "button_actions()"[237] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[238] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[239] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message.

[235] *See, e.g.*, display/display.cpp at L1015-1037 and L1076-1115.
[236] *See, e.g.*, display/buttons.cpp at L594-598 and L1344-1346.
[237] *See, e.g.*, display/buttons.cpp at L1344-2365.
[238] *See, e.g.*, display/msgproc.cpp at L170-325.
[239] *See, e.g.*, display/msgproc.cpp at L615-682.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "display/msgproc.cpp" the functions "send_message_bulk()" and "send_message_direct()"[240] are implemented where both have IP transmit capability with the addition of "send_message_direct()" also having SMS transmission capability. Within the IP transmit capability, both functions make calls to the "sendTCPMessage()" function. In the file "csif.cpp" the "sendTCPMessage()"[241] function is implemented where it utilizes the "tcpConnection" variable to call the "enqueueForSending()" function. The "tcpConnection" variable holds the necessary information to complete an IP based transmission. In the file "csif.cpp" the function "initTCP()"[242] is implemented where the "tcpConnection" variable is instantiated. In this function a network management object "nm" from the "NetworkMgmt" class is used to call the "connectIPNetwork()" function and implement an "ipMonitor" variable to monitor the IP network. Also the "tcpConnection" variable is set to a new object by a call to "TCPServerConnection()" function.
13. The method of claim 1, further comprising identifying, by the first device, user interaction with the display selecting a particular user-selectable symbol positioned on the second georeferenced map and corresponding to a particular second device, wherein identifying the user interaction selecting the particular user-selectable symbol	The AGIS LifeRing product practices "identifying, by the first device, user interaction with the display selecting a particular user-selectable symbol positioned on the second georeferenced map and corresponding to a particular second device, wherein identifying the user interaction selecting the particular user-selectable symbol comprises: detecting user selection of a portion of the interactive display corresponding to a position on the second georeferenced map; based at least in part on coordinates of the selected position on the second georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the selected position on the second georeferenced map; and identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients.

[240] *See, e.g.*, display/msgproc.cpp at L47-127.
[241] *See, e.g.*, csif.cpp at L378-386.
[242] *See, e.g.*, csif.cpp at L150-204.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
comprises: detecting user selection of a portion of the interactive display corresponding to a position on the second georeferenced map; based at least in part on coordinates of the selected position on the second georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the selected position on the second georeferenced map; and identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates	In the file "buttons.cpp" the function "button_actions()"[243] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[244] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[245] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[246] and "num_on_que_list()"[247] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[248] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[249] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[250] and subsequently to "form_net_line()"[251] forms the list of participants to

[243] *See, e.g.*, buttons.cpp at L1344-2365.
[244] *See, e.g.*, display/ftext.cpp at L176-244.
[245] *See, e.g.*, netselect.cpp at L152-221.
[246] *See, e.g.*, quelist.cpp at L16-22.
[247] *See, e.g.*, quelist.cpp at L60-71.
[248] *See, e.g.*, display/msgproc.cpp at L328-399.
[249] *See, e.g.*, netselect.cpp at L391-492.
[250] *See, e.g.*, netselect.cpp at L373-384.
[251] *See, e.g.*, netselect.cpp at L352-370.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
represented by the selected position.	receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[252] and subsequently "send_net_msg_participants()"[253] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[254] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[255] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[256] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
14. The method of claim 13, wherein identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the	The AGIS LifeRing product practices "wherein identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position comprises: searching a database of entities for an entity located nearest to the spatial coordinates represented by the selected position, wherein the entities represented by data in the database include the second devices, wherein the database data include locations of the respective entities, and wherein the database is searchable by location; and based on a result of searching the database, identifying the particular second device as the

[252] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[253] *See, e.g.*, display/msgproc.cpp at L828-922.
[254] *See, e.g.*, photo.cpp at L189-292.
[255] *See, e.g.*, server/msgproc.cpp at L377-474.
[256] *See, e.g.*, server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
selected position comprises: searching a database of entities for an entity located nearest to the spatial coordinates represented by the selected position, wherein the entities represented by data in the database include the second devices, wherein the database data include locations of the respective entities, and wherein the database is searchable by location; and based on a result of searching the database, identifying the particular second device as the entity located nearest to the spatial coordinates represented by the selected position, wherein the particular user-selectable symbol corresponds to the	entity located nearest to the spatial coordinates represented by the selected position, wherein the particular user-selectable symbol corresponds to the particular second device." The device utilizes buttons on the display limited to other entities in a latitude and longitude limited space to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "tact.cpp" the function "refresh_tact()"[257] is implemented to draw the tactical display including when some change to location or extent of the display is made. In this function a call is made to the function "track_draw()" to draw the extent of the tracks. In the file "tact.cpp" the function "track_draw()"[258] is implemented where based on a latitude and longitude value position of a track the function "screen_clip_check()" is called and if the entity is not on the display that the processing loop exits the loop not drawing the track leaving only tracks corresponding to other entities that are close to a selected position is drawn and subsequently the user can interact with. The retrieval of the symbols from the server is performed for the displayed map location. In the file "buttons.cpp" the function "button_actions()"[259] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[260] is implemented where the display is configured to receive user interface

[257] *See, e.g.*, tact.cpp at L200-277.
[258] *See, e.g.*, tact.cpp at L557-742.
[259] *See, e.g.*, buttons.cpp at L1344-2365.
[260] *See, e.g.*, display/ftext.cpp at L176-244.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
particular second device.	for communicating through free text. The call to “build_sending_text()” builds the receiving participants and in file “netselect.cpp” the “build_sending_text()”[261] function is implemented. List of participants is organized in the “indiv” array or the que_list based on the usage of the build and que_list is used via the “on_que_list()” and “num_on_que_list()” functions. The file “quelist.cpp” implements the “on_que_list()”[262] and “num_on_que_list()”[263] functions where the “que_list” array is maintained. Once the message is compiled the message is send via “send_ftext_msg()” function. In the file “display/msgproc.cpp” the function “send_ftext_msg()”[264] is implemented where the free text message is transmitted directly or through a server as described above through the use of the “send_message_direct()” and “send_message_bulk()” using the “server_ip” variable. Similarly for the case of “BUTTON_NET” a call is made to “create_netlist()”. In the file netselect.cpp” the function “create_netlist()”[265] is implemented where the display is configured for net communication. The call to “send_to_display_net_list()”[266] and subsequently to “form_net_line()”[267] forms the list of participants to receive the communication. In the file “display/msgproc.cpp” the function “process_net_msg()”[268] and subsequently “send_net_msg_participants()”[269] the message is transmitted directly or through a server as described above through the use of the “send_message_direct()” and “send_message_bulk()” using the “server_ip” variable. For the case of “BUTTON_MSG_PHOTO” and “BUTTON_ MSG_VIDEO” the call to “create_photo()” is made. In the file “photo.cpp” the function “create_photo()”[270] is implemented where the directly listing of either photo or video files are gathered and displayed

[261] *See, e.g.*, netselect.cpp at L152-221.
[262] *See, e.g.*, quelist.cpp at L16-22.
[263] *See, e.g.*, quelist.cpp at L60-71.
[264] *See, e.g.*, display/msgproc.cpp at L328-399.
[265] *See, e.g.*, netselect.cpp at L391-492.
[266] *See, e.g.*, netselect.cpp at L373-384.
[267] *See, e.g.*, netselect.cpp at L352-370.
[268] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[269] *See, e.g.*, display/msgproc.cpp at L828-922.
[270] *See, e.g.*, photo.cpp at L189-292.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[271] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[272] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
15. The method of claim 14, wherein the entity is a first entity, and wherein the method further comprises performing by the first device: receiving user input via user interaction with the interactive display of the first device, the user input specifying a location and a symbol corresponding to a second entity other than the first device	The AGIS LifeRing product practices "wherein the entity is a first entity, and wherein the method further comprises performing by the first device: receiving user input via user interaction with the interactive display of the first device, the user input specifying a location and a symbol corresponding to a second entity other than the first device and the second devices; and based on the user input, adding the user-specified symbol to the interactive display at a position on the second georeferenced map corresponding to the user-specified location of the second entity." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5.

[271] *See, e.g.,* server/msgproc.cpp at L377-474.
[272] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
and the second devices; and based on the user input, adding the user-specified symbol to the interactive display at a position on the second georeferenced map corresponding to the user-specified location of the second entity.	In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[273], "BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY", "BUTTON_TRACK_TYPE_VEHICLE", "BUTTON_TRACK_TYPE_INF", "BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. In the file "buttons.cpp" the function "button_actions()"[274] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[275] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[276] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the

[273] *See, e.g.*, buttons.h at L117-132.
[274] *See, e.g.*, buttons.cpp at L1344-2365.
[275] *See, e.g.*, display/ftext.cpp at L176-244.
[276] *See, e.g.*, netselect.cpp at L152-221.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
	"on_que_list()"[277] and "num_on_que_list()"[278] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[279] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[280] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[281] and subsequently to "form_net_line()"[282] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[283] and subsequently "send_net_msg_participants()"[284] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[285] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[286] is implemented as described above with various conditions. For the default condition catching messages for

[277] *See, e.g.*, quelist.cpp at L16-22.
[278] *See, e.g.*, quelist.cpp at L60-71.
[279] *See, e.g.*, display/msgproc.cpp at L328-399.
[280] *See, e.g.*, netselect.cpp at L391-492.
[281] *See, e.g.*, netselect.cpp at L373-384.
[282] *See, e.g.*, netselect.cpp at L352-370.
[283] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[284] *See, e.g.*, display/msgproc.cpp at L828-922.
[285] *See, e.g.*, photo.cpp at L189-292.
[286] *See, e.g.*, server/msgproc.cpp at L377-474.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[287] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
16. The method of claim 15, further comprising performing by the first device: transmitting the user-specified symbol and location of the second entity to the second devices for addition of the user-specified symbol to respective interactive displays of the second devices at respective positions on respective georeferenced maps corresponding to the user-specified location of the second entity.	The AGIS LifeRing product practices "transmitting the user-specified symbol and location of the second entity to the second devices for addition of the user-specified symbol to respective interactive displays of the second devices at respective positions on respective georeferenced maps corresponding to the user-specified location of the second entity." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5. In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[288], "BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY", "BUTTON_TRACK_TYPE_VEHICLE", "BUTTON_TRACK_TYPE_INF", "BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a

[287] *See, e.g.,* server/msgproc.cpp at L 51-61.
[288] *See, e.g.,* buttons.h at L117-132.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. In the file "buttons.cpp" the function "button_actions()"[289] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[290] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[291] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[292] and "num_on_que_list()"[293] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[294] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[295] is implemented where the display is configured for

[289] *See, e.g.*, buttons.cpp at L1344-2365.
[290] *See, e.g.*, display/ftext.cpp at L176-244.
[291] *See, e.g.*, netselect.cpp at L152-221.
[292] *See, e.g.*, quelist.cpp at L16-22.
[293] *See, e.g.*, quelist.cpp at L60-71.
[294] *See, e.g.*, display/msgproc.cpp at L328-399.
[295] *See, e.g.*, netselect.cpp at L391-492.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	net communication. The call to "send_to_display_net_list()"[296] and subsequently to "form_net_line()"[297] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[298] and subsequently "send_net_msg_participants()"[299] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[300] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[301] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[302] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
17. The method of claim 16, wherein the user input further specifies information	The AGIS LifeRing product practices "wherein the user input further specifies information associated with the second entity, and wherein the method further comprises performing, by the first device: transmitting the user-specified information associated with the second entity to the second devices."

[296] *See, e.g.*, netselect.cpp at L373-384.
[297] *See, e.g.*, netselect.cpp at L352-370.
[298] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[299] *See, e.g.*, display/msgproc.cpp at L828-922.
[300] *See, e.g.*, photo.cpp at L189-292.
[301] *See, e.g.*, server/msgproc.cpp at L377-474.
[302] *See, e.g.*, server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
associated with the second entity, and wherein the method further comprises performing, by the first device: transmitting the user-specified information associated with the second entity to the second devices.	The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5. The user can add information about the added/entered symbol and send to other users. Exhibit 4 at 5. In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[303], "BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY", "BUTTON_TRACK_TYPE_VEHICLE", "BUTTON_TRACK_TYPE_INF", "BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. In the file "buttons.cpp" the function "button_actions()"[304] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action.

[303] *See, e.g.,* buttons.h at L117-132.
[304] *See, e.g.,* buttons.cpp at L1344-2365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "display/ftext.cpp" the function "create_ftext()"[305] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[306] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[307] and "num_on_que_list()"[308] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[309] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[310] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[311] and subsequently to "form_net_line()"[312] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[313] and subsequently "send_net_msg_participants()"[314] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file

[305] *See, e.g.*, display/ftext.cpp at L176-244.
[306] *See, e.g.*, netselect.cpp at L152-221.
[307] *See, e.g.*, quelist.cpp at L16-22.
[308] *See, e.g.*, quelist.cpp at L60-71.
[309] *See, e.g.*, display/msgproc.cpp at L328-399.
[310] *See, e.g.*, netselect.cpp at L391-492.
[311] *See, e.g.*, netselect.cpp at L373-384.
[312] *See, e.g.*, netselect.cpp at L352-370.
[313] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[314] *See, e.g.*, display/msgproc.cpp at L828-922.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
	"photo.cpp" the function "create_photo()"[315] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[316] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[317] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
18. The method of claim 17, further comprising performing by the first device: adding data representing the spatial coordinates of the location of the second entity and data representing the information associated with the second entity to the database.	The AGIS LifeRing product practices "adding data representing the spatial coordinates of the location of the second entity and data representing the information associated with the second entity to the database." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5. The user can add information about the added/entered symbol and send to other users. Exhibit 4 at 5. In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[318], "BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY",

[315] *See, e.g.*, photo.cpp at L189-292.
[316] *See, e.g.*, server/msgproc.cpp at L377-474.
[317] *See, e.g.*, server/msgproc.cpp at L 51-61.
[318] *See, e.g.*, buttons.h at L117-132.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"BUTTON_TRACK_TYPE_VEHICLE", "BUTTON_TRACK_TYPE_INF", "BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function "initGPS()"[319] is implemented where the variable "comport" is the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[320] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[321] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[322] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[323] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[324] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event

[319] *See, e.g.*, csif/csif.cpp at L126-148.
[320] *See, e.g.*, csif/gpslistener.cpp at L36-51.
[321] *See, e.g.*, csif/gpslistener.cpp at L101-235.
[322] *See, e.g.*, csif/gpslistener.cpp at L158.
[323] *See, e.g.*, csif/gpslistener.cpp at L76-99.
[324] *See, e.g.*, display/display.cpp at L274-581.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	capture routine "WndProc()"[325] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[326] is implemented where the queued GPS messages are processed incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[327] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[328] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[329] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[330] and are implemented in the function "button_actions()"[331] the case for

[325] *See, e.g.*, display/display.cpp at L584-969.
[326] *See, e.g.*, display/gps.cpp at L346-365.
[327] *See, e.g.*, display/gps.cpp at L292-336.
[328] *See, e.g.*, display/gps.cpp at L77-126.
[329] *See, e.g.*, display/display.cpp at L1015-1037 and L1076-1115.
[330] *See, e.g.*, display/buttons.cpp at L594-598 and L1344-1346.
[331] *See, e.g.*, display/buttons.cpp at L1344-2365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[332] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[333] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. As a further example of receiving messages from the second devices that include location information, each time a message is received by the server from a device, the location information from that device is updated at the server. The location information for eah device in the group (e.g. the second devices) is then transmitted as part of the update message to the first device, as depicted in nmsg.numMapEntries. msgproc.cpp at ll. 862-892. The "send_track_msg()"function is used under the condition where "add_track()"[334] and "update_track_id()"[335] functions in the "display/track.cpp" is invoked. The "update_track_id()" is used in user interface as views in the "buttons.cpp" file in the main routine where

[332] *See, e.g.*, display/msgproc.cpp at L170-325.
[333] *See, e.g.*, display/msgproc.cpp at L615-682.
[334] *See, e.g.*, display/track.cpp at L373-477.
[335] *See, e.g.*, display/track.cpp at L482-510.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	button actions are implemented "button_actions()" as discussed above where the case for "BUTTON_TRACK_ID_UNKN", "BUTTON_TRACK_ID_FRND" and "BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[336] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent the "DDL_MSG_OWN_POSITION" case where the message is passed through by a call to "pass_thru_msg()" function. The file "server/server.cpp" implements the main function "WndProc()"[337] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through. In the file "buttons.cpp" the function "button_actions()"[338] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[339] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[340] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the

[336] *See, e.g.*, server/msgproc.cpp at L377-474.
[337] *See, e.g.*, server/server.cpp at L228-386.
[338] *See, e.g.*, buttons.cpp at L1344-2365.
[339] *See, e.g.*, display/ftext.cpp at L176-244.
[340] *See, e.g.*, netselect.cpp at L152-221.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"on_que_list()"[341] and "num_on_que_list()"[342] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[343] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[344] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[345] and subsequently to "form_net_line()"[346] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[347] and subsequently "send_net_msg_participants()"[348] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[349] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[350] is implemented as described above with various conditions. For the default condition catching messages for

[341] *See, e.g.*, quelist.cpp at L16-22.
[342] *See, e.g.*, quelist.cpp at L60-71.
[343] *See, e.g.*, display/msgproc.cpp at L328-399.
[344] *See, e.g.*, netselect.cpp at L391-492.
[345] *See, e.g.*, netselect.cpp at L373-384.
[346] *See, e.g.*, netselect.cpp at L352-370.
[347] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[348] *See, e.g.*, display/msgproc.cpp at L828-922.
[349] *See, e.g.*, photo.cpp at L189-292.
[350] *See, e.g.*, server/msgproc.cpp at L377-474.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[351] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
19. The method of claim 15, wherein the portion of the interactive display is a first portion, wherein the position of the symbol corresponding to the particular second device is a first position, and wherein receiving the user input specifying the location of the second entity comprises: detecting user selection of a second portion of the interactive display corresponding to a second position on the second georeferenced map; and based at least in part on coordinates of the second position on the	The AGIS LifeRing product practices "wherein the portion of the interactive display is a first portion, wherein the position of the symbol corresponding to the particular second device is a first position, and wherein receiving the user input specifying the location of the second entity comprises: detecting user selection of a second portion of the interactive display corresponding to a second position on the second georeferenced map; and based at least in part on coordinates of the second position on the second georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the second position on the second georeferenced map, wherein the location represented by the second position is the location of the second entity." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5. The user can add information about the added/entered symbol and send to other users. Exhibit 4 at 5. In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[352], "BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY", "BUTTON_TRACK_TYPE_VEHICLE",

[351] *See, e.g.,* server/msgproc.cpp at L 51-61.
[352] *See, e.g.,* buttons.h at L117-132.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
second georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the second position on the second georeferenced map, wherein the location represented by the second position is the location of the second entity.	"BUTTON_TRACK_TYPE_INF", "BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. In the file "buttons.cpp" the function "button_actions()"[353] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[354] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[355] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[356] and "num_on_que_list()"[357] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[358] is implemented where the free text message is transmitted directly or through a server as described above through the use of the

[353] *See, e.g.*, buttons.cpp at L1344-2365.
[354] *See, e.g.*, display/ftext.cpp at L176-244.
[355] *See, e.g.*, netselect.cpp at L152-221.
[356] *See, e.g.*, quelist.cpp at L16-22.
[357] *See, e.g.*, quelist.cpp at L60-71.
[358] *See, e.g.*, display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[359] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[360] and subsequently to "form_net_line()"[361] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[362] and subsequently "send_net_msg_participants()"[363] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[364] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[365] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[366] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list.

[359] *See, e.g.*, netselect.cpp at L391-492.
[360] *See, e.g.*, netselect.cpp at L373-384.
[361] *See, e.g.*, netselect.cpp at L352-370.
[362] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[363] *See, e.g.*, display/msgproc.cpp at L828-922.
[364] *See, e.g.*, photo.cpp at L189-292.
[365] *See, e.g.*, server/msgproc.cpp at L377-474.
[366] *See, e.g.*, server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
20. The method of claim 14, wherein the database is stored on the first device.	The AGIS LifeRing product practices "wherein the database is stored on the first device." See claim 14.
21. The method of claim 14, wherein the database is stored on the server.	The AGIS LifeRing product practices "wherein the database is stored on the server." See claim 14.
22. The method of claim 1, wherein the spatial coordinates comprise latitude and longitude coordinates.	The AGIS LifeRing product practices "wherein the spatial coordinates comprise latitude and longitude coordinates." The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function "initGPS()"[367] is implemented where the variable comport is as the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[368] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[369] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[370] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[371] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued.

[367] *See, e.g.*, csif/csif.cpp at L126-148.
[368] *See, e.g.*, csif/gpslistener.cpp at L36-51.
[369] *See, e.g.*, csif/gpslistener.cpp at L101-235.
[370] *See, e.g.*, csif/gpslistener.cpp at L158.
[371] *See, e.g.*, csif/gpslistener.cpp at L76-99.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "display.cpp" inside the initialization function "InitInstance()"[372] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[373] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[374] is implemented where the queued GPS messages are process incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[375] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[376] is implemented. This function utilizes the "gps_pos_decode()" function to convert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[377] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states.

[372] *See, e.g.*, display/display.cpp at L274-581.
[373] *See, e.g.*, display/display.cpp at L584-969.
[374] *See, e.g.*, display/gps.cpp at L346-365.
[375] *See, e.g.*, display/gps.cpp at L292-336.
[376] *See, e.g.*, display/gps.cpp at L77-126.
[377] *See, e.g.*, display/display.cpp at L1015-1037 and L1076-1115.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file “display/buttons.cpp” in the main routine where it is documented that button actions are processed by “button_actions”[378] and are implemented in the function “button_actions()”[379] the case for pressing the “BUTTON_PING:” is implemented where calls are made to “SendOwnPosString()” and “send_all_tracks()”. In the file “display/msgproc.cpp” the function “SendOwnPosString()”[380] is implemented where the “op” struct variable of type “OWN_POSITION” is populated with location information such as latitude and longitude. Then the “dmsg” variable of type “DDLMSG” is set to have a “msgType” of “DDL_MSG_OWN_POSITION” and variable “op” set to one of its other member variables. Then the “dmsg” is set to a buffer with the call to “put_ddl_msg()” and a call to “send_message_bulk()” when the transmit method “xmit_method” is set to “SERVER” and using a “server_ip” variable. Alternatively, direct transmission may occur with the call to “send_message_direct()” Similarly, in the file “display/msgproc.cpp” the function “send_track_msg()”[381] is implemented which includes location information along with previous location information. Similar to the previous function, the “dmsg” is set to type “DDL_MSG_TRACK_REPORT” and “tmsg” variable with location information is set to its member variable. The functions “put_ddl_msg()” followed by “send_message_bulk()” or “send_message_direct()” transmit the message. The “send_track_msg()”function is used under the condition where “add_track()”[382] and “update_track_id()”[383] functions in the “display/track.cpp” is invoked. The “update_track_id()” is used in user interface as views in the “buttons.cpp” file in the main routine where button actions are implemented “button_actions()” as discussed above where the case for “BUTTON_TRACK_ID_UNKN”, “BUTTON_TRACK_ID_FRND” and

[378] *See, e.g.*, display/buttons.cpp at L594-598 and L1344-1346.
[379] *See, e.g.*, display/buttons.cpp at L1344-2365.
[380] *See, e.g.*, display/msgproc.cpp at L170-325.
[381] *See, e.g.*, display/msgproc.cpp at L615-682.
[382] *See, e.g.*, display/track.cpp at L373-477.
[383] *See, e.g.*, display/track.cpp at L482-510.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[384] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent the "DDL_MSG_OWN_POISITION" case where the message is passed through by a call to "pass_thru_msg()" function. The file "server/server.cpp" implements the main function "WndProc()"[385] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through.
23. The method of claim 1, further comprising performing, by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating voice-over-IP (VOIP) communication with the particular	The AGIS LifeRing product practices "performing, by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating voice-over-IP (VOIP) communication with the particular second device." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[386] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. See also

[384] *See, e.g.*, server/msgproc.cpp at L377-474.
[385] *See, e.g.*, server/server.cpp at L228-386.
[386] *See, e.g.*, buttons.cpp at L1344-2365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
second device.	BUTTON_CALL and BUTTON_CCALL in buttons.h. and buttons.cpp. In the file "display/ftext.cpp" the function "create_ftext()"[387] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[388] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[389] and "num_on_que_list()"[390] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[391] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[392] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[393] and subsequently to "form_net_line()"[394] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[395] and subsequently "send_net_msg_participants()"[396] the message is transmitted directly or through a server as described above through the use of the

[387] *See, e.g.*, display/ftext.cpp at L176-244.
[388] *See, e.g.*, netselect.cpp at L152-221.
[389] *See, e.g.*, quelist.cpp at L16-22.
[390] *See, e.g.*, quelist.cpp at L60-71.
[391] *See, e.g.*, display/msgproc.cpp at L328-399.
[392] *See, e.g.*, netselect.cpp at L391-492.
[393] *See, e.g.*, netselect.cpp at L373-384.
[394] *See, e.g.*, netselect.cpp at L352-370.
[395] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[396] *See, e.g.*, display/msgproc.cpp at L828-922.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[397] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[398] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[399] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
24. A system comprising: a first device programmed to perform operations comprising:	See claim 1.
receiving a message from a second device, wherein the message relates to joining a group;	See claim 1.

[397] *See, e.g.,* photo.cpp at L189-292.
[398] *See, e.g.,* server/msgproc.cpp at L377-474.
[399] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group;	See claim 1.
presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the first georeferenced map at respective	See claim 1.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial coordinates;	
sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location;	See claim 1.
receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates;	See claim 1.
presenting, via the interactive display of the first device, the second	See claim 1.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices;	
and identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first device does not have access to respective Internet	See claim 1.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
Protocol addresses of the second devices.	
25. The system of claim 24, wherein the data includes a short message service message, a text message, an image, or a video.	See claim 2.
26. The system of claim 24, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC).	See claim 3.
27. The system of claim 24, wherein the second map is a satellite image.	See claim 4.
28. The system of claim 24, wherein the operations further comprise sending updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information	See claim 5.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
comprising a previous location of the first device, displacement of the first device by a predetermined distance relative to a previous location of the first device, or both.	
29. The system of claim 24, wherein the operations further comprise identifying second user interaction with the interactive display selecting at least one of the user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device.	See claim 6.
30. The system of claim 24, wherein the message from the second device is a Short Message Service (SMS) message or a text	See claim 7.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
message.	
31. The system of claim 24, wherein participating in the group further includes sending first status information to the server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second location information comprising a plurality of battery levels of the respective plurality of second devices included in the group, a plurality of signal strengths of wireless signals of the respective plurality of second	See claim 8.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a combination thereof.	
32. The system of claim 24, wherein the first device is a smart phone.	See claim 9.
33. The system of claim 24, wherein the operations further include: with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality of second devices; and based on transmitting the group identifier associated with the second group, participating in the second group, wherein participating in the second group includes receiving third location information from the server, the third	See claim 10.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
location information comprising a plurality of locations of the respective second plurality of second devices included in the second group.	
34. The system of claim 24, wherein the data includes a voice recording.	See claim 11.
35. The system of claim 24, wherein the first device includes a Global Positioning Satellite (GPS) receiver, and wherein the operations further include: using the GPS receiver to obtain data indicative of the location of the first device, wherein sending the first location information to the server comprises using the Internet Protocol (IP) to send the first location information to the server.	See claim 12.

The evidence submitted with this response is sufficient to overcome Haney. Based on the arguments and evidence presented with this response, Patent Owner submits that Haney is not prior art to the '251 patent due to prior invention. Patent Owner respectfully requests withdrawal of Rejections 2-5 based on the Haney combinations.

CONCLUSION

For these reasons, Patent Owner respectfully requests withdrawal of the rejections and issuance of a reexamination certificate allowing all of the Challenged Claims.

Dated: March 24, 2021

Respectfully submitted,

FABRICANT LLP

/Peter Lambrianakos /
Peter Lambrianakos, Reg. No. 58,279
Attorney for Owner
Email: plambrinakos@fabricantllp.com

FABRICANT LLP
411 Theodore Fremd Road, Suite 206 South
Rye, NY 10580
Tel.: 212-257-5797
Fax: 212-257-5796

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

U.S. PATENT NO.:	9,445,251	ART UNIT:	3992
CONTROL NUMBER:	90/014,509	CONF. NO.:	477
FILING DATE:	May 15, 2020	EXAMINER:	Hotaling, John M.

TITLE: **METHOD TO PROVIDE AD HOC AND PASSWORD PROTECTED DIGITAL AND VOICE NETWORKS**

FILED ELECTRONICALLY

Mail Stop *Ex Parte* Reexam
ATTN: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF MALCOLM K. BEYER JR. IN SUPPORT OF REPLY TO OFFICE ACTION

I, Malcolm K. Beyer, Jr., being duly sworn, hereby state as follows:

1. I am the Chief Executive Officer ("CEO") of Plaintiff AGIS Software Development LLC ("AGIS"). I am also the first-named inventor on U.S. Patent No. 9,445,251 (the "'251 patent"). I submit this declaration based on my personal knowledge and in support of AGIS's response to the non-final office action issued by the Office in its reexamination of the '251 patent (Control No. 90/014,509).

Background

2. I graduated from the U.S. Naval Academy in 1962 and was commissioned as a Second Lieutenant in the U.S. Marine Corps. I later attended the U.S. Navy's programming

school and was the lead programmer for the first automated Marine Corps Tactical Operations Center Link-11 Navy Interface.

3. After leaving active service, I worked at a number of well-known technology companies, including System Development Corporation (considered the world's first computer software company) and Litton Industries. I then started several businesses which provided technology and engineering solutions supporting defense and military customers.

4. In 1987, I co-founded Advanced Programming Concepts, Inc. ("APC"), a Texas corporation based in Austin, Texas. APC operated primarily out of its main business location in Austin, Texas. APC specialized in designing, building, and supporting systems for enabling integration and sharing time-critical information across dissimilar military and defense applications. At APC, I was the majority shareholder and Chairman until we sold the business to Ultra Electronics in July 1999.

5. On June 30, 2004, I founded Advanced Ground Information Systems, Inc. ("AGIS Inc."). AGIS Inc.'s main business location is located at 92 Lighthouse Drive, Jupiter, Florida. AGIS Inc. also maintains business locations in Austin, TX and Kansas City, KS. AGIS Inc. employs approximately 15 people. In 2017, the Board of AGIS, Inc. approved a reorganization plan which organized AGIS and AGIS, Inc. under its current parent, AGIS Holdings, Inc. AGIS, Inc. continues to sell the AGIS LifeRing solutions while AGIS maintains a data center located at 1005 Stuart Lane, Marshall, Texas 75672. AGIS's Texas data center hosts servers, code, applications, and services necessary to run operations for AGIS Inc.'s LifeRing products and solutions and is used for research and development projects.

6. As the CEO of AGIS and AGIS Inc. and first-named inventor of the '251 patent, I have knowledge of all facets of the businesses, including the conception and reduction to

practice of the ’251 patent and the diligence involved in reducing to practice the claimed inventions.

7. Since 2004, AGIS Inc.’s primary business has revolved around offering the “LifeRing” products and solutions which include client-based applications and a server-based solutions for, generally, enabling smartphone, tablet, and PC users to easily and rapidly establish secure ad hoc digital networks. LifeRing 5.0 and its predecessor versions have been offered and sold to military, defense, and first-responder customers, as well as private industry customers.

8. AGIS LifeRing software was developed by United States military veterans as a response to the September 11th, 2001 terrorist bombings. Since that time, the company has continued its development efforts; leveraging the technological advances in modern PCs and Smartphones as well as communications methods. AGIS’s LifeRing software enables PC, iPhone, and Android Tablet and Smartphone users to easily establish ad hoc COP networks where many people need to coordinate and collaborate with many others.

9. AGIS LifeRing provides users with the present location and status of others. LifeRing software uses GPS to provide location data. Users can seamlessly enter geo-locations of events or objects by selecting the desired map location and then the appropriate symbol. Information about other networked users and symbols can be obtained by touching symbols on the map.

10. AGIS LifeRing provides an interactive map display. Because accuracy is necessary to ensure the success of life and death missions, AGIS’s maps are geo-referenced so that users and events are displayed at their correct locations in real-time. To interact and communicate with other users in a one or more groups, users simply touch the display at the

desired map location and then select the appropriate or desired symbol, which appears at the correct map location, to initiate an exchange. LifeRing supports 80 different map types.

11. Over the years, LifeRing has been successfully deployed and tested in operational environments, including:

- New York Emergency Operations Center Test
- National Incident Management System (NIMS)Test
- Coalition Warrior Interoperability Demonstration (CWID)
- Army Network Integration Evaluation (NIE 12.1, 12.2 and13.1)
- Numerous US Joint Chiefs of Staff Exercises
- The Defense Intelligence Agency (LifeRing won 4 Stars in the DIA's PLUGFEST)
- SOCOM TNT Exercises (2012 &2013)
- US NATO Bold Quest (2012, 2013)
- Joint-Interagency Field Experimentation (JIFX) Exercises
- Army Expeditionary Warrior Experiment (AEWE 2013)
- Jolted Tactics
- US Navy
- International Partners (e.g., Australian, Dutch Defense Forces)

12. I am the first-named inventor of the '251 patent and its claimed inventions. I am very familiar with the claimed inventions of the '251 patent as I am the main point of contact for patent prosecution matters at AGIS (and formerly at AGIS, Inc.) and I personally reviewed and approved of the issued claims 1-35 of the '251 patent.

13. In addition to me, Mr. Christopher R. Rice is identified as a co-inventor on the '251 patent. With respect to the '251 patent, Mr. Rice's contributions included the client-server communications. For example, Mr. Rice contributed to the portion of the claimed invention related to the following recited limitations: "sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location" and "receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates."

14. In this reexamination proceeding, I understand that the Office has issued several rejections based on U.S. Patent No. 7,353,034 ("Haney"). I understand that Haney's application filing date is April 4, 2005. However, I invented the claimed inventions prior to April 4, 2005.

15. The claimed inventions of the '251 patent were conceived of by at least January 19, 2005 and were reduced to practice by October 22, 2005. The October 22, 2005 reduction to practice was the product of diligent work on the part of a team of engineers and programmers under my direction and supervision. Below, I identify and submit substantial evidence regarding the conception and reduction to practice of the claimed inventions of the '251 patent.

16. The claimed inventions recited in the '251 patent were conceived of by at least January 19, 2005. In August 2004, AGIS and its customer Raytheon engaged in discussions to develop a homeland security solution based on AGIS's LifeRing solution. In Exhibit 1 to this declaration, I provide a copy of an email summarizing an outline of the planned phases of development. **Exhibit 1.** After these August 2004 discussions, AGIS submitted a proposal for a system of PDA/cell phone devices for use in a homeland defense system, and AGIS reduced its proposal to a statement of work (SOW) on or around August 29, 2004. The August 29, 2004

SOW outlined the system using PDA cell phone devices for transmitting and displaying tracked participants on maps, i.e., "correctly geographically superimposed on a map" and for interacting with the map to exchange data. **Exhibit 2.**

17. On January 19, 2005 I prepared a presentation for Raytheon attached hereto as **Exhibit 3**, and this presentation was presented to Raytheon personnel, including Kurt Winckler, on January 21, 2005. The January 19, 2005 presentation attached as **Exhibit 3** is evidence that I and Mr. Rice had conceived of the claimed inventions as early as January 19, 2005.

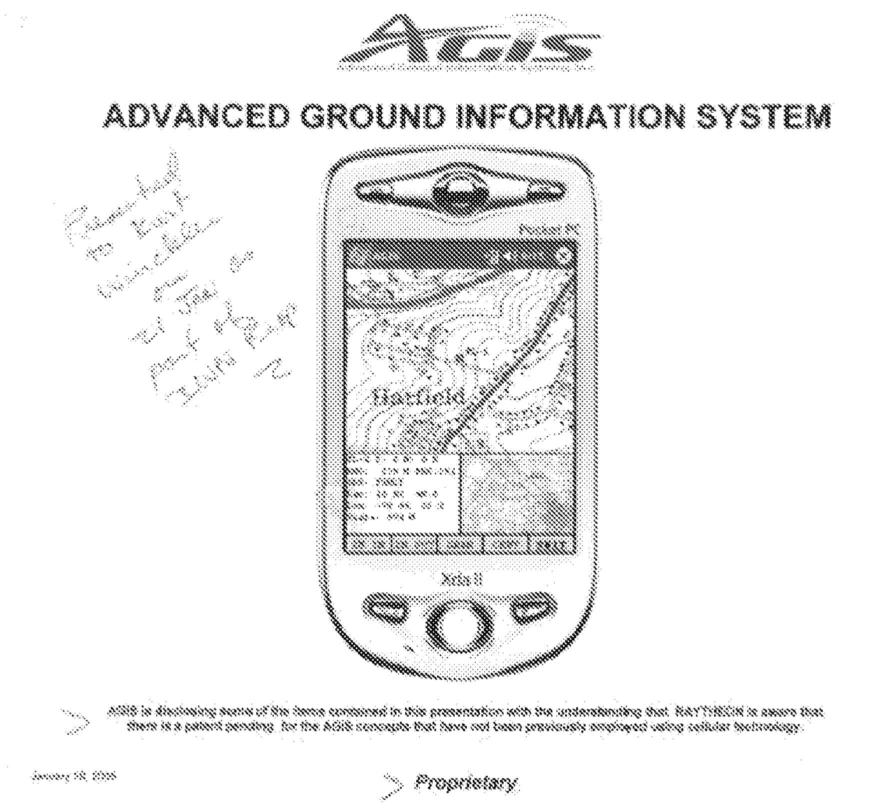


Exhibit 3 at 1.

18. The January 21, 2005 presentation describes cell phone/PDA devices programmed with software to permit users to display different maps and overlay maps, zoom and

offset maps, enter other entities (track symbols), assign information associated with entities, obtain location and information of other AGIS units and tracking them, displaying tracking histories, and selecting different maps (e.g., fixed versus moving).

AGIS OPERATION
Display

The AGIS operator can:
- Display different maps and overlay maps
- Zoom and Offset maps
- Enter other entities (track symbols) such as accidents, fires, trucks, military units, etc., and assign information associated with them
- Obtain range and bearing readings to other locations
- Obtain location and information of other AGIS units and the tracks that they have entered
- Display the track histories
- Select a fixed map (his symbol moves across the map)
- Select a moving map (his symbol stays at the center of the display and the map moves underneath)
- Specify that the map and the symbols occupy most of the display

Proprietary

Exhibit 3 at 3.

19. The January 21, 2005 presentation describes that the AGIS software provides each user with the following features: display user's phone's and other users' phones' location, status, heading, speed, altitude, phone signal strength, and GPS status; displaying phones' locations, identities, and status information on different maps; establishing voice and data communications with other participants by interacting with the symbols; and transmitting messages and multimedia between participants.

AGIS OPERATION
Display

The AGIS operator can:

- Display different maps and overlay maps
- Zoom and Offset maps
- Enter other entities (track symbols) such as accidents, fires, trucks, military units, etc., and assign information associated with them
- Obtain range and bearing readings to other locations
- Obtain location and information of other AGIS units and the tracks that they have entered
- Display the track histories
- Select a fixed map (his symbol moves across the map)
- Select a moving map (his symbol stays at the center of the display and the map moves underneath)
- Specify that the map and the symbols occupy most of the display

Proprietary

Exhibit 3 at 3.

AGIS SOFTWARE

PROVIDES EACH USER:

- Display of his and other phones' location, status, heading, speed , altitude, phone signal strength and GPS status (on, off, 2D or 3D).
- His and other AGIS participants' location, identity and status information superimposed on a variety of maps, aerial photographs, and satellite images.
- Multiple, easily accessible, layered *SoftSwitches* organized by category to enable the AGIS operator to quickly operate the system.
- The ability quickly and easily to establish voice and data communications (*point to talk*) with any other AGIS net participant without losing the tactical picture by simply hooking other site's symbols.
- The ability quickly to transmit free text, fixed formatted messages, photographs and video clips between each user.

Proprietary 5

Exhibit 3 at 5.

AGIS SOFTWARE
Continued

AGIS FURTHER PROVIDES EACH USER:

- The ability to enter friendly and hostile forces (or other information such as fires, emergencies, accidents, pilot boats, fire boats, etc.*) locations on a map and to associate data with their symbol. This information is then transmitted along with the AGIS participants location on the AGIS cellular net to all other AGIS equipped users.
- The ability to receive and transmit free text, fixed formatted, photograph**, and video clip** messages by pointing at the AGIS user to whom they are to be sent.
- The ability to establish range and bearing data from the user to any point.
- The ability for each AGIS user to view his own and other tracks' history trail and last heading data.
- The ability to also operate from a PC and a Tablet*.

Proprietary

Exhibit 3 at 6.

20. The January 21, 2005 presentation describes the formation and use of groups or "nets" of participants.

AGIS OPERATION

The operator controls the AGIS with either a Stylus or using Finger On Glass

The operator interacts with the AGIS through the use of layered *SoftSwitches* which are located at the bottom of the display.

When initializing the system, the AGIS operator specifies from a list those AGIS users with whom he desires to net location data and with whom he desires to be able to conduct rapid voice and data communications.

When the AGIS is prompted to respond with its location, it sends its location and status to all that are part of the net.

MilStd 2525 symbols (or Homeland Defense defined symbols*) are associated with each user and appear at the correct location on each of the displays.

The AGIS operator obtains information concerning other symbols appearing on the display by touching the screen at their location thereby "hooking" them. Data concerning the other symbol then appears on the lower part of the AGIS display.

* Part of the proposed contract

[illegible] *Proprietary* 8

Exhibit 3 at 8.

AGIS OPERATION
Cellular phone calls

- When the AGIS operator decides to call another AGIS user, he simply hooks the other user's symbol and selects the Call SoftSwitch. The phone of the AGIS receiving call then hears a ring and sees a box appear around the sender's symbol.
- The AGIS operator can setup pre-established nets of up to six AGIS net participants. When he desires to talk to that group of users he selects the appropriate net number and a conference call is automatically made.
- The AGIS operator can conference an almost unlimited number of AGIS participants by using the AGIS's 800 conferencing capability. Again the AGIS operator assigns AGIS participants to a net number. When he desires to make the conference call, he selects the net number. This action causes a message to be sent to all of the AGIS participants, causing their phones to automatically call an 800 number and automatically entering their participant code.

Proprietary

Exhibit 3 at 9.

21. AGIS's LifeRing solution was under development throughout 2004 and 2005. During this period, I conceived and reduced to practice both non-server and server-based implementations. As I discussed above, Mr. Rice's contributions included the client-server communications, including "sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location" and "receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates." In the non-server implementations, AGIS's maps were uploaded to and called from databases and the transmissions occurred over SMS only. However, in the server-based implementations, which were conceived of by January 19,

2005, the devices communicated with intermediary servers and the maps requested and received from servers. The January 21, 2005 presentation describes an AGIS server implementation and the development and testing of preproduction server-based implementations were underway.

AGIS SERVER

- When AGIS is operating in a SMS (slow speed) mode of operation, communications are from the AGIS phone to the telephone company and then from the phone company to all other AGIS phones.
- When the AGIS is operating in a high speed GRPS / CDMA2000 1xEVDO mode, communications are from the AGIS phone to the telephone phone company, then to the TCP/IP Server and from the TCP/IP Server to all the other AGIS phones.
- The TCP/IP Server pushes the received data to the AGIS phones so that information is received within approximately 5 seconds.
- The TCP/IP Server provides interoperability between the GRPS / CDMA2000 1xEV-DO communications.
- The AGIS accounts for when the phones go into Vcice mode and then transmits data in SMS until high speed communications are available.
- The TCP/IP Server is operating on a 24 hour basis. We are still in the process of resolving Server Trouble Reports.

Proprietary

Exhibit 3 at 14.

AGIS STATUS

THE PROTOTYPE AGIS HAS UNDERGONE A WEEK OF TESTING AT THE NAVAL RESEARCH LABORATORY'S (NRL) ANTITERRIORISM GROUP. DURING THE TEST PERIOD, THE AGIS OPERATED WITHOUT FAILURE. THE RESULTS OF THIS TESTING WAS POSITIVE. IT WAS REQUESTED THAT:

- AGIS FUNCTION SWITCHES BE MADE EASIER TO USE AND REMEMBER*
- MULTIPLE MAP SETS CAPABILITY BE PROVIDED*
- DIRECTION UP MAPS AND MULTIPLE LOCATION MAPS BE PROVIDED**

WE ARE UNDER CONTRACT TO ADD THE OTHER FEATURES (OTHER THAN SECURITY) DISCUSSED IN THIS PRESENTATION.

THE PREPRODUCTION AGIS IS DUE TO BE TESTED AT NRL IN FEB 2005.

ALL THE FEATURES (EXCEPT SECURITY) WILL BE INCORPORATED INTO THE SEATTLE DELIVERY

* Completed. ** Will be available in January or early February

Proprietary

Exhibit 3 at 15.

22. This is consistent with AGIS's plans to provide "over the air" georeferenced maps via servers. **Exhibit 4 at 8-9.**

and Figure 19 depicts selection of coastlines and geopolitical boundaries. As part of a

Proprietary

Advanced Ground Information Systems, Inc. Topic Number N05-069 Proposal Number N051-069-0725 **Proprietary** 9

separate non government funded effort, AGIS plans to incorporate "over the air" downloading of georeferenced maps.

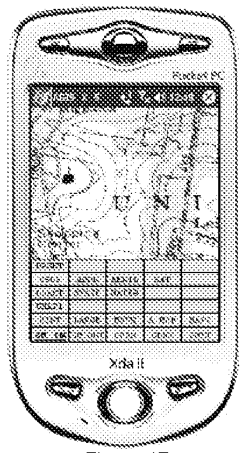


Figure 17

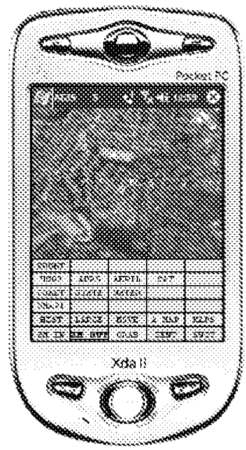


Figure 18

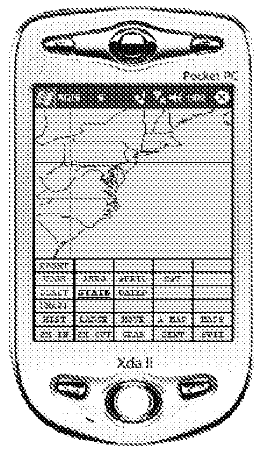


Figure 19

Exhibit 4 at 8-9.

23. Under my direction and supervision, AGIS was diligent in reducing to practice the claimed inventions of the '251 patent. Immediately after conception, I assembled a team to begin development and testing of various implementations. The AGIS project timeline consisted of multiple phases, and the project timeline was updated continuously over the course of development. **Exhibits 5-8** describe multiple timeline updates based on changes to the server-based implementation. From January 2005 to October 2005, AGIS dedicated hundreds of hours to the development and testing of the server-based implementations of the AGIS LifeRing solution. The development team included myself, Christopher R. Rice, Sandel Blackwell,

Dennis Hoff, Scott Brown, and Jason Cardamone. **Exhibits 9-14** include timesheets from the individuals involved in the development and evaluation of the AGIS LifeRing solution during the period of April 2005 through October 2005. These timesheets demonstrate that I was diligent in reducing to practice the January 19, 2005 concepts for the server-based implementation of the AGIS LifeRing solution. Please note that these timesheets indicate billable hours only, which reflects a subset of the actual time dedicated to the research and development and testing of the LifeRing solution.

24. In April 2005, Mr. Rice dedicated at least 39 hours to fixing the SMS transmission and TCP protocol portions of LifeRing. In April 2005, Mr. Blackwell billed 89 hours for work including, for example, updating the change rate of the displayed information, grouping and selecting participant groups, and user location updates. Mr. Blackwell was also involved in testing TCP and SMS communications issues and server issues including fixing database scroll issues on the server-based implementation. Mr. Blackwell also worked on addressing memory leakage issues. Mr. Blackwell also worked on integrating and developing Global Mapper features and other different maps used in LifeRing. Mr. Blackwell communicated with and worked closely with Mr. Rice to address these problems. Mr. Blackwell also worked on finalizing several versions of code. In April 2005, Mr. Hoff billed 17 hours for work on testing portions of LifeRing and developing code for the client side of LifeRing. Mr. Hoff was involved in testing location tracking and the loading of the client side application. Prior to April, Mr. Hoff was involved in development and testing for the location tracking, mapping data, interaction with maps and symbols, displaying of locations, maps and symbols, data and multimedia transfers, memory issues, and other issues related to both client and server side issues and had billed over 190 hours for such work between January 2005's conception date

and April 2005. In April 2005, Mr. Brown billed 160 hours for work on testing the LifeRing system. Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. Mr. Brown also assisted Mr. Blackwell, Mr. Rice and myself on documentation of the LifeRing reports, manuals, and versions.

25. In May 2005, Mr. Rice billed at least 60 hours for work related to fixing problems related to SMS and voice transitions, photo transmissions, and SMS transmissions. Mr. Rice also spent considerable time creating and testing UDP communications protocol for establishing communications between applications and servers, including testing the new implementation of the protocol with a device. Mr. Rice also worked on the development of the CSIF code. During this time, Mr. Rice communicated frequently with Mr. Blackwell, who was also working on updates to the code. In May 2005, Mr. Hoff billed 11 hours for work on testing portions of LifeRing and working with Mr. Rice and Mr. Blackwell on testing features. In May 2005, Mr. Cardamone billed 80 hours for work related to the code and configuration of an SVN code repository. Mr. Cardamone also worked on the communications and display issues for LifeRing. Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. In May 2005, Mr. Brown billed 80 hours for work on testing the LifeRing system. Mr. Brown also assisted Mr. Blackwell, Mr. Rice and myself on documentation of the LifeRing reports, manuals, and versions.

26. In June 2005, Mr. Rice billed at least 60 hours for work related to transmission issues with SMS and phone/conference calls. He worked on enhancing the stability of our SMS transmissions and supported our efforts on data reductions, UDP communications protocol, and CSIF code. Mr. Rice also made changes to the code for location tracking and message handling.

Mr. Rice also worked on network configuration for our development server. During this time, Mr. Rice communicated frequently with Mr. Blackwell, who was also working on updates to the code. In June 2005, Mr. Hoff billed 10 hours for work on testing new versions of LifeRing. Mr. Hoff was involved in testing SMS features of LifeRing. In June 2005, Mr. Cardamone billed 160 hours for work related to the LifeRing code repository and data reductions/transmissions related to LifeRing data. Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. In June 2005, Mr. Brown billed over 160 hours for work on testing the LifeRing system. Mr. Brown also assisted Mr. Blackwell, Mr. Rice and myself on documentation of the LifeRing reports, manuals, and versions.

27. In July 2005, Mr. Rice billed over 40 billable hours on communications infrastructure issues and network configuration and server migration issues. In July 2005, Mr. Blackwell billed over 80 billable hours on various issues related to LifeRing including communications issues, display issues, and location tracking issues. Mr. Blackwell performed research to solve some issues that we were having with our server-based implementation and he worked on developing the new maps that had been integrated and tested in April and May 2005. Again, during this period, Mr. Blackwell remained in constant communication with Mr. Rice in order to resolve issues with the development of LifeRing. In July 2005, Mr. Cardamone billed 120 hours for work on data processing and data reduction relating to the georeferenced maps and georeferenced map data for presentation and use in LifeRing. In April 2005, Mr. Brown billed 80 hours for work on testing the LifeRing system. Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on

such items. Mr. Brown also assisted Mr. Blackwell, Mr. Rice and myself on documentation of the LifeRing reports, manuals, and versions.

28. In August 2005, Mr. Rice billed over 80 hours for work on the LifeRing database systems, including databases used for location tracking data. Mr. Rice's August 2005 work also included work on the processing output (maps) for LifeRing. During August 2005, we ran into problems with the startup and shutdown of the LifeRing application. Mr. Rice worked on researching ways to reduce delays in starting the program and obtaining new maps and location data immediately upon startup. This included the retrieval of maps in both non-server and server-based implementations. In August 2005, Mr. Blackwell billed over 150 hours for work on the development of LifeRing. During this time, Mr. Blackwell was intimately involved in researching and development of the server-based implementation. He also spent considerable time on testing communications issues and updating the display of information on the client. In August 2005, Mr. Cardamone billed 200 hours for work on data processing and data reduction relating to the georeferenced maps and georeferenced map data for presentation and use in LifeRing. During this time, Mr. Cardamone also worked on code processing output for display (e.g., georeferenced maps) and various display filter issues in the LifeRing code. In August 2005, Mr. Brown billed 160 hours for work on testing the LifeRing system. Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. Mr. Brown also assisted Mr. Blackwell, Mr. Rice and myself on documentation of the LifeRing reports, manuals, and versions. Mr. Brown also assisted on preparing the LifeRing product for presentation to customers in Seattle.

29. In September 2005, Mr. Rice billed over 70 hours for work on the LifeRing location tracking database and location history issues. Mr. Rice also began to develop code to

more efficiently manage and distribute processing tasks in order to improve the delay issues that we were having in retrieving and processing maps and location information while also maintaining a reliable data communications flow. These improvements were being developed to support our non-server and server-based implementations. In September 2005, Mr. Blackwell billed over 150 hours for work on the development of LifeRing. Considerable time was spent on the server-based implementation, including on display, mapping, and communications tests and integrating the former with our existing non-server implementation. Mr. Blackwell spent a considerable amount of time researching whether and how to migrate our existing data and programs to the server and the impact of "over the air" transfer on the performance of the client. Mr. Blackwell worked closely with Mr. Rice to press for an October 2005 release of the server-based implementation. In September 2005, Mr. Brown billed over 190 hours for work on testing the LifeRing system. Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. Mr. Brown also assisted Mr. Blackwell, Mr. Rice and myself on documentation of the LifeRing reports, manuals, and versions.

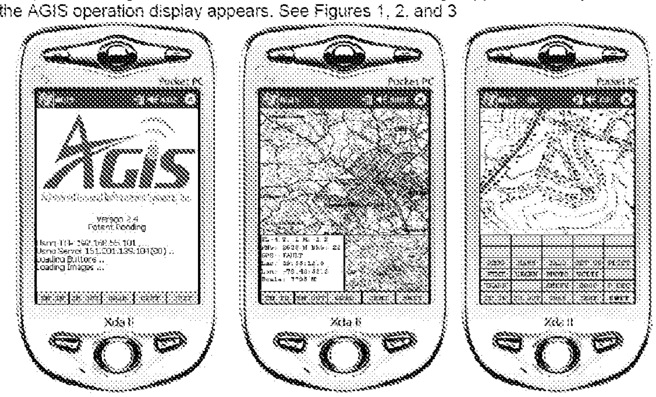
30. In October 2005, Mr. Rice billed over 85 hours for work on the LifeRing development. In addition to working on the location tracking database issues identified in our bug reports, Mr. Rice worked on finalizing the development of our image server implementation, which was the project for requesting/receiving maps from servers. Mr. Rice's work on this included writing and testing code for communicating with different servers for providing georeferenced maps to the LifeRing client applications on devices. In October 2005, Mr. Blackwell billed over 140 hours for work on the development of LifeRing. Again, Mr. Blackwell spent considerable time on the server-based implementation and working to finalize

the new release with Mr. Rice. This was the culmination of significant effort as the requested maps come in various formats and protocols from different servers. Mr. Rice and Mr. Blackwell were deeply involved in finalizing the October 22, 2005 code base for the server-based implementation and addressing issues found in bug reports and testing services. In October 2005, Mr. Brown billed 180 hours for work on testing the LifeRing system. Mr. Brown was responsible for testing all portions of the code and client side program to find bugs and errors and to track reporting on such items. Mr. Brown also assisted Mr. Blackwell, Mr. Rice and myself on documentation of the LifeRing reports, manuals, and versions. Mr. Brown spent considerable amount of time testing the October 22, 2005 server-based implementation.

31. The claimed inventions of the ’251 patent were reduced to practice by at least October 22, 2005. Exhibit 15 consists of code files that were checked into and present in AGIS’s code repository by October 22, 2005, and Exhibit 16 is a description of the map request function for the October 22, 2005 server-based implementation. The features presented in Exhibit 4 were also implemented in the October 22, 2005 server-based implementation. The October 22, 2005 server-based implementation included messaging protocols for joining group(s) and sending location information of its devices among the devices. The system’s maps were georeferenced maps with the devices and other entities represented as symbols on the maps. The system’s maps were real-time maps which included generating for presentation new maps with (1) updated locations, (2) different map types, and (3) different zoom levels and panned areas. By interacting with the system’s maps, users could touch symbols to send communications and multimedia to other users. Prior to October 22, 2005, AGIS’s LifeRing maps were generated from map data kept in databases resident on the device or on a connected peripheral device from

which maps were downloaded. Substantial and diligent efforts culminated in the claimed inventions as implemented in the October 22, 2005 version of LifeRing.

32. The below chart describes exemplary[1] correspondences between claims 1-35 of the ’251 patent and the October 22, 2005 version of AGIS LifeRing. I have confirmed that the each of the code files in Exhibit 15, which are the code files cited in the below chart, were checked in to the AGIS code repository by October 22, 2005.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
1. A computer-implemented method comprising:	The AGIS LifeRing product practices the computer implemented method of claim 1. The AGIS LifeRing product includes systems and methods for executing an application on PDA/cell phones, now referred to as a smartphone devices. See, e.g., “call.cpp” references to “phone” and “other phones.” The operator can then send voice conference or digital data to those AGIS equipped units by simply selecting their assigned NET SoftSwitch. The AGIS application software does not affect the operation of the PDA, the Cell phone or the GPS until it is activated. When the AGIS application software is selected, the AGIS PDA Cell phone is turned on, the GPS is automatically connected to the PDA / Cell phone through a Bluetooth interface and the AGIS Logo appears. Shortly thereafter the AGIS operation display appears. See Figures 1, 2, and 3  Figure 1 Figure 2 Figure 3 Exhibit 4 to the Beyer Declaration, depicting AGIS PDA/cell phones.

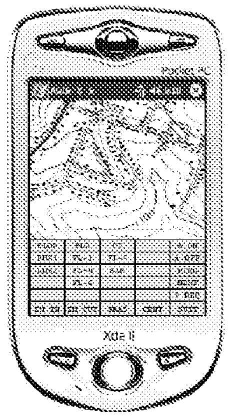
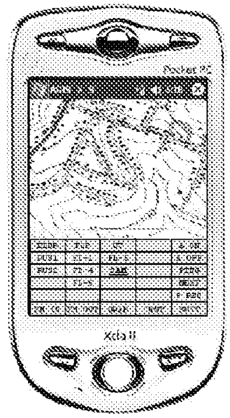
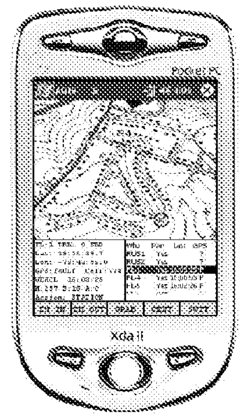
[1] This is not intended to be an exhaustive identification of correspondence of the LifeRing product to the claims.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
with a first device, receiving a message from a second device, wherein the message relates to joining a group;	The AGIS LifeRing product practices "with a first device, receiving a message from a second device, wherein the message relates to joining a group." The first device is programmed to receive a message from a second device related to joining a group. The process of joining a group is performed by sending messages to server and other users to join a group. The file "display/msgproc.cpp" implements the function "process_net_msg()"[2] where the logging function indicates "rcvd net msg" indicative of receiving a network message. The "nmsg" variable of type "NETUSERS" which is input to this function is tested against its member variable requestType which is specified in the file "csiftranscode.h"[3] and states to have "J)oining or R)equest current participants". The "nmsg.requestType" is then branched for cases of "I" for initial joining, "J" for joining, "R" for requesting and "X" for exiting the network. For the case of "J" for joining, the function makes a call to the "update_address_book()" function with sender information as the input parameters. The file "initfiles.cpp" implements the "update_address_book()"[4] where the sender information is used to find its index in the address book by a call to "find_addr_number()". The return from this call is used to update the array variable "csif_addr[]" by either adding the address if it does not exist indicative of a return value of -1 or by copying the "ipaddr" into a struct member of the indexed variable. The "csif_addr[]" array is an array of type "ADDRESS_ENTRY"[5] which is specified in the file "csiftranscode.h"[6] The members of this struct includes "name", "number", "group", "type", "level" and "ipaddr". The marking of the "csif_addr[]" variable array constitutes the joining of a group. The server further provides additional functionality in file "server/msgproc.cpp" implements the function "process_net_msg()"[7] where the logging function indicates "rcvd net msg" indicative of

[2] *See, e.g.,* display/msgproc.cpp at L1234-1292..
[3] *See, e.g.,* csiftranscode/csiftranscode.h at L254-263.
[4] *See, e.g.,* display/initfiles.cpp at L701-721.
[5] *See, e.g.,* display/initfiles.h at L 35
[6] *See, e.g.,* csiftranscode/csiftranscode.h at L45-54.
[7] *See, e.g.,* server/msgproc.cpp at L235-293.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	receiving a network message. The variables "from" of type character string and "nmsg" of type "NETUSERS" as described above is input to this function. Similar to described above, the "nmsg.requestType" is then branched for cases of "J" and not "J" described to mean of type "R" are handled. For the case of joining a group where "nmsg.requestType" equals "J", the function "update_address_book()" is called with the sender and receiver information and returns with an index into the address book array "csif_addr" as discussed above. Then the logging command indicates that the "address book [address book index] updated with ipaddr [address book ip address]" and writes "user [name] joined network" to the display. Followed by two calls to "send_net_msg_participants()" using a wildcard designator "-1" indicating everything in the respective field is to be sent. The first call sends everything in the table to the currently added user. The second call sends to everyone the currently added user. In the same file the function "send_net_msg_participants()"[8] documents that the user of "-1" in the first field indicates a "send to all" and the use of "-1" in the second field indicates a "send all net participants" condition. Thus, one of the objectives of the server function "send_net_msg_participants" is to process messages that are received from the second device related to joining a group and to send messages to the first device related to joining a group. For example, the source code reflects the parameter "nmsg.requetType='R'" that reflects a request to join a group being transmitted from a second device to a first device via the server. **2.5 AGIS Net Assignment and Ping** The AGIS permits the person in charge of the network to specify which AGIS cell phones are to be part of the network. This is accomplished by the AGIS operator selecting the P REQ SoftSwitch which causes a matrix of the AGIS phones to appear. Note that this is a multilayered matrix and, if there are more than 28 phones, the operator can select the NEXT SoftSwitch which will cause yet another group of participants to appear. The operator can drop highlighted AGIS units from the net by selecting their SoftSwitch or add new ones to the net by selecting them. In Figure 22 note that AGIS Participant RUS 2 is not part of the AGIS communications net. In Figure 23 note that RUS 2 has been made part of the AGIS communications net and FL 5 and SAM have been dropped from the AGIS communications net. The AGIS operator can also select to immediately transmit his location and all his tracks to another AGIS unit and cause that unit to immediately report its location and all its tracks by selecting the PING SoftSwitch To determine the status of the various AGISs in the communications net, the AGIS operator selects the PLIST SoftSwitch which causes display of the current AGIS communications net participants to be displayed in the lower left of the AGIS display. See Figure 24.

[8] *See, e.g.,* server/msgproc.cpp at L108-191.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	   Figure 22 Figure 23 Figure 24 Exhibit 4 to the Beyer Declaration, at 10-11. Similarly, the server further provides functionality to support the server side function similar to the client side in the file "server/initfiles.cpp" by implementing the function "update_address_book()"[9] where similar to the description above the call to "find_addr_number()" function retrieves an address index. If the address does not exist indicative of a return index value of "-1" calls the function "add_address_book()" otherwise updates the "csif_addr" array with the ip address input to the function. In the same file the function "add_address_book()"[10] is implemented where the "csif_addr" array is populated with the "name", "number", "group", "ipaddr" and a "isValid" member variable.
based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information	The AGIS LifeRing product practices "based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group." Once the devices have formed a group they share their location with each other. The AGIS software captures location information on the device through a serial communication link configured to continuously process GPS information messages. Messages such as location, precision and constellation are captured on an independent thread and when a valid

[9] *See, e.g.,* server/initfiles.cpp at L194-211.
[10] *See, e.g.,* server/initfiles.cpp at L163-191.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group;	message is received the thread triggers an event indicating the presence of a new message. The software then sends the new location to others through either a server or direct connection. The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function "initGPS()"[11] is implemented where the variable "comport" is the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[12] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[13] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[14] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[15] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[16] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[17] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[18] is implemented where the queued GPS messages are processed incrementally with a call to the "process_csif_gps_msg()". The function

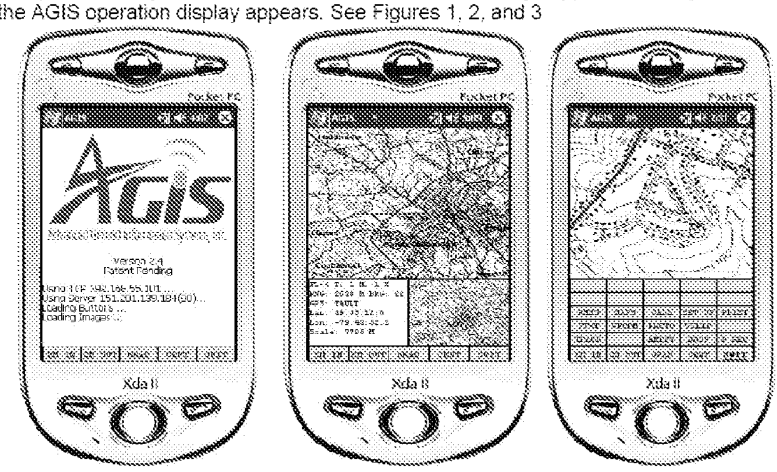
[11] *See, e.g.,* csif/csif.cpp at L126-148.
[12] *See, e.g.,* csif/gpslistener.cpp at L36-51.
[13] *See, e.g.,* csif/gpslistener.cpp at L101-235.
[14] *See, e.g.,* csif/gpslistener.cpp at L158.
[15] *See, e.g.,* csif/gpslistener.cpp at L76-99.
[16] *See, e.g.,* display/display.cpp at L274-581.
[17] *See, e.g.,* display/display.cpp at L584-969.
[18] *See, e.g.,* display/gps.cpp at L346-365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"process_csif_gps_msg()"[19] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[20] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[21] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[22] and are implemented in the function "button_actions()"[23] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[24] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to

[19] *See, e.g.,* display/gps.cpp at L292-336.
[20] *See, e.g.,* display/gps.cpp at L77-126.
[21] *See, e.g.,* display/display.cpp at L1015-1037 and L1076-1115.
[22] *See, e.g.,* display/buttons.cpp at L594-598 and L1344-1346.
[23] *See, e.g.,* display/buttons.cpp at L1344-2365.
[24] *See, e.g.,* display/msgproc.cpp at L170-325.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[25] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. As a further example of receiving messages from the second devices that include location information, each time a message is received by the server from a device, the location information from that device is updated at the server. The location information for each device in the group (e.g. the second devices) is then transmitted as part of the update message to the first device, as depicted in nmsg.numMapEntries. msgproc.cpp at ll. 862-892. The "send_track_msg()"function is used under the condition where "add_track()"[26] and "update_track_id()"[27] functions in the "display/track.cpp" is invoked. The "update_track_id()" is used in user interface as views in the "buttons.cpp" file in the main routine where button actions are implemented "button_actions()" as discussed above where the case for "BUTTON_TRACK_ID_UNKN", "BUTTON_TRACK_ID_FRND" and "BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[28] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent the "DDL_MSG_OWN_POSITION" case where the message is passed through by a call to "pass_thru_msg()" function.

[25] *See, e.g.,* display/msgproc.cpp at L615-682.
[26] *See, e.g.,* display/track.cpp at L373-477.
[27] *See, e.g.,* display/track.cpp at L482-510.
[28] *See, e.g.,* server/msgproc.cpp at L377-474.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	The file "server/server.cpp" implements the main function "WndProc()"[29] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through.
presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial coordinates;	The AGIS LifeRing product practices "presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial coordinates." The devices running the software display an interactive display with locations of group devices, georeferenced and shown in relation to others. The display further draws tracks and other features related to the display of devices. An exemplary interface from the AGIS client running on a mobile device is depicted below: The AGIS application software does not affect the operation of the PDA, the Cell phone or the GPS until it is activated. When the AGIS application software is selected, the AGIS PDA Cell phone is turned on, the GPS is automatically connected to the PDA / Cell phone through a Bluetooth interface and the AGIS Logo appears. Shortly thereafter the AGIS operation display appears. See Figures 1, 2, and 3  Figure 1 Figure 2 Figure 3

[29] *See, e.g.,* server/server.cpp at L228-386.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Exhibit 4 to Beyer Declaration, at 4. The code to render this UI, which includes the capability to display each of the devices on a map with data relating the positions to spatial coordinates is found primarily in the display/display.cpp code file. In the file "display/display.cpp" the main method "WndProc()"[30] where display events are captured and processed, the case for "WM_PAINT" where the display is drawn is implemented. When a "WM_PAINT" event occurs calls are made to "refresh_tact()", "refresh_stat()" and "refresh_inset()" passing the display window handle where the map, locations and other information gets displayed. In the file "tact.cpp" the function "refresh_tact()"[31] is implemented. There are three types of this function, two with an input handle and one without. The one without the handle first retrieves the current handle and then calls the "refresh_tact()" function version with the handle. In this function, among other features, the system draws the map by a call to "ww_draw_map()" and draws locations by a call to "track_draw()". A special "refresh_tact()" implementation only draws the tactical area which is documented to correspond to the display's WM_PAINT event and only accepts a window handle of type HWND. The function "refresh_tact()" is used throughout the code to signal an update to the view within "buttons.cpp", "display.cpp", "images.cpp", "inset.cpp", "maps.cpp", "msgproc.cpp", "track.cpp" and others to signal the update of the graphical window. In the file "maps.cpp" the function "ww_draw_map()"[32] is implemented. The function utilizes georeferenced latitude and longitude information to position the map using variables "urlat", "urlon", "lllat" and "lllon" for the upper right and lower left latitude and longitude. The variable array "ww_map" is used to hold the map data and is read in the same file using the function "ww_read_map()"[33] from local files. The function "ww_read_maps()"[34] makes individual calls to "ww_read_map()" to read the "world.pnt", "plit.pnt", "water.pnt" and a local map through invocation of the "local_read_map()" function pointed to by the "test user.ump" file. The

[30] *See, e.g.,* display/display.cpp at L584-969.
[31] *See, e.g.,* display/tact.cpp at L200-306.
[32] *See, e.g.,* maps.cpp at L213-276.
[33] *See, e.g.,* maps.cpp at L69-153.
[34] *See, e.g.,* maps.cpp at L726-732.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"ww_read_maps()" function is called in the file "display/display.cpp" by the function "InitInstance()"[35] when the system has started. In the file "stat.cpp" the function "refresh_stat()"[36] is implemented. There are two types of this function, one with an input handle and one without. The one without the handle first retrieves the current handle and then calls the "refresh_stat()" function version with the handle. In this function, among other features, the system "refresh status window" by a call to "stat_template_refresh()". There are other methods that the system implements such as "show_status_window()"[37] and the main window process in the "WndStatProc()"[38] callback function "WM_PAINT" event which calls the "refresh_stat()" and in turn "stat_template_refresh()". Upon creation of the stat view by use of the function "create_stat()"[39] to create the graphical interface followed by a call to the "stat_template_refresh()". Furthermore, the function "refresh_stat()" is used throughout the code within "buttons.cpp", "display.cpp", "gps.cpp", "msgproc.cpp", "partlist.cpp" to signal the update of the stat window. The function "stat_template_refresh()"[40] is implemented with a call to "status_hook()". The function "status_hook()"[41] is implemented where various information gets presented such as track information, current latitude, current longitude, GPS status amongst others. In the file "inset.cpp" the function "refresh_inset()"[42] is implemented. There are two types of this function, one with an input handle and one without. Both versions retrieve the Inset window handle through either a global handle or passed in handle and make a call to the "refresh_inset_window()" function. The function "refresh_inset_window()"[43] is implemented in the same file with calls made to "refresh_inset_map()", "refresh_inset_gps()" and "refresh_inset_trackamp()" based on the value held in the "inset_mode" variable set to "INSETWIN_MAP_USE", "INSETWIN_GPS_USE" or "INSETWIN_TRACKAMP_USE".

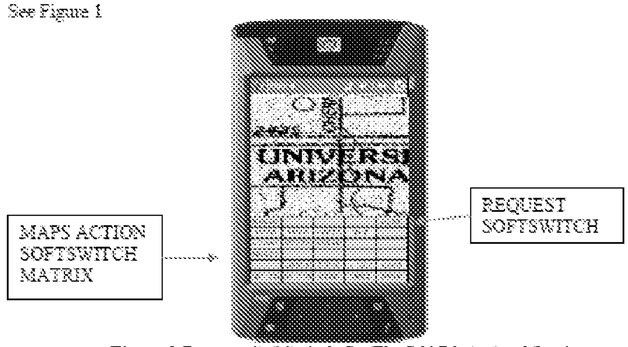
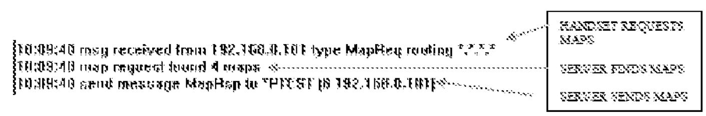
[35] *See, e.g.,* display/display.cpp at L268-581.
[36] *See, e.g.,* stat.cpp at L321-371.
[37] *See, e.g.,* stat.cpp at L387-403.
[38] *See, e.g.,* stat.cpp at L39-115.
[39] *See, e.g.,* stat.cpp at L119-166.
[40] *See, e.g.,* stat.cpp at L296-317.
[41] *See, e.g.,* stat.cpp at L207-292
[42] *See, e.g.,* inset.cpp at L708-737.
[43] *See, e.g.,* inset.cpp at L684-705.

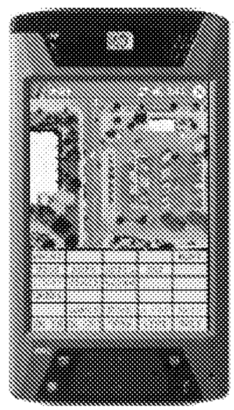
U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	The function "refresh_inset_map()"[44] is implemented in the same file and the georeferenced latitude and longitude values such as "inset_maplllon" or "inset_mapurlat" that are used to draw the map with the track information used from the "track_file" variable. The function draws the map using the system "Bit_Blt()" or call to "ww_draw_image_inset() and then calls the "inset_track_draw()" function which then draws all tracks. The function "ww_draw_image_inset()"[45] is implemented in the "display/images.cpp" and utilizes georeferenced latitude and longitude values. The function "inset_track_draw()"[46] is documented that to "plot all tracks as dots in inset window" and utilizes a loop across all tracks to draw all tracks. The function "refresh_inset_gps()"[47] is implemented in the same file and primarily provides information about the gps satelites and their respective signal strengths. The function "refresh_inset_trackamp()"[48] updates the amplified track mode equipped with its own series of functions including messaging, location and amplified views.
sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location;	The AGIS LifeRing Product practices "sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location." The first device makes a request for a georeferenced map by interacting with the device's user interface. In the file "buttons.cpp" the function "button_actions()"[49] implements the main routine for capturing the button press event and processing based on the button pressed. For the case of the "BUTTON_IMAGE_REQUEST" a call is made to the "image_server_request()" function. In the file "display/images.cpp" the function "image_server_request()"[50] is implemented. In this function the coordinate boundaries are retrieved by a call to "coord_bounds()" and sent to the function "SendImageRequest()". In the file "coords.cpp" the function "coord_bounds()"[51] is implemented where the latitude and longitude are used to generate the values required. This code is at least

[44] *See, e.g.,* inset.cpp at L285-393.
[45] *See, e.g.,* display/images.cpp at L534-596.
[46] *See, e.g.,* inset.cpp at L220-282.
[47] *See, e.g.,* inset.cpp at L395-512.
[48] *See, e.g.,* inset.cpp at L535-657.
[49] *See, e.g.,* buttons.cpp at L1344-2365.
[50] *See, e.g.,* display/images.cpp at L752-761.
[51] *See, e.g.,* coords.cpp at L88-98.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	an example of a request made to a server for a georeferenced map where the request specifies a map location, e.g., the latitude and longitude of the boundaries for the requested map. In the file "display/msgproc.cpp" the function "SendImageRequest()"[52] is implemented where a message is composed using the "DDL_MSG_MAP_REQUEST" type. The message is then processed by the "put_ddl_msg()" routine as described above and sent to the server with the call to "send_message_bulk()" as described above using the "server_ip" variable as the target for the server. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[53] is implemented to process received messages. The condition "DDL_MSG_MAP_REQUEST" is captured and a call is made to "process_map_request()". The function "process_map_request()"[54] is implemented in the same file where a message of type "DDL_MSG_MAP_RESPONSE" is formed and returns a map by calls to "put_ddl_msg()" and "send_message()". In the file "display/msgproc.cpp" the function "process_csif_msg()"[55] is implemented where the condition "DDL_MSG_MAP_RESPONSE" is processed by a call to "process_image_server_response()". In the file "display/images.cpp" the function "process_image_server_response()"[56] is implemented where it is logged that with the "response from image server".

[52] *See, e.g.,* display/msgproc.cpp at L963-996.
[53] *See, e.g.,* server/msgproc.cpp at L377-474.
[54] *See, e.g.,* server/msgproc.cpp at L312-332.
[55] *See, e.g.,* display/msgproc.cpp at L1391-1472.
[56] *See, e.g.,* display/images.cpp at L763-768.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	MAP REQUEST RQST Selecting the Map SoftSwitch from The AGIS handset's Maps Function SoftSwitch Matrix will cause the AGIS Maps Action SoftSwitch matrix to appear. The AGIS User can access more detailed map media by zooming to a specific area and selecting the RQST (REQUEST) SoftSwitch. See Figure 1  **Figure 1 Request SoftSwitch On The MAPS Action Matrix** Once selected, the AGIS Handset sends a map request via the internet to the AGIS SERVER requesting the best available map image to illustrate the center point of the Handset's tactical map area. Using the Handsets GPS coordinates, the AGIS Server searches the MAPSERVER directory and returns to the AGIS Handset four (4) appropriate aerial photographs for the correct zoom level. The AGIS SERVER's online DATA REDUCTION FILE illustrates this process. See figure 2  **Figure 2 AGIS SERVER Online Data Reduction** (The AGIS SERVER receives the request from the AGIS hand set and returns four map files.) Maps are transferred as data, exactly like photographs and video clips. Unlike that data, however, the Transferred Map media is automatically received and displayed on the Handsets Tactical Map Area, upon receipt. See **Figure 3**.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	 Figure 3 The AGIS Handset Has Displayed the Aerial Photograph on the TACTICAL MAP AREA Exhibit 16 to Beyer Declaration.
receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates;	The AGIS LifeRing product practices "receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates." The first device makes a request for a georeferenced map by interacting with the device's user interface. In the file "buttons.cpp" the function "button_actions()"[57] implements the main routine for capturing the button press event and processing based on the button pressed. For the case of the "BUTTON_IMAGE_REQUEST" a call is made to the "image_server_request()" function. In the file "display/images.cpp" the function "image_server_request()"[58] is implemented. In this function the coordinate boundaries are retrieved by a call to "coord_bounds()" and sent to the function "SendImageRequest()". In the file "coords.cpp" the function "coord_bounds()"[59] is implemented where the latitude and longitude are used to generate the values required. In the file "display/msgproc.cpp" the function "SendImageRequest()"[60] is implemented where a message is composed using the

[57] *See, e.g.*, buttons.cpp at L1344-2365.
[58] *See, e.g.*, display/images.cpp at L752-761.
[59] *See, e.g.*, coords.cpp at L88-98.
[60] *See, e.g.*, display/msgproc.cpp at L963-996.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
	"DDL_MSG_MAP_REQUEST" type. The message is then processed by the "put_ddl_msg()" routine as described above and sent to the server with the call to "send_message_bulk()" as described above using the "server_ip" variable as the target for the server. The response from the server includes the map at the specified location. Furthermore, the maps utilized by AGIS were each georeferenced and represent locations in real world locations. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[61] is implemented to process received messages. The condition "DDL_MSG_MAP_REQUEST" is captured and a call is made to "process_map_request()". The function "process_map_request()"[62] is implemented in the same file where a message of type "DDL_MSG_MAP_RESPONSE" is formed and returns a map by calls to "put_ddl_msg()" and "send_message()". In the file "display/msgproc.cpp" the function "process_csif_msg()"[63] is implemented where the condition "DDL_MSG_MAP_RESPONSE" is processed by a call to "process_image_server_response()". In the file "display/images.cpp" the function "process_image_server_response()"[64] is implemented where it is logged that with the "response from image server".

[61] *See, e.g.,* server/msgproc.cpp at L377-474.
[62] *See, e.g.,* server/msgproc.cpp at L312-332.
[63] *See, e.g.,* display/msgproc.cpp at L1391-1472.
[64] *See, e.g.,* display/images.cpp at L763-768.

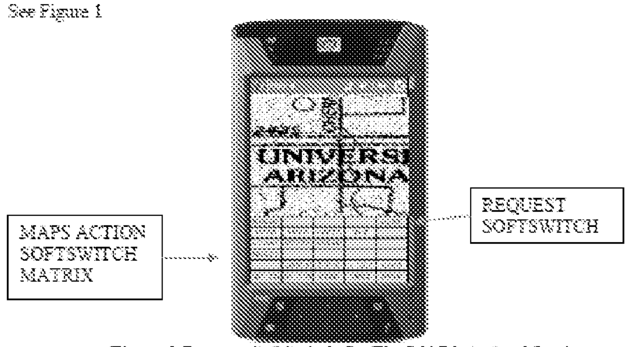
<table>
<tr><th>U.S. Patent No. 9,445,251</th><th>October 22, 2005 AGIS LIFERING</th></tr>
<tr><td></td><td>
MAP REQUEST
RQST
Selecting the Map SoftSwitch from The AGIS handset's Maps Function SoftSwitch Matrix will cause the AGIS Maps Action SoftSwitch matrix to appear. The AGIS User can access more detailed map media by zooming to a specific area and selecting the RQST (REQUEST) SoftSwitch.
See Figure 1


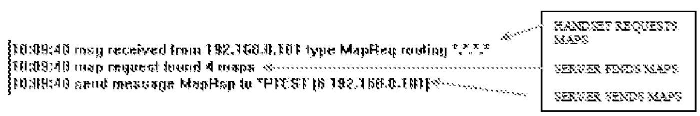
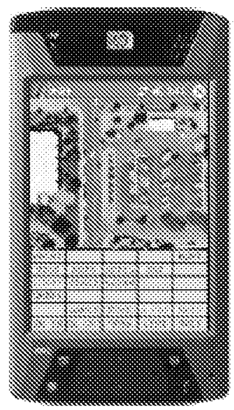
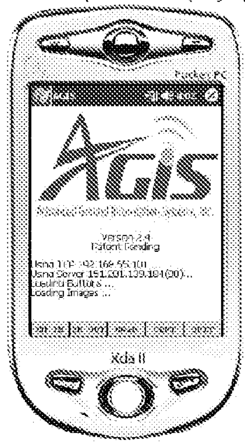
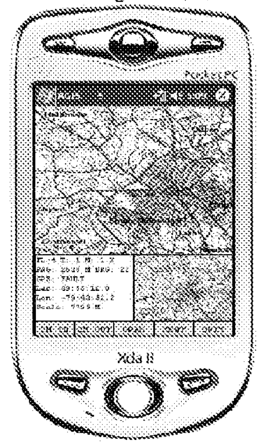
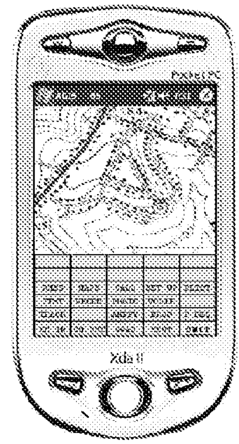
Figure 1 Request SoftSwitch On The MAPS Action Matrix
Once selected, the AGIS Handset sends a map request via the internet to the AGIS SERVER requesting the best available map image to illustrate the center point of the Handset's tactical map area.
Using the Handsets GPS coordinates, the AGIS Server searches the MAPSERVER directory and returns to the AGIS Handset four (4) appropriate serial photographs for the correct zoom level. The AGIS SERVER's online DATA REDUCTION FILE illustrates this process. See figure 2


Figure 2 AGIS SERVER Online Data Reduction
(The AGIS SERVER receives the request from the AGIS hand set and returns four map files.)
Maps are transferred as data, exactly like photographs and video clips. Unlike that data, however, the Transferred Map media is automatically received and displayed on the Handsets Tactical Map Area, upon receipt. See Figure 3.
</td></tr>
</table>

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	 Figure 3 The AGIS Handset Has Displayed the Aerial Photograph on the TACTICAL MAP AREA Exhibit 16 to Beyer Declaration.
presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices; and	The AGIS LifeRing product practices "presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map at respective positions corresponding to the locations of the second devices." The device has an interactive display showing the locations of other devices overlayed on a georeferenced map along with its own. The display of locations and tracks are described above. As depicted above, the user client of the AGIS software includes a user interface display with a map overlayed with symbols representing other members of a group at their geographic locations on the georeferenced map.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	The AGIS application software does not affect the operation of the PDA, the Cell phone or the GPS until it is activated. When the AGIS application software is selected, the AGIS PDA Cell phone is turned on, the GPS is automatically connected to the PDA / Cell phone through a Bluetooth interface and the AGIS Logo appears. Shortly thereafter the AGIS operation display appears. See Figures 1, 2, and 3  Figure 1  Figure 2  Figure 3 Exhibit 4 to Beyer Declaration, at 4.

<table>
<tr><th>U.S. Patent No. 9,445,251</th><th>October 22, 2005 AGIS LIFERING</th></tr>
<tr><td></td><td>

MAP REQUEST

RQST

Selecting the Map SoftSwitch from The AGIS handset's Maps Function SoftSwitch Matrix will cause the AGIS Maps Action SoftSwitch matrix to appear. The AGIS User can access more detailed map media by zooming to a specific area and selecting the RQST (REQUEST) SoftSwitch.
See Figure 1

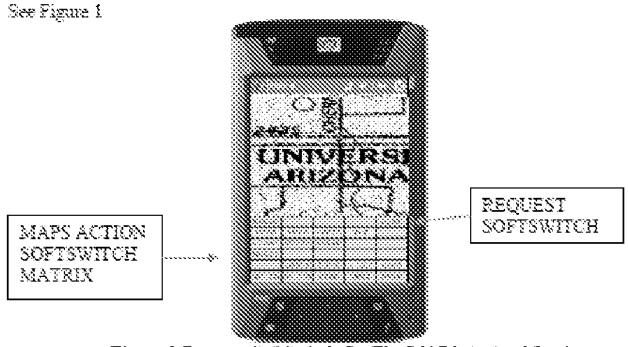


Figure 1 Request SoftSwitch On The MAPS Action Matrix

Once selected, the AGIS Handset sends a map request via the internet to the AGIS SERVER requesting the best available map image to illustrate the center point of the Handset's tactical map area.
Using the Handsets GPS coordinates, the AGIS Server searches the MAPSERVER directory and returns to the AGIS Handset four (4) appropriate aerial photographs for the correct zoom level. The AGIS SERVER's online DATA REDUCTION FILE illustrates this process. See figure 2

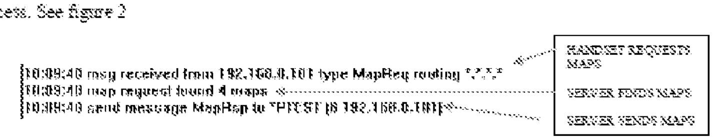
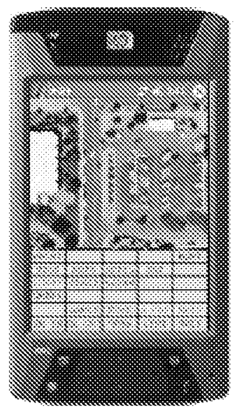


Figure 2 AGIS SERVER Online Data Reduction
(The AGIS SERVER receives the request from the AGIS hand set and returns four map files.)

Maps are transferred as data, exactly like photographs and video clips. Unlike that data, however, the Transferred Map media is automatically received and displayed on the Handsets Tactical Map Area, upon receipt. See Figure 3.

</td></tr>
</table>

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
	 Figure 3 The AGIS Handset Has Displayed the Aerial Photograph on the TACTICAL MAP AREA Exhibit 16 to Beyer Declaration. In the file "display/display.cpp" the main method "WndProc()"[65] where display events are captured and processed, the case for "WM_PAINT" where the display is drawn is implemented. When a "WM_PAINT" event occurs calls are made to "refresh_tact()", "refresh_stat()" and "refresh_inset()" passing the display window handle where the map, locations and other information gets displayed. In the file "tact.cpp" the function "refresh_tact()"[66] is implemented. There are three types of this function, two with an input handle and one without. The one without the handle first retrieves the current handle and then calls the "refresh_tact()" function version with the handle. In this function, among other features, the system draws the map by a call to "ww_draw_map()" and draws locations by a call to "track_draw()". A special "refresh_tact()" implementation only draws the tactical area which is documented to correspond to the display's WM_PAINT event and only accepts a window handle of type HWND. The function "refresh_tact()" is used throughout the code to signal an update to the view within "buttons.cpp", "display.cpp", "images.cpp", "inset.cpp",

[65] *See, e.g.,* display/display.cpp at L584-969.
[66] *See, e.g.,* display/tact.cpp at L200-306.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"maps.cpp", "msgproc.cpp", "track.cpp" and others to signal the update of the graphical window. In the file "maps.cpp" the function "ww_draw_map()"[67] is implemented. The function utilizes georeferenced latitude and longitude information to position the map using variables "urlat", "urlon", "lllat" and "lllon" for the upper right and lower left latitude and longitude. The variable array "ww_map" is used to hold the map data and is read in the same file using the function "ww_read_map()"[68] from local files. The function "ww_read_maps()"[69] makes individual calls to "ww_read_map()" to read the "world.pnt", "plit.pnt", "water.pnt" and a local map through invocation of the "local_read_map()" function pointed to by the "test user.ump" file. The "ww_read_maps()" function is called in the file "display/display.cpp" by the function "InitInstance()"[70] when the system has started. In the file "stat.cpp" the function "refresh_stat()"[71] is implemented. There are two types of this function, one with an input handle and one without. The one without the handle first retrieves the current handle and then calls the "refresh_stat()" function version with the handle. In this function, among other features, the system "refresh status window" by a call to "stat_template_refresh()". There are other methods that the system implements such as "show_status_window()"[72] and the main window process in the "WndStatProc()"[73] callback function "WM_PAINT" event which calls the "refresh_stat()" and in turn "stat_template_refresh()". Upon creation of the stat view by use of the function "create_stat()"[74] to create the graphical interface followed by a call to the "stat_template_refresh()". Furthermore, the function "refresh_stat()" is used throughout the code within "buttons.cpp", "display.cpp", "gps.cpp", "msgproc.cpp", "partlist.cpp" to signal the update of the stat window. The function "stat_template_refresh()"[75] is implemented with a call to "status_hook()". The function

[67] *See, e.g.,* maps.cpp at L213-276.
[68] *See, e.g.,* maps.cpp at L69-153.
[69] *See, e.g.,* maps.cpp at L726-732.
[70] *See, e.g.,* display/display.cpp at L268-581.
[71] *See, e.g.,* stat.cpp at L321-371.
[72] *See, e.g.,* stat.cpp at L387-403.
[73] *See, e.g.,* stat.cpp at L39-115.
[74] *See, e.g.,* stat.cpp at L119-166.
[75] *See, e.g.,* stat.cpp at L296-317.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"status_hook()"[76] is implemented where various information gets presented such as track information, current latitude, current longitude, GPS status amongst others. In the file "inset.cpp" the function "refresh_inset()"[77] is implemented. There are two types of this function, one with an input handle and one without. Both versions retrieve the Inset window handle through either a global handle or passed in handle and make a call to the "refresh_inset_window()" function. The function "refresh_inset_window()"[78] is implemented in the same file with calls made to "refresh_inset_map()", "refresh_inset_gps()" and "refresh_inset_trackamp()" based on the value held in the "inset_mode" variable set to "INSETWIN_MAP_USE", "INSETWIN_GPS_USE" or "INSETWIN_TRACKAMP_USE". The function "refresh_inset_map()"[79] is implemented in the same file and the georeferenced latitude and longitude values such as "inset_maplllon" or "inset_mapurlat" that are used to draw the map with the track information used from the "track_file" variable. The function draws the map using the system "Bit_Blt()" or call to "ww_draw_image_inset() and then calls the "inset_track_draw()" function which then draws all tracks. The function "ww_draw_image_inset()"[80] is implemented in the "display/images.cpp" and utilizes georeferenced latitude and longitude values. The function "inset_track_draw()"[81] is documented that to "plot all tracks as dots in inset window" and utilizes a loop across all tracks to draw all tracks. The function "refresh_inset_gps()"[82] is implemented in the same file and primarily provides information about the gps satelites and their respective signal strengths. The function "refresh_inset_trackamp()"[83] updates the amplified track mode equipped with its own series of functions including messaging, location and amplified views.
and identifying user interaction with the	The AGIS LifeRing product practices "identifying user interaction with the interactive display selecting one or more of the user-selectable

[76] *See, e.g.,* stat.cpp at L207-292
[77] *See, e.g.,* inset.cpp at L708-737.
[78] *See, e.g.,* inset.cpp at L684-705.
[79] *See, e.g.,* inset.cpp at L285-393.
[80] *See, e.g.,* display/images.cpp at L534-596.
[81] *See, e.g.,* inset.cpp at L220-282.
[82] *See, e.g.,* inset.cpp at L395-512.
[83] *See, e.g.,* inset.cpp at L535-657.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first device does not have access to respective Internet Protocol addresses of the second devices.	symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first device does not have access to respective Internet Protocol addresses of the second devices." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[84] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[85] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[86] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[87] and "num_on_que_list()"[88] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[89] is implemented where the free text message is transmitted directly or

[84] *See, e.g.,* buttons.cpp at L1344-2365.
[85] *See, e.g.,* display/ftext.cpp at L176-244.
[86] *See, e.g.,* netselect.cpp at L152-221.
[87] *See, e.g.,* quelist.cpp at L16-22.
[88] *See, e.g.,* quelist.cpp at L60-71.
[89] *See, e.g.,* display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[90] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[91] and subsequently to "form_net_line()"[92] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[93] and subsequently "send_net_msg_participants()"[94] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[95] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[96] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[97] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list.

[90] *See, e.g.,* netselect.cpp at L391-492.
[91] *See, e.g.,* netselect.cpp at L373-384.
[92] *See, e.g.,* netselect.cpp at L352-370.
[93] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[94] *See, e.g.,* display/msgproc.cpp at L828-922.
[95] *See, e.g.,* photo.cpp at L189-292.
[96] *See, e.g.,* server/msgproc.cpp at L377-474.
[97] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
2. The method of claim 1, wherein the data includes a short message service message, a text message, an image, or a video.	The AGIS LifeRing product practices "wherein the data includes a short message service message, a text message, an image, or a video." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[98] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[99] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[100] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[101] and "num_on_que_list()"[102] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[103] is implemented where the free text message is transmitted directly or

[98] *See, e.g.,* buttons.cpp at L1344-2365.
[99] *See, e.g.,* display/ftext.cpp at L176-244.
[100] *See, e.g.,* netselect.cpp at L152-221.
[101] *See, e.g.,* quelist.cpp at L16-22.
[102] *See, e.g.,* quelist.cpp at L60-71.
[103] *See, e.g.,* display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[104] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[105] and subsequently to "form_net_line()"[106] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[107] and subsequently "send_net_msg_participants()"[108] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[109] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[110] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[111] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list.

[104] *See, e.g.*, netselect.cpp at L391-492.
[105] *See, e.g.*, netselect.cpp at L373-384.
[106] *See, e.g.*, netselect.cpp at L352-370.
[107] *See, e.g.*, display/msgproc.cpp at L1234-1292.
[108] *See, e.g.*, display/msgproc.cpp at L828-922.
[109] *See, e.g.*, photo.cpp at L189-292.
[110] *See, e.g.*, server/msgproc.cpp at L377-474.
[111] *See, e.g.*, server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
3. The method of claim 1, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC).	The AGIS LifeRing product practices "wherein the first device is a personal digital assistant (PDA) or a personal computer (PC)." The AGIS LifeRing product includes systems and methods for executing an application on PDA/cell phones, now referred to as smartphone devices. See, e.g., "call.cpp" references to "phone" and "other phones."
4. The method of claim 1, wherein the second map is a satellite image.	The AGIS LifeRing product practices "wherein the second map is a satellite image." See, e.g., "satelliteID" in "csiftranscode.h" and "display\gps.cpp."
5. The method of claim 1, further comprising sending, by the first device, updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information comprising a previous location of the first device, displacement of the first device by a predetermined distance relative to a	The AGIS LifeRing product practices "sending, by the first device, updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information comprising a previous location of the first device, displacement of the first device by a predetermined distance relative to a previous location of the first device, or both." The AGIS software captures location information on the device through a serial communication link configured to continuously process GPS information messages. Messages such as location, precision and constellation are captured on an independent thread and when a valid message is received the thread triggers an event indicating the presence of a new message. The software then sends the new location to others through either a server or direct connection. The AGIS operator can also assign AGIS reporting intervals by time period and distance traveled since the last report. In Figure 26 the time interval for reporting has been set to every 5 minutes and distance traveled that will cause a report (even if the 5 minute interval has not lapsed) is set to 1 mile. Exhibit 4 at 11. The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
previous location of the first device, or both.	"initGPS()"[112] is implemented where the variable comport is as the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[113] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[114] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[115] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[116] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[117] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[118] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[119] is implemented where the queued GPS messages are process incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[120] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()"

[112] *See, e.g.,* csif/csif.cpp at L126-148.
[113] *See, e.g.,* csif/gpslistener.cpp at L36-51.
[114] *See, e.g.,* csif/gpslistener.cpp at L101-235.
[115] *See, e.g.,* csif/gpslistener.cpp at L158.
[116] *See, e.g.,* csif/gpslistener.cpp at L76-99.
[117] *See, e.g.,* display/display.cpp at L274-581.
[118] *See, e.g.,* display/display.cpp at L584-969.
[119] *See, e.g.,* display/gps.cpp at L346-365.
[120] *See, e.g.,* display/gps.cpp at L292-336.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[121] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" variables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[122] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[123] and are implemented in the function "button_actions()"[124] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[125] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()"

[121] *See, e.g.,* display/gps.cpp at L77-126.
[122] *See, e.g.,* display/display.cpp at L1015-1037 and L1076-1115.
[123] *See, e.g.,* display/buttons.cpp at L594-598 and L1344-1346.
[124] *See, e.g.,* display/buttons.cpp at L1344-2365.
[125] *See, e.g.,* display/msgproc.cpp at L170-325.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[126] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. The "send_track_msg()"function is used under the condition where "add_track()"[127] and "update_track_id()"[128] functions in the "display/track.cpp" is invoked. The "update_track_id()" is used in user interface as views in the "buttons.cpp" file in the main routine where button actions are implemented "button_actions()" as discussed above where the case for "BUTTON_TRACK_ID_UNKN", "BUTTON_TRACK_ID_FRND" and "BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[129] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent the "DDL_MSG_OWN_POISITION" case where the message is passed through by a call to "pass_thru_msg()" function. The file "server/server.cpp" implements the main function "WndProc()"[130] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through.
6. The method of claim 1, further comprising identifying second user interaction with the interactive	The AGIS LifeRing product practices "identifying second user interaction with the interactive display selecting at least one of the user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device."

[126] *See, e.g.,* display/msgproc.cpp at L615-682.
[127] *See, e.g.,* display/track.cpp at L373-477.
[128] *See, e.g.,* display/track.cpp at L482-510.
[129] *See, e.g.,* server/msgproc.cpp at L377-474.
[130] *See, e.g.,* server/server.cpp at L228-386.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
display selecting at least one of the user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device.	The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[131] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[132] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[133] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[134] and "num_on_que_list()"[135] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[136] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function

[131] *See, e.g.,* buttons.cpp at L1344-2365.
[132] *See, e.g.,* display/ftext.cpp at L176-244.
[133] *See, e.g.,* netselect.cpp at L152-221.
[134] *See, e.g.,* quelist.cpp at L16-22.
[135] *See, e.g.,* quelist.cpp at L60-71.
[136] *See, e.g.,* display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"create_netlist()"[137] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[138] and subsequently to "form_net_line()"[139] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[140] and subsequently "send_net_msg_participants()"[141] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[142] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. See also BUTTON_CALL and BUTTON_CCALL in buttons.h. and buttons.cpp for phone calls and conferences. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[143] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[144] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.

[137] *See, e.g.,* netselect.cpp at L391-492.
[138] *See, e.g.,* netselect.cpp at L373-384.
[139] *See, e.g.,* netselect.cpp at L352-370.
[140] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[141] *See, e.g.,* display/msgproc.cpp at L828-922.
[142] *See, e.g.,* photo.cpp at L189-292.
[143] *See, e.g.,* server/msgproc.cpp at L377-474.
[144] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
7. The method of claim 1, wherein the message from the second device is a Short Message Service (SMS) message or a text message.	The AGIS LifeRing product practices "wherein the message from the second device is a Short Message Service (SMS) message or a text message." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[145] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[146] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[147] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[148] and "num_on_que_list()"[149] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[150] is implemented where the free text message is transmitted directly or through a server as described above through the use of the

[145] *See, e.g.,* buttons.cpp at L1344-2365.
[146] *See, e.g.,* display/ftext.cpp at L176-244.
[147] *See, e.g.,* netselect.cpp at L152-221.
[148] *See, e.g.,* quelist.cpp at L16-22.
[149] *See, e.g.,* quelist.cpp at L60-71.
[150] *See, e.g.,* display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[151] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[152] and subsequently to "form_net_line()"[153] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[154] and subsequently "send_net_msg_participants()"[155] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[156] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[157] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[158] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list.

[151] *See, e.g.,* netselect.cpp at L391-492.
[152] *See, e.g.,* netselect.cpp at L373-384.
[153] *See, e.g.,* netselect.cpp at L352-370.
[154] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[155] *See, e.g.,* display/msgproc.cpp at L828-922.
[156] *See, e.g.,* photo.cpp at L189-292.
[157] *See, e.g.,* server/msgproc.cpp at L377-474.
[158] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	As described above the final leg of transmission is done by the “pass_thru_msg()” function implemented in the “server/msgproc.cpp” file.
8. The method of claim 1, wherein participating in the group further includes sending first status information to the server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second location information comprising a plurality of battery levels of the respective plurality of second devices included in the	The AGIS LifeRing product practices “wherein participating in the group further includes sending first status information to the server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second location information comprising a plurality of battery levels of the respective plurality of second devices included in the group, a plurality of signal strengths of wireless signals of the respective plurality of second devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a combination thereof.” The device utilizes a GPS receiver to locate a first device and the first device sends its location to the server using IP. In the file “gpslistener.cpp” the “start()”[159] method is implemented where a thread is created for processing in the background. The thread performs the function “process()”[160] and uses a loop to scan the comport documented as “read in a message from serial port, will only wait for 1 second for data”[161] and once a message is detected that it calls the “processGPSMessage()” passing the message and length of the message. The function “processGPSMessage()”[162] calls the “enQueueGPSMsg()” of the “DDLDatabase” class where the GPS message is queued. In the file “display.cpp” inside the initialization function “InitInstance()”[163] the GPS message event processor is registered through the call to the “registerGPSListener()” much like the DDL messages. The GPS messages are then processed by the callback event capture routine “WndProc()”[164] where the “ID_TIMER_BLINK” event

[159] *See, e.g.,* csif/gpslistener.cpp at L36-51.
[160] *See, e.g.,* csif/gpslistener.cpp at L101-235.
[161] *See, e.g.,* csif/gpslistener.cpp at L158.
[162] *See, e.g.,* csif/gpslistener.cpp at L76-99.
[163] *See, e.g.,* display/display.cpp at L274-581.
[164] *See, e.g.,* display/display.cpp at L584-969.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
group, a plurality of signal strengths of wireless signals of the respective plurality of second devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a combination thereof.	documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[165] is implemented where the queued GPS messages are process incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[166] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[167] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[168] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[169] and are implemented in the function "button_actions()"[170] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()".

[165] *See, e.g.,* display/gps.cpp at L346-365.
[166] *See, e.g.,* display/gps.cpp at L292-336.
[167] *See, e.g.,* display/gps.cpp at L77-126.
[168] *See, e.g.,* display/display.cpp at L1015-1037 and L1076-1115.
[169] *See, e.g.,* display/buttons.cpp at L594-598 and L1344-1346.
[170] *See, e.g.,* display/buttons.cpp at L1344-2365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "display/msgproc.cpp" the function "SendOwnPosString()"[171] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[172] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. In the file "display/msgproc.cpp" the functions "send_message_bulk()" and "send_message_direct()"[173] are implemented where both have IP transmit capability with the addition of "send_message_direct()" also having SMS transmission capability. Within the IP transmit capability, both functions make calls to the "sendTCPMessage()" function. In the file "csif.cpp" the "sendTCPMessage()"[174] function is implemented where it utilizes the "tcpConnection" variable to call the "enqueueForSending()" function. The "tcpConnection" variable holds the necessary information to complete an IP based transmission. In the file "csif.cpp" the function "initTCP()"[175] is implemented where the "tcpConnection" variable is instantiated. In this function a network management object "nm" from the "NetworkMgmt" class is used to call the "connectIPNetwork()" function and implement an "ipMonitor" variable to monitor the IP network. Also the "tcpConnection" variable is set to a new object by a call to "TCPServerConnection()" function.

[171] *See, e.g.,* display/msgproc.cpp at L170-325.
[172] *See, e.g.,* display/msgproc.cpp at L615-682.
[173] *See, e.g.,* display/msgproc.cpp at L47-127.
[174] *See, e.g.,* csif.cpp at L378-386.
[175] *See, e.g.,* csif.cpp at L150-204.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	See "getGPSStatus" and getDDLStatus The devices process battery levels to populate a navbar display that includes the display of the battery levels that are transmitted to the server for display and sharing with other devices. In the file "display.cpp" the function "InitInstance()"[176] and the function "WndProc()"[177] are implemented for display initialization and run time processing where calls to "display_nav_bar()" and "timed_display_nav_bar()" are made respectively. Also, there are calls to "check_ip_address()" in both initialization and run time processing where subsequent uses of navbar data is present. In the file "navbar.cpp" the function "timed_display_nav_bar()"[178] is implemented where it calls the function "display_nav_bar()" . The function "display_nav_bar()"[179] is also implemented where it is documented to "display all indicators from previous update". The function stores the result of the function "battery_check()" in the variable "bat_ret" and builds the "str" array at location [10] to empty for full battery, "b" for very low level and "B" for low level battery. The "str" variable is then copied to the "outstr" array and is used to display the result with calls to either "SHSetNavBarText()" or "system_readout()" based on the type of processor running the code where it stores the navbar data in the system memory. In the file "display.cpp" the function "check_ip_address()"[180] is implemented for two different cases of "PPC2003" or else. The case of "PPC2003" indicative of a mobile handset having battery levels the call to "display_nav_bar()" is made followed by "SendOwnPosString()" which is described above. The code is also conditioned on the use of "SMS" or not, indicative of a server. The position and battery level data are used as part of a message to be sent to either the server or other devices directly.
9. The method of claim 1, wherein the	The AGIS LifeRing product practices "wherein the first device is a smart phone."

[176] *See, e.g.,* display.cpp at L268-581.
[177] *See, e.g.,* display.cpp at L584-969.
[178] *See, e.g.,* navbar.cpp at L163-173.
[179] *See, e.g.,* navbar.cpp at L69-161.
[180] *See, e.g.,* display.cpp at L1014-1117.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
first device is a smart phone.	The AGIS LifeRing product includes systems and methods for executing an application on PDA/cell phones, now referred to as smartphone devices. See, e.g., "call.cpp" references to "phone" and "other phones."
10. The method of claim 1, further comprising: with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality of second devices; and based on transmitting the group identifier associated with the second group, participating in the second group, wherein participating in the second group includes receiving third location information from the server, the third location information comprising a plurality of locations of the respective second plurality of second	The AGIS LifeRing product practices "with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality of second devices; and based on transmitting the group identifier associated with the second group, participating in the second group, wherein participating in the second group includes receiving third location information from the server, the third location information comprising a plurality of locations of the respective second plurality of second devices included in the second group." The file "display/msgproc.cpp" implements the function "process_net_msg()"[181] where the logging function indicates "rcvd net msg" indicative of receiving a network message. The "nmsg" variable of type "NETUSERS" which is input to this function is tested against its member variable requestType which is specified in the file "csiftranscode.h"[182] and states to have "J)oining or R)equest current participants". The "nmsg.requestType" is then branched for cases of "I" for initial joining, "J" for joining, "R" for requesting and "X" for exiting the network. For the case of "J" for joining, the function makes a call to the "update_address_book()" function with sender information as the input parameters. The file "initfiles.cpp" implements the "update_address_book()"[183] where the sender information is used to find its index in the address book by a call to "find_addr_number()". The return from this call is used to update the array variable "csif_addr[]" by either adding the address if it does not exist indicative of a return value of -1 or by copying the "ipaddr" into a struct member of the indexed variable. The "csif_addr[]" array is an array of type "ADDRESS_ENTRY"[184] which is specified in the file "csiftranscode.h"[185] The members of this

[181] *See, e.g.,* display/msgproc.cpp at L1234-1292..
[182] *See, e.g.,* csiftranscode/csiftranscode.h at L254-263.
[183] *See, e.g.,* display/initfiles.cpp at L701-721.
[184] *See, e.g.,* display/initfiles.h at L 35
[185] *See, e.g.,* csiftranscode/csiftranscode.h at L45-54.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
devices included in the second group.	struct includes "name", "number", "group", "type", "level" and "ipaddr". The marking of the "csif_addr[]" variable array constitutes the joining of a group. The server further provides additional functionality in file "server/msgproc.cpp" implements the function "process_net_msg()"[186] where the logging function indicates "rcvd net msg" indicative of receiving a network message. The variables "from" of type character string and "nmsg" of type "NETUSERS" as described above is input to this function. Similar to described above, the "nmsg.requestType" is then branched for cases of "J" and not "J" described to mean of type "R" are handled. For the case of joining a group where "nmsg.requestType" equals "J", the function "update_address_book()" is called with the sender and receiver information and returns with an index into the address book array "csif_addr" as discussed above. Then the logging command indicates that the "address book [address book index] updated with ipaddr [address book ip address]" and writes "user [name] joined network" to the display. Followed by two calls to "send_net_msg_participants()" using a wildcard designator "-1" indicating everything in the respective field is to be sent. The first call sends everything in the table to the currently added user. The second call sends to everyone the currently added user. In the same file the function "send_net_msg_participants()"[187] documents that the user of "-1" in the first field indicates a "send to all" and the use of "-1" in the second field indicates a "send all net participants" condition. Similarly, the server further provides functionality to support the server side function similar to the client side in the file "server/initfiles.cpp" by implementing the function "update_address_book()"[188] where similar to the description above the call to "find_addr_number()" function retrieves an address index. If the address does not exist indicative of a return index value of "-1" calls the function "add_address_book()" otherwise updates the "csif_addr" array with the ip address input to the function. In the same file the function "add_address_book()"[189] is implemented where the "csif_addr" array is populated with the "name", "number", "group", "ipaddr" and a "isValid" member variable.

[186] *See, e.g.,* server/msgproc.cpp at L235-293.
[187] *See, e.g.,* server/msgproc.cpp at L108-191.
[188] *See, e.g.,* server/initfiles.cpp at L194-211.
[189] *See, e.g.,* server/initfiles.cpp at L163-191.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
11. The method of claim 1, wherein the data includes a voice recording.	The AGIS LifeRing product practices "wherein the data includes a voice recording." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[190] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[191] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[192] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[193] and "num_on_que_list()"[194] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[195] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable.

[190] *See, e.g.,* buttons.cpp at L1344-2365.
[191] *See, e.g.,* display/ftext.cpp at L176-244.
[192] *See, e.g.,* netselect.cpp at L152-221.
[193] *See, e.g.,* quelist.cpp at L16-22.
[194] *See, e.g.,* quelist.cpp at L60-71.
[195] *See, e.g.,* display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[196] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[197] and subsequently to "form_net_line()"[198] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[199] and subsequently "send_net_msg_participants()"[200] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[201] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[202] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[203] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.

[196] *See, e.g.,* netselect.cpp at L391-492.
[197] *See, e.g.,* netselect.cpp at L373-384.
[198] *See, e.g.,* netselect.cpp at L352-370.
[199] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[200] *See, e.g.,* display/msgproc.cpp at L828-922.
[201] *See, e.g.,* photo.cpp at L189-292.
[202] *See, e.g.,* server/msgproc.cpp at L377-474.
[203] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Once the devices have formed a group they share their location with each others. The AGIS software captures location information on the device through a serial communication link configured to continuously process GPS information messages. Messages such as location, precision and constellation are captured on an independent thread and when a valid message is received the thread triggers an event indicating the presence of a new message. The software then sends the new location to others through either a server or direct connection. The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function "initGPS()"[204] is implemented where the variable comport is as the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[205] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[206] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[207] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[208] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[209] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[210] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()".

[204] *See, e.g.,* csif/csif.cpp at L126-148.
[205] *See, e.g.,* csif/gpslistener.cpp at L36-51.
[206] *See, e.g.,* csif/gpslistener.cpp at L101-235.
[207] *See, e.g.,* csif/gpslistener.cpp at L158.
[208] *See, e.g.,* csif/gpslistener.cpp at L76-99.
[209] *See, e.g.,* display/display.cpp at L274-581.
[210] *See, e.g.,* display/display.cpp at L584-969.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "gps.cpp" the function "check_gps_msgs()"[211] is implemented where the queued GPS messages are process incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[212] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[213] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[214] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[215] and are implemented in the function "button_actions()"[216] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[217] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location

[211] *See, e.g.,* display/gps.cpp at L346-365.
[212] *See, e.g.,* display/gps.cpp at L292-336.
[213] *See, e.g.,* display/gps.cpp at L77-126.
[214] *See, e.g.,* display/display.cpp at L1015-1037 and L1076-1115.
[215] *See, e.g.,* display/buttons.cpp at L594-598 and L1344-1346.
[216] *See, e.g.,* display/buttons.cpp at L1344-2365.
[217] *See, e.g.,* display/msgproc.cpp at L170-325.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[218] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. The "send_track_msg()"function is used under the condition where "add_track()"[219] and "update_track_id()"[220] functions in the "display/track.cpp" is invoked. The "update_track_id()" is used in user interface as views in the "buttons.cpp" file in the main routine where button actions are implemented "button_actions()" as discussed above where the case for "BUTTON_TRACK_ID_UNKN", "BUTTON_TRACK_ID_FRND" and "BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[221] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent the "DDL_MSG_OWN_POISITION" case where the message is passed through by a call to "pass_thru_msg()" function. The file "server/server.cpp" implements the main function "WndProc()"[222] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to

[218] *See, e.g.,* display/msgproc.cpp at L615-682.
[219] *See, e.g.,* display/track.cpp at L373-477.
[220] *See, e.g.,* display/track.cpp at L482-510.
[221] *See, e.g.,* server/msgproc.cpp at L377-474.
[222] *See, e.g.,* server/server.cpp at L228-386.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through.
12. The method of claim 1, further comprising: using a Global Positioning Satellite (GPS) receiver of the first device to obtain data indicative of the location of the first device, wherein sending the first location information to the server comprises using the Internet Protocol (IP) to send the first location information to the server.	The AGIS LifeRing product practices "using a Global Positioning Satellite (GPS) receiver of the first device to obtain data indicative of the location of the first device, wherein sending the first location information to the server comprises using the Internet Protocol (IP) to send the first location information to the server." The device utilizes a GPS receiver to locate a first device and the first device sends it's location to the server using IP. In the file "gpslistener.cpp" the "start()"[223] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[224] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[225] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[226] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[227] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[228] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[229] is implemented where the queued GPS messages are process

[223] *See, e.g.*, csif/gpslistener.cpp at L36-51.
[224] *See, e.g.*, csif/gpslistener.cpp at L101-235.
[225] *See, e.g.*, csif/gpslistener.cpp at L158.
[226] *See, e.g.*, csif/gpslistener.cpp at L76-99.
[227] *See, e.g.*, display/display.cpp at L274-581.
[228] *See, e.g.*, display/display.cpp at L584-969.
[229] *See, e.g.*, display/gps.cpp at L346-365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[230] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[231] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[232] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[233] and are implemented in the function "button_actions()"[234] the case for pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[235] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its

[230] *See, e.g.,* display/gps.cpp at L292-336.
[231] *See, e.g.,* display/gps.cpp at L77-126.
[232] *See, e.g.,* display/display.cpp at L1015-1037 and L1076-1115.
[233] *See, e.g.,* display/buttons.cpp at L594-598 and L1344-1346.
[234] *See, e.g.,* display/buttons.cpp at L1344-2365.
[235] *See, e.g.,* display/msgproc.cpp at L170-325.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[236] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. In the file "display/msgproc.cpp" the functions "send_message_bulk()" and "send_message_direct()"[237] are implemented where both have IP transmit capability with the addition of "send_message_direct()" also having SMS transmission capability. Within the IP transmit capability, both functions make calls to the "sendTCPMessage()" function. In the file "csif.cpp" the "sendTCPMessage()"[238] function is implemented where it utilizes the "tcpConnection" variable to call the "enqueueForSending()" function. The "tcpConnection" variable holds the necessary information to complete an IP based transmission. In the file "csif.cpp" the function "initTCP()"[239] is implemented where the "tcpConnection" variable is instantiated. In this function a network management object "nm" from the "NetworkMgmt" class is used to call the "connectIPNetwork()" function and implement an "ipMonitor" variable to monitor the IP network. Also the "tcpConnection" variable is set to a new object by a call to "TCPServerConnection()" function.
13. The method of claim 1, further comprising identifying, by the first device, user interaction with the	The AGIS LifeRing product practices "identifying, by the first device, user interaction with the display selecting a particular user-selectable symbol positioned on the second georeferenced map and corresponding to a particular second device, wherein identifying the user interaction selecting the particular user-selectable symbol comprises: detecting user selection of a portion of the interactive display corresponding to a

[236] *See, e.g.,* display/msgproc.cpp at L615-682.
[237] *See, e.g.,* display/msgproc.cpp at L47-127.
[238] *See, e.g.,* csif.cpp at L378-386.
[239] *See, e.g.,* csif.cpp at L150-204.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
display selecting a particular user-selectable symbol positioned on the second georeferenced map and corresponding to a particular second device, wherein identifying the user interaction selecting the particular user-selectable symbol comprises: detecting user selection of a portion of the interactive display corresponding to a position on the second georeferenced map; based at least in part on coordinates of the selected position on the second georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented	position on the second georeferenced map; based at least in part on coordinates of the selected position on the second georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the selected position on the second georeferenced map; and identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[240] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[241] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[242] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[243] and "num_on_que_list()"[244] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[245] is

[240] *See, e.g.,* buttons.cpp at L1344-2365.
[241] *See, e.g.,* display/ftext.cpp at L176-244.
[242] *See, e.g.,* netselect.cpp at L152-221.
[243] *See, e.g.,* quelist.cpp at L16-22.
[244] *See, e.g.,* quelist.cpp at L60-71.
[245] *See, e.g.,* display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
by the selected position on the second georeferenced map; and identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position.	implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[246] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[247] and subsequently to "form_net_line()"[248] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[249] and subsequently "send_net_msg_participants()"[250] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[251] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[252] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[253] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list.

[246] *See, e.g.,* netselect.cpp at L391-492.
[247] *See, e.g.,* netselect.cpp at L373-384.
[248] *See, e.g.,* netselect.cpp at L352-370.
[249] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[250] *See, e.g.,* display/msgproc.cpp at L828-922.
[251] *See, e.g.,* photo.cpp at L189-292.
[252] *See, e.g.,* server/msgproc.cpp at L377-474.
[253] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
14. The method of claim 13, wherein identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position comprises: searching a database of entities for an entity located nearest to the spatial coordinates represented by the selected position, wherein the entities represented by data in the database include the second devices, wherein the database data include locations of the respective entities, and wherein the database is searchable by location; and based on a result of searching the database, identifying the particular second	The AGIS LifeRing product practices "wherein identifying the particular user-selectable symbol based, at least in part, on the spatial coordinates represented by the selected position comprises: searching a database of entities for an entity located nearest to the spatial coordinates represented by the selected position, wherein the entities represented by data in the database include the second devices, wherein the database data include locations of the respective entities, and wherein the database is searchable by location; and based on a result of searching the database, identifying the particular second device as the entity located nearest to the spatial coordinates represented by the selected position, wherein the particular user-selectable symbol corresponds to the particular second device." The device utilizes buttons on the display limited to other entities in a latitude and longitude limited space to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "tact.cpp" the function "refresh_tact()"[254] is implemented to draw the tactical display including when some change to location or extent of the display is made. In this function a call is made to the function "track_draw()" to draw the extent of the tracks. In the file "tact.cpp" the function "track_draw()"[255] is implemented where based on a latitude and longitude value position of a track the function "screen_clip_check()" is called and if the entity is not on the display that the processing loop exits the loop not drawing the track leaving only tracks corresponding to other entities that are close to a selected position is drawn and subsequently the user can interact with. The retrieval of the symbols from the server is performed for the displayed map location.

[254] *See, e.g.,* tact.cpp at L200-277.
[255] *See, e.g.,* tact.cpp at L557-742.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
device as the entity located nearest to the spatial coordinates represented by the selected position, wherein the particular user-selectable symbol corresponds to the particular second device.	In the file "buttons.cpp" the function "button_actions()"[256] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[257] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[258] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[259] and "num_on_que_list()"[260] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[261] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[262] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[263] and subsequently to "form_net_line()"[264] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the

[256] *See, e.g.,* buttons.cpp at L1344-2365.
[257] *See, e.g.,* display/ftext.cpp at L176-244.
[258] *See, e.g.,* netselect.cpp at L152-221.
[259] *See, e.g.,* quelist.cpp at L16-22.
[260] *See, e.g.,* quelist.cpp at L60-71.
[261] *See, e.g.,* display/msgproc.cpp at L328-399.
[262] *See, e.g.,* netselect.cpp at L391-492.
[263] *See, e.g.,* netselect.cpp at L373-384.
[264] *See, e.g.,* netselect.cpp at L352-370.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	function "process_net_msg()"[265] and subsequently "send_net_msg_participants()"[266] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[267] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[268] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[269] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
15. The method of claim 14, wherein the entity is a first entity, and wherein the method further comprises performing by the	The AGIS LifeRing product practices "wherein the entity is a first entity, and wherein the method further comprises performing by the first device: receiving user input via user interaction with the interactive display of the first device, the user input specifying a location and a symbol corresponding to a second entity other than the first device and the second devices; and based on the user input, adding the user-specified symbol to the interactive display at a position on the second

[265] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[266] *See, e.g.,* display/msgproc.cpp at L828-922.
[267] *See, e.g.,* photo.cpp at L189-292.
[268] *See, e.g.,* server/msgproc.cpp at L377-474.
[269] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
first device: receiving user input via user interaction with the interactive display of the first device, the user input specifying a location and a symbol corresponding to a second entity other than the first device and the second devices; and based on the user input, adding the user-specified symbol to the interactive display at a position on the second georeferenced map corresponding to the user-specified location of the second entity.	georeferenced map corresponding to the user-specified location of the second entity." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5. In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[270], "BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY", "BUTTON_TRACK_TYPE_VEHICLE", "BUTTON_TRACK_TYPE_INF", "BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. In the file "buttons.cpp" the function "button_actions()"[271] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action.

[270] *See, e.g.,* buttons.h at L117-132.
[271] *See, e.g.,* buttons.cpp at L1344-2365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "display/ftext.cpp" the function "create_ftext()"[272] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[273] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[274] and "num_on_que_list()"[275] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[276] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[277] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[278] and subsequently to "form_net_line()"[279] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[280] and subsequently "send_net_msg_participants()"[281] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[282] is implemented where the

[272] *See, e.g.,* display/ftext.cpp at L176-244.
[273] *See, e.g.,* netselect.cpp at L152-221.
[274] *See, e.g.,* quelist.cpp at L16-22.
[275] *See, e.g.,* quelist.cpp at L60-71.
[276] *See, e.g.,* display/msgproc.cpp at L328-399.
[277] *See, e.g.,* netselect.cpp at L391-492.
[278] *See, e.g.,* netselect.cpp at L373-384.
[279] *See, e.g.,* netselect.cpp at L352-370.
[280] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[281] *See, e.g.,* display/msgproc.cpp at L828-922.
[282] *See, e.g.,* photo.cpp at L189-292.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[283] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[284] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
16. The method of claim 15, further comprising performing by the first device: transmitting the user-specified symbol and location of the second entity to the second devices for addition of the user-specified symbol to respective interactive displays of the second devices at respective positions on respective georeferenced maps corresponding to the user-specified location of the	The AGIS LifeRing product practices "transmitting the user-specified symbol and location of the second entity to the second devices for addition of the user-specified symbol to respective interactive displays of the second devices at respective positions on respective georeferenced maps corresponding to the user-specified location of the second entity." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5. In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[285], "BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY", "BUTTON_TRACK_TYPE_VEHICLE", "BUTTON_TRACK_TYPE_INF",

[283] *See, e.g.*, server/msgproc.cpp at L377-474.
[284] *See, e.g.*, server/msgproc.cpp at L 51-61.
[285] *See, e.g.*, buttons.h at L117-132.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
second entity.	"BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. In the file "buttons.cpp" the function "button_actions()"[286] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[287] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[288] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[289] and "num_on_que_list()"[290] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[291] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable.

[286] *See, e.g.,* buttons.cpp at L1344-2365.
[287] *See, e.g.,* display/ftext.cpp at L176-244.
[288] *See, e.g.,* netselect.cpp at L152-221.
[289] *See, e.g.,* quelist.cpp at L16-22.
[290] *See, e.g.,* quelist.cpp at L60-71.
[291] *See, e.g.,* display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[292] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[293] and subsequently to "form_net_line()"[294] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[295] and subsequently "send_net_msg_participants()"[296] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[297] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[298] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[299] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.

[292] *See, e.g.,* netselect.cpp at L391-492.
[293] *See, e.g.,* netselect.cpp at L373-384.
[294] *See, e.g.,* netselect.cpp at L352-370.
[295] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[296] *See, e.g.,* display/msgproc.cpp at L828-922.
[297] *See, e.g.,* photo.cpp at L189-292.
[298] *See, e.g.,* server/msgproc.cpp at L377-474.
[299] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
17. The method of claim 16, wherein the user input further specifies information associated with the second entity, and wherein the method further comprises performing, by the first device: transmitting the user-specified information associated with the second entity to the second devices.	The AGIS LifeRing product practices "wherein the user input further specifies information associated with the second entity, and wherein the method further comprises performing, by the first device: transmitting the user-specified information associated with the second entity to the second devices." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5. The user can add information about the added/entered symbol and send to other users. Exhibit 4 at 5. In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[300], "BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY", "BUTTON_TRACK_TYPE_VEHICLE", "BUTTON_TRACK_TYPE_INF", "BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. In the file "buttons.cpp" the function "button_actions()"[301] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform

[300] *See, e.g.,* buttons.h at L117-132.
[301] *See, e.g.,* buttons.cpp at L1344-2365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[302] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[303] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[304] and "num_on_que_list()"[305] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[306] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[307] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[308] and subsequently to "form_net_line()"[309] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[310] and subsequently "send_net_msg_participants()"[311] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable.

[302] *See, e.g.,* display/ftext.cpp at L176-244.
[303] *See, e.g.,* netselect.cpp at L152-221.
[304] *See, e.g.,* quelist.cpp at L16-22.
[305] *See, e.g.,* quelist.cpp at L60-71.
[306] *See, e.g.,* display/msgproc.cpp at L328-399.
[307] *See, e.g.,* netselect.cpp at L391-492.
[308] *See, e.g.,* netselect.cpp at L373-384.
[309] *See, e.g.,* netselect.cpp at L352-370.
[310] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[311] *See, e.g.,* display/msgproc.cpp at L828-922.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[312] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[313] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[314] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
18. The method of claim 17, further comprising performing by the first device: adding data representing the spatial coordinates of the location of the second entity and data representing the information associated with the second entity to the database.	The AGIS LifeRing product practices "adding data representing the spatial coordinates of the location of the second entity and data representing the information associated with the second entity to the database." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5. The user can add information about the added/entered symbol and send to other users. Exhibit 4 at 5. In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[315],

[312] *See, e.g.,* photo.cpp at L189-292.
[313] *See, e.g.,* server/msgproc.cpp at L377-474.
[314] *See, e.g.,* server/msgproc.cpp at L 51-61.
[315] *See, e.g.,* buttons.h at L117-132.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY", "BUTTON_TRACK_TYPE_VEHICLE", "BUTTON_TRACK_TYPE_INF", "BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function "initGPS()"[316] is implemented where the variable "comport" is the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[317] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[318] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[319] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[320] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[321] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL

[316] *See, e.g.,* csif/csif.cpp at L126-148.
[317] *See, e.g.,* csif/gpslistener.cpp at L36-51.
[318] *See, e.g.,* csif/gpslistener.cpp at L101-235.
[319] *See, e.g.,* csif/gpslistener.cpp at L158.
[320] *See, e.g.,* csif/gpslistener.cpp at L76-99.
[321] *See, e.g.,* display/display.cpp at L274-581.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	messages. The GPS messages are then processed by the callback event capture routine "WndProc()"[322] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[323] is implemented where the queued GPS messages are processed incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[324] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[325] is implemented. This function utilizes the "gps_pos_decode()" function to covert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[326] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[327] and are implemented in the function "button_actions()"[328] the case for

[322] *See, e.g.,* display/display.cpp at L584-969.
[323] *See, e.g.,* display/gps.cpp at L346-365.
[324] *See, e.g.,* display/gps.cpp at L292-336.
[325] *See, e.g.,* display/gps.cpp at L77-126.
[326] *See, e.g.,* display/display.cpp at L1015-1037 and L1076-1115.
[327] *See, e.g.,* display/buttons.cpp at L594-598 and L1344-1346.
[328] *See, e.g.,* display/buttons.cpp at L1344-2365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[329] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[330] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. As a further example of receiving messages from the second devices that include location information, each time a message is received by the server from a device, the location information from that device is updated at the server. The location information for eah device in the group (e.g. the second devices) is then transmitted as part of the update message to the first device, as depicted in nmsg.numMapEntries. msgproc.cpp at ll. 862-892. The "send_track_msg()"function is used under the condition where "add_track()"[331] and "update_track_id()"[332] functions in the "display/track.cpp" is invoked. The "update_track_id()" is used in user interface as views in the "buttons.cpp" file in the main routine where button actions are implemented "button_actions()" as discussed above

[329] *See, e.g.,* display/msgproc.cpp at L170-325.
[330] *See, e.g.,* display/msgproc.cpp at L615-682.
[331] *See, e.g.,* display/track.cpp at L373-477.
[332] *See, e.g.,* display/track.cpp at L482-510.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	where the case for "BUTTON_TRACK_ID_UNKN", "BUTTON_TRACK_ID_FRND" and "BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[333] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent the "DDL_MSG_OWN_POSITION" case where the message is passed through by a call to "pass_thru_msg()" function. The file "server/server.cpp" implements the main function "WndProc()"[334] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through. In the file "buttons.cpp" the function "button_actions()"[335] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[336] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[337] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the

[333] *See, e.g.,* server/msgproc.cpp at L377-474.
[334] *See, e.g.,* server/server.cpp at L228-386.
[335] *See, e.g.,* buttons.cpp at L1344-2365.
[336] *See, e.g.,* display/ftext.cpp at L176-244.
[337] *See, e.g.,* netselect.cpp at L152-221.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"on_que_list()"[338] and "num_on_que_list()"[339] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[340] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[341] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[342] and subsequently to "form_net_line()"[343] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[344] and subsequently "send_net_msg_participants()"[345] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[346] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[347] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the

[338] *See, e.g.,* quelist.cpp at L16-22.
[339] *See, e.g.,* quelist.cpp at L60-71.
[340] *See, e.g.,* display/msgproc.cpp at L328-399.
[341] *See, e.g.,* netselect.cpp at L391-492.
[342] *See, e.g.,* netselect.cpp at L373-384.
[343] *See, e.g.,* netselect.cpp at L352-370.
[344] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[345] *See, e.g.,* display/msgproc.cpp at L828-922.
[346] *See, e.g.,* photo.cpp at L189-292.
[347] *See, e.g.,* server/msgproc.cpp at L377-474.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[348] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
19. The method of claim 15, wherein the portion of the interactive display is a first portion, wherein the position of the symbol corresponding to the particular second device is a first position, and wherein receiving the user input specifying the location of the second entity comprises: detecting user selection of a second portion of the interactive display corresponding to a second position on the second georeferenced map; and based at least in part on coordinates of the second position on the second	The AGIS LifeRing product practices "wherein the portion of the interactive display is a first portion, wherein the position of the symbol corresponding to the particular second device is a first position, and wherein receiving the user input specifying the location of the second entity comprises: detecting user selection of a second portion of the interactive display corresponding to a second position on the second georeferenced map; and based at least in part on coordinates of the second position on the second georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the second position on the second georeferenced map, wherein the location represented by the second position is the location of the second entity." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. The user can add/enter symbols on the display map and transmit them to other users. Exhibit 4 at 5. The user can add information about the added/entered symbol and send to other users. Exhibit 4 at 5. In the file "buttons.h" the button types are defined as "BUTTON_TRACK_TYPE_UNKN_GRD "[349], "BUTTON_TRACK_TYPE_ARMOR", "BUTTON_TRACK_TYPE_ARTILLARY", "BUTTON_TRACK_TYPE_VEHICLE", "BUTTON_TRACK_TYPE_INF",

[348] *See, e.g.,* server/msgproc.cpp at L 51-61.
[349] *See, e.g.,* buttons.h at L117-132.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
georeferenced map and on the data relating positions on the second georeferenced map to spatial coordinates, determining spatial coordinates of a location represented by the second position on the second georeferenced map, wherein the location represented by the second position is the location of the second entity.	"BUTTON_TRACK_TYPE_UNKN_SEA", "BUTTON_TRACK_TYPE_MIL_SEA", "BUTTON_TRACK_TYPE_COM_SEA", "BUTTON_TRACK_TYPE_PRV_SEA", "BUTTON_TRACK_TYPE_UNKN_AIR", "BUTTON_TRACK_TYPE_MIL_AIR" and "BUTTON_TRACK_TYPE_COM_AIR" where the user can assign a button type to a track, gets assigned to a track and during the process of update to other devices the track type in terms of a button symbol is also communicated. In the file "buttons.cpp" the function "button_actions()"[350] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. In the file "display/ftext.cpp" the function "create_ftext()"[351] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the "build_sending_text()"[352] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[353] and "num_on_que_list()"[354] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[355] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable.

[350] *See, e.g.,* buttons.cpp at L1344-2365.
[351] *See, e.g.,* display/ftext.cpp at L176-244.
[352] *See, e.g.,* netselect.cpp at L152-221.
[353] *See, e.g.,* quelist.cpp at L16-22.
[354] *See, e.g.,* quelist.cpp at L60-71.
[355] *See, e.g.,* display/msgproc.cpp at L328-399.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[356] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[357] and subsequently to "form_net_line()"[358] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[359] and subsequently "send_net_msg_participants()"[360] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[361] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above. In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[362] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[363] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.

[356] *See, e.g.,* netselect.cpp at L391-492.
[357] *See, e.g.,* netselect.cpp at L373-384.
[358] *See, e.g.,* netselect.cpp at L352-370.
[359] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[360] *See, e.g.,* display/msgproc.cpp at L828-922.
[361] *See, e.g.,* photo.cpp at L189-292.
[362] *See, e.g.,* server/msgproc.cpp at L377-474.
[363] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
20. The method of claim 14, wherein the database is stored on the first device.	The AGIS LifeRing product practices "wherein the database is stored on the first device." See claim 14.
21. The method of claim 14, wherein the database is stored on the server.	The AGIS LifeRing product practices "wherein the database is stored on the server." See claim 14.
22. The method of claim 1, wherein the spatial coordinates comprise latitude and longitude coordinates.	The AGIS LifeRing product practices "wherein the spatial coordinates comprise latitude and longitude coordinates." The "csif" module provides a service to start a GPSListener thread that communicates with a serial port. In the file "csif/csif.cpp" the function "initGPS()"[364] is implemented where the variable comport is as the argument. The variable "gpsListener" is then assigned to a new GPSListener class using the comport variable followed by calling the "GPSListener->start()" call. In the file "gpslistener.cpp" the "start()"[365] method is implemented where a thread is created for processing in the background. The thread performs the function "process()"[366] and uses a loop to scan the comport documented as "read in a message from serial port, will only wait for 1 second for data"[367] and once a message is detected that it calls the "processGPSMessage()" passing the message and length of the message. The function "processGPSMessage()"[368] calls the "enQueueGPSMsg()" of the "DDLDatabase" class where the GPS message is queued. In the file "display.cpp" inside the initialization function "InitInstance()"[369] the GPS message event processor is registered through the call to the "registerGPSListener()" much like the DDL messages. The GPS messages are then processed by the callback event

[364] *See, e.g.,* csif/csif.cpp at L126-148.
[365] *See, e.g.,* csif/gpslistener.cpp at L36-51.
[366] *See, e.g.,* csif/gpslistener.cpp at L101-235.
[367] *See, e.g.,* csif/gpslistener.cpp at L158.
[368] *See, e.g.,* csif/gpslistener.cpp at L76-99.
[369] *See, e.g.,* display/display.cpp at L274-581.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	capture routine "WndProc()"[370] where the "ID_TIMER_BLINK" event documented to represent a 1 second timer makes a call to "check_gps_msgs()", "check_for_internal_msgs()" and "check_for_db_msgs()". In the file "gps.cpp" the function "check_gps_msgs()"[371] is implemented where the queued GPS messages are process incrementally with a call to the "process_csif_gps_msg()". The function "process_csif_gps_msg()"[372] is implemented where the message is retrieved by a call to "get_gps_msg()" and based on one of three choices of a message type captured by the "msgType" member of the message variable "gmsg" that the position, precision or constellation of the gps message is processed. For the case where the "msgType" is of "GPS_MSG_POSITION" type the call is made to "process_gps_pos()" passing the "gmsg.body.position" variable member. In the same file the function "process_gps_pos()"[373] is implemented. This function utilizes the "gps_pos_decode()" function to convert the input message's members such as "gpsp.latitude" and "gpsp.longitude" to a floating point format in the "lat" and "lon" vaiables. Further into the function, the "lat" and "lon" variables are used to populate the members of the "temptrk" struct variable which is of type "track_file_struct". In the file "display/display.cpp" the function "check_ip_address()"[374] is an example of communication with transmission of location information. This function is conditioned to use certain code for the case where the "PPC2003" is defined, indicative of SMS capability, or not defined. In both routines similar functionality is performed where calls to functions "send_net_msg_netmgmt()" to join a network group and "SendOwnPosString()" to send location information under various connection states. In the file "display/buttons.cpp" in the main routine where it is documented that button actions are processed by "button_actions"[375] and are implemented in the function "button_actions()"[376] the case for

[370] *See, e.g.,* display/display.cpp at L584-969.
[371] *See, e.g.,* display/gps.cpp at L346-365.
[372] *See, e.g.,* display/gps.cpp at L292-336.
[373] *See, e.g.,* display/gps.cpp at L77-126.
[374] *See, e.g.,* display/display.cpp at L1015-1037 and L1076-1115.
[375] *See, e.g.,* display/buttons.cpp at L594-598 and L1344-1346.
[376] *See, e.g.,* display/buttons.cpp at L1344-2365.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	pressing the "BUTTON_PING:" is implemented where calls are made to "SendOwnPosString()" and "send_all_tracks()". In the file "display/msgproc.cpp" the function "SendOwnPosString()"[377] is implemented where the "op" struct variable of type "OWN_POSITION" is populated with location information such as latitude and longitude. Then the "dmsg" variable of type "DDLMSG" is set to have a "msgType" of "DDL_MSG_OWN_POSITION" and variable "op" set to one of its other member variables. Then the "dmsg" is set to a buffer with the call to "put_ddl_msg()" and a call to "send_message_bulk()" when the transmit method "xmit_method" is set to "SERVER" and using a "server_ip" variable. Alternatively, direct transmission may occur with the call to "send_message_direct()" Similarly, in the file "display/msgproc.cpp" the function "send_track_msg()"[378] is implemented which includes location information along with previous location information. Similar to the previous function, the "dmsg" is set to type "DDL_MSG_TRACK_REPORT" and "tmsg" variable with location information is set to its member variable. The functions "put_ddl_msg()" followed by "send_message_bulk()" or "send_message_direct()" transmit the message. The "send_track_msg()"function is used under the condition where "add_track()"[379] and "update_track_id()"[380] functions in the "display/track.cpp" is invoked. The "update_track_id()" is used in user interface as views in the "buttons.cpp" file in the main routine where button actions are implemented "button_actions()" as discussed above where the case for "BUTTON_TRACK_ID_UNKN", "BUTTON_TRACK_ID_FRND" and "BUTTON_TRACK_ID_HOSTILE" cases are handled for various features including the call to "update_track_id()". On the server side, the file "server/msgproc.cpp" implements the function "process_csif_ddl_msg()"[381] where the incoming message uses the "msgType" to branch to a section of the code depicted to represent

[377] *See, e.g.,* display/msgproc.cpp at L170-325.
[378] *See, e.g.,* display/msgproc.cpp at L615-682.
[379] *See, e.g.,* display/track.cpp at L373-477.
[380] *See, e.g.,* display/track.cpp at L482-510.
[381] *See, e.g.,* server/msgproc.cpp at L377-474.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	the "DDL_MSG_OWN_POISITION" case where the message is passed through by a call to "pass_thru_msg()" function. The file "server/server.cpp" implements the main function "WndProc()"[382] as a callback where a receipt of a message triggers its execution. The case of "CSIF_DDL_MSG_AVAILABLE" is used to call the "getNextDDLMessage()" with a variable "route" that is used to call the "process_csif_ddl_msg()" function as described above to make the call for pass through.
23. The method of claim 1, further comprising performing, by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating voice-over-IP (VOIP) communication with the particular second device.	The AGIS LifeRing product practices "performing, by the first device: identifying user interaction with the interactive display selecting a particular user-selectable symbol corresponding to a particular second device and user interaction with the display specifying an action and, based thereon, initiating voice-over-IP (VOIP) communication with the particular second device." The device utilizes buttons on the display to send messages, make calls and send photo or video. The device transmits a message to the server using a participant list and the server makes the final delivery of messages and the device does not have access to the Internet Protocol addresses of the recipients. In the file "buttons.cpp" the function "button_actions()"[383] implements the main routine for capturing the button press event and processing based on the button pressed. For example, cases for "BUTTON_MSG_FREETEXT", "BUTTON_NET", "BUTTON_MSG_PHOTO" and "BUTTON_MSG_VIDEO" perform an action to use Internet Protocol to send data via a server. Calls are made to "create_ftext()", "create_netlist()" and "create_photo()" perform these actions based on the user action. See also BUTTON_CALL and BUTTON_CCALL in buttons.h. and buttons.cpp. In the file "display/ftext.cpp" the function "create_ftext()"[384] is implemented where the display is configured to receive user interface for communicating through free text. The call to "build_sending_text()" builds the receiving participants and in file "netselect.cpp" the

[382] *See, e.g.,* server/server.cpp at L228-386.
[383] *See, e.g.,* buttons.cpp at L1344-2365.
[384] *See, e.g.,* display/ftext.cpp at L176-244.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	"build_sending_text()"[385] function is implemented. List of participants is organized in the "indiv" array or the que_list based on the usage of the build and que_list is used via the "on_que_list()" and "num_on_que_list()" functions. The file "quelist.cpp" implements the "on_que_list()"[386] and "num_on_que_list()"[387] functions where the "que_list" array is maintained. Once the message is compiled the message is send via "send_ftext_msg()" function. In the file "display/msgproc.cpp" the function "send_ftext_msg()"[388] is implemented where the free text message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. Similarly for the case of "BUTTON_NET" a call is made to "create_netlist()". In the file netselect.cpp" the function "create_netlist()"[389] is implemented where the display is configured for net communication. The call to "send_to_display_net_list()"[390] and subsequently to "form_net_line()"[391] forms the list of participants to receive the communication. In the file "display/msgproc.cpp" the function "process_net_msg()"[392] and subsequently "send_net_msg_participants()"[393] the message is transmitted directly or through a server as described above through the use of the "send_message_direct()" and "send_message_bulk()" using the "server_ip" variable. For the case of "BUTTON_MSG_PHOTO" and "BUTTON_ MSG_VIDEO" the call to "create_photo()" is made. In the file "photo.cpp" the function "create_photo()"[394] is implemented where the directly listing of either photo or video files are gathered and displayed to the user eventually followed by the call to the function "build_sending_text()" similar to the free text mode above.

[385] *See, e.g.,* netselect.cpp at L152-221.
[386] *See, e.g.,* quelist.cpp at L16-22.
[387] *See, e.g.,* quelist.cpp at L60-71.
[388] *See, e.g.,* display/msgproc.cpp at L328-399.
[389] *See, e.g.,* netselect.cpp at L391-492.
[390] *See, e.g.,* netselect.cpp at L373-384.
[391] *See, e.g.,* netselect.cpp at L352-370.
[392] *See, e.g.,* display/msgproc.cpp at L1234-1292.
[393] *See, e.g.,* display/msgproc.cpp at L828-922.
[394] *See, e.g.,* photo.cpp at L189-292.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
	In the file "server/msgproc.cpp" the function "process_csif_ddl_msg()"[395] is implemented as described above with various conditions. For the default condition catching messages for Free Text, Net, Photo and Video messaging, the code calls the "pass_thru_msg()" function. In the file "server/msgproc.cpp" the function "pass_thru_msg()"[396] is implemented where the "trouting" variable is parsed to populate the address list for the recipients using the "parse_routing()" function followed by the call to "send_message()" to each individual in the list. As described above the final leg of transmission is done by the "pass_thru_msg()" function implemented in the "server/msgproc.cpp" file.
24. A system comprising: a first device programmed to perform operations comprising:	See claim 1.
receiving a message from a second device, wherein the message relates to joining a group;	See claim 1.
based on receiving the message from the second device, participating in the group, wherein participating in the group includes sending first location information to a server and receiving second	See claim 1.

[395] *See, e.g.,* server/msgproc.cpp at L377-474.
[396] *See, e.g.,* server/msgproc.cpp at L 51-61.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
location information from the server, the first location information comprising a location of the first device, the second location information comprising a plurality of locations of a respective plurality of second devices included in the group;	
presenting, via an interactive display of the first device, a first interactive, georeferenced map and a plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the first georeferenced map at respective positions corresponding to the locations of the second devices, and wherein the first georeferenced map includes data relating positions on the first georeferenced map to spatial	See claim 1.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
coordinates;	
sending, from the first device to the server, a request for a second georeferenced map different from the first georeferenced map, wherein the request specifies a map location;	See claim 1.
receiving, from the server, the second georeferenced map, wherein the second georeferenced map includes the requested location and data relating positions on the second georeferenced map to spatial coordinates;	See claim 1.
presenting, via the interactive display of the first device, the second georeferenced map and the plurality of user-selectable symbols corresponding to the plurality of second devices, wherein the symbols are positioned on the second georeferenced map	See claim 1.

U.S. Patent No. 9,445,251	**October 22, 2005 AGIS LIFERING**
at respective positions corresponding to the locations of the second devices;	
and identifying user interaction with the interactive display selecting one or more of the user-selectable symbols corresponding to one or more of the second devices and positioned on the second georeferenced map and user interaction with the display specifying an action and, based thereon, using an Internet Protocol to send data to the one or more second devices via the server, wherein the first device does not have access to respective Internet Protocol addresses of the second devices.	See claim 1.
25. The system of claim 24, wherein the data includes a short message service message, a text message, an	See claim 2.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
image, or a video.	
26. The system of claim 24, wherein the first device is a personal digital assistant (PDA) or a personal computer (PC).	See claim 3.
27. The system of claim 24, wherein the second map is a satellite image.	See claim 4.
28. The system of claim 24, wherein the operations further comprise sending updated location information comprising an updated location of the first device, the updated location information being sent based on passage of a predetermined time interval since sending previous location information comprising a previous location of the first device, displacement of the first device by a predetermined distance relative to a previous location of the first device, or	See claim 5.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
both.	
29. The system of claim 24, wherein the operations further comprise identifying second user interaction with the interactive display selecting at least one of the user-selectable symbols corresponding to at least one of the second devices and user interaction with the display specifying an action and, based thereon, initiating a phone call or phone conference with the at least one second device.	See claim 6.
30. The system of claim 24, wherein the message from the second device is a Short Message Service (SMS) message or a text message.	See claim 7.
31. The system of claim 24, wherein participating in the group further includes sending first status information to the	See claim 8.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
server and receiving second status information from the server, the first status information comprising a battery level of the first device, a signal strength of a wireless signal of the first device, a status of a Global Positioning Satellite (GPS) receiver of the first device, or a combination thereof, the second location information comprising a plurality of battery levels of the respective plurality of second devices included in the group, a plurality of signal strengths of wireless signals of the respective plurality of second devices included in the group, a plurality of statuses of GPS receivers of the respective plurality of second devices included in the group, or a combination thereof.	
32. The system of claim 24, wherein	See claim 9.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
the first device is a smart phone.	
33. The system of claim 24, wherein the operations further include: with the first device, transmitting a group identifier associated with a second group, the second group including a second plurality of second devices; and based on transmitting the group identifier associated with the second group, participating in the second group, wherein participating in the second group includes receiving third location information from the server, the third location information comprising a plurality of locations of the respective second plurality of second devices included in the second group.	See claim 10.
34. The system of claim 24, wherein the data includes a	See claim 11.

U.S. Patent No. 9,445,251	October 22, 2005 AGIS LIFERING
voice recording.	
35. The system of claim 24, wherein the first device includes a Global Positioning Satellite (GPS) receiver, and wherein the operations further include: using the GPS receiver to obtain data indicative of the location of the first device, wherein sending the first location information to the server comprises using the Internet Protocol (IP) to send the first location information to the server.	See claim 12.

I declare under the penalty of perjury that the foregoing is true and correct to the best of my knowledge.

Executed this 24th day of March, 2021 in Jupiter, FL.

Malcolm K. Beyer, Jr.

EXHIBIT 1

From:	BeyerM@aol.com
Sent:	Thursday, August 12, 2004 9:10 PM
To:	sropson@raytheon.com
Subject:	AGIS SMS STDS
Attachments:	smsmsg_icd - for merge1.doc

Steve,

It was a pleasure to meet you telephonically earlier today. As agreed, attached you will find a copy of our current AGIS SMS message standards.

As we discussed, I will generate a Statement of Work based on an incremental approach with demonstrate able milestones at the end of each increment. Each increment will have a brief (less than 5 page) test plan, which will describe the incremental steps to meet the milestone. The test plan will also describe the jointly derived criteria for meeting the milestone. The first three increments will be independently priced; the fourth increment will require further definition prior to costing.

If you agree, there will be four increments:

Increment One

Transmission of Raytheon data to AGIS for processing and display. This phase will consist of first sending SMS messages (in AGIS format) and then TCP/IP messages from the Raytheon system over a cellular network to an AGIS which will receive, process and display the data.

Increment Two

Transmission (using TCP/IP) of AGIS unit locations and simulated targets observed by the AGIS operator to the Raytheon system.

Increment Three

Transmission of snapshots and video clips from Raytheon to an AGIS over a cellular network. (The snapshots and film clips will be in a jointly agreed Micro Soft compatible format.) AGIS will receive and display the data on the same display that is used for the tactical communications and permit the operator to access the snapshots and video in a rapid manner and permit the AGIS operator to easily shift between the images and the tactical situation display.

Increment Four

Transmission of Commands from the Raytheon system to the AGIS operator and responses to those commands by the AGIS operator (this phase is largely undefined as the data to be exchanged has yet to be defined)

EXHIBIT 2

RAYTHEON / ADVANCED GROUND INFORMATION SYSTEMS (AGIS)
STATEMENT OF WORK

Raytheon has developed a Home Land defense system that is capable of receiving, processing, fusing and displaying data from various sensors. AGIS proposes to provide Raytheon a PDA/Phone system that will permit the reception and display of data transmitted by this system to an in the field user. AGIS will also modify its system to transmit back to the Raytheon system position location, status and local imagery information from each of the AGIS units. This fixed price milestone effort will be broken down into demonstration increments. At the end of each increment, a test will be conducted according to mutually agreed to procedures. Progression from one phase to another is dependent on the successful completion of an increment. The deliverables of each phase will consist of software and an operator's manual that describes the operation of the PDA/Phone/GPS system (referred to as AGIS)

Increment One - Transmission of Raytheon data to an AGIS cell phone / PDA for processing and display.

This phase will consist of first sending SMS messages (in an existing AGIS format) and then TCP/IP messages from the Raytheon system over a cellular network to an AGIS which will receive, process and display the data. The proposed progression of this increment is as follows:

1. A copy of the proprietary AGIS message standards will be delivered to Raytheon in an electronic format. (This has already been done)
2. Raytheon will use the standards to create a Word document that will contain two samples of each type of AGIS message type. Raytheon will then E mail these messages to AGIS.
3. Raytheon will develop software to convert their system's internal data to an AGIS format and transmit that data first via an E mail (for testing) and then via cell phone track SMS messages to an AGIS cell phone / PDA.
4. The AGIS will receive and display the Raytheon SMS track data and correctly geographically superimposed on a map. Proof that the system functions will consist of AGIS screen shots and hook read of the track information
5. Raytheon will then send the same or different tracks to AGIS using TCP/IP communications.
6. The AGIS will receive and display the Raytheon SMS track data and correctly geographically superimposed on a map. Proof that the system functions will consist of AGIS screen shots and hook read of the track information

Increment Two - TCP/IP transmission of AGIS unit locations and simulated targets observed by the AGIS operator to the Raytheon system.

This increment will consist of the AGIS transmitting multiple AGIS locations and locally derived target information to Raytheon via cell phone. The AGIS units will report their location as they move in accordance with a variable interval depending on their speed of movement. When the AGIS units are stationary, they will report their positions in accordance with a pre-established rate.

The proposed progression of this increment is as follows:

1. AGIS will write a data reduction program which will record all and reduce AGIS message traffic for analysis. This data reduction software will be provided to Raytheon to use in analyzing the AGIS message transmissions. AGIS will send a copy of actual message traffic to Raytheon for data reduction and concurrence that the messages are in the correct format and are sent at the correct rate.
2. AGIS will set up a test bed consisting of three phones and will cause the phones to first operate at a fixed location and then move. Data will be collected at AGIS and at Raytheon to document the messages that were transmitted.

An option to Increment Two is to provide the AGIS operator the ability to enter building floor number where he is located.

The proposed progression of this option is as follows: a) find and purchase a head sets that are advertised to operate in a high noise environment, b) test the head set with Micro Soft voice recognition software, c) decide whether to proceed based on the outcome of the testing, d) if the voice recognition appears to work in a high noise environment develop the software that will cause floor data to be entered into AGIS and generate a position update message upon recognition of floor level.

Increment Three – Exchange of snapshots and video clips

This increment will consist of Raytheon and AGIS exchanging (via ???) snapshots and film clips (in a jointly agreed Micro Soft compatible format) Modifications will be made to the AGIS display to permit the AGIS operator easily shift from the tactical display to take and display snapshots and video clips and to then easily shift back to the tactical situation display.

The proposed progression of this increment is as follows:

1. Raytheon will E mail a file of an image to AGIS, AGIS will process and display the image. AGIS will E mail an image to Raytheon who will process and display the image.
2. Raytheon will E mail a file of a video clip to AGIS, AGIS will process and display the video clip. AGIS will E mail a video clip to Raytheon who will process and display the image.
3. Raytheon and AGIS will exchange (via???) images and video clips

Increment Four – Encryption

This increment will consist of encrypting the exchanged track, image and video clip information that is exchanged between Raytheon and AGIS. Raytheon and AGIS will mutually agree on an encryption method that is compatible with cellular communications. Once that has been achieved, tests similar to the above will be conducted to prove that data can be exchanged in the same manner whether the systems are operating in an encrypted mode of in the clear.

Increment Five - Transmission of Commands from the Raytheon system to the AGIS operator and responses to those commands by the AGIS operator.

This increment is largely undefined as the data to be exchanged has yet to be defined.

EXHIBIT 4

From: BeyerM@aol.com
To: sandelb@yahoo.com
Subject: SBIR
Date: Thursday, January 13, 2005 2:02:59 PM
Attachments: SBIR12a.pdf

Sandel,

Attached is a copy of the final submittal.

I will be sending you separate file of the costing.

Regards,

Cap

1.0 Identification and Significance of the Applicability of AGIS to Enhancing Tactical Decision-Making in Navy Seal Operations

Advanced Ground Information Systems (AGIS), Inc. is developing for first responders and military security applications PDA / Cell phone / GPS software that can be readily enhanced to meet the Navy's SBIR N05-069 Enhancing Tactical Decision-Making Requirements. The problem addressed by this proposed SBIR Phase 1 effort is how to meet the Seal Tactical Decision-Making operational requirements in a generalized manner that provides an enhanced product for Navy Seal Operations while at the same time permitting the previously developed technology that has been enhanced by our SBIR Phase I effort to be also utilized for military security and first responder requirements. Our approach will offer the Seals the ability to use small light weight, low power usage PDAs to pass their location and other data between themselves and to their Toughbook PCs which connect to the ViaSat terminals through a ViaSat interface card. Our proposal has many color photographs of the proposed system. Unfortunately, we have found out that the evaluators will not be able to clearly see these color photographs. We will be most pleased to provide these color photographs at any time.

2.0 Phase I Technical Objectives

To understand our technical objectives it is first necessary to understand AGIS, Inc.'s Advanced Ground Information System (AGIS) PDA/Phone/Camera/GPS personnel collaboration software. Our AGIS software can reside in one of many commercial off the shelf PDA/Phone/Camera/GPS Microsoft devices. Our AGIS software enables all users to rapidly view their location (depicted using **Mil-Std-2525** symbols superimposed on a map) and status information and the location and status information of all other AGIS software equipped cell phones. One's location data is obtained from a small GPS unit which is connected (via a Bluetooth link) to the phone. This location data and the identification of and status of the AGIS unit are then automatically digitally transmitted to all other AGIS units.

The AGIS software is designed to provide a means to rapidly establish voice, data, photographic and video clip communications between the AGIS software users. The AGIS operator selects to display and transmit information by touching switches drawn on the PDA display (a SoftSwitch) or by touching (hooking) symbols appearing on the LCD using a stylus or his/her finger. This action activates the SoftSwitch or causes information concerning the hooked item to appear in an auxiliary readout area at the bottom of the display. The operator can enter and automatically transmit items of interest (track) data to other AGIS users by pointing at the map location where he wants the track to appear. After the operator assigns characteristics (Unknown, Friend, Hostile; Infantry, Armor, Artillery, etc.) using the appropriate SoftSwitches, the track is transmitted to all other netted AGIS units. The AGIS operator can send free text, formatted messages, photographs and video clips to other users by hooking their map

Symbols. Furthermore, he can rapidly and selectively voice conference AGIS users by hooking their location and, using a similar method, can specify subgroups to which he desires to send free text messages, fixed formatted messages, photographs and video clips. If there are established subgroups with whom the AGIS user normally has voice communications and sends digital data (i.e. his higher headquarters, and adjacent units), he can assign them to one of four communications nets.

The operator can then send voice conference or digital data to those AGIS equipped units by simply selecting their assigned NET SoftSwitch.

The AGIS application software does not affect the operation of the PDA, the Cell phone or the GPS until it is activated. When the AGIS application software is selected, the AGIS PDA Cell phone is turned on, the GPS is automatically connected to the PDA / Cell phone through a Bluetooth interface and the AGIS Logo appears. Shortly thereafter the AGIS operation display appears. See Figures 1, 2, and 3

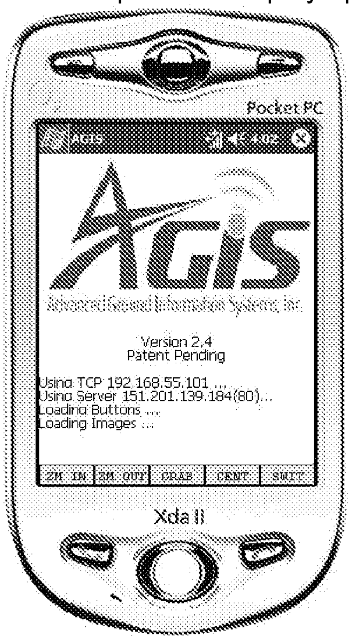


Figure 1

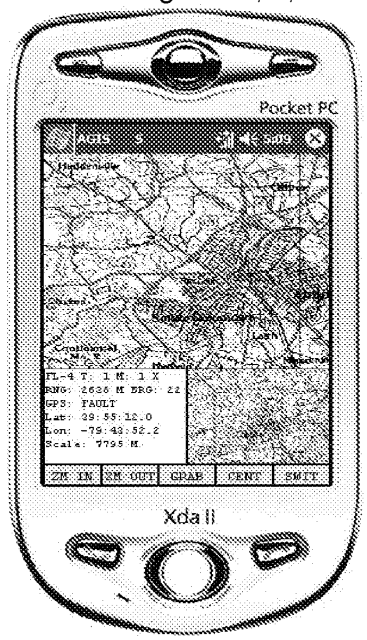


Figure 2

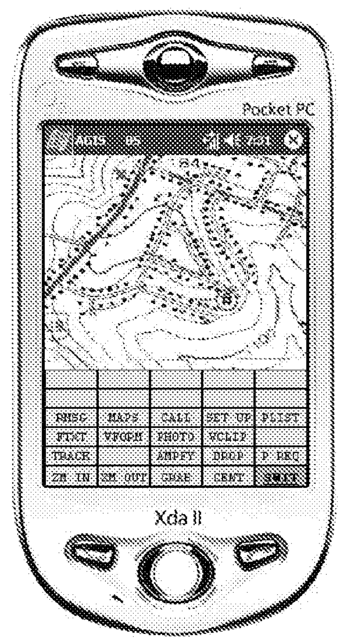


Figure 3

The five SoftSwitches at the bottom of the display (Figures1,2 and 3) are always present. They provide the operator the ability to zoom in or out on the map, to GRAB and adjust the map or to center the map immediately with a single SoftSwitch action. However, the SWIT SoftSwitch at the bottom right of the PDA display is different. When touched, it causes an AGIS SoftSwitch Function Matrix to appear. See Figure 3. The SoftSwitch Function Matrix allows the AGIS operator to select the type of function he desires to perform (i.e. enter a track, send a digital message, etc.). When the AGIS operator selects a Function Matrix SoftSwitch, the Function Matrix changes to the SoftSwitch Action Matrix. These Action Matrix SoftSwitches are used to accomplish the desired task. As can be seen above, there are 12 blank Function Matrix SoftSwitches. During the Phase 1 SBIR, AGIS, Inc. proposes to add additional Function Matrix SoftSwitches to provide rapid access to the additional software that is necessary to meet the stated Seal operational specific requirements. Furthermore, AGIS, Inc. proposes to generalize the software associated with adding these SoftSwitch functions so that the underlying code can be used for first responder and other military security requirements.

2.1 Use of SoftSwitches for Entering Acquired Data and Sharing Information

The best way to explain how AGIS SoftSwitches processing will be used to meet the Navy Seal Operational requirements is to explain AGIS's current capability to rapidly provide a collaborative means for acquiring, processing and sharing updated information for making improved tactical decisions.

How AGIS SoftSwitches are used to provide the ability to quickly transmit tracks between AGIS net participants is described in the following scenario. Upon activating their AGIS's, Units FL 1 and FL 4 commence automatically exchanging location and status information. In the example below, the FL 1 AGIS operator has hooked his symbol (indicated by the circle surrounding his symbol) and has detected a tank and wants to report its location to AGIS FL 4 (the green Infantry symbol next to the road). To report the location of the tank, the AGIS FL1 operator selects the SWIT SoftSwitch. See Figure 6. This action causes the SoftSwitch Function Matrix to appear. See Figure 7. The operator selects the TRACK SoftSwitch and touches the point on the map display where he sees the tank is located and selects the HOST, ARMR, and MOVE SoftSwitches. See Figure 8. This action causes a tracked vehicle (the red symbol) to be reported at the correct location to AGIS FL 4. The FL1 operator then decides to amplify the information by returning to the SoftSwitch matrix and selecting the AMPFY SoftSwitch and enters the desired data into the text entry using the Microsoft soft keyboard that is depicted in Figure 9. When the operator is finished entering the desired data, he selects OK which causes the amplifying information to appear on his and FL2's AGIS displays. See Figure 10. When the tank is destroyed, the FL1 AGIS operator selects the DROP SoftSwitch which causes the track to disappear. See Figure 11.

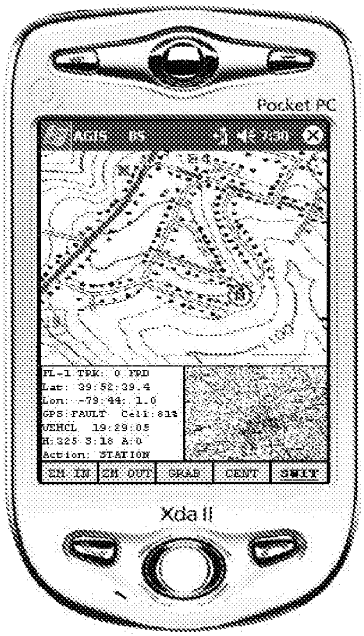


Figure 6

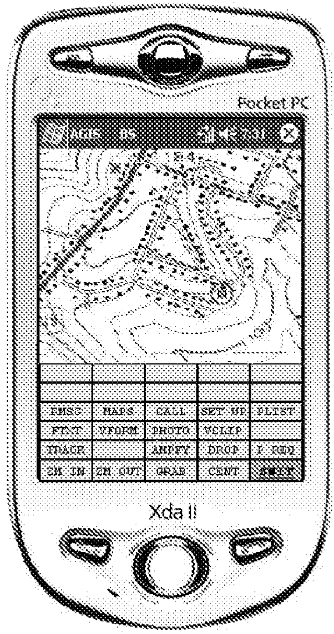


Figure 7

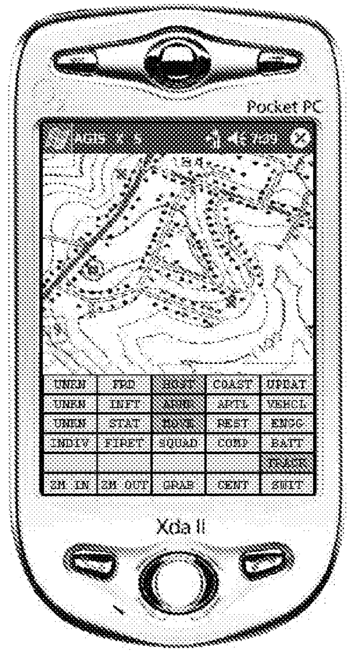
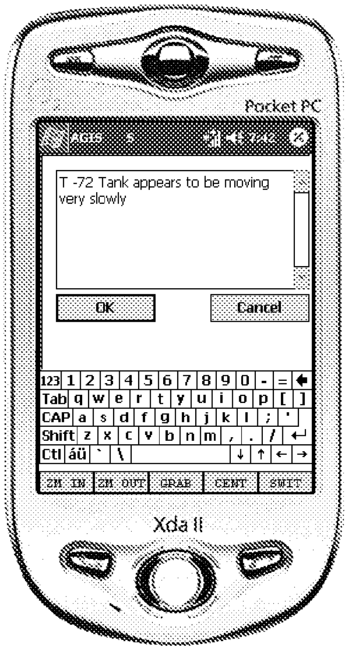
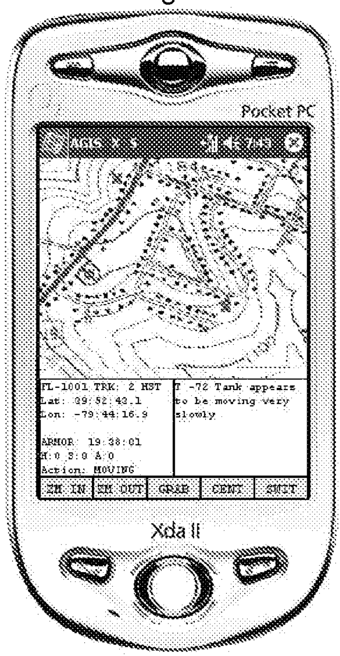


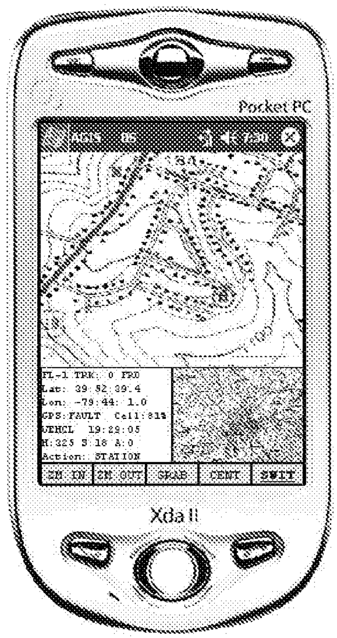
Figure 8



FL 1
Figure 9



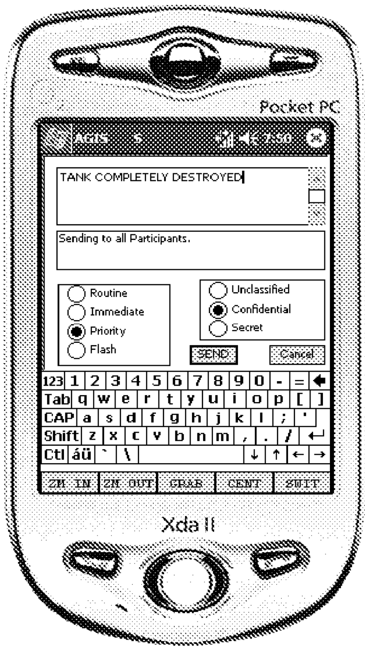
FL 4
Figure 10



FL 1
Figure 11

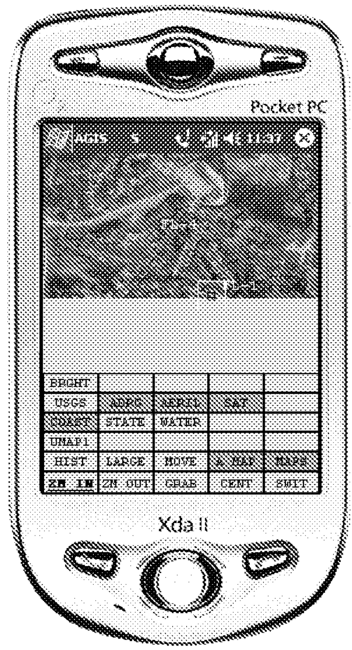
2.2 Transmission and Receipt of Digital Messages Between AGISs

To transmit Free Text messages to other AGIS net participants, the AGIS operator hooks the AGIS to whom he desires to send a Free Text message and then selects the FXTX SoftSwitch. If he desires to send the same message to more than one AGIS net participant, he sequentially hooks the AGIS units, to which the Free Text message is to be transmitted. If the operator doesn't hook any of the AGIS units, the AGIS transmits the digital data to all other AGIS units. Upon selecting the FXTX SoftSwitch, the Free Text message form appears. In this example FL1 is sending to all AGIS net participants the message confirming that the observed tank has been completely destroyed. FL 4 then receives an audio alert (audio can be silenced) informing him that he has received a message. Additionally, a box appears around FL1 indicating the message to FL 4 and the identity of the message sender. See Figure 13. When the FL 4 operator hears that a message has been received and / or sees the box appear, he selects the Read Message (RMSG) SoftSwitch to activate that function. See Figure 14. The FL 4 operator selects the most recently received message by touching the screen and views the contents of the message in the right hand readout area at the bottom of the display.



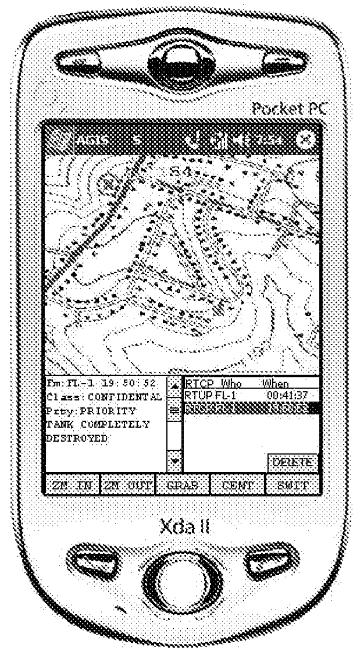
FL 1

Figure 12



FL 4

Figure 13

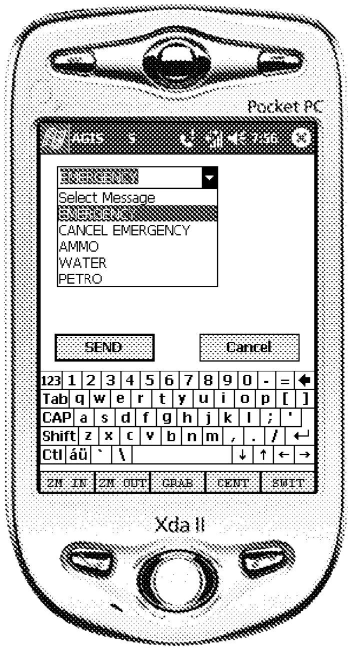


FL 4

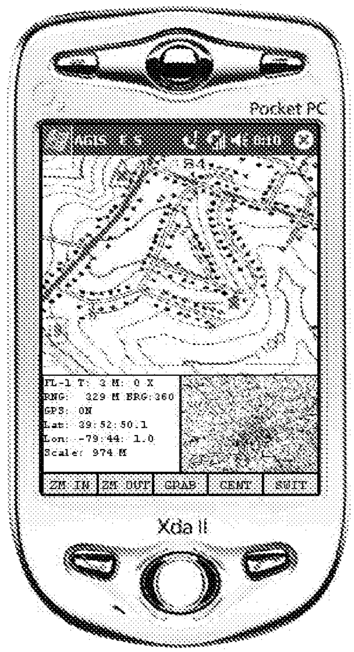
Figure 14

If the AGIS operator is constantly sending Fixed Text messages to the same group of AGIS net participants, he can assign them to a Net and then transmit to that group by selecting that net prior to entering the Free Text message.

Transmission of Variable Formatted messages is implemented in the same manner as the transmission of Free Text data. The AGIS operator selects the VFORM SoftSwitch and the message selection appears. See Figure 15. Currently, only the EMERGENCY and CANCEL EMERGENCY messages are implemented. The other messages are implemented but not transmitted. In this example, the FL 1 operator selects to send the Emergency message by touching that function and then selecting SEND. When the Emergency message is received by FL 4, it causes a red triangle to appear around FL 1. See Figure 16



FL 1
Figure 16



FL 4
Figure 16

2.2 Transmission of Photographs and Video Clips Between AGISs

The Transmission of Photographs and Video Clips is underway and expected to be operational prior to award of the Phase 1 SBIR. The transmission of the photographs and video clips will function in much the same manner as the transmission and receipt of Digital Messages.

2.3 AGIS Map Processing

Our current AGIS map processing enables the display of up to 29 maps which can be of three broad classifications (Raster, Vector line maps, and georeferenced aerial or satellite photographs). AGIS automatically selects the map resolution that best correlates with the range of the operator's display. Currently, maps are loaded through use of a PDA cradle and Microsoft ActiveSync. If the AGIS operator prefers, he can select the type of map that he desires to be displayed. Figure 17 depicts a USGS or ADRG map. Figure 18 depicts a commercially available georeferenced satellite image and Figure 19 depicts selection of coastlines and geopolitical boundaries. As part of a separate non government funded effort, AGIS plans to incorporate "over the air" downloading of georeferenced maps.

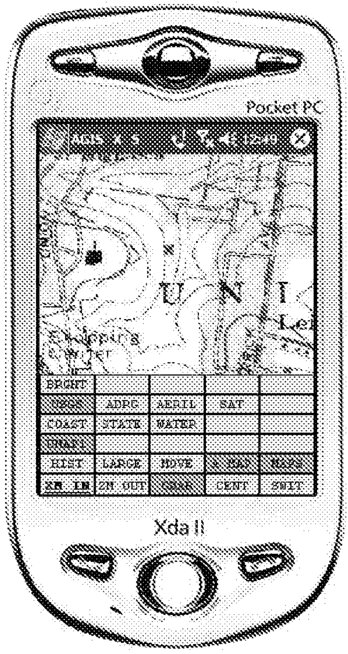


Figure 17

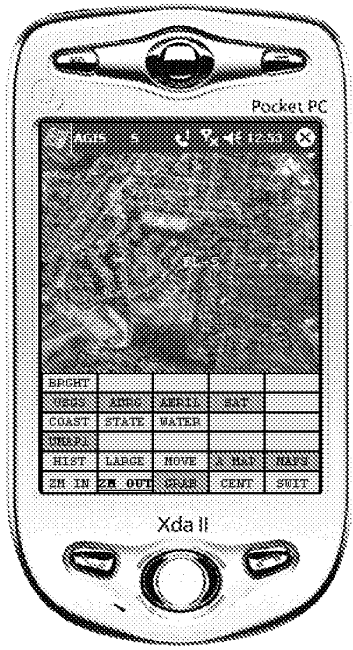


Figure 18

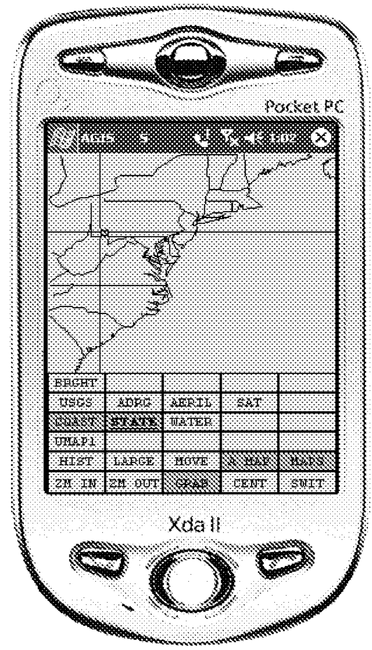


Figure 19

2.4 AGIS Voice Calls

To make a voice call the AGIS operator first hooks the AGIS unit to whom the call is to be placed and selects the CALL SoftSwitch, which causes the call to be placed to that AGIS unit. When in the CALL SoftSwitch mode, the hook symbol appears as a dashed box indicating the AGIS unit to whom the call is to be placed. See Figure 20. The AGIS operator can also place conference calls with up to six participants by sequentially hooking them. Alternatively, he can select NET1, 2, 3 or 4 which will conference the AGIS units that were pre-assigned to that NET. See Figure 21. If the AGIS operator desires to conference more than six users, he can place an 800 conference call. This action causes each of the remote AGISs to automatically call an 800 conference number and to automatically enter a participant code. As this process occurs, the participants are automatically conferenced together.

To set up the AGIS for operations, the AGIS operator selects the SETUP SoftSwitch which causes the set up SoftSwitch matrix to appear. The SoftSwitch matrix contains the NET assignment SoftSwitches. Selecting a NET SoftSwitch causes a matrix of SoftSwitches to appear with the identifier of each of the AGIS net participants. The operator can then add or drop participants for the NET SoftSwitch that was chosen. There are also reporting interval and distance traveled SoftSwitches which control the rate at which the AGIS reports its location and status information. The GPS SoftSwitch causes a display of the number of GPS satellites from which data is being received to appear in the lower right readout area. See Figures 21 and 22.

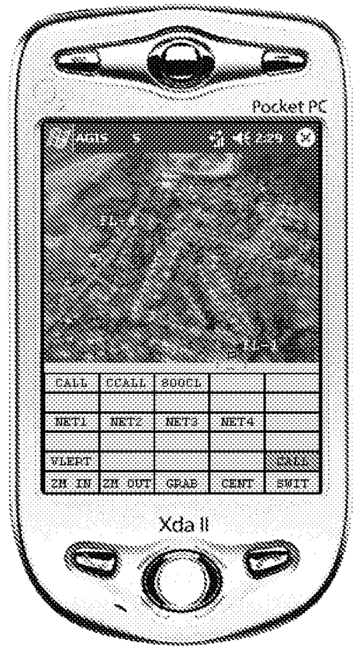


Figure 20

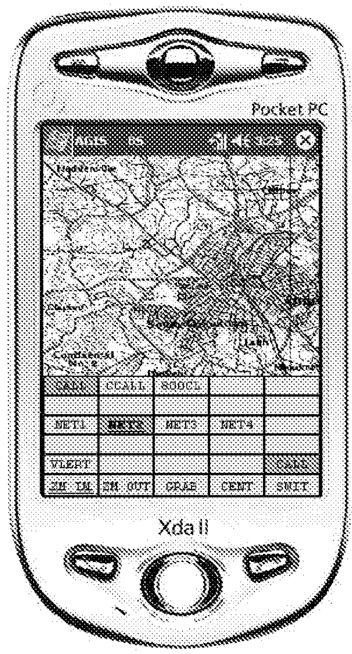


Figure 21

2.5 AGIS Net Assignment and Ping

The AGIS permits the person in charge of the network to specify which AGIS cell phones are to be part of the network. This is accomplished by the AGIS operator selecting the P REQ SoftSwitch which causes a matrix of the AGIS phones to appear. Note that this is a multilayered matrix and, if there are more than 28 phones, the operator can select the NEXT SoftSwitch which will cause yet another group of participants to appear. The operator can drop highlighted AGIS units from the net by selecting their SoftSwitch or add new ones to the net by selecting them. In Figure 22 note that AGIS Participant RUS 2 is not part of the AGIS communications net. In Figure 23 note that RUS 2 has been made part of the AGIS communications net and FL 5 and SAM have been dropped from the AGIS communications net. The AGIS operator can also select to immediately transmit his location and all his tracks to another AGIS unit and cause that unit to immediately report its location and all its tracks by selecting the PING SoftSwitch

To determine the status of the various AGISs in the communications net, the AGIS operator selects the PLIST SoftSwitch which causes display of the current AGIS communications net participants to be displayed in the lower left of the AGIS display. See Figure 24.

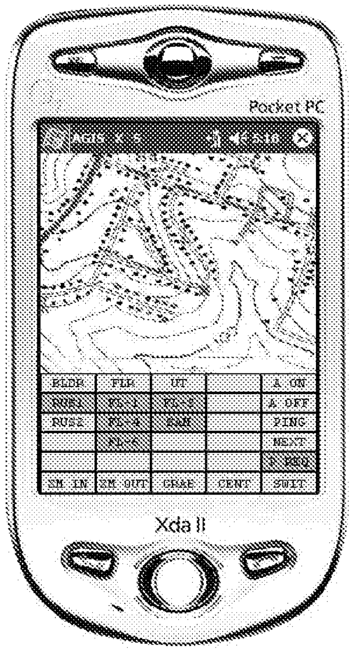


Figure 22

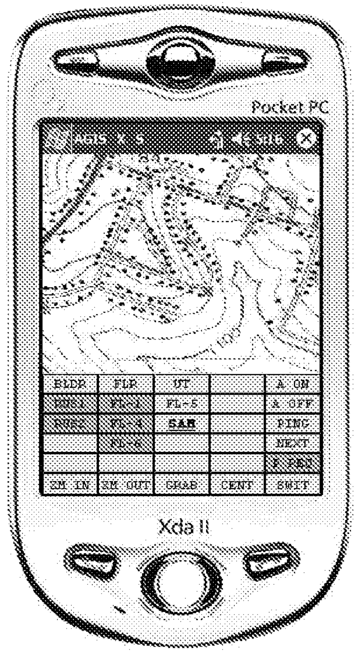


Figure 23

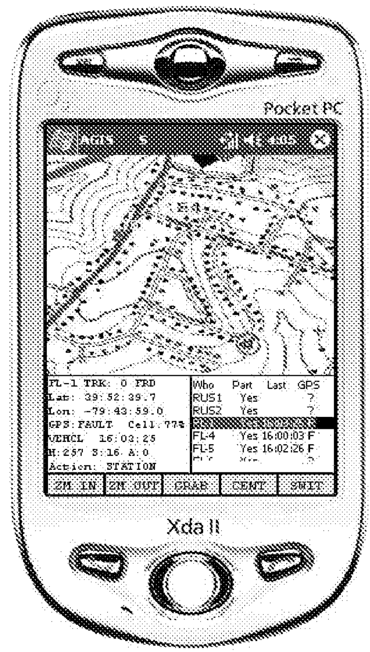


Figure 24

2.6 AGIS Set Up

The AGIS SETUP SoftSwitch enables the AGIS operator to control the digital communications intervals and other AGIS functions. To assign who is to be part of the AGIS voice and digital data transmission nets, the AGIS operator selects the appropriate AGIS units for each NET number. See Figure 25. The AGIS operator can also assign AGIS reporting intervals by time period and distance traveled since the last report. In Figure 26 the time interval for reporting has been set to every 5 minutes and distance traveled that will cause a report (even if the 5 minute interval has not lapsed) is set to 1 mile. The AGIS operator can also select to view which GPS satellites are being received by the GPS. See Figure 27.

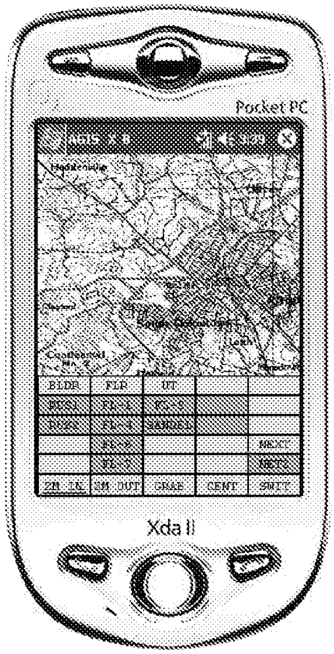


Figure 25

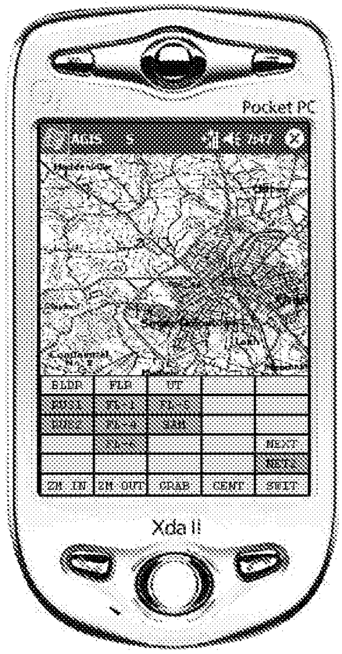


Figure 26

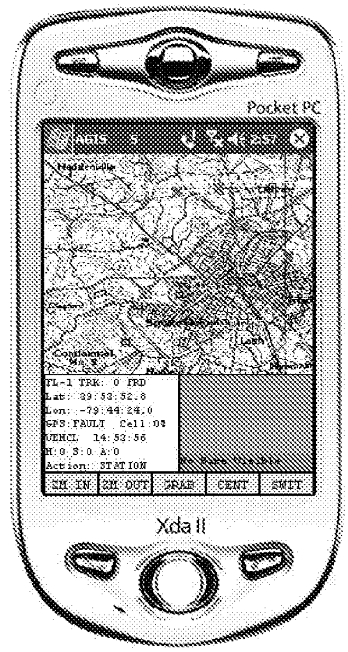


Figure 27

2.7 AGIS PC and Tablet PC

AGIS, Inc. is in the process of porting the AGIS software that operates under Pocket PC 2003 to a Microsoft XP environment. The AGIS PC will have all the capabilities of the AGIS PDA that have been discussed except the display screens will be horizontally oriented instead of the vertical orientation of the PocketPC. It is anticipated that this effort will be completed in March 2005. See Figure 28.

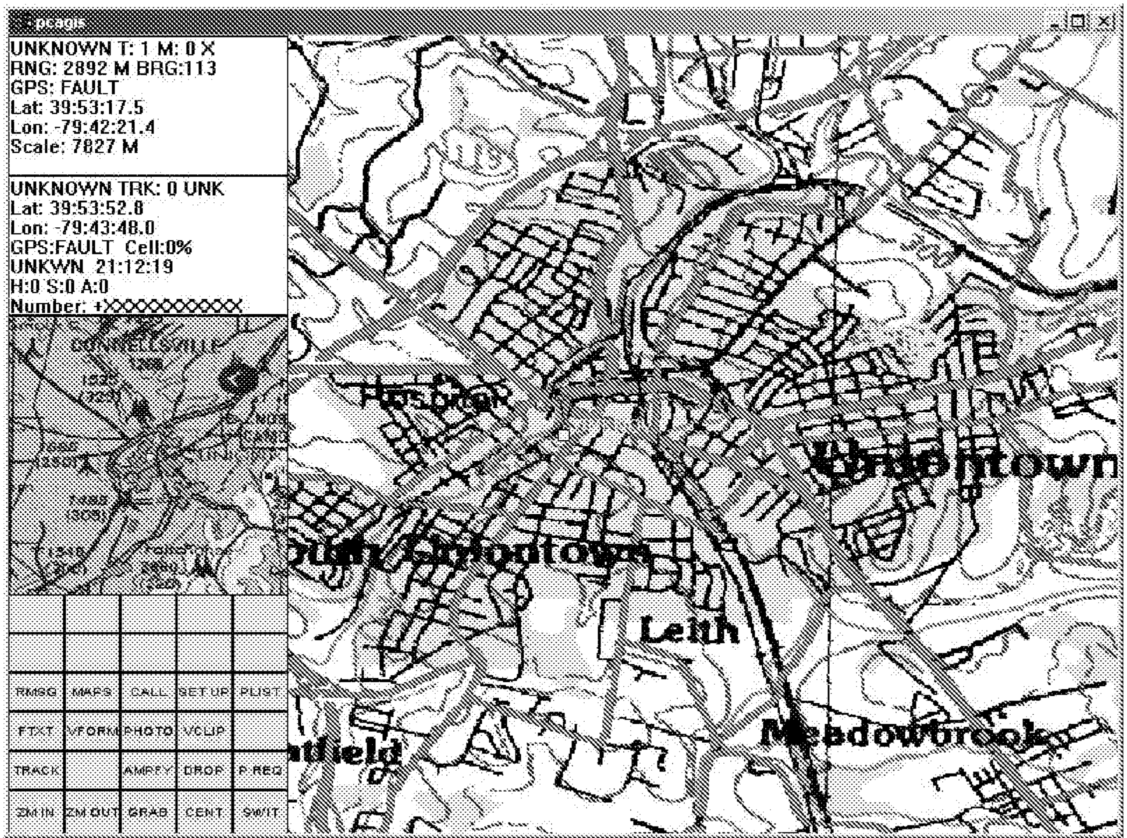


Figure 28

AGIS has filed a patent for several aspects associated with the AGIS. To this existing software AGIS plans to add software to meet the Navy Seal SBIR Phase 1 requirements.

3.0 Phase I Work Plan to Meet Seal Operational Requirements

We will demonstrate our ability to utilize the existing AGIS communications software to net three PDA/Cell phones /Cameras and the Panasonic Toughbook used by the Navy Seals (using an equivalent PC) immediately upon contract award. We will demonstrate the use of (and allow government SBIR Phase I project personnel to exercise) the above discussed capabilities and any others that are available at the time of contract award. The demonstration will enable an immediate evaluation of our SBIR Phase1 approach and provide the basis for our Phase 1 kickoff meeting. The kick off meeting can be held at our office in Jupiter, FL or combined with other business in Washington, DC where we often travel on other business. See Schedule Task 1.

As a first step, AGIS, Inc. proposes to add to the existing AGIS software the Navy Seal SWAMP application software and Paragon Imaging's ELT 3500 software. We will then use this configuration as the basis for our SBIR Phase 1 development which will use our collaborative PDA and PC software to enhance Tactical Decision-Making in Navy Seal Operations.

It is our understanding that the Navy Seals also utilize Paragon's pocketELT software that has been modified to include Navy Seal data entry. We propose to purchase this same software and host it on our AGIS PDA. The NITF processing software and the Report Generation software will both be assigned to one of the spare SoftSwitch Function Matrix switches. The Report Form Generation software will initially appear as depicted in Figure 29.

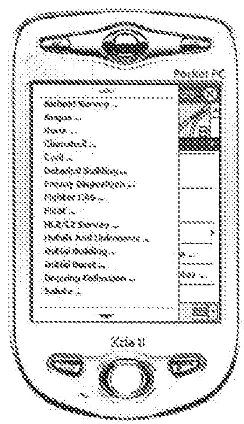


Figure 29

Along with the Paragon Imaging software we will host Navy Seal Special Warfare Automated Mission Planning (SWAMP) software on the PC and add additional off the shelf software to the PDA in order to meet Navy Seal specific requirements. The SWAMP software will be directly accessible through one of the spare PC SoftSwitch Function Matrix switches. (Note: AGIS, Inc. will need access to the SWAMP software.) . See Schedule Task 2.

The Tactical Operations Center (TOC) and the Joint Operations Center (JOC) utilize Joint Variable Message Formats (JVMF) ("K") messages. During Phase I, to enable the AGIS to communicate with the TOC and with the JOC, AGIS will be modified to add a subset of the JVMF ("K") messages to our processing. Those JVMF messages will replace all but the EMERGENCY and the CANCEL EMERGENCY AGIS Variable Message Formats messages previously discussed. The JVMF messages will be available through the AGIS VFORM SoftSwitch. AGIS will provide a cellular interface between our AGIS PDAs and our AGIS Toughbook compatible PC. See Schedule Task 3.

Real time weather information will be transmitted from the AGIS/ELT 3500 PC to the AGIS PDA and as a map image and superimposed on the AGIS maps in the same manner as **Mil-Std-2525** symbology and vector lines are currently superimposed on our raster maps. The AGIS software currently provides the GPS coordinates for any point on the display. The weather information will be accessed through an additional PC and PDA WEATH SoftSwitch. (Note: AGIS, Inc. will need to discuss which of several ways the Navy Seals want to process the weather information.) See Schedule Task 4.

Target ingress/egress points georeferenced to the AGIS maps will be added to the AGIS PC and PDA software. The ingress/egress points will be part of a fixed site data base that will also include such items as police departments and fire stations. These fixed points will be directly available through a SoftSwitch Function Matrix switch. Information concerning the fixed site will be available (as is the **Mil-Std-2525** symbology) by simply hooking the symbol. See Schedule Task 5.

Comparison of historical data to new data will be accomplished through use of off the shelf Paragon Imaging PC ELT 3500 software which performs image comparison. The ELT 3500 will be available through use of a single PC AGIS SoftSwitch Function Matrix switch. See Schedule Task 6.

Tides data / currents / phases of the moon will be accomplished by using off the shelf software that will be accessible through a single AGIS PDA SoftSwitch Function Matrix switch. See Schedule Task 7.

Medical information (Merck Manual and Physician's Desk Reference (PDR) will be accomplished by using off the shelf software that will be accessible through a single PDA SoftSwitch Function Matrix switch. See Schedule Task 8.

Remote Access to a certified doctor will be accomplished using voice communications, free text messages and the under development AGIS ability to rapidly transmit and receive photographs and video clips (due to be completed in Spring 2005). Development of this software is underway and is part of the baseline AGIS. See Schedule Task 9.

Survival maps relevant to the area of operation will be loaded and added as a SoftSwitch under the PC and PDA AGIS MAPS function. The current AGIS software requires that map downloads be accomplished using a PDA cradle and Microsoft ActiveSync software. As part of the SBIR Phase I effort, AGIS, Inc. will modify this software to enable over the air down loading of georeferenced maps. See Schedule Task 10.

Information on indigenous plants and poisonous snakes will be loaded into a Microsoft PDA compatible database and will be accessible through a single SoftSwitch Function Matrix switch selection. (Note: AGIS, Inc. will need more information on the Navy Seal existing indigenous plant and poisonous snake data.) See Schedule Task 11.

Tables of GPS location coordinates, satellite frequencies, phone numbers and identification codes will be made available from databases that will be accessible through a single AGIS PC and PDA SoftSwitch Function Matrix switch. See Schedule Task 12.

Real-time video feeds from satellites will be discussed and planned for during Phase I and implemented in Phase II. (Note: AGIS, Inc. will require additional data as to the data rates and formats of these feeds.)

AWACS data is normally transmitted using TADIL – J formats. AGIS, Inc. is very knowledgeable as to these formats and our personnel have on numerous occasions developed software to process TADIL – J data. TADIL – J data processing and video processing will be planned for in Phase I but implemented in Phase II. (Note: AGIS, Inc. will need additional data rate and format information of the AWACS video feeds. We are aware that receiving AWACS TADIL –J data would require use of a JTIDS terminal and antenna, AWACS TADIL – J data, however can also be available via a satellite feed.)

AGIS has made provisions to use a Jane's demonstration CD that contains a small subset of the military systems and equipment (e.g. weapons, armament, ship and aircraft profiles) CDs. This data is compatible with Microsoft XP and will run on the AGIS PC and Tablet software. AGIS, Inc. has made arrangements with Jane's for a demonstration CD so that we can demonstrate that this software operates on a PC.

During Phase I, to simulate military communications, AGIS, Inc. will use the SBIR Phase 1 software enhanced AGIS PDA / Cell phones to simulate the Navy Seal requirements in our laboratory or in the field (but not in a rugged environment). We will use our simple but effective test bed that consists of three PDA / Cell Phone / Camera / GPS devices and our PC (simulating a Seal Toughbook) to demonstrate that we have met our SBIR Phase 1 deliverables.

It should be noted that many military hardware manufacturers are making Microsoft software compatible ruggedized and Mil Spec PDAs, Tablets and small PCs. We will coordinate with the Seals to select a device that meets the Seal operational requirements. AGIS personnel have worked with many different NSA approved encryption devices. Some of the new very small military radios (including the Raytheon MicroLight) have provisions for integrated encryption.

SBIR PHASE I WORK PLAN DEVELOPMENT SCHEDULE

	Task Name	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6
1	PHASE I KICKOFF. DEMO						
2	ADD SWAMP & PARAGON SOFTWARE						
3	JVMF ("K") MESSAGE						
4	ADD REAL TIME WEATHER INFO						
5	TARGET INGRESS AND EGRESS						
6	COMPARISON OF HISTORICAL DATA						
7	ADD TIDE / PHASES OF THE MOON						
8	MEDICAL INFORMATION						
9	REMOTE ACCESS TO DOCTOR						
10	SURVIVAL MAPS						
11	INDIG PLANTS & POISONOUS SNAKE INFO						
12	TABLES OF GPS, SAT, PHONE & ID CODE						

4.0 Related Work

"AGIS has no prior, current, or pending support for proposed work"

AGIS has been totally developed using entirely corporate funds. However, last year, the Naval Research Laboratory (NRL) contracted with AGIS, Inc. through Assurance Technology Corporation to test (no development work) AGIS for use with their Anti-Terrorism Group. The prototype AGIS successfully passed the test and operated for a week without a failure. The prototype AGIS used four switches drawn at the bottom of the display and, in lieu of the SoftSwitch matrix, used the Navigation Button located at the center bottom of the phone to control the display. See Figure 30. As a result of the test, NRL provided AGIS, Inc. with a list of four desired additions: (1) Use a method of accessing additional switches that is easier to use than the Navigation Button, (2) Provide more than one map that can be accessed by the operator, (3) Integrate voice and data communications better and (4) Enable maps to rotate so that the map is selectively oriented heading up. All of these requests have been accomplished. NRL is not paying for any of the costs associated with these requested features. The Assurance Technology Corporation NRL Subcontract Number is N00173-01-C-2006. The point of contact for this ongoing testing effort is Mr.

Bob Eisenhauer (202) 767-1621. AGIS, Inc. will be returning to NRL for additional AGIS testing in the near future.

Figure 30

5.0 Relationship with Future Research or Research and Development

The work the AGIS performs in SBIR Phase 1 will be immediately useable to AGIS customers as can be seen in our commercialization strategy. We firmly believe that our completion of the Phase 1 effort will result in a versatile and revolutionary first responder collaboration tool. The communications netted AGIS PDA / Cell Phone / Camera will substantially expand current mobile group collaboration technology. AGIS has the distinct possibility of becoming a standard for civilian collaboration applications. At the completion of the Phase 1,effort, AGIS, Inc. will commence a marketing campaign advertising the AGIS in the applicable first responder association publications.

Furthermore, successful completion of SBIR Phase 1 will provide a strong base from which to commence Phase 2 Navy Seal collaborative effort activities. Specifically in SBIR Phase II we propose to: (a) modify our AGIS software so that it is capable of utilizing encrypted military communications including the Raytheon Micro Light radio (we have signed a Non-Disclosure Agreement with Raytheon in order to interface to this device) in addition to the currently AGIS utilized high speed cellular GSM and CDMA technologies, (b) expand the number of JVMF messages implemented in Phase 1, (c) connect to a small ViaSat or similar military satellite terminal. This activity will enable us to exchange JVMF messages with the TOC and JOC and (d) if desired, add TADIL J software so that AGIS is able to interface with the JTIDS terminal or over the horizon TADIL – J message traffic.

6.0 Commercialization Strategy

AGIS, Inc. is not aware of any other company offering a distributed product where each user of the PDA / Cell Phone / Camera can quickly exchange data with others without having a centralized command center. Because of prior company funded work, the development software required for the Phase 1 part of the SBIR N05-069 effort is relatively small. However, there are many SBIR Phase 1 requirements that properly implemented will <u>substantially</u> add to the commercialization of our system. We intend to implement these changes to the AGIS software in a manner that will improve the ability to coordinate the activities of groups of people on the move. First responders (policemen, firemen and emergency medical) military security, park service, border patrol, construction industry, civilian security companies, etc. will all benefit. Specifically these changes are:

a. The software necessary to enable our AGIS to access commercial databases through a single SoftSwitch will implemented. This feature will enable users to utilize their own custom databases or commercially available ones while at the same time retaining the AGIS positional databases.

b. The requirement to provide a database of easily accessible pre-established GPS points will be implemented in a manner that permits the same software to be able to load and access facilities like fire stations, police stations, emergency rooms, etc. Furthermore our design will permit that when the phone numbers associated with those facilities are included in the database, the AGIS software will enable the user to automatically call these locations by pointing at them.

c. The requirement to provide real time weather will be implemented in a manner that permits AGIS to transmit and receive georeferenced map, aerial and satellite images and to store them as maps that can be accessed through use of AGIS's MAP SoftSwitches

The availability of these features add to the number of customers who will find AGIS to be a low cost, light weight collaborative system that substantially enhances the user's ability to communicate and coordinate with other personnel.

Work that is being done under this Phase 1 SBIR that will be of particular interest to other US military services and to other governments (when approved for foreign sales) is the addition of an ability to process JVMF as this data link has been established as the standard for US ground operations.

7.0 Key Personnel

The following key personnel have been working for AGIS, Inc. for the last year and are the key personnel who have developed the AGIS software. Their education, job assignments and US Security clearances are listed below. All are AGIS, Inc. stockholders.

Mr. Beyer has been the key person in defining and specifying the designing of over 15 defense systems and developed the specifications for the AGIS. Mr. Beyer is being specified as the Principal Investigator.

Malcolm K. Beyer, Jr.

EDUCATION: *United States Naval Academy*
University of California, Los Angeles, Postgraduate work in Business and Computer Science

PROFESSIONAL EXPERIENCE:

Mr. Beyer has been involved in defense systems for the last 35 years. Mr. Beyer has specialized in founding small defense firms and growing them in the face of much larger competitors. Mr. Beyer's specific expertise is in the area of developing military Command and Control systems and simulators. Mr. Beyer has served as a corporate director of five firms and has consulted to many other companies.

Mr. Beyer's last position was as founder, principal stockholder and Chairman of Advanced Programming Concepts, Inc. (APC). APC specialized in processing National Asset data and transforming it into a form that permitted its display and translation. The Patriot missile system, AWACS, and most of our naval vessels then use this transformed data. As an example, APC's systems were used during the Gulf War to alert Patriot as to the firing and arrival of the Iraqi SCUD missiles. In 1999 Mr. Beyer sold APC to a public British firm.

Prior to founding APC, Mr. Beyer was the founder, principal stockholder, CEO and Chairman of Command Control and Communications Corporation (4C). 4C specialized in the development of data link simulators for the United States and other friendly countries. One of these simulators was, at the time, one of the world's largest. It comprised 99 mini computers, hundreds of boards of specially developed electronics, and over 50,000 miles of dedicated telephone lines. In 1985 Mr. Beyer sold 4C to a NYSE firm.

In 2004, Mr. Beyer started Advanced Ground information Systems, Inc. (AGIS), a company organized to provide position location information exchange and voice and digital (text, photograph and video clip) communications between personnel with a Cell phone / PDA / Camera.

Mr. Beyer has held SI/TK and other special clearances and currently has a Secret clearance.

Sandel S. Blackwell

EDUCATION: *M.S. Applied Mathematical Sciences, Rice University*
B.S. Computer Science and Mathematics, Trinity University
Certified Six Sigma "Black Belt Level"

PROFESSIONAL EXPERIENCE:

At General Electric, Mr. Blackwell was responsible for IT internet and intranet technologies developments. Mr. Blackwell lead teams and oversaw development and deployment of systems for global message exchange.

At Raytheon, Mr. Blackwell was functionally and technically responsible for all engineering at a site that encompassed product development (hardware, systems and software), test and evaluation, product transition and support to manufacturing (250 people, 6 managers). This site designed hardware (graphic design/layout, on-site and off-site card manufacturing, full box/system solutions), implemented box and full system software, and qualified/tested for production. Mr. Blackwell was responsible for ensuring process and product performance across product line as well as Six Sigma and SEI certifications.

At Aerojet, Mr. Blackwell was responsible for overseeing all software development of a Classified project. Aerojet is a certified SEI level 3 development facility.

At Advanced Programming Concepts, Inc., Mr. Blackwell was the Vice President of R&D and Chief Programmer responsible for new product development. Mr. Blackwell was deeply involved in business development and software design. Prior to that Mr. Blackwell was the Program Manager and software designer for the Air Defense Systems Integrator (ADSI) including TADIL –J an Air Force Tactical Exploitation of National Capabilities (TENCAP) project. More than 350 of these systems have been delivered.

Mr. Blackwell currently holds an SI/TK Polygraph level Clearance.

Christopher R. Rice

EDUCATION: *B.S. Computer Science, The University of Texas at Austin*

PROFESSIONAL EXPERIENCE
Principal Software Engineer
As the tech lead for a "Fast Feature Smart Cell Phone Development" group whose mission is to provide a total software solution by developing custom software around existing company product offerings. As a member of the Architecture Team he was responsible to assist individual development teams in integrating newer technologies into their projects.

As the technical lead software engineer Classified Program for communication interfaces. Personally created the overall software design using a real-time data/event driven methodologies (documented in OMT notation). Applications were segmented and distributed across multiple hosts. Wrote reusable libraries to isolate the software applications from the operating system. Mr. Rice also re-implemented the data link message processing software to support updated interface specifications.

In addition, Mr. Rice developed software in C++ for multiprocessor workstations using SGI's integrated development environment.

As the senior programmer for the Air Defense Communications Platform (ADCP) project, Mr. Rice developed software in ANSI C for Sun Sparc 20s and 486 Pentium PCs under a proprietary real-time OS. The OS and CPU (byte ordering) independent layers were written so the same source code compiled and ran on both platforms. Wrote several military data link applications (GBDL, IBDL, PPDL, TADIL-J), a multi-data link track correlation package, and a basic linear algebra package (add, multiply, inverse, Cholesky decomposition, etc). Mr. Rice also designed and developed software to Interface to SINCGARS radios, to RS-423, RS-422 devices, and to many types of modems.

Additionally, Mr. Rice developed software for the Air Defense Communications Platform and created several data link and simulator applications (GBDL, PPDL, IBDL) written in C for a 32-bit DOS protected mode. Ported this same software to Solaris 2.4 running on Sparc 20s.

Mr. Rice has held a Secret clearance with special programs access.

8.0 Facilities/Equipment

AGIS will use its facility at 92 Lighthouse Drive, Jupiter, FL 33469 and also its remote sites where the personnel whose resumes are located and where the development work for much of the AGIS design and software has taken place. AGIS already has the equipment (i.e. PCs and PDA / Cell phones, printers, copiers, scanners, etc.) necessary for this effort and will not require additional equipment.

9.0 Subcontractors/Consultants

Paragon Imaging (Woburn, MA) is a leading supplier of image processing software for government and commercial applications. Paragon Imaging's ELT®(Electronic Light Table) Series software is the most widely used commercial image processing software among U.S. intelligence and related government analysts. With ELT® Series software, users can easily view and exploit digital images, link associated information with images, and shares their analyses with decision-makers and other imagery users. ELT® Series software is available for UNIX, Windows and Web platforms.

To expedite our development for the AGIS Navy Seal requirement development process, we propose to purchase Paragon Imaging ELT 3500 software and their Pocket ELT software for National Image Transmission Format (NITF) processing. We will also purchase their PDA pocketELT Report Generation software (that was developed for the US Navy Seals) and have their assurance that they will provide us their knowledge of the interface to the ViaSat terminal.

No consultants are presently foreseen for the Phase I program. If a need should arise, AGIS, Inc. knows of many available consultants.

10.0 Prior, Current, or Pending Support of Similar Proposals or Awards.

"AGIS has no prior, current, or pending support for proposed work"

11. SBIR Phase 1 Option

As a SBIR Phase 1 Option, AGIS, Inc. proposes to provide the Navy Seals an AGIS training course and three AGIS PDA / Cell Phones / Cameras with their associated waterproof cases. So that the Navy Seals have the ability to experiment with the system in a near operational form we will also connect the completed SBIR Phase 1 AGIS to a ViaSat terminal using a Navy Seal Toughbook and transmit to and receive from the TOC and JOC two types of JVMF messages which will then be sent to the AGIS PDA's using cellular communications.

EXHIBIT 5

Raytheon Task

Scope Increase for Server

- Add Server to AGIS architecture
  - Budget
    - Capital
      - $1000 - PC Server
      - $20/month - increase Blackwell house to business DSL to get Static IP
    - Travel
      - Rus Rice travel to Uniontown for server install and integration
    - Labor
      - Increase of $6000 to fixed cost budget

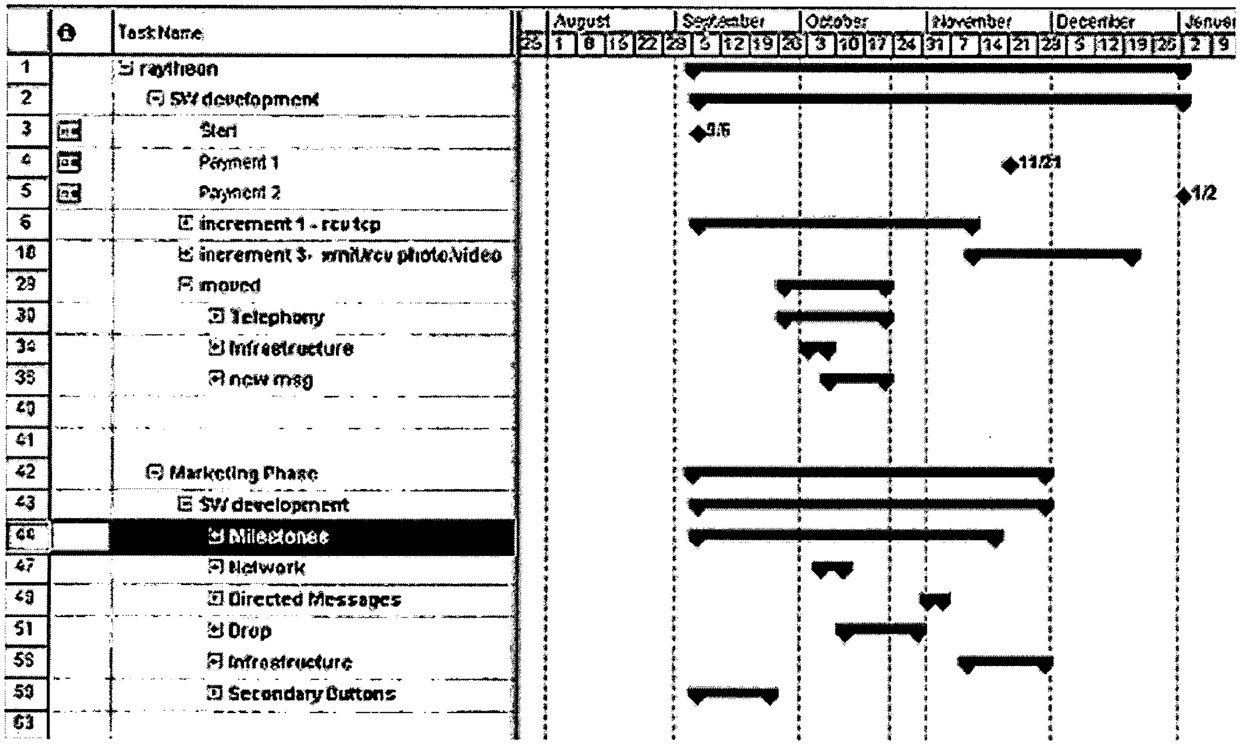
Server Work tasks

- Purchase and configure hardware for server.
- Change handset to add another transport method - handset to static IP.
- Program server for listening and forwarding AGIS messages between handsets. Server to decode Network Message and hold participants persistent. Server to transmit Network Participants message.
- Integrate with current Raytheon Phase I delivery.

Server Addition Schedule Effect

- Raytheon Phase 1 moved from Oct 31 -> Nov 19
- Marketing Phase Oct 31 and Nov 31 deliveries to be made Nov 19
- Raytheon Phase III moved to Jan 2, 2005

Schedule Updated



Current Labor Current Commitment

- Raytheon Phase 1 - $37197 (Nov 19)
  - Raytheon Phase 1 - $20323
  - Marketing moved tasks -$10874
  - Scope change for server - $6000
- Raytheon Phase III - $13157 (Jan 2, 2005)
- Marketing Phase - $21170 (Nov 19)

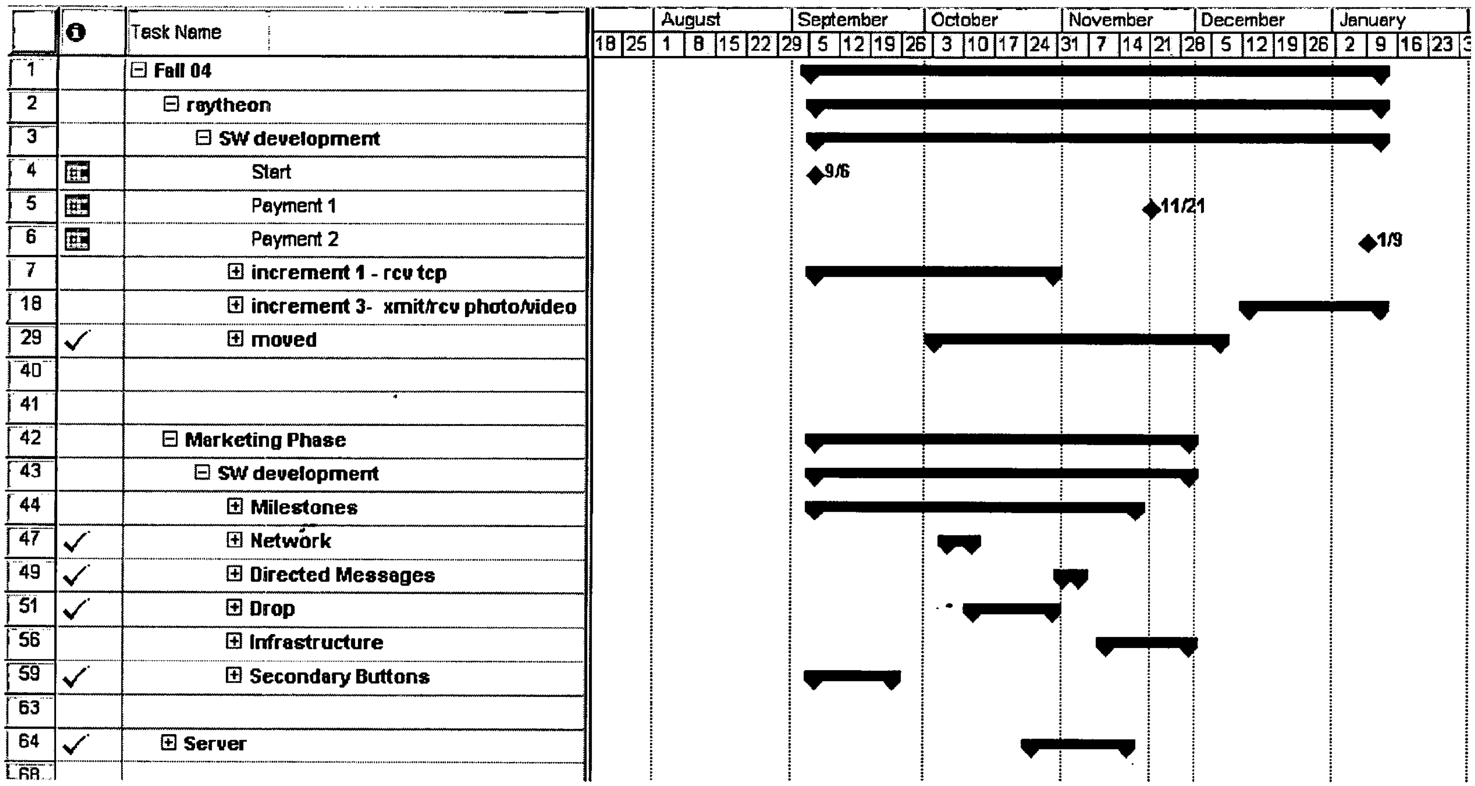
Numbers updated and consistent with SOW Aug 29, 2004.

EXHIBIT 6

Raytheon Phase I Marketing Phase I Delivery

November 21, 2004

Current Schedule



3 Week Slip in first delivery, 4 week Slip in second delivery

Current Financials

			Rus	Sandel		
Raytheon						
	Phase I	$31,197	$15,599	$15,599		
	Server Addition	$6,000	$4,200	$1,800		
	Phase III				Due January	$13,157
Marketing						
	Phase I	$11,904		$11,904		
	Phase II				Due Janauary	$6,266
Total			$19,799	$29,303		
Equity Addtions			1%	3%	Due January	3%
Equity Totals			6%	8%	Total after January	17%

Email delivery, current executable

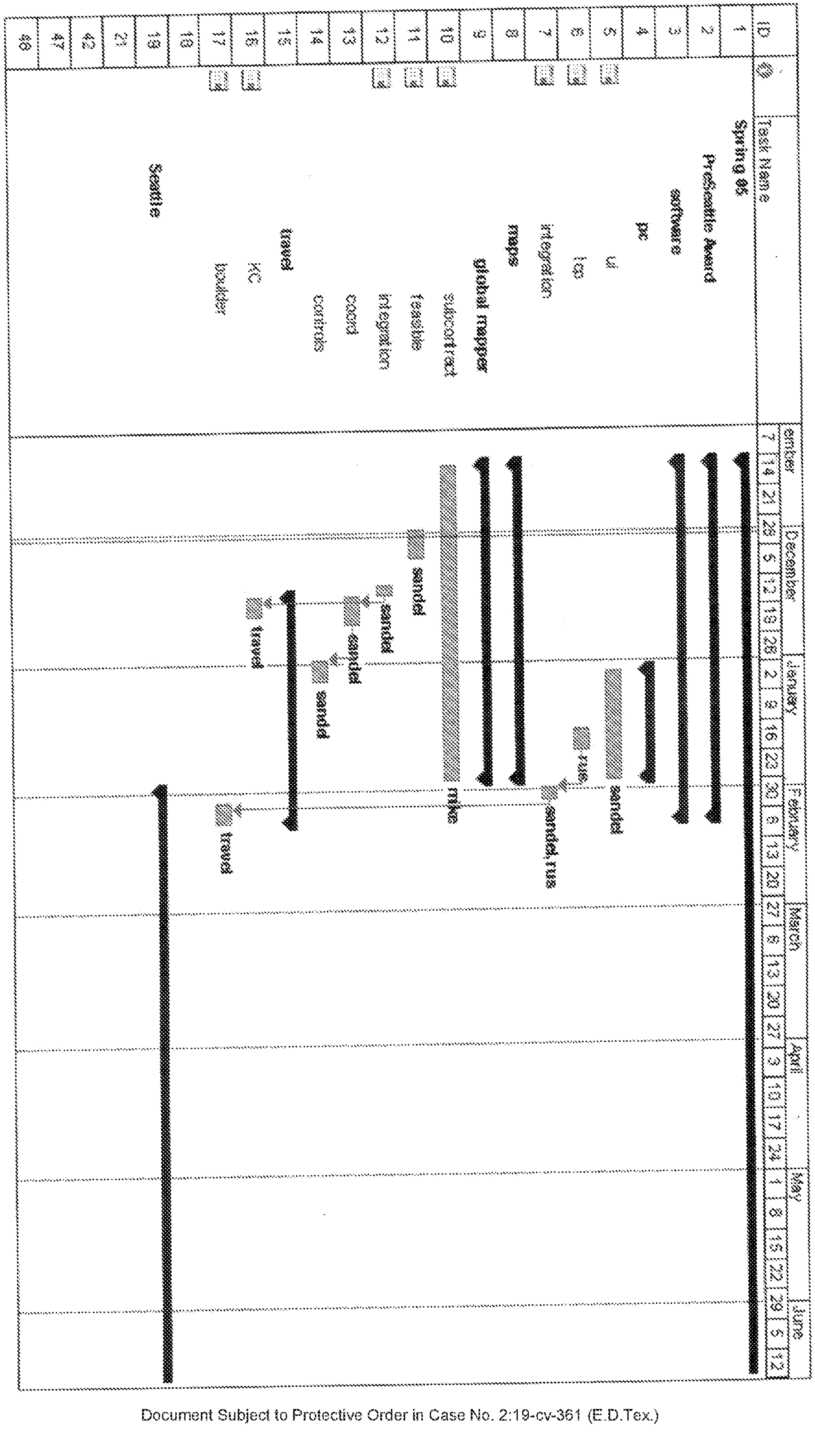
CD with full source to be delivered before December 1.

EXHIBIT 7

Spring 05

- Pre Seattle (Risk) Dec-Jan
  - Global Mapper integration
    - Global Mapper SOW $20,130
    - AGIS integration $8,400 (lower cost after relook)
  - PC AGIS $15,000
- Seattle $129,526 Feb-Jun (needs to be scrubbed)

Pre Seattle VERY aggressive. Could be as late as 30 days.
Advantage is much earlier delivery for Seattle.


ID
Task Name
Spring 05
PreSeattle Award
software
pc
ui
tcp
integration
maps
global mapper
subcontract
feasible
integration
coord
controls
travel
KC
boulder
Seattle
ember
December
January
February
March
April
May
June
sandel
sandel
sandel
sandel
travel
rus
sandel
mike
sandel,rus
travel

Global Mapper Breakdown

- Subcontract
  - Mike Child to perform against SOW as signed
- Feasible
  - Image load from dataset generated from Global Mapper PC
  - Basic zoom/pan
  - Two map usage, tactical and inset
  - Determine map projecctions
- Integration
  - Check functionality
  - Demo other features not yet integrated
  - Work out layer switching for display based on best fit
  - Determine strategy for image types to standardize based on timing
- Coords
  - Convert current projection and coordinate conversion to use Global Mapper functions
  - Coord system choicesand selecctions
- Control
  - Add rotation other features from demo not included in feasible
  - Any additional switches for GlobalMapper features

PC AGIS

- PC AGIS port
  - Common software baseline with Pocket PC display code
  - Window support for Switches, Inset, Readout and Tactical area
  - Direct feature port, no changes
  - TCP for PC (Client)

What I am asking for

- Approval to risk fund Global Mapper task and PC AGIS Task.
- Current Risk funded items not billed
  - Marketing Phase II
  - Raytheon Phase III

A very good first year.

ID		Task Name	Duration	Start	Finish
1					
2		Base Capability	87 days	Thu 4/1/04	Fri 7/30/04
3		SMS	65 days	Mon 5/3/04	Fri 7/30/04
4		User Interface Upgrade	98 days	Mon 8/2/04	Wed 12/15/04
5		TCP	56 days	Mon 8/30/04	Wed 11/17/04
6		Server	34 days	Fri 10/1/04	Wed 11/17/04
7		Maping Upgrade	55 days	Mon 11/15/04	Fri 1/28/05
8		Photo/Video exchange	20 days	Mon 1/3/05	Fri 1/28/05
9		PC Version	41 days	Mon 1/3/05	Mon 2/28/05
10		Abstracted Symbols	31 days	Fri 4/1/05	Fri 5/13/05
11		Server Interface Upgrade	22 days	Fri 5/13/05	Mon 6/13/05

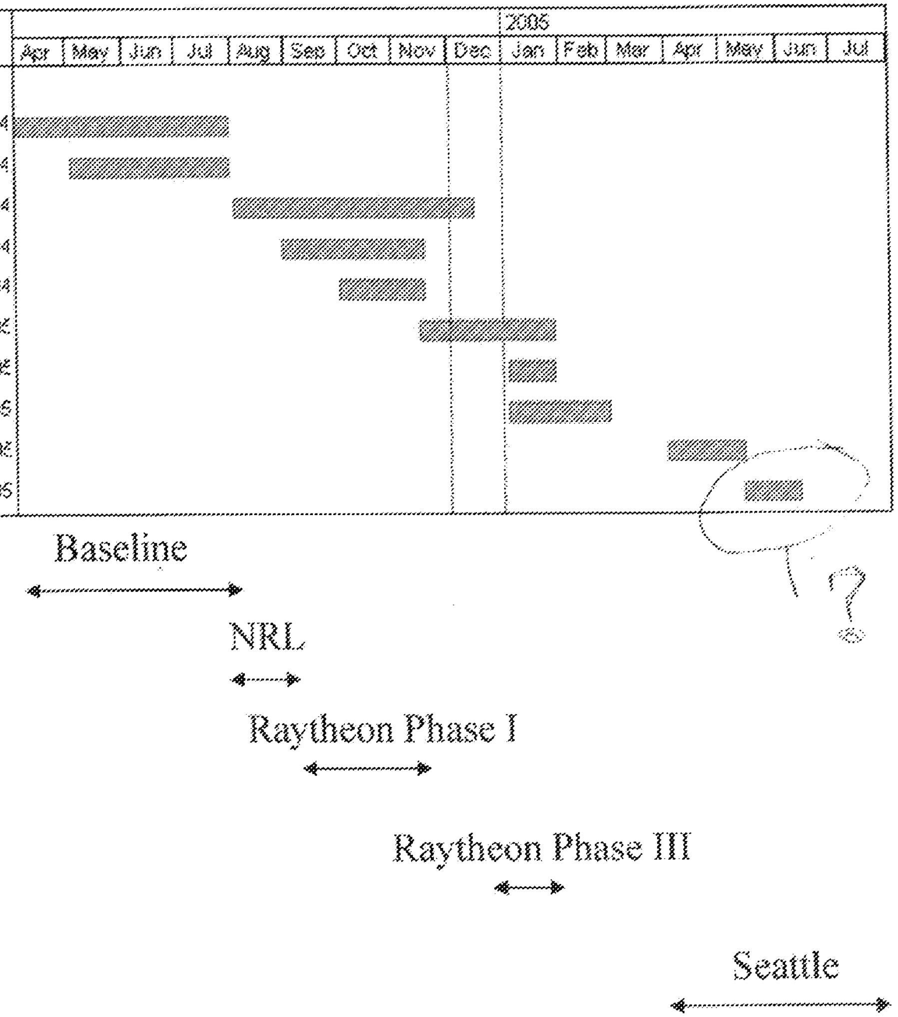


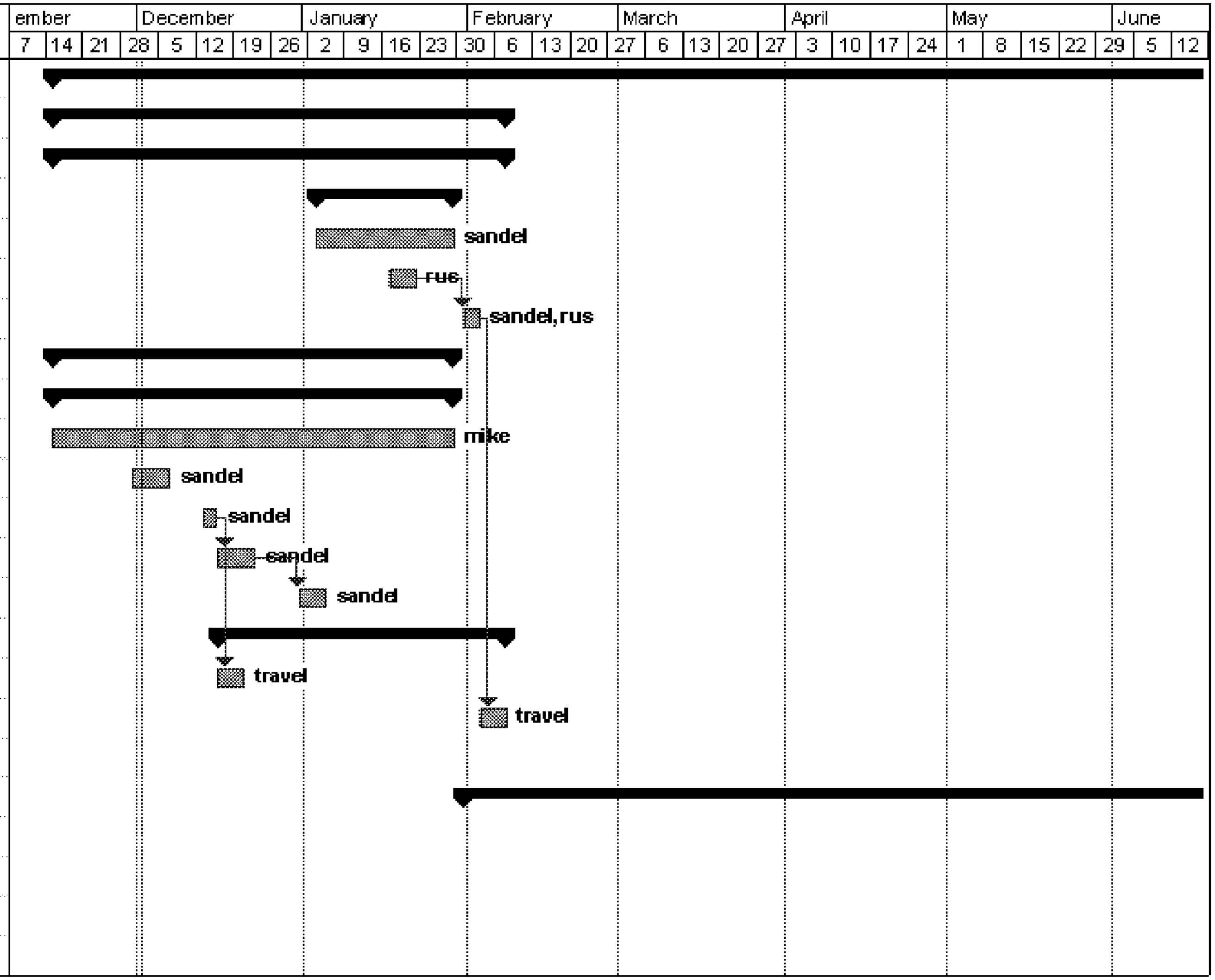
EXHIBIT 8

Spring 05

- Pre Seattle (Risk) Dec-Jan
  - Global Mapper integration
    - Global Mapper SOW $20,130
    - AGIS integration $8,400 (lower cost after relook)
  - PC AGIS $15,000
- Seattle $129,526 Feb-Jun (needs to be scrubbed)

Pre Seattle VERY aggressive. Could be as late as 30 days. Advantage is much earlier delivery for Seattle.

ID	Task Name
1	**Spring 05**
2	**PreSeattle Award**
3	**software**
4	**pc**
5	ui
6	tcp
7	integration
8	**maps**
9	**global mapper**
10	subcontract
11	feasible
12	integration
13	coord
14	controls
15	**travel**
16	KC
17	boulder
18	
19	**Seattle**
21	
42	
47	
48	



Document Subject to Protective Order in Case No. 2:19-cv-361 (E.D.Tex.)

Global Mapper Breakdown

- Subcontract
  - Mike Child to perform against SOW as signed
- Feasible
  - Image load from dataset generated from Global Mapper PC
  - Basic zoom/pan
  - Two map usage, tactical and inset
  - Determine map projecctions
- Integration
  - Check functionality
  - Demo other features not yet integrated
  - Work out layer switching for display based on best fit
  - Determine strategy for image types to standardize based on timing
- Coords
  - Convert current projection and coordinate conversion to use Global Mapper functions
  - Coord system choicesand selecctions
- Control
  - Add rotation other features from demo not included in feasible
  - Any additional switches for GlobalMapper features

PC AGIS

- PC AGIS port
  - Common software baseline with Pocket PC display code
  - Window support for Switches, Inset, Readout and Tactical area
  - Direct feature port, no changes
  - TCP for PC (Client)

What I am asking for

- Approval to risk fund Global Mapper task and PC AGIS Task.
- Current Risk funded items not billed
  - Marketing Phase II
  - Raytheon Phase III

EXHIBIT 9

Invoice 2-14-2005

Rice Computer Services CR Rice

AGIS 2/14/2005

Invoice		Rus	Sandel
Travel			
Air	$598.40		598.4
Car,gas	$143.38		143.38
Hotel	$284.72		284.72
Parking,Mileage	$75.57		75.57
Meals	$117.43		117.43
Capital			
6 SD Cards	$1,112.20		1112.2
Milestone			
Global Mapper Integration	$8,400.00		8400 Previously billed 1/15/05.
Raytheon - Phase III (Photo)	$13,157.00	$6,578.50	$6,578.50
Total Billed	$22,669.20	$6,578.50	$16,090.70

Please remit one check for Sandel Blackwell $16090.70 and one check for Rice Computing Solutions $6578.5.

PCAGIS beta released, to be billed in early March.

RHW

Pls Issue (Labor) Check to

"Rice Computer Services"

for $6,578.50

for Raytheon - Phase III (Photo) milestone

mail to: C. R. Rice
136 Damascus Rd
Golden, CO 80403

#1033 [illegible]

6657

C R Rice 3-30-2005

AGIS Hours Time & Materials Based Efforts

Date	Hours	Task Description	
RUS			
3/6/2005	2	Download, install Skype for PocketPC. Run test with AGIS and Skype call simultaneously.	
3/25/2005 3/26/2005 3/27/2005	15	Verizon phone support. Improve EVDO/RTT data support. Work on making connection establishment and retry more robust. Work on SMS integration issues. SMS/TCP Implement rolled log files for CSIF to capture debug and errors for Telephony, File Xfer and Investigate memory leak, installed Entrek analysis tool to look for leaks. Seems to be a problem	
Rus Total	17		$1,275.00
Sandel			
25-Mar	2	Reintegrate, latest version, new comm log	
26-Mar	9	New O2 ROM - font change, IP setup, periodic rewrite, verizon phone changes	
27-Mar	6	memory leak check, planning meeting, map/change	
30-Mar	2	NRL slides	
Sandel Total	19		$1,425.00

Rice Computer Services 4/9/05

~~(E.R. Rice)~~

Labor

AGIS	4/9/2005			
Invoice			Rus	Sandel
Travel				
Florida 3/25-3/27	Air	$233.18		233.18
	Car,gas,toll	$113.58		113.58
	Hotel			
	Parking,Mileage	$54.10		54.1
	Meals	$13.59		13.59
KC 4/1-4/2,DC 4/3-4/4	Air	$487.60		487.6
	Car,gas	$109.97		109.97
	Hotel	$146.17		146.17
	Parking,Mileage	$94.57		94.57
	Meals	$65.64		65.64
* Labor				
March	T&M	$2,700.00	$1,275.00	$1,425.00
Milestone				
PC AGIS	Fixed Price	$18,600.00	$4,800.00	$13,800.00
Total Billed		$21,300.00	$6,075.00	$16,543.40
			Total Labor	Pd ~~#~~ 1075 5-2-2005

6657

ok mwtb 4/29/05

AGIS TIMECARD

NAME Rus Rice
PERIOD 4/4/2005 to 4/30/2005

POOL	PROJECT	TARGET	AREA	ACTIVITY	HOURS	COMMENT
10	0	10	501	40	10	SMS stabili
10	0	10	502	40	10	TCP "stale
10	0	10	550	10	1	
10	0	10	550	20	14	Integrated i
10	0	10	550	30	2	Refactor in
20	30	20	500	10	2	UDP
TOTAL					**39**	

ck
Pd #1083

Name: Christopher R. Rice
Time Period: 7/10/2005 to 7/23/2005

		S	M	T	W	X	F	S
Description	WBS	7/10	7/11	7/12	7/13	7/14	7/15	7/16
Support - Jason, DX, UDP, CSIF	20.31.20.0.10		5	1				
NRL VTS Phase 1 Delivery	20.31.20.401.500							
NRL Infrastructure dev	20.31.20.300.10							
NRL Infrastructure dev	20.31.20.300.20							
Network configuration of Dev Server	20.31.1.600.40							
NRL VTS Phase 1 Delivery	20.31.20.0.90							
Patent docs, SLOC tool	30.0.1.1000.70							
CM	20.31.1.0.50				1			
Network configuration/server migration	20.31.0.550.40							
	Total	0	5	1	1	0	0	0

Pd ck #1167
7-25-2005

Rec
Page # 2

S	M	T	W	X	F	S	
7/17	7/18	7/19	7/20	7/21	7/22	7/23	Total
							0
							0
				0.5	1		7.5
5		5					10
2	2						4
1	1						2
							0
4	8	4					16
		1.5				2	3.5
						1	2
						2	2
							0
							0
							0
							0
							0
							0
							0
							0
12	11	10.5	0	0.5	1	5	47

AGIS Timesheet
Name:
Christopher R. Rice

Pay Period:
8-6-05 to 8-19-05

			8/6 S	8/7 S	8/8 M	8/9 T	8/10 W	8/11 X	8/12 F	8/13 S	8/14 S	8/15 M	8/16 T	8/17 W	8/18 X	8/19 F	Total
Description	GL Code	Job,Phase,Cost Code							2.00								2.00
CSIF Transcode	50000	32,550,40				2.00	2.00				4.00	1.00		3.00			12.00
Track DB	50000	32,300,20				2.00	1.00		3.00			2.00	1.00	1.00			10.00
Track DB	60000	32,300,30			1.00							1.00					2.00
CM	60000	32,300,50		2.00	1.00												3.00
Group 1	70500											3.00	3.00	2.00	2.00	3.00	13.00
Output Processor	50000	32,500,20															-
Paid Time Off	61500																-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
Note: Fill in shaded areas only		Total	-	2.00	2.00	4.00	3.00	-	5.00	-	4.00	7.00	4.00	6.00	2.00	3.00	42.00

Pd
CK 1202
8-22-05

ACI? Timesheet

Name:

Christopher R. Rice

Pay Period:

8-20-05 to 9-2-05

Description	GL Code	Job,Phase,Cost Code	8/20 S	8/21 S	8/22 M	8/23 T	8/24 W	8/25 X	8/26 F	8/27 S	8/28 S	8/29 M	8/30 T	8/31 W	9/1 X	9/2 F	Total
AIS Transcode	50000	32,104,10					1.00		2.00								3.00
Track DB	50000	32,300,20			1.00							5.00	1.00				7.00
Track DB	50000	32,300,40			3.00			3.00				1.00	3.00				10.00
CM	50000	32,300,50								1.00							1.00
Startup/Shutdown Research	50000	32,550,10					1.00		1.00	1.00							3.00
Output Processor	50000	32,500,20		8.00		3.00											11.00
Paid Time Off	61500																-
Track DB	50000	32,300,30													2.00		2.00
Track DB	50000	32,300,10										1.00					1.00
Jason termination	70500	[illegible]														3.00	3.00
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
Note: fill in shaded areas only		Total	-	8.00	4.00	3.00	2.00	3.00	3.00	2.00	-	7.00	4.00	-	2.00	3.00	41.00

pd ck #0124
9-6-2021

AGIS Timesheet

Name: Christopher R. Rice

Pay Period: 9/3/2005 to 9/16/2005

Description	GL Code	Job,Phase,Cost Code	9/3 S	9/4 S	9/5 M	9/6 T	9/7 W	9/8 X	9/9 F	9/10 S	9/11 S	9/12 M	9/13 T	9/14 W	9/15 X	9/16 F	Total
	50000	32,104,10															-
TrackDB/History	50000	32,300,20		1.00								1.00					2.00
TrackDB/History	50000	32,300,40			3.00					3.00						2.00	8.00
CM	50000	32,300,50															-
TrackDB/History	50000	32,300,30										3.00	1.00				4.00
Multi-Reader DistQueue	50000	32,500,20		3.00					2.00	5.00							10.00
Paid Time Off	61500																-
Multi-Reader DistQueue	50000	32,500,30								1.00							1.00
Multi-Reader DistQueue	50000	32,500,10		1.00													1.00
Support/Setup Router for Cap	70500					1.00		2.00								1.00	4.00
DISCO interview, travel	70500						3.00										3.00
Resumes/Interviews	70500									1.00			1.00			1.00	3.00
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
		Total	-	5.00	3.00	1.00	3.00	2.00	2.00	10.00	-	4.00	2.00	-	-	4.00	36.00

Note: Fill in shaded areas only

Pd ck #158
9-20-2005

AGIS Timesheet

Name: Christopher R. Rice

Pay Period: 9/17/2005 - 9/30/2005

Description	GL Code	Job,Phase,Cost Code	9/17 S	9/18 S	9/19 M	9/20 T	9/21 W	9/22 X	9/23 F	9/24 S	9/25 S	9/26 M	9/27 T	9/28 W	9/29 X	9/30 F	Total
	50000																-
TrackDB/History	50000	32,300,20			1.00												1.00
TrackDB/History	50000	32,300,40					2.00				2.00						4.00
CM	50000	32,300,50						1.00									1.00
TrackDB/History	50000	32,300,30			3.00	2.00	4.00					1.00					10.00
Multi-Reader DistQueue	50000	32,500,20															-
Paid Time Off	61500																-
Multi-Reader DistQueue	50000	32,500,30															-
Multi-Reader DistQueue	50000	32,500,10															-
Support AGIS Server/Scotty	70500							2.00									2.00
DISCO interview, travel	50000	32,800,999							3.00						1.00		4.00
Resumes/Interviews	70500				1.00			1.00		1.00							3.00
HP hx4700 integration	70500												5.00	6.00	1.00	5.00	17.00
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
																	-
Note: fill in shaded areas only		Total	-	-	5.00	2.00	6.00	4.00	3.00	1.00	2.00	1.00	5.00	6.00	2.00	5.00	42.00

1.00

Pd ck #165
10-3-2005

16 = III

AGIS Timesheet

Name: Christopher R. Rice

Pay Period: 10/1/05 - 10/14/05

Description	GL Code	Job,Phase,Cost Code	10/1 S	10/2 S	10/3 M	10/4 T	10/5 W	10/6 X	10/7 F	10/8 S	10/9 S	10/10 M	10/11 T	10/12 W	10/13 X	10/14 F	Total
	50000																-
TrackDB/History	50000	33,300,20															-
TrackDB/History	50000	33,300,40										2.00			2.00	0.50	4.50
CM	50000	33,300,50															-
TrackDB/History	50000	33,300,30															-
Multi-Reader DistQueue	50000	33,500,20															-
Paid Time Off	61500																-
Multi-Reader DistQueue	50000	33,500,30															-
Multi-Reader DistQueue	50000	33,500,10															-
Support Scotty/Audiovox	70500					2.00									2.00		4.00
DISCO interview, travel	50000	33,800,999															-
New Employee Setup	70500													0.50			0.50
HP hx4700 integration/Tradeshow	78000	30,600,40 … Thru			5.00		2.00	1.00				1.00					9.00
ImageServer	50000	33,610,10	Communications					2.00		0.50							2.50
ImageServer	50000	33,610,20	Maint. Labor					1.00		3.00	8.00					4.00	16.00
ImageServer	50000	33,610,30								0.50	1.00					1.00	2.50
PocketPC Release version	50000	33,200,40											1.00				1.00
DX for imagery msgs	50000	33,210,20									1.00						1.00
																	-
																	-
																	-
																	-
																	-
																	-
																	-
Note: fill in shaded areas only		Total	-	-	5.00	2.00	2.00	4.00	-	4.00	10.00	3.00	1.00	0.50	4.00	5.50	41.00

Pdck #197
10-17-2005