CLAIMS APPENDIX

Listing of the claims involved in the appeal:

1. (previously presented) A method of displaying an instant message conversation on an electronic device, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, the method comprising:

receiving a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmitting a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associating each instant message with a corresponding time stamp;

displaying the incoming textual portion of each displayed incoming instant message horizontally aligned at a same first horizontal position;

displaying the outgoing textual portion of each displayed outgoing instant message horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position; and

displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.

2. (original) The method of claim 1 further comprising displaying an incoming symbol with each displayed incoming instant message.

3. (original) The method of claim 1 further comprising displaying an outgoing symbol with each displayed outgoing instant message.

4. (original) The method of claim 1, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

5. (original) The method of claim 1, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

6. (original) The method of claim 1, further comprising:

detecting an interruption in the instant message conversation; and

refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

7. (original) The method of claim 1, further comprising refraining from displaying at least one of the corresponding time stamps.

8. (original) The method of claim 7, wherein the refraining is performed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

9. (previously presented) An electronic device for displaying an instant message conversation, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, the electronic device comprising:

a display;

a memory; and

a processor electronically coupled with the display and the memory, the processor configured to:

receive a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmit a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associate each instant message with a corresponding time stamp;

display the incoming textual portion of each displayed incoming instant message horizontally aligned at a same first horizontal position;

display the outgoing textual portion of each displayed outgoing instant message horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position; and

display time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.

10. (original) The electronic device of claim 9, wherein the processor is further configured to display an incoming symbol with each displayed incoming instant message.

11. (original) The electronic device of claim 9, wherein the processor is further configured to display an outgoing symbol with each displayed outgoing instant message.

12. (original) The electronic device of claim 9, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

13. (original) The electronic device of claim 9, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

14. (original) The electronic device of claim 9, wherein the processor is further configured to:

detect an interruption in the instant message conversation; and

refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

15. (original) The electronic device of claim 9, wherein the processor is further configured to refrain from displaying at least one of the corresponding time stamps.

16. (original) The electronic device of claim 15, wherein the processor refrains from displaying the at least one of the corresponding time stamps if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

17. (previously presented) A non-transitory computer readable medium comprising computer executable instructions embedded thereon for execution by a processor of an electronic device for displaying an instant message conversation upon a display of the electronic device, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, such that when executed, the processor:

receives a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmits a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associates each instant message with a corresponding time stamp;

displays the incoming textual portion of each displayed incoming instant message horizontally aligned at a same first horizontal position;

displays the outgoing textual portion of each displayed outgoing instant message horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position; and

displays time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.

18. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to display an incoming symbol with each displayed incoming instant message.

19. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to display an outgoing symbol with each displayed outgoing instant message.

20. (previously presented) The non-transitory computer readable medium of claim 17, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

21. (previously presented) The non-transitory computer readable medium of claim 17, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

22. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to:

detect an interruption in the instant message conversation; and

refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

23. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps.

24. (previously presented) The non-transitory computer readable medium of claim 23, wherein the computer instructions that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps are executed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

25. (previously presented) The method of claim 1, the input comprising movement of a pointing device in proximity to the at least one instant message of the incoming and outgoing instant messages.

26. (previously presented) The electronic device of claim 9, the input comprising movement of a pointing device in proximity to the at least one instant message of the incoming and outgoing instant messages.

27. (previously presented) The non-transitory computer readable medium of claim 17, the input comprising movement of a pointing device in proximity to the at least one instant message of the incoming and outgoing instant messages.

EVIDENCE APPENDIX

NONE

RELATED PROCEEDINGS APPENDIX

NONE

22238497.1

Electronic Patent Application Fee Transmittal

Application Number:	13111734
Filing Date:	19-May-2011
Title of Invention:	Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment
First Named Inventor/Applicant Name:	Gerhard D. Klassen
Filer:	Brett Joseph Slaney/Judith Martin
Attorney Docket Number:	70314/00568

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Filing a brief in support of an appeal	1402	1	630	630
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**630**

Electronic Acknowledgement Receipt	
EFS ID:	13942692
Application Number:	13111734
International Application Number:	
Confirmation Number:	6081
Title of Invention:	Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment
First Named Inventor/Applicant Name:	Gerhard D. Klassen
Customer Number:	91704
Filer:	Brett Joseph Slaney/Judith Martin
Filer Authorized By:	Brett Joseph Slaney
Attorney Docket Number:	70314/00568
Receipt Date:	09-OCT-2012
Filing Date:	19-MAY-2011
Time Stamp:	16:30:11
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$630
RAM confirmation Number	3595
Deposit Account	022553
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees) Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Appeal Brief Filed	11144-US-CNT3_Appeal_Brief.pdf	1117681 d4430589e7c48b08889308e8bb888e2ad54dbd16	no	22
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30266 bc51c7be875f1dcbb94e9ef4b86431f77256b444	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	1147947		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

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.
13/111,734	05/19/2011	Gerhard D. Klassen	70314/00568	6081

91704 7590 08/30/2012

Blake, Cassels & Graydon LLP
199 BAY STREET , SUITE 4000
COMMERCE COURT WEST
TORONTO, ON M5L 1A9
CANADA

EXAMINER	
LAI, MICHAEL C	
ART UNIT	PAPER NUMBER
2457	

NOTIFICATION DATE	DELIVERY MODE
08/30/2012	ELECTRONIC

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.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

rimpatent@blakes.com
brett.slaney@blakes.com
portfolioprosecution@rim.com

Advisory Action Before the Filing of an Appeal Brief	**Application No.** *13/111,734*	**Applicant(s)** KLASSEN ET AL.
	Examiner MICHAEL C. LAI	**Art Unit** 2457

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

THE REPLY FILED 09 August 2012 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

NO NOTICE OF APPEAL FILED

1. ☐ The reply was filed after a final rejection. No Notice of Appeal has been filed. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance;

   (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114 if this is a utility or plant application. Note that RCEs are not permitted in design applications. The reply must be filed within one of the following time periods:

   a) ☐ The period for reply expires _____months from the mailing date of the final rejection.

   b) ☐ The period for reply expires on: (1) the mailing date of this Advisory Action; or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.

   c) ☐ A prior Advisory Action was mailed more than 3 months after the mailing date of the final rejection in response to a first after-final reply filed within 2 months of the mailing date of the final rejection. The current period for reply expires months from the mailing date of *the prior Advisory Action* or SIX MONTHS from the mailing date of the final rejection, whichever is earlier.

   Examiner Note: If box 1 is checked, check either box (a), (b) or (c). ONLY CHECK BOX (b) WHEN THIS ADVISORY ACTION IS THE FIRST RESPONSE TO APPLICANT'S FIRST AFTER-FINAL REPLY WHICH WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. ONLY CHECK BOX (c) IN THE LIMITED SITUATION SET FORTH UNDER BOX (c). See MPEP 706.07(f).

Extensions of time may be obtained under 37 CFR 1.136(a). The date on which the petition under 37 CFR 1.136(a) and the appropriate extension fee have been filed is the date for purposes of determining the period of extension and the corresponding amount of the fee. The appropriate extension fee under 37 CFR 1.17(a) is calculated from: (1) the expiration date of the shortened statutory period for reply originally set in the final Office action; or (2) as set forth in (b) or (c) above, if checked. Any reply received by the Office later than three months after the mailing date of the final rejection, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

NOTICE OF APPEAL

2. ☒ The Notice of Appeal was filed on *09 August 2012*. A brief in compliance with 37 CFR 41.37 must be filed within two months of the date of filing the Notice of Appeal (37 CFR 41.37(a)), or any extension thereof (37 CFR 41.37(e)), to avoid dismissal of the appeal. Since a Notice of Appeal has been filed, any reply must be filed within the time period set forth in 37 CFR 41.37(a).

AMENDMENTS

3. ☐ The proposed amendments filed after a final rejection, but prior to the date of filing a brief, will not be entered because

   a) ☐ They raise new issues that would require further consideration and/or search (see NOTE below);

   b) ☐ They raise the issue of new matter (see NOTE below);

   c) ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or

   d) ☐ They present additional claims without canceling a corresponding number of finally rejected claims.

   NOTE: _____. (See 37 CFR 1.116 and 41.33(a)).

4. ☐ The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).

5. ☐ Applicant's reply has overcome the following rejection(s): _____.

6. ☐ Newly proposed or amended claim(s) _____ would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).

7. ☒ For purposes of appeal, the proposed amendment(s): (a) ☐ will not be entered, or (b) ☒ will be entered, and an explanation of how the new or amended claims would be rejected is provided below or appended.

AFFIDAVIT OR OTHER EVIDENCE

8. ☐ The affidavit or other evidence filed after final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).

9. ☐ The affidavit or other evidence filed after the date of filing the Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing of good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).

10. ☐ The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. ☒ The request for reconsideration has been considered but does NOT place the application in condition for allowance because: See Continuation Sheet.

12. ☐ Note the attached Information *Disclosure Statement*(s). (PTO/SB/08) Paper No(s). _____

13. ☐ Other: _____.

STATUS OF CLAIMS

14. The status of the claim(s) is (or will be) as follows:

Claim(s) allowed: .
Claim(s) objected to: .
Claim(s) rejected: 1-27.
Claim(s) withdrawn from consideration: .

/ARIO ETIENNE/ Supervisory Patent Examiner, Art Unit 2457	/MICHAEL C LAI/ Examiner, Art Unit 2457

Continuation of 11. does NOT place the application in condition for allowance because: Applicant's arguments are not persuasive. In the remarks, the applicant argues in substance that:
A) Maurille does not teach or suggest balancing the competing objectives of conserving display space and providing useful time information by enabling a selective display of time information.
B) MacPhail does not teach or suggest selectively displaying a time stamp associated with an incoming or outgoing message in a messaging environment.
C) MacPhail does not teach: "displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages".

In response to A), it is noted that the features upon which applicant relies (i.e., balancing…providing…by enabling…) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See In re Van Geuns, 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).
In response to B), it is noted that the features upon which applicant relies (i.e., selectively displaying…) are not recited in the rejected claim(s). Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims.
In response to C), Appelman discloses "displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp" (see at least Figs. 17 and 18). The Examiner relies on MacPhail for the feature of "in response to an input, displaying time information according to the correspnding time stamp". It is the combination of Appelman, Maurille, and MacPhail that teach the limitation, not just MacPhail.

OK TO ENTER: /M.L./

Application No. 13/111,734
Amendment Dated: August 9, 2012
Reply to Office Action of: June 22, 2012

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Appl. No.:	**13/111,734**
Applicant:	**KLASSEN, Gerhard D. et al.**
Filed:	**May 19, 2011**
Title:	**Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment**
Art Unit:	**2457**
Examiner:	**LAI, Michael C.**
Docket No.:	**70314/00568**

Mail Stop AF
U.S. Patent & Trademark Office
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE AFTER FINAL REJECTION

Sir:

This is further to the Office Action dated June 22, 2012. Applicant wishes to present the following remarks:

Listing of Claims: begins on page 2 of this paper.

Remarks: begin on page 7 of this paper.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Appl. No.:	**13/111,734**
Applicant:	**KLASSEN, Gerhard D. et al.**
Filed:	**May 19, 2011**
Title:	**Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment**
Art Unit:	**2457**
Examiner:	**LAI, Michael C.**
Docket No.:	**70314/00568**

Mail Stop AF
U.S. Patent & Trademark Office
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE AFTER FINAL REJECTION

Sir:

This is further to the Office Action dated June 22, 2012. Applicant wishes to present the following remarks:

Listing of Claims: begins on page 2 of this paper.

Remarks: begin on page 7 of this paper.

Previously Presented Claims

Listing of claims:

1. (previously presented) A method of displaying an instant message conversation on an electronic device, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, the method comprising:

receiving a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmitting a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associating each instant message with a corresponding time stamp;

displaying the incoming textual portion of each displayed incoming instant message horizontally aligned at a same first horizontal position;

displaying the outgoing textual portion of each displayed outgoing instant message horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position; and

displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.

2. (original) The method of claim 1 further comprising displaying an incoming symbol with each displayed incoming instant message.

3. (original) The method of claim 1 further comprising displaying an outgoing symbol with each displayed outgoing instant message.

4. (original) The method of claim 1, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

5. (original) The method of claim 1, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

6. (original) The method of claim 1, further comprising:

detecting an interruption in the instant message conversation; and

refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

7. (original) The method of claim 1, further comprising refraining from displaying at least one of the corresponding time stamps.

8. (original) The method of claim 7, wherein the refraining is performed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

9. (previously presented) An electronic device for displaying an instant message conversation, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, the electronic device comprising:

a display;

a memory; and

a processor electronically coupled with the display and the memory, the processor configured to:

receive a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmit a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associate each instant message with a corresponding time stamp;

display the incoming textual portion of each displayed incoming instant message horizontally aligned at a same first horizontal position;

display the outgoing textual portion of each displayed outgoing instant message horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position; and

display time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.

10. (original) The electronic device of claim 9, wherein the processor is further configured to display an incoming symbol with each displayed incoming instant message.

11. (original) The electronic device of claim 9, wherein the processor is further configured to display an outgoing symbol with each displayed outgoing instant message.

12. (original) The electronic device of claim 9, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

13. (original) The electronic device of claim 9, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

14. (original) The electronic device of claim 9, wherein the processor is further configured to:

detect an interruption in the instant message conversation; and

refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

15. (original) The electronic device of claim 9, wherein the processor is further configured to refrain from displaying at least one of the corresponding time stamps.

16. (original) The electronic device of claim 15, wherein the processor refrains from displaying the at least one of the corresponding time stamps if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

17. (previously presented) A non-transitory computer readable medium comprising computer executable instructions embedded thereon for execution by a processor of an electronic device for displaying an instant message conversation upon a display of the electronic device, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, such that when executed, the processor:

receives a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmits a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associates each instant message with a corresponding time stamp;

displays the incoming textual portion of each displayed incoming instant message horizontally aligned at a same first horizontal position;

displays the outgoing textual portion of each displayed outgoing instant message horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position; and

displays time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.

18. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to display an incoming symbol with each displayed incoming instant message.

19. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to display an outgoing symbol with each displayed outgoing instant message.

20. (previously presented) The non-transitory computer readable medium of claim 17, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

21. (previously presented) The non-transitory computer readable medium of claim 17, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

22. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to:

detect an interruption in the instant message conversation; and

refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

23. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps.

24. (previously presented) The non-transitory computer readable medium of claim 23, wherein the computer instructions that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps are executed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

25. (previously presented) The method of claim 1, the input comprising movement of a pointing device in proximity to the at least one instant message of the incoming and outgoing instant messages.

26. (previously presented) The electronic device of claim 9, the input comprising movement of a pointing device in proximity to the at least one instant message of the incoming and outgoing instant messages.

27. (previously presented) The non-transitory computer readable medium of claim 17, the input comprising movement of a pointing device in proximity to the at least one instant message of the incoming and outgoing instant messages.

REMARKS

Applicant thanks the Examiner for reviewing the present application and notes that the claims have not been amended.

Claims 1-5, 9-13, 17-21, and 25-27 have been rejected as being unpatentable over Appelman (U.S. Patent No. 7,181,497) in view of Maurille (U.S. Patent No. 6,484,196) and MacPhail (U.S. Patent No. 6,636,243). Applicant respectfully traverses the rejections as follows.

As previously argued, Applicant has found that selective display of time information in response to an input in association with particular communications saves space on the display, which is particularly important in handheld or wireless communication devices. In addition to these considerations, the present application discloses displaying incoming and outgoing textual portions aligned at first and second horizontal positions wherein the first and second horizontal positions are different from each other. Such horizontal alignment facilitates distinguishing between incoming and outgoing messages, e.g., those messages the user sent and those they received. When considering the selective display of time information, the ability to differentiate between incoming and outgoing messages in this way further facilitates associating an input for causing time information to be displayed with the message for which the user actually wants such time information.

Claim 1 as previously presented recites in part:

> "displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages".

Applicant respectfully submits that none of the cited references teach or fairly suggest the above features, let alone to address the same problems faced by Applicant.

Therefore, even if combined, Appelman, Maurille, and MacPhail fail to teach what is recited in claim 1 and provide no motivation to modify either reference in a way that renders claim 1 obvious.

Appelman may show, among other things, a conversation screen (see FIG. 18) that includes multiple outgoing (T) and multiple incoming (F) messages. However, in Appelman, the incoming and outgoing messages are all aligned at the same exact horizontal position. Moreover, Appelman clearly shows displaying a timestamp with each message, and thus illustrates exactly what the method of claim 1 is trying to avoid, namely the potential cluttering of a display by providing too much time information.

Although Maurille may show indenting email replies, Maurille has not addressed the competing objectives of conserving display space and providing useful time information. In fact, similar to Appelman, Maurille includes time information with each message. Maurille does not teach or suggest balancing the competing objectives of conserving display space and providing useful time information by enabling a selective display of time information, let alone as recited in claim 1.

The Examiner acknowledges that Appelman and Maurille do not teach: "displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages". MacPhail has been cited as teaching what is missing from Appelman and Maurille.

Applicant respectfully submits that the Examiner has read too much into MacPhail and that a person skilled in the art would not likely consider MacPhail in combination with Appelman and Maurille. Applicant further submits that even if a person skilled in the art were to consider MacPhail in combination with Appelman and Maurille, the combination still fails to teach each and every element that is recited in claim 1 for at least the reason that MacPhail does not teach or suggest selectively displaying a time stamp associated with an incoming or outgoing message in a messaging environment.

MacPhail teaches a system in which icons representing critical indicators are displayed in superposition with a reference shape. The reference shape is divided into "higher-interest" and "lower-interest" portions such that display of an icon over the higher-interest portion of the reference shape indicates a higher-interest value of the corresponding value. MacPhail therefore addresses problems in an entirely different environment than the present application. MacPhail has nothing to do with displaying time information in a messaging environment. MacPhail has also failed to recognize, let alone address, the competing objectives of providing useful time information and conserving display space, which is important in messaging environments on electronic devices. For at least these reasons, Applicant respectfully submits that a person skilled in the art would not consider MacPhail in combination with Appelman and Maurille.

Even if, for the sake of argument, a person skilled in the art were to consider Appelman, Maurille, and MacPhail in combination, Applicant respectfully submits that the combination still fails to teach what is recited in claim 1.

For example, in FIG. 4(a), which is relied upon by the Examiner, MacPhail teaches a series of icons displayed with reference shapes. Timestamps (88 and 90) are displayed in the vicinity of blinking icons (82 and 86). Timestamp (88) shows the time at which the variable corresponding to icon (82) most recently changed status and timestamp (90) shows the time at which the variable

corresponding to icon (86) changed status.

Accordingly, not only may there be multiple timestamps for a particular icon, the timestamps displayed by MacPhail may in fact change over time, i.e., according to a most recent change in status. In the present application, the time stamp is meant to be in association with a message such that a user can determine information about when the message was sent or received. If the timestamp was related to the status of the message as taught by MacPhail then, for example, the time at which the message was sent would not be available and thus the objective of the method of claim 1 would not be achieved. MacPhail therefore teaches displaying time stamps for a different purpose, in a different manner, with different results. Moreover, the different results may vary over time. Therefore, time stamping of messages is clearly not contemplated by MacPhail.

Although MacPhail may suggest displaying timestamps to indicate a change in status, there is nothing in MacPhail that teaches displaying a timestamp for a message in a messaging environment. MacPhail also fails to teach or suggest that the time stamps associated with the status changes could be adapted to such a scenario. MacPhail is entirely silent in that regard.

In summary:

a) MacPhail is silent regarding selective display of time stamps in association with messages in a messaging environment. MacPhail only teaches enabling time stamps to be displayed for status changes for an icon.

b) MacPhail clearly shows more than one time stamp in FIG. 4(a). Therefore, MacPhail does not necessarily restrict the display of timestamps, let alone by displaying only a timestamp for a most recent status change.

c) Since the icons in MacPhail are meant to represent critical indicators and to express when the variables have reached critical thresholds, the icons may be associated with several timestamps. This is quite dissimilar to selectively displaying time information for a message. For example, if several critical thresholds have been reached at different times, for multiple variables, multiple timestamps for multiple icons (i.e. many timestamps) would be displayed by MacPhail.

For at least the above reasons, not only is MacPhail an unlikely combination with Appelman and Maurille, even if combined, MacPhail fails to teach what is missing from Appelman and Maurille.

Accordingly, Applicant respectfully submits that claim 1 is patentable over Appelman, Maurille, and MacPhail. Similar arguments apply to claims 9 and 17 and those claims dependent thereon.

Claims 6-8, 14-16, and 22-24 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Appelman, Maurille and MacPhail, in further view of Yamada (U.S. Patent No. 6,889,063). Applicant respectfully traverses the rejections as follows.

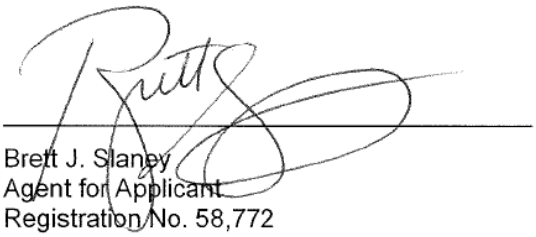
Although Yamada may teach providing an alert after a predetermined amount of time has passed since an in-absence incoming call message has been displayed, Yamada fails to teach or suggest displaying incoming and outgoing textual portions aligned at first and second horizontal positions wherein the first and second horizontal positions are different from each other.

Yamada also fails to teach "displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages". As such, Applicant respectfully submits that Yamada fails to teach what is missing from Appelman, Maurille, and MacPhail per the above and thus claims 6-8, 14-16, and 22-24 are patentable over Appelman, Maurille, and MacPhail, in further view of Yamada.

* * *

In view of the foregoing, Applicant respectfully submits that the claims as amended are patentable over the cited reference and thus the present application is in condition for allowance. Applicant therefore requests early reconsideration and allowance of the present application.

Respectfully submitted,



Brett J. Slaney
Agent for Applicant
Registration No. 58,772

Date: August 9, 2012

BLAKE, CASSELS & GRAYDON LLP
199 Bay Street
Suite 4000, Commerce Court West
Toronto ON M5L 1A9
Canada

Tel: 416-863-2518

BS/jmmm/dm

Electronic Patent Application Fee Transmittal

Application Number:	13111734
Filing Date:	19-May-2011
Title of Invention:	Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment
First Named Inventor/Applicant Name:	Gerhard D. Klassen
Filer:	Brett Joseph Slaney/Sandy Barbieri
Attorney Docket Number:	70314/00568

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Notice of appeal	1401	1	620	620
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**620**

Electronic Acknowledgement Receipt	
EFS ID:	13462048
Application Number:	13111734
International Application Number:	
Confirmation Number:	6081
Title of Invention:	Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment
First Named Inventor/Applicant Name:	Gerhard D. Klassen
Customer Number:	91704
Filer:	Brett Joseph Slaney/Sandy Barbieri
Filer Authorized By:	Brett Joseph Slaney
Attorney Docket Number:	70314/00568
Receipt Date:	09-AUG-2012
Filing Date:	19-MAY-2011
Time Stamp:	17:27:40
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$620
RAM confirmation Number	4344
Deposit Account	022553
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees) Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Notice of Appeal Filed	11144-US-CNT3_notice_of_appeal.pdf	72063 911875b81c2a877abcb182817e73a3208a642377	no	1
Warnings:					
Information:					
2		11144-US-CNT3_response_after_final_rejection.pdf	504143 1b0584c3d862af613a1548e730a65dec638523d8	yes	10
	Multipart Description/PDF files in .zip description				
	Document Description		**Start**	**End**	
	Amendment After Final		1	1	
	Claims		2	6	
	Applicant Arguments/Remarks Made in an Amendment		7	10	
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	30337 42791db1869bf9fc417c88420e2d6cb363f9a74e	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	606543		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/31 (07-09)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

NOTICE OF APPEAL FROM THE EXAMINER TO THE BOARD OF PATENT APPEALS AND INTERFERENCES

Docket Number (Optional): 70314/00568

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)]
on _____

Signature _____

Typed or printed name _____

In re Application of: KLASSEN, Gerhard D. et al

Application Number	Filed
13/111,734	May 19, 2011

For: Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment

Art Unit	Examiner
2457	LAI, Michael C.

Applicant hereby **appeals** to the Board of Patent Appeals and Interferences from the last decision of the examiner.

The fee for this Notice of Appeal is (37 CFR 41.20(b)(1)) $ 620

☐ Applicant claims small entity status. See 37 CFR 1.27. Therefore, the fee shown above is reduced by half, and the resulting fee is: $ _____

☐ A check in the amount of the fee is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☑ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 02-2553 .

☐ A petition for an extension of time under 37 CFR 1.136(a) (PTO/SB/22) is enclosed.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.
(Form PTO/SB/96)

☑ attorney or agent of record. Registration number 58,772

☐ attorney or agent acting under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34. _____

Signature

Brett J. SLANEY
Typed or printed name

416-863-2518
Telephone number

August 9, 2012
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required.
Submit multiple forms if more than one signature is required, see below*.

☐ *Total of _____ forms are submitted.

This collection of information is required by 37 CFR 41.31. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11, 1.14 and 41.6. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

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.
13/111,734	05/19/2011	Gerhard D. Klassen	70314/00568	6081

91704 7590 06/22/2012

Blake, Cassels & Graydon LLP
199 BAY STREET , SUITE 4000
COMMERCE COURT WEST
TORONTO, ON M5L 1A9
CANADA

EXAMINER	
LAI, MICHAEL C	
ART UNIT	PAPER NUMBER
2457	

NOTIFICATION DATE	DELIVERY MODE
06/22/2012	ELECTRONIC

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.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

rimpatent@blakes.com
brett.slaney@blakes.com
portfolioprosecution@rim.com

Office Action Summary	Application No.	Applicant(s)	
	13/111,734	KLASSEN ET AL.	
	Examiner	Art Unit	
	MICHAEL C. LAI	2457	

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

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 April 2012*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *2/28/2012*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is responsive to communication filed on 04/09/2012.

Response to Amendment

2. The examiner has acknowledged the amended claims 1, 9, 17, and new claims 25-27. Claims 1-27 are pending.

Response to Arguments

3. Applicant's arguments with respect to the claims have been considered but are moot in view of the new ground(s) of rejection.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-5, 9-13, 17-21, and 25-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over Appelman et al. (US 7,181,497 B1, hereinafter Appelman), in view of Maurille **(**US 6,484,196 B1, hereinafter Maurille) and MacPhail (US 6,636,243 B1, hereinafter MacPhail).

Regarding claim 1, Appelman discloses a method of displaying an instant message conversation on an electronic device [see at least Fig. 7, User 1; Fig. 16, the user], the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device [see at least Fig. 7, User 2; Fig. 16, mroe1934], the method comprising:

receiving a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48];

transmitting a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34];

associating each instant message with a corresponding time stamp [see at least Figs. 16, 18; col. 9, lines 23-48];

displaying the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48]; and

displaying the outgoing textual portion of each displayed outgoing instant message being horizontally aligned at a same second horizontal position [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34].

Appelman does not disclose the second horizontal position being different from the first horizontal position. However, Maurille discloses “Each incoming e-mail message is treated by the server application 114 as a level one message that begins a respective e-mail thread. Replies to the incoming e-mail messages are indented on the mail screen 900, visually indicating their position as second level messages in their associated thread” [col. 20, lines 9-15]. It would have been obvious to a person with ordinary skill in the art at the time the invention

was made to incorporate Maurille's teaching into Appelman's method for the purpose of visually differentiating between incoming and outgoing messages by aligning/indenting incoming and outgoing textual portions at different horizontal positions, thereby providing a more user friendly Instant Messaging system.

Appelman and Maurille disclose the claimed invention except for displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages. However, MacPhail teaches a timestamp indicating the most recent time of a status change could be displayed only upon request by a viewer (or "user"), as illustrated in FIG. 4(b), icon 80. MacPhail further discloses by simply positioning a pointer over an icon (read as a request or an input) may be sufficient to cause display of the timestamp [see col. 10, lines 3-22]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate MacPhail's teaching into Appelman's and Maurille's method for the purpose of providing displays on-demand by displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages, thereby providing effective displays on devices having diverse sizes including those with small,

monochrome displays and capabilities [see at least the abstract and col. 2, lines 53-56].

Regarding claim 2, Appelman further discloses displaying an incoming symbol with each displayed incoming instant message [see at least Figs. 16, 18, “F>” (from) is the incoming symbol; col. 9, lines 35-48].

Regarding claim 3, Appelman further discloses displaying an outgoing symbol with each displayed outgoing instant message [see at least Figs. 16, 18, “T>” (to) is the outgoing symbol; col. 9, lines 23-34].

Regarding claim 4, Appelman further discloses wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message [see at least Figs. 16, 18, “13:20:27”; col. 9, lines 35-48].

Regarding claim 5, Appelman further discloses wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message [see at least Figs. 16, 18, “13:20:05”; col. 9, lines 23-34].

Regarding claim 9, Appelman discloses an electronic device for displaying an instant message conversation, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device [see at least Fig. 7, User 1; Fig. 16, the user] and a second electronic device [see at least Fig. 7, User 2; Fig. 16, mroe1934], the electronic device comprising:

a display [Fig. 1, 107, and Fig. 6];

a memory [Fig. 1, 109]; and

a processor [Fig. 1, 121] electronically coupled with the display and the memory, the processor configured to:

receive a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48];

transmit a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34];

associate each instant message with a corresponding time stamp [see at least Figs. 16, 18; col. 9, lines 23-48];

display the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48]; and

display the outgoing textual portion of each displayed outgoing instant message being horizontally aligned at a same second horizontal position [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34].

Appelman does not disclose the second horizontal position being different from the first horizontal position. However, Maurille discloses “Each incoming e-mail message is treated by the server application 114 as a level one message that begins a respective e-mail thread. Replies to the incoming e-mail messages

are indented on the mail screen 900, visually indicating their position as second level messages in their associated thread" [col. 20, lines 9-15]. It would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Maurille's teaching into Appelman's device for the purpose of visually differentiating between incoming and outgoing messages by aligning/indenting incoming and outgoing textual portions at different horizontal positions, thereby providing a more user friendly Instant Messaging system.

Appelman and Maurille disclose the claimed invention except for displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages. However, MacPhail teaches a timestamp indicating the most recent time of a status change could be displayed only upon request by a viewer (or "user"), as illustrated in FIG. 4(b), icon 80. MacPhail further discloses by simply positioning a pointer over an icon (read as a request or an input) may be sufficient to cause display of the timestamp [see col. 10, lines 3-22]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate MacPhail's teaching into Appelman's and Maurille's method for the purpose of providing displays on-demand by displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant

message of the incoming and outgoing instant messages, thereby providing effective displays on devices having diverse sizes including those with small, monochrome displays and capabilities [see at least the abstract and col. 2, lines 53-56].

Regarding claim 10, Appelman further discloses wherein the processor is further configured to display an incoming symbol with each displayed incoming instant message [see at least Figs. 16, 18, “F>” (from) is the incoming symbol; col. 9, lines 35-48].

Regarding claim 11, Appelman further discloses wherein the processor is further configured to display an outgoing symbol with each displayed outgoing instant message [see at least Figs. 16, 18, “T>” (to) is the outgoing symbol; col. 9, lines 23-34].

Regarding claim 12, Appelman further discloses wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message [see at least Figs. 16, 18, “13:20:27”; col. 9, lines 35-48].

Regarding claim 13, Appelman further discloses wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message [see at least Figs. 16, 18, “13:20:05”; col. 9, lines 23-34].

Regarding claim 17, Appelman discloses a non-transitory computer readable medium comprising computer executable instructions embedded thereon for execution by a processor of an electronic device for displaying an instant

message conversation upon a display of the electronic device, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device [see at least Fig. 7, User 1; Fig. 16, the user] and a second electronic device [see at least Fig. 7, User 2; Fig. 16, mroe1934], such that when executed, the processor:

receives a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48];

transmits a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34];

associates each instant message with a corresponding time stamp [see at least Figs. 16, 18; col. 9, lines 23-48];

displays the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48]; and

displays the outgoing textual portion of each displayed outgoing instant message being horizontally aligned at a same second horizontal position [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34].

Appelman does not disclose the second horizontal position being different from the first horizontal position. However, Maurille discloses “Each incoming e-

mail message is treated by the server application 114 as a level one message that begins a respective e-mail thread. Replies to the incoming e-mail messages are indented on the mail screen 900, visually indicating their position as second level messages in their associated thread" [col. 20, lines 9-15]. It would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Maurille's teaching into Appelman's system for the purpose of visually differentiating between incoming and outgoing messages by aligning/indenting incoming and outgoing textual portions at different horizontal positions, thereby providing a more user friendly Instant Messaging system.

Appelman and Maurille disclose the claimed invention except for displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages. However, MacPhail teaches a timestamp indicating the most recent time of a status change could be displayed only upon request by a viewer (or "user"), as illustrated in FIG. 4(b), icon 80. MacPhail further discloses by simply positioning a pointer over an icon (read as a request or an input) may be sufficient to cause display of the timestamp [see col. 10, lines 3-22]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate MacPhail's teaching into Appelman's and Maurille's method for the purpose of providing displays on-demand by displaying time information along with at least one instant message of

the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages, thereby providing effective displays on devices having diverse sizes including those with small, monochrome displays and capabilities [see at least the abstract and col. 2, lines 53-56].

Regarding claim 18, Appelman further discloses computer instructions such that when executed cause the processor to display an incoming symbol with each displayed incoming instant message [see at least Figs. 16, 18, “F>” (from) is the incoming symbol; col. 9, lines 35-48].

Regarding claim 19, Appelman further discloses computer instructions such that when executed cause the processor to display an outgoing symbol with each displayed outgoing instant message [see at least Figs. 16, 18, “T>” (to) is the outgoing symbol; col. 9, lines 23-34].

Regarding claim 20, Appelman further discloses wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message [see at least Figs. 16, 18, “13:20:27”; col. 9, lines 35-48].

Regarding claim 21, Appelman further discloses wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message [see at least Figs. 16, 18, “13:20:05”; col. 9, lines 23-34].

Regarding claims 25-27, Appelman, Maurille, and MacPhail disclose the claimed invention including the input comprising movement of a pointing device

in proximity to the at least one instant message of the incoming and outgoing instant messages [MacPhail: col. 10, lines 3-22]. See claim 1 for motivation.

6. Claims 6-8, 14-16, and 22-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Appelman, Maurille, and MacPhail as applied to claim 1, and further in view of Yamada (US 6,889,063 B2, hereinafter Yamada).

Regarding claim 6, Appelman, Maurille, and MacPhail disclose the method of claim 1 but are silent about detecting an interruption in the instant message conversation; and refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time. However, Appelman teaches displaying time stamp on messages [see at least claim 1] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's, Maurille's, and MacPhail's method in order to alert the user only

when the timer expires before next message by refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time [see the abstract].

Regarding claim 7, Appelman, Maurille, and MacPhail disclose the method of claim 1 but are silent about refraining from displaying at least one of the corresponding time stamps. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's, Maurille's, and MacPhail's method in order to alert the user only when the timer expires before next message by refraining from displaying at least one of the corresponding time stamps [see the abstract].

Regarding claim 8, Appelman, Maurille, MacPhail, and Yamada disclose the method of claim 7, including wherein the refraining is performed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time [Yamada, see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. See claim 6 for motivation.

Regarding claim 14, Appelman, Maurille, and MacPhail disclose the electronic device of claim 9 but is silent about wherein the processor is further configured to: detect an interruption in the instant message conversation; and refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence

incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's, Maurille's, and MacPhail's electronic device in order to alert the user only when the timer expires before next message by refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time [see the abstract].

Regarding claim 15, Appelman, Maurille, and MacPhail disclose the electronic device of claim 9 but are silent about wherein the processor is further configured to refrain from displaying at least one of the corresponding time stamps. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's, Maurille's, and MacPhail's electronic device

in order to alert the user only when the timer expires before next message by refraining from displaying at least one of the corresponding time stamps [see the abstract].

Regarding claim 16, Appelman, Maurille, MacPhail, and Yamada disclose the electronic device of claim 15, including wherein the processor refrains from displaying the at least one of the corresponding time stamps if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time [Yamada, see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. See claim 14 for motivation.

Regarding claim 22, Appelman, Maurille, and MacPhail disclose the computer readable medium of claim 17 but is silent about computer instructions such that when executed cause the processor to: detect an interruption in the instant message conversation; and refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence

incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's, Maurille's, and MacPhail's computer readable medium in order to alert the user only when the timer expires before next message by refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time [see the abstract].

Regarding claim 23, Appelman, Maurille, and MacPhail disclose the computer readable medium of claim 17 but is silent about comprising computer instructions such that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call

[see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's, Maurille's, and MacPhail's computer readable medium in order to alert the user only when the timer expires before next message by refraining from displaying at least one of the corresponding time stamps [see the abstract].

Regarding claim 24, Appelman, Maurille, MacPhail, and Yamada disclose the computer readable medium of claim 23, including wherein the computer instructions that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps are executed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a 5 predetermined duration of time [Yamada, see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. See claim 22 for motivation.

Conclusion

Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Examiner's Note: Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. In the case of amending the claimed invention, Applicant is respectfully requested to indicate the portion(s) of the specification which dictate(s) the structure relied on for proper interpretation and also to verify and ascertain the metes and bounds of the claimed invention.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL C. LAI whose telephone number is (571)270-3236. The examiner can normally be reached on M-F 9:00 - 5:30 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571) 272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Michael C. Lai
Art Unit 2457
Phone: (571) 270-3236
Fax: (571) 270-4236

/ARIO ETIENNE/
Supervisory Patent Examiner, Art Unit 2457

Notice of References Cited	Application/Control No. 13/111,734	Applicant(s)/Patent Under Reexamination KLASSEN ET AL.	
	Examiner MICHAEL C. LAI	Art Unit 2457	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-5,896,567 A	04-1999	Ogushi, Masuo	455/421
*	B	US-2004/0205775 A1	10-2004	Heikes et al.	719/318
*	C	US-6,889,063 B2	05-2005	Yamada, Hironori	455/567
*	D	US-6,940,407 B2	09-2005	Miranda-Knapp et al.	340/572.1
*	E	US-7,181,497 B1	02-2007	Appelman et al.	709/206
*	F	US-6,484,196	11-2002	Maurille, Mithras C.	709/206
*	G	US-6,636,243	10-2003	MacPhail, Margaret Gardner	715/772
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13111734	KLASSEN ET AL.
	Examiner	Art Unit
	MICHAEL C LAI	2457

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	07/28/2011	01/04/2012	06/04/2012						
	1	✓	✓	✓						
	2	✓	✓	✓						
	3	✓	✓	✓						
	4	✓	✓	✓						
	5	✓	✓	✓						
	6	✓	✓	✓						
	7	✓	✓	✓						
	8	✓	✓	✓						
	9	✓	✓	✓						
	10	✓	✓	✓						
	11	✓	✓	✓						
	12	✓	✓	✓						
	13	✓	✓	✓						
	14	✓	✓	✓						
	15	✓	✓	✓						
	16	✓	✓	✓						
	17	✓	✓	✓						
	18	✓	✓	✓						
	19	✓	✓	✓						
	20	✓	✓	✓						
	21	✓	✓	✓						
	22	✓	✓	✓						
	23	✓	✓	✓						
	24	✓	✓	✓						
	25			✓						
	26			✓						
	27			✓						

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13111734	KLASSEN ET AL.
	Examiner	Art Unit
	MICHAEL C LAI	2457

SEARCHED			
Class	Subclass	Date	Examiner
Inventor search		7/25/11	Lai
709	206, 207	7/26/11	Lai

SEARCH NOTES		
Search Notes	Date	Examiner
EAST text search	7/26/11	Lai
EAST	1/3/12	Lai
Consulted Joni Hsu for display search	1/4/12	Lai
EAST	6/4/12	Lai

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	4	"6636243"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/06/04 16:47
S1	11346	709/206.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 11:57
S2	2237	709/207.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 11:58
S3	1028	S1 S2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 11:58
S4	13	"7181497"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 12:00
S5	137	"6539421"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 12:17
S6	4	"5896567"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:02
S7	492201	(without or no or interrupt$4 or discontinu$4 or disconnect$4) with (communication or conversation or messag$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:55
S8	300348	time near2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:56
S9	1409	S7 with S8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:56

S10	27935	(display$4 or output$4) near4 time near2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S11	56	S9 same S10	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S12	11346	709/206.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S13	2237	709/207.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S14	12555	S12 or S13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S15	3	S11 S14	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S16	25	S11 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 16:15
S17	26	"7007085"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 16:36
S18	3	"6889063"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 18:40
S19	125	(without or refrain$4 or stop$4) adj2 (display$4 or output$4) near4 time near2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:27
S20	492201	(without or no or interrupt$4 or discontinu$4 or disconnect$4) with (communication or conversation or messag$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:27
S21	12	S19 S20	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:27

S22	19	(without or refrain$4 or stop$4) adj2 (display$4 or output$4) adj4 time adj2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:36
S23	19	S22 not S21	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:37
S24	13	S23 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:37
S25	66	S19 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:49
S26	53	S25 not S24	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:49
S27	13	"6940407"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/27 10:36
S28	3	"20040205775"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/27 10:58
S29	4	"5896567"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/27 11:00
S30	262224	(without or no or interrupt$4 or discontinu$4 or disconnect$4) near4 (communication or conversation or messag$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:49
S31	11361	709/206.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50
S32	2238	709/207.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50
S33	12571	S31 or S32	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50

S34	5972	S30 S33	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50
S35	2017	S34 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50
S36	39382	(interrupt$4) near4 (communication or conversation or messag$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:51
S37	12571	S33 S33	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:51
S38	441	S33 S36	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:52
S39	157	S38 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:53
S40	27960	(display$4 or output$4) near4 time near2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:53
S41	2	S39 S40	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:53
S42	443	(interrupt$4) near4 (IM or instant adj1 messag$4 or chat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:56
S43	443435	timer or time$1stamp$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:57
S44	15	S42 same S43	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:57
S45	11	S44 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:57

S46	94	S33 S42	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:00
S47	6999	S30 same S43	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:02
S48	142	S47 S33	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:02
S49	47	S48 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:02
S50	7125	(without or no or interrupt$4 or discontinu$4 or disconnect$4) near4 (IM or chat$4 or instant adj1 messag$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:19
S51	68	S50 same S43	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:20
S52	12	S51 S33	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:20
S53	1	S52 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:21
S54	105	(without or no or interrupt$4 or discontinu$4 or disconnect$4) near4 (IM or chat$4 or instant adj1 messag$4) near4 conversation	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 16:15
S55	18	S54 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 16:16
S56	711	display$4 with incoming with outgoing with message$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:12
S57	0	align$4 with different$2 with horizontaly with position	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:14

S58	11135	align$4 with different$2 with position	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:15
S59	16761	align$4 with different$2 with position$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:15
S60	1	S56 S59	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:15
S61	474793	align$4 with position$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:17
S62	1	S56 S60	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:17
S63	5936	display$4 with receiving with (send$4 or transmit$4) with message$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:18
S64	0	S63 S60	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:19
S65	3	S56 S61	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:19
S66	44	S63 S61	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:21
S67	44	S66 not S65	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:21
S68	16761	align$4 with different$2 with position$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:28
S69	7202	display$4 with receiving with (send$4 or transmit$4) with (text$4 or message$1)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:29

S70	4	S68 S69	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:29
S71	0	display$4 with to with (from) with (text$4 or message$1)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:37
S72	9	indent$6 with incoming with message$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 14:19
S73	0	WO 02/21413	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:13
S74	0	WO 2002/21413	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:14
S75	1	WO "200221413"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:14
S76	2	WO "2002021413"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:14
S77	118	"7130885"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:16

EAST Search History (Interference)

<This search history is empty>

6/4/2012 4:52:02 PM
C:\Users\mlai\Documents\EAST\Workspaces\13111734.wsp

PTO/SB/08A (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	13/111,734
Filing Date	May 19, 2011
First named Inventor	KLASSEN, Gerhard D.
Art Unit	2457
Examiner Name	LAI, Michael C.
Attorney Docket Number	70314/00568

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/M.L./		US-2003/0001890 A1	01-02-2003	BRIN	
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]	Number[4]	Kind-Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]

Examiner Signature	/Michael Lai/	Date Considered	06/04/2012

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] **Applicants' unique citation designation number** (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Appl. No.:	**13/111,734**
Applicant:	**KLASSEN, Gerhard D. et al.**
Filed:	**May 19, 2011**
Title:	**Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment**
Art Unit:	**2457**
Examiner:	**LAI, Michael C.**
Docket No.:	**70314/00568**

Mail Stop Amendment
U.S. Patent & Trademark Office
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE

Sir:

This is further to the Office Action dated January 19. 2012. Applicant wishes to amend the above-identified application as follows:

Amendments to the Claims: are reflected in the listing of claims which begins on page 2 of this paper.

Remarks: begin on page 7 of this paper.

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of claims:

1. (currently amended) A method of displaying an instant message conversation on an electronic device, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, the method comprising:

receiving a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmitting a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associating each instant message with a corresponding time stamp;

displaying the incoming textual portion ~~of at least one of the incoming instant messages along with a respective time stamp, the incoming textual portion~~ of each displayed incoming instant message ~~being~~ horizontally aligned at a same first horizontal position; [[and]]

displaying the outgoing textual portion ~~of at least one of the outgoing instant messages along with a respective time stamp, the outgoing textual portion~~ of each displayed outgoing instant message ~~being~~ horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position[[.]]; and

displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.

2. (original) The method of claim 1 further comprising displaying an incoming symbol with each displayed incoming instant message.

3. (original) The method of claim 1 further comprising displaying an outgoing symbol with each displayed outgoing instant message.

4. (original) The method of claim 1, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

5. (original) The method of claim 1, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

6. (original) The method of claim 1, further comprising:

detecting an interruption in the instant message conversation; and

refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

7. (original) The method of claim 1, further comprising refraining from displaying at least one of the corresponding time stamps.

8. (original) The method of claim 7, wherein the refraining is performed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

9. (currently amended) An electronic device for displaying an instant message conversation, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, the electronic device comprising:

a display;

a memory; and

a processor electronically coupled with the display and the memory, the processor configured to:

receive a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmit a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associate each instant message with a corresponding time stamp;

display the incoming textual portion ~~of at least one of the incoming instant messages along with a respective time stamp, the incoming textual portion~~ of each displayed incoming instant message ~~being~~ horizontally aligned at a same first horizontal position; [[and]]

display the outgoing textual portion ~~of at least one of the outgoing instant messages along with a respective time stamp, the outgoing textual portion~~ of each displayed outgoing instant

message ~~being~~ horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position[[.]]; and

display time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.

10. (original) The electronic device of claim 9, wherein the processor is further configured to display an incoming symbol with each displayed incoming instant message.

11. (original) The electronic device of claim 9, wherein the processor is further configured to display an outgoing symbol with each displayed outgoing instant message.

12. (original) The electronic device of claim 9, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

13. (original) The electronic device of claim 9, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

14. (original) The electronic device of claim 9, wherein the processor is further configured to:

detect an interruption in the instant message conversation; and

refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

15. (original) The electronic device of claim 9, wherein the processor is further configured to refrain from displaying at least one of the corresponding time stamps.

16. (original) The electronic device of claim 15, wherein the processor refrains from displaying the at least one of the corresponding time stamps if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

17. (currently amended) A non-transitory computer readable medium comprising

computer executable instructions embedded thereon for execution by a processor of an electronic device for displaying an instant message conversation upon a display of the electronic device, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, such that when executed, the processor:

receives a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmits a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associates each instant message with a corresponding time stamp;

displays the incoming textual portion ~~of at least one of the incoming instant messages along with a respective time stamp, the incoming textual portion~~ of each displayed incoming instant message ~~being~~ horizontally aligned at a same first horizontal position; [[and]]

displays the outgoing textual portion ~~of at least one of the outgoing instant messages along with a respective time stamp, the outgoing textual portion~~ of each displayed outgoing instant message ~~being~~ horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position[[.]]; and

displays time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.

18. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to display an incoming symbol with each displayed incoming instant message.

19. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to display an outgoing symbol with each displayed outgoing instant message.

20. (previously presented) The non-transitory computer readable medium of claim 17, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

21. (previously presented) The non-transitory computer readable medium of claim 17, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant

message.

22. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to:

detect an interruption in the instant message conversation; and

refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

23. (previously presented) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps.

24. (previously presented) The non-transitory computer readable medium of claim 23, wherein the computer instructions that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps are executed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

25. (new) The method of claim 1, the input comprising movement of a pointing device in proximity to the at least one instant message of the incoming and outgoing instant messages.

26. (new) The electronic device of claim 9, the input comprising movement of a pointing device in proximity to the at least one instant message of the incoming and outgoing instant messages.

27. (new) The non-transitory computer readable medium of claim 17, the input comprising movement of a pointing device in proximity to the at least one instant message of the incoming and outgoing instant messages.

REMARKS

Applicant wishes to thank the Examiner for reviewing the present application.

Claim Amendments

Claim 1 has been amended removing reference to displaying "a respective time stamp" for each of the incoming and outgoing textual portions and inserting the following operation: "displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages".

Support for these amendments can be found in at least the examples described in paragraphs [0050] and [0051] and shown in FIG. 6a of the application as filed.

Claims 9 and 17 have been amended in a manner consistent with claim 1 as amended.

Claims 25-27 have been added and are directed to an example wherein the input comprises: "movement of a pointing device in proximity to the at least one of the incoming or outgoing instant messages".

Support for new claims 25-27 can be found in at least paragraph [0050] of the application as filed.

Claim Rejections – 35 U.S.C. 103

Claims 1-5, 9-13, and 17-21 have been rejected as being unpatentable over Appelman (U.S. Patent No. 7,181,497) in view of Maurille (U.S. Patent No. 6,484,196). Applicant respectfully traverses the rejections as follows.

The present application describes and claims methods, computer readable media, and electronic devices, operable to balance the competing objectives of providing useful information (such as a time stamp for a particular message) and conserving space on the display, which is particularly important in handheld or wireless communication devices. By enabling the displaying of time stamps in the way recited in claims 1, 14, and 27, these competing objectives can be balanced.

The present application also addresses the issue that although the availability of certain timing aspects in a data communication exchange is desirable, the amount of space required to do so can be a competing objective. Applicant has recognized that the timing of messages is particularly important when there is a non-responsive or relatively older message that has been

displayed. In a normal back-to-back exchange, the indication of the time at which it was sent may not be deemed to be significant.

However, when there is a gap in the conversation or a user wishes to determine a time associated with a message previously displayed, a time stamp associated with a particular message or indication of a message can be desirable.

Therefore, it has been found that selective display of time information in response to an input in association with particular communications saves space on the display, which is particularly important in handheld or wireless communication devices.

For example, Fig. 6b illustrates that, by determining a particular amount of time has passed since a message indicating "I'll pick you up in an hour", and then displaying a time associated with that message (e.g. 5:04 pm as shown), meaningful information is conveyed since a reference point for "an hour" is provided.

In addition to the above considerations, the present application discloses displaying incoming and outgoing textual portions aligned at first and second horizontal positions wherein the first and second horizontal positions are different from each other.

Such horizontal alignment facilitates distinguishing between incoming and outgoing messages, e.g., those messages the user sent and those they received. When considering the selective display of time information, the ability to differentiate between incoming and outgoing messages in this way further facilitates associating an input for causing time information to be displayed with the message for which the user actually wants such time information.

Claim 1 as amended recites in part:

> "displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages.".

Applicant respectfully submits that neither Appelman nor Maurille teach or fairly suggest the above features, let alone to address the same problems faced by Applicant.

Therefore, even if combined, Appelman and Maurille fail to teach what is recited in the claimed invention and provide no motivation to modify either reference in a way that renders claim 1 obvious.

As previously argued, Appelman may show, among other things, a conversation screen (see

FIG. 18) that includes multiple outgoing (T) and multiple incoming (F) messages. However, in Appelman, the incoming and outgoing messages are all aligned at the same exact horizontal position.

Moreover, Appelman clearly shows displaying a timestamp with each message, and thus illustrates exactly what the claimed invention is trying to avoid, namely the potential cluttering of a display by providing too much time information.

Although Maurille may shown indenting email replies, Maurille has not addresses the competing objectives of conserving display space and providing useful time information. In fact, similar to Appelman, Maurille includes time information with each message.

Maurille does not teach or suggest balancing the competing objectives of conserving display space and providing useful time information by enabling a selective display of time information, let alone as recited in claim 1.

For at least the above reasons, Applicant respectfully submits that claim 1 is patentable over Appelman in view of Maurille. Similar arguments apply to claims 9 and 17 and those claims dependent thereon.

Claims 6-8, 14-16, and 22-24 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Appelman and Maurille, in further view of Yamada (U.S. Patent No. 6,889,063). Applicant respectfully traverses the rejections as follows.

Although Yamada may teach providing an alert after a predetermined amount of time has passed since an in-absence incoming call message has been displayed, Yamada fails to teach or suggest displaying incoming and outgoing textual portions aligned at first and second horizontal positions wherein the first and second horizontal positions are different from each other and thus cannot anticipate.

Yamada also fails to teach "displaying time information along with at least one instant message of the incoming and outgoing instant messages according to the corresponding time stamp, in response to an input in association with the at least one instant message of the incoming and outgoing instant messages". As such, Applicant respectfully submits that Yamada fails to teach what is missing from Appelman and Maurille per the above and thus claims 6-8, 14-16, and 22-24 are patentable over Appelman and Maurille, in further view of Yamada.

* * *

In view of the foregoing, Applicant respectfully submits that the claims as amended are patentable over the cited reference and thus the present application is in condition for allowance. Applicant therefore requests early reconsideration and allowance of the present application.

Respectfully submitted,

Brett J. Slaney
Agent for Applicant
Registration No. 58,772

Date: April 9, 2012

BLAKE, CASSELS & GRAYDON LLP
199 Bay Street
Suite 4000, Commerce Court West
Toronto ON M5L 1A9
Canada

Tel: 416-863-2518

BS/jmmm

22221833.1

Electronic Patent Application Fee Transmittal

Application Number:	13111734
Filing Date:	19-May-2011
Title of Invention:	Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment
First Named Inventor/Applicant Name:	Gerhard D. Klassen
Filer:	Brett Joseph Slaney/Judith Martin
Attorney Docket Number:	70314/00568

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Claims in excess of 20	1202	3	60	180
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
			Total in USD ($)	**180**

Electronic Acknowledgement Receipt	
EFS ID:	12501081
Application Number:	13111734
International Application Number:	
Confirmation Number:	6081
Title of Invention:	Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment
First Named Inventor/Applicant Name:	Gerhard D. Klassen
Customer Number:	91704
Filer:	Brett Joseph Slaney/Judith Martin
Filer Authorized By:	Brett Joseph Slaney
Attorney Docket Number:	70314/00568
Receipt Date:	09-APR-2012
Filing Date:	19-MAY-2011
Time Stamp:	17:41:41
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$180
RAM confirmation Number	4309
Deposit Account	022553
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows: Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees) Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		11144-US-CNT3_OA-Response.pdf	462289 47045c9e04f8a8311b88d47ef874f0442c44d20d	yes	10

Multipart Description/PDF files in .zip description		
Document Description	**Start**	**End**
Amendment/Req. Reconsideration-After Non-Final Reject	1	1
Claims	2	6
Applicant Arguments/Remarks Made in an Amendment	7	10

Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	30323 38f4c4bfa62272cd9a75f51bcff7172f8f96c21c	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	492612		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
13/111,734	05/19/2011	

APPLICATION AS FILED – PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY ☐ RATE ($)	SMALL ENTITY ☐ FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT 04/09/2012

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 27	Minus	** 24	= 3	X $ =		OR	X $60=	180
Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $ =		OR	X $250=	0
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	180

AMENDMENT

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/LAVINIA JOHNSON/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08A (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	13/111,734
Filing Date	May 19, 2011
First named Inventor	KLASSEN, Gerhard D.
Art Unit	2457
Examiner Name	LAI, Michael C.
Attorney Docket Number	70314/00568

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-2003/0001890 A1	01-02-2003	BRIN	
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]	Number[4]	Kind-Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]

Examiner Signature		Date Considered	

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] **Applicants' unique citation designation number** (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

Electronic Acknowledgement Receipt	
EFS ID:	12182829
Application Number:	13111734
International Application Number:	
Confirmation Number:	6081
Title of Invention:	Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment
First Named Inventor/Applicant Name:	Gerhard D. Klassen
Customer Number:	91704
Filer:	Brett Joseph Slaney/Judith Martin
Filer Authorized By:	Brett Joseph Slaney
Attorney Docket Number:	70314/00568
Receipt Date:	28-FEB-2012
Filing Date:	19-MAY-2011
Time Stamp:	18:05:45
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		11144-US-CNT3_IDS.pdf	137768 00ddc4d0e691f2319fd6458d3737448a83e1be56	yes	3

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Transmittal Letter	1	2
	Information Disclosure Statement (IDS) Form (SB08)	3	3
Warnings:			
Information:			
	Total Files Size (in bytes):	137768	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Appl. No.: **13/111,734**

Applicant: **KLASSEN, Gerhard D. et al.**

Filed: **May 19, 2011**

Title: **Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment**

Art Unit: **2457**

Examiner: **LAI, Michael C.**

Docket No.: **70314/00568**

Mail Stop Amendment
U.S. Patent & Trademark Office
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

SECOND SUPPLEMENTAL
INFORMATION DISCLOSURE STATEMENT

Pursuant to the duty to disclose under 37 CFR §1.56, Applicant submits herewith a Form PTO/SB/08 listing references of which the Applicant is aware and which are brought to the attention of the Examiner.

Pursuant to 35 USC §120, this application relies on the earlier filing date(s) of the following prior application(s):

Serial Number	Filing Date
10/944,925	September 20, 2004

The filing of this IDS shall not be construed as a representation that a search has been made, an admission that the information cited is, or is considered to be, material for patentability, or that no other material information exists. This filing shall not be construed as an admission against interest in any matter.

This IDS is being submitted pursuant to 37 CFR 1.97(c) prior to the issuance of a final action or a Notice of Allowance. **Applicant hereby certifies that each item of information contained in the present Information Disclosure Statement was cited in a communication from a foreign Patent Office in a counterpart foreign application not more than 3 months prior to the filing of**

the present Statement. Accordingly, no fee is believed to be due for consideration of the documents submitted herewith.

Applicant respectfully requests consideration of the items listed and requests the Examiner to return a copy of the attached Form PTO/SB/08 after being marked as being considered by the Examiner.

Respectfully submitted,

Date: Feb. 28/12

Brett J. Slaney
Registration No. **58,772**
Agent for Applicant

BLAKE, CASSELS & GRAYDON LLP
199 Bay Street
Suite 2800, Commerce Court West
Toronto, Ontario, M5L 1A9
Canada

Tel 416-863-2518
Fax 416-863-2653

BSL/jm

(✓) encl.

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 03/06/2012

MBYARS SALE #00000002 Mailroom Dt: 02/28/2012 022553 13111734
01 FC : 1806 180.00 DA

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 07/03/2012

CSIAS1 ADJ #00000001 Mailroom Dt: 02/28/2012
Seq No: 2 Sales Acctg Dt: 03/06/2012 022553 13111734
01 FC : 1806 180.00 CR

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.
13/111,734	05/19/2011	Gerhard D. Klassen	70314/00568	6081

91704 7590 01/19/2012

Blake, Cassels & Graydon LLP
199 BAY STREET , SUITE 4000
COMMERCE COURT WEST
TORONTO, ON M5L 1A9
CANADA

EXAMINER	
LAI, MICHAEL C	
ART UNIT	PAPER NUMBER
2457	

NOTIFICATION DATE	DELIVERY MODE
01/19/2012	ELECTRONIC

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.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

rimpatent@blakes.com
brett.slaney@blakes.com
portfolioprosecution@rim.com

Office Action Summary	Application No.	Applicant(s)	
	13/111,734	KLASSEN ET AL.	
	Examiner	Art Unit	
	MICHAEL C. LAI	2457	

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

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10/24/2011*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-24* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-24* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *9/9/2011*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This office action is responsive to communication filed on 10/24/2011.

Response to Amendment

2. The examiner has acknowledged the amended specification and the amended claims 17-24. The objection to the specification has been corrected and withdrawn accordingly. The 101 rejection to claims 17-24 has been corrected and withdrawn accordingly. Claims 1-24 are pending.

Response to Arguments

3. Applicant's arguments, see pages 8-9, filed 10/24/2011, with respect to the rejection(s) of claim(s) 1-5, 9-13, and 17-21 under 102(e) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of Maurille **(**US 6,484,196 B1**)**.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-5, 9-13, and 17-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Appelman et al. (US 7,181,497 B1, hereinafter Appelman), in view of Maurille **(**US 6,484,196 B1, hereinafter Maurille).

Regarding claim 1, Appelman discloses a method of displaying an instant message conversation on an electronic device [see at least Fig. 7, User 1; Fig. 16, the user], the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device [see at least Fig. 7, User 2; Fig. 16, mroe1934], the method comprising:

receiving a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48];

transmitting a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34];

associating each instant message with a corresponding time stamp [see at least Figs. 16, 18; col. 9, lines 23-48];

displaying the incoming textual portion of at least one of the incoming instant messages along with a respective time stamp, the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48]; and

displaying the outgoing textual portion of at least one of the outgoing instant messages along with a respective time stamp, the outgoing textual portion of each displayed outgoing instant message being horizontally aligned

at a same second horizontal position [see at least Figs. 16, 18, "T>" (to) messages; col. 9, lines 23-34].

Appelman discloses the claimed invention except for the second horizontal position being different from the first horizontal position. However, Maurille discloses "Each incoming e-mail message is treated by the server application 114 as a level one message that begins a respective e-mail thread. Replies to the incoming e-mail messages are indented on the mail screen 900, visually indicating their position as second level messages in their associated thread" [col. 20, lines 9-15]. It would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Maurille's teaching into Appelman's method for the purpose of visually differentiating between incoming and outgoing messages by aligning/indenting incoming and outgoing textual portions at different horizontal positions, thereby providing a more user friendly Instant Messaging system.

Regarding claim 2, Appelman further discloses displaying an incoming symbol with each displayed incoming instant message [see at least Figs. 16, 18, "F>" (from) is the incoming symbol; col. 9, lines 35-48].

Regarding claim 3, Appelman further discloses displaying an outgoing symbol with each displayed outgoing instant message [see at least Figs. 16, 18, "T>" (to) is the outgoing symbol; col. 9, lines 23-34].

Regarding claim 4, Appelman further discloses wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message [see at least Figs. 16, 18, “13:20:27”; col. 9, lines 35-48].

Regarding claim 5, Appelman further discloses wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message [see at least Figs. 16, 18, “13:20:05”; col. 9, lines 23-34].

Regarding claim 9, Appelman discloses an electronic device for displaying an instant message conversation, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device [see at least Fig. 7, User 1; Fig. 16, the user] and a second electronic device [see at least Fig. 7, User 2; Fig. 16, mroe1934], the electronic device comprising:

a display [Fig. 1, 107, and Fig. 6];

a memory [Fig. 1, 109]; and

a processor [Fig. 1, 121] electronically coupled with the display and the memory, the processor configured to:

receive a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48];

transmit a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34];

associate each instant message with a corresponding time stamp [see at least Figs. 16, 18; col. 9, lines 23-48];

display the incoming textual portion of at least one of the incoming instant messages along with a respective time stamp, the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48]; and

display the outgoing textual portion of at least one of the outgoing instant messages along with a respective time stamp, the outgoing textual portion of each displayed outgoing instant message being horizontally aligned at a same second horizontal position [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34].

Appelman discloses the claimed invention except for the second horizontal position being different from the first horizontal position. However, Maurille discloses “Each incoming e-mail message is treated by the server application 114 as a level one message that begins a respective e-mail thread. Replies to the incoming e-mail messages are indented on the mail screen 900, visually indicating their position as second level messages in their associated thread” [col. 20, lines 9-15]. It would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Maurille’s teaching into Appelman’s device for the purpose of visually differentiating between incoming

and outgoing messages by aligning/indenting incoming and outgoing textual portions at different horizontal positions, thereby providing a more user friendly Instant Messaging system.

Regarding claim 10, Appelman further discloses wherein the processor is further configured to display an incoming symbol with each displayed incoming instant message [see at least Figs. 16, 18, “F>” (from) is the incoming symbol; col. 9, lines 35-48].

Regarding claim 11, Appelman further discloses wherein the processor is further configured to display an outgoing symbol with each displayed outgoing instant message [see at least Figs. 16, 18, “T>” (to) is the outgoing symbol; col. 9, lines 23-34].

Regarding claim 12, Appelman further discloses wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message [see at least Figs. 16, 18, “13:20:27”; col. 9, lines 35-48].

Regarding claim 13, Appelman further discloses wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message [see at least Figs. 16, 18, “13:20:05”; col. 9, lines 23-34].

Regarding claim 17, Appelman discloses a non-transitory computer readable medium comprising computer executable instructions embedded thereon for execution by a processor of an electronic device for displaying an instant message conversation upon a display of the electronic device, the instant

message conversation comprising a plurality of instant messages exchanged between the electronic device [see at least Fig. 7, User 1; Fig. 16, the user] and a second electronic device [see at least Fig. 7, User 2; Fig. 16, mroe1934], such that when executed, the processor:

receives a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48];

transmits a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34];

associates each instant message with a corresponding time stamp [see at least Figs. 16, 18; col. 9, lines 23-48];

displays the incoming textual portion of at least one of the incoming instant messages along with a respective time stamp, the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position [see at least Figs. 16, 18, “F>” (from) messages; col. 9, lines 35-48]; and

displays the outgoing textual portion of at least one of the outgoing instant messages along with a respective time stamp, the outgoing textual portion of each displayed outgoing instant message being horizontally aligned at a same second horizontal position [see at least Figs. 16, 18, “T>” (to) messages; col. 9, lines 23-34].

Appelman discloses the claimed invention except for the second horizontal position being different from the first horizontal position. However, Maurille discloses "Each incoming e-mail message is treated by the server application 114 as a level one message that begins a respective e-mail thread. Replies to the incoming e-mail messages are indented on the mail screen 900, visually indicating their position as second level messages in their associated thread" [col. 20, lines 9-15]. It would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Maurille's teaching into Appelman's system for the purpose of visually differentiating between incoming and outgoing messages by aligning/indenting incoming and outgoing textual portions at different horizontal positions, thereby providing a more user friendly Instant Messaging system.

Regarding claim 18, Appelman further discloses computer instructions such that when executed cause the processor to display an incoming symbol with each displayed incoming instant message [see at least Figs. 16, 18, "F>" (from) is the incoming symbol; col. 9, lines 35-48].

Regarding claim 19, Appelman further discloses computer instructions such that when executed cause the processor to display an outgoing symbol with each displayed outgoing instant message [see at least Figs. 16, 18, "T>" (to) is the outgoing symbol; col. 9, lines 23-34].

Regarding claim 20, Appelman further discloses wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message [see at least Figs. 16, 18, “13:20:27”; col. 9, lines 35-48].

Regarding claim 21, Appelman further discloses wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message [see at least Figs. 16, 18, “13:20:05”; col. 9, lines 23-34].

6. Claims 6-8, 14-16, and 22-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Appelman et al. (US 7,181,497 B1, hereinafter Appelman) and Maurille **(**US 6,484,196 B1, hereinafter Maurille) as applied to claim 1, and further in view of Yamada (US 6,889,063 B2, hereinafter Yamada).

Regarding claim 6, Appelman and Maurille disclose the method of claim 1 but are silent about detecting an interruption in the instant message conversation; and refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time. However, Appelman teaches displaying time stamp on messages [see at least claim 1] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call

[see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's and Maurille's method in order to alert the user only when the timer expires before next message by refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time [see the abstract].

Regarding claim 7, Appelman and Maurille disclose the method of claim 1 but are silent about refraining from displaying at least one of the corresponding time stamps. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to

incorporate Yamada's teaching into Appelman's and Maurille's method in order to alert the user only when the timer expires before next message by refraining from displaying at least one of the corresponding time stamps [see the abstract].

Regarding claim 8, Appelman, Maurille, and Yamada disclose the method of claim 7, including wherein the refraining is performed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time [Yamada, see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. See claim 6 for motivation.

Regarding claim 14, Appelman and Maurille disclose the electronic device of claim 9 but is silent about wherein the processor is further configured to: detect an interruption in the instant message conversation; and refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to

the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's and Maurille's electronic device in order to alert the user only when the timer expires before next message by refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time [see the abstract].

Regarding claim 15, Appelman and Maurille disclose the electronic device of claim 9 but are silent about wherein the processor is further configured to refrain from displaying at least one of the corresponding time stamps. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in

the art at the time the invention was made to incorporate Yamada's teaching into Appelman's and Maurille's electronic device in order to alert the user only when the timer expires before next message by refraining from displaying at least one of the corresponding time stamps [see the abstract].

Regarding claim 16, Appelman, Maurille, and Yamada disclose the electronic device of claim 15, including wherein the processor refrains from displaying the at least one of the corresponding time stamps if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time [Yamada, see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. See claim 14 for motivation.

Regarding claim 22, Appelman and Maurille disclose the computer readable medium of claim 17 but is silent about computer instructions such that when executed cause the processor to: detect an interruption in the instant message conversation; and refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a

timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert the user to the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's and Maurille's computer readable medium in order to alert the user only when the timer expires before next message by refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time [see the abstract].

Regarding claim 23, Appelman and Maurille disclose the computer readable medium of claim 17 but is silent about comprising computer instructions such that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps. However, Appelman teaches displaying time stamp on messages [see at least Fig. 12, item 624] and Yamada teaches displaying an in-absence incoming call message on a display if the user of the cellular phone does not answer an incoming call [read as detecting an interruption, see the abstract]. Yamada further discloses starting a timer with a predetermined period of time at the same time the in-absence incoming call message is displayed. When the timer does not expire, no alert tone/vibration/light is outputted to alert

the user to the in-absence incoming call [see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. Thus it would have been obvious to a person with ordinary skill in the art at the time the invention was made to incorporate Yamada's teaching into Appelman's and Maurille's computer readable medium in order to alert the user only when the timer expires before next message by refraining from displaying at least one of the corresponding time stamps [see the abstract].

Regarding claim 24, Appelman, Maurille, and Yamada disclose the computer readable medium of claim 23, including wherein the computer instructions that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps are executed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a 5 predetermined duration of time [Yamada, see Figure 2 and col. 2 line 7 through col. 3 line 22, only when the timer expires, an alert tone/vibration/light is outputted to alert the user to the in-absence incoming call]. See claim 22 for motivation.

Conclusion

Examiner's Note: Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and

figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. In the case of amending the claimed invention, Applicant is respectfully requested to indicate the portion(s) of the specification which dictate(s) the structure relied on for proper interpretation and also to verify and ascertain the metes and bounds of the claimed invention.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MICHAEL C. LAI whose telephone number is (571)270-3236. The examiner can normally be reached on M-F 9:00 - 5:30 EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571) 272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-

free). If you would like assistance from a USPTO Customer Service

Representative or access to the automated information system, call 800-786-

9199 (IN USA OR CANADA) or 571-272-1000.

Michael C. Lai
Art Unit 2457
Phone: (571) 270-3236
Fax: (571) 270-4236

/YVES DALENCOURT/
Primary Examiner, Art Unit 2457

Notice of References Cited	Application/Control No. 13/111,734	Applicant(s)/Patent Under Reexamination KLASSEN ET AL.	
	Examiner MICHAEL C. LAI	Art Unit 2457	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-5,896,567 A	04-1999	Ogushi, Masuo	455/421
*	B	US-2004/0205775 A1	10-2004	Heikes et al.	719/318
*	C	US-6,889,063 B2	05-2005	Yamada, Hironori	455/567
*	D	US-6,940,407 B2	09-2005	Miranda-Knapp et al.	340/572.1
*	E	US-7,181,497 B1	02-2007	Appelman et al.	709/206
*	F	US-6,484,196	11-2002	Maurille, Mithras C.	709/206
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	11346	709/206.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 11:57
S2	2237	709/207.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 11:58
S3	1028	S1 S2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 11:58
S4	13	"7181497"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 12:00

S5	137	"6539421"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 12:17
S6	4	"5896567"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:02
S7	492201	(without or no or interrupt$4 or discontinu$4 or disconnect$4) with (communication or conversation or messag$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:55
S8	300348	time near2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:56
S9	1409	S7 with S8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:56

S10	27935	(display$4 or output $4) near4 time near2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S11	56	S9 same S10	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S12	11346	709/206.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S13	2237	709/207.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S14	12555	S12 or S13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58

S15	3	S11 S14	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 15:58
S16	25	S11 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 16:15
S17	26	"7007085"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 16:36
S18	3	"6889063"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 18:40
S19	125	(without or refrain$4 or stop$4) adj2 (display$4 or output $4) near4 time near2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:27

S20	492201	(without or no or interrupt$4 or discontinu$4 or disconnect$4) with (communication or conversation or messag$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:27
S21	12	S19 S20	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:27
S22	19	(without or refrain$4 or stop$4) adj2 (display$4 or output $4) adj4 time adj2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:36
S23	19	S22 not S21	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:37
S24	13	S23 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:37

S25	66	S19 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:49
S26	53	S25 not S24	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/26 22:49
S27	13	"6940407"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/27 10:36
S28	3	"20040205775"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/27 10:58
S29	4	"5896567"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/27 11:00

S30	262224	(without or no or interrupt$4 or discontinu$4 or disconnect$4) near4 (communication or conversation or messag$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:49
S31	11361	709/206.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50
S32	2238	709/207.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50
S33	12571	S31 or S32	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50
S34	5972	S30 S33	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50

S35	2017	S34 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:50
S36	39382	(interrupt$4) near4 (communication or conversation or messag$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:51
S37	12571	S33 S33	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:51
S38	441	S33 S36	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:52
S39	157	S38 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:53

S40	27960	(display$4 or output $4) near4 time near2 (stamp or indicat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:53
S41	2	S39 S40	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:53
S42	443	(interrupt$4) near4 (IM or instant adj1 messag$4 or chat$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:56
S43	443435	timer or time$1stamp $4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:57
S44	15	S42 same S43	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:57

S45	11	S44 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 14:57
S46	94	S33 S42	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:00
S47	6999	S30 same S43	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:02
S48	142	S47 S33	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:02
S49	47	S48 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:02

S50	7125	(without or no or interrupt$4 or discontinu$4 or disconnect$4) near4 (IM or chat$4 or instant adj1 messag $4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:19
S51	68	S50 same S43	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:20
S52	12	S51 S33	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:20
S53	1	S52 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 15:21

S54	105	(without or no or interrupt$4 or discontinu$4 or disconnect$4) near4 (IM or chat$4 or instant adj1 messag $4) near4 conversation	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 16:15
S55	18	S54 @ad < "20030919"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2011/07/28 16:16
S56	711	display$4 with incoming with outgoing with message $1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:12
S57	0	align$4 with different $2 with horizontaly with position	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:14

S58	11135	align$4 with different $2 with position	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:15
S59	16761	align$4 with different $2 with position$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:15
S60	1	S56 S59	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:15
S61	474793	align$4 with position $1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:17
S62	1	S56 S60	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:17

S63	5936	display$4 with receiving with (send$4 or transmit$4) with message$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:18
S64	0	S63 S60	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:19
S65	3	S56 S61	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:19
S66	44	S63 S61	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:21
S67	44	S66 not S65	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:21

S68	16761	align$4 with different $2 with position$1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:28
S69	7202	display$4 with receiving with (send$4 or transmit$4) with (text$4 or message$1)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:29
S70	4	S68 S69	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:29
S71	0	display$4 with to with (from) with (text$4 or message$1)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/03 17:37
S72	9	indent$6 with incoming with message $1	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 14:19

S73	0	WO 02/21413	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:13
S74	0	WO 2002/21413	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:14
S75	1	WO "200221413"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:14
S76	2	WO "2002021413"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:14
S77	118	"7130885"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	AND	OFF	2012/01/04 18:16

EAST Search History (Interference)

<This search history is empty>

1/ 4/ 2012 9:31:32 PM
C:\ Documents and Settings\ mlai\ My Documents\ EAST\ Workspaces\ 13111734.wsp

PTO/SB/08A (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	13/111,734
Filing Date	May 19, 2011
First named Inventor	KLASSEN, Gerhard D.
Art Unit	2457
Examiner Name	LAI, Michael C.
Attorney Docket Number	70314/00568

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]	Number[4]	Kind-Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
/M.L./		WO	02/21413	A2	03-14-2002	ZAPLET, INC.		

Examiner Signature	/Michael Lai/	Date Considered	01/04/2012

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] **Applicants' unique citation designation number** (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13111734	KLASSEN ET AL.
	Examiner	Art Unit
	MICHAEL C LAI	2457

SEARCHED			
Class	Subclass	Date	Examiner
Inventor search		7/25/11	Lai
709	206, 207	7/26/11	Lai

SEARCH NOTES		
Search Notes	Date	Examiner
EAST text search	7/26/11	Lai
EAST	1/3/12	Lai
Consulted Joni Hsu for display search	1/4/12	Lai

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	13111734	KLASSEN ET AL.
	Examiner	Art Unit
	MICHAEL C LAI	2457

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	07/28/2011	01/04/2012							
	1	✓	✓							
	2	✓	✓							
	3	✓	✓							
	4	✓	✓							
	5	✓	✓							
	6	✓	✓							
	7	✓	✓							
	8	✓	✓							
	9	✓	✓							
	10	✓	✓							
	11	✓	✓							
	12	✓	✓							
	13	✓	✓							
	14	✓	✓							
	15	✓	✓							
	16	✓	✓							
	17	✓	✓							
	18	✓	✓							
	19	✓	✓							
	20	✓	✓							
	21	✓	✓							
	22	✓	✓							
	23	✓	✓							
	24	✓	✓							

IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Appl. No.:	**13/111,734**
Applicant:	**KLASSEN, Gerhard D. et al.**
Filed:	**May 19, 2011**
Title:	**Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment**
Art Unit:	**2457**
Examiner:	**LAI, Michael C.**
Docket No.:	**70314/00568**

Mail Stop Amendment
U.S. Patent & Trademark Office
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSE

Sir:

This is further to the Office Action dated August 9, 2011. Applicant wishes to amend the above-identified application as follows:

Amendments to the Specification: begin on page 2 of this paper.

Amendments to the Claims: are reflected in the listing of claims which begins on page 3 of this paper.

Remarks: begin on page 8 of this paper.

Amendments to the Specification

Please replace paragraph [0001] of the application as filed with the following replacement paragraph:

[0001] This application is a continuation of U.S. Patent Application No. 10/944,925 filed on September 20, 2004 (now U.S. Patent No. 7,970,849), which claims the benefit of U.S. Provisional Application No. 60/504,379 entitled filed on Sep. 19, 2003, both of which are hereby incorporated into the present application by reference.

Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of claims:

1. (original) A method of displaying an instant message conversation on an electronic device, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, the method comprising:

receiving a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmitting a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associating each instant message with a corresponding time stamp;

displaying the incoming textual portion of at least one of the incoming instant messages along with a respective time stamp, the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position; and

displaying the outgoing textual portion of at least one of the outgoing instant messages along with a respective time stamp, the outgoing textual portion of each displayed outgoing instant message being horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position.

2. (original) The method of claim 1 further comprising displaying an incoming symbol with each displayed incoming instant message.

3. (original) The method of claim 1 further comprising displaying an outgoing symbol with each displayed outgoing instant message.

4. (original) The method of claim 1, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

5. (original) The method of claim 1, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

6. (original) The method of claim 1, further comprising:

detecting an interruption in the instant message conversation; and

refraining from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

7. (original) The method of claim 1, further comprising refraining from displaying at least one of the corresponding time stamps.

8. (original) The method of claim 7, wherein the refraining is performed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

9. (original) An electronic device for displaying an instant message conversation, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, the electronic device comprising:

a display;

a memory; and

a processor electronically coupled with the display and the memory, the processor configured to:

receive a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmit a plurality of outgoing instant messages to the second electronic device, each outgoing instant message having an outgoing textual portion;

associate each instant message with a corresponding time stamp;

displaying the incoming textual portion of at least one of the incoming instant messages along with a respective time stamp, the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position; and

display the outgoing textual portion of at least one of the outgoing instant messages along with a respective time stamp, the outgoing textual portion of each displayed outgoing instant message being horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position.

10. (original) The electronic device of claim 9, wherein the processor is further configured

to display an incoming symbol with each displayed incoming instant message.

11. (original) The electronic device of claim 9, wherein the processor is further configured to display an outgoing symbol with each displayed outgoing instant message.

12. (original) The electronic device of claim 9, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

13. (original) The electronic device of claim 9, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

14. (original) The electronic device of claim 9, wherein the processor is further configured to:

detect an interruption in the instant message conversation; and

refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

15. (original) The electronic device of claim 9, wherein the processor is further configured to refrain from displaying at least one of the corresponding time stamps.

16. (original) The electronic device of claim 15, wherein the processor refrains from displaying the at least one of the corresponding time stamps if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

17. (currently amended) A non-transitory computer readable medium comprising computer executable instructions embedded thereon for execution by a processor of an electronic device for displaying an instant message conversation upon a display of the electronic device, the instant message conversation comprising a plurality of instant messages exchanged between the electronic device and a second electronic device, such that when executed, the processor:

receives a plurality of incoming instant messages from the second electronic device, each incoming instant message having an incoming textual portion;

transmits a plurality of outgoing instant messages to the second electronic device, each

outgoing instant message having an outgoing textual portion;

associates each instant message with a corresponding time stamp;

displays the incoming textual portion of at least one of the incoming instant messages along with a respective time stamp, the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position; and

displays the outgoing textual portion of at least one of the outgoing instant messages along with a respective time stamp, the outgoing textual portion of each displayed outgoing instant message being horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position.

18. (currently amended) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to display an incoming symbol with each displayed incoming instant message.

19. (currently amended) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to display an outgoing symbol with each displayed outgoing instant message.

20. (currently amended) The non-transitory computer readable medium of claim 17, wherein at least one first time stamp is displayed adjacent to its corresponding incoming instant message.

21. (currently amended) The non-transitory computer readable medium of claim 17, wherein at least one second time stamp is displayed adjacent to its corresponding outgoing instant message.

22. (currently amended) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to:

detect an interruption in the instant message conversation; and

refrain from displaying a time stamp associated with a next one of an incoming instant message and an outgoing instant message if the interruption is less than a predetermined duration of time.

23. (currently amended) The non-transitory computer readable medium of claim 17, further comprising computer instructions such that when executed cause the processor to refrain

from displaying at least one of the corresponding time stamps.

24. (currently amended) The non-transitory computer readable medium of claim 23, wherein the computer instructions that when executed cause the processor to refrain from displaying at least one of the corresponding time stamps are executed if an amount of time that has lapsed between the at least one corresponding time stamp and a previous corresponding time stamp is less than a predetermined duration of time.

REMARKS

Applicant thanks the Examiner for reviewing the present application.

Amendments to the Specification

Paragraph [0001] has been amended to identify that U.S. Patent Application No. 10/944,925 has now issued as U.S. Patent No. 7,970,849.

Claim Amendments

Claims 17-24 have been amended inserting "non-transitory" in the preamble.

Applicant respectfully submits that no new subject matter has been added by way of these amendments.

Specification

As suggested by the Examiner, paragraph [0001] has been amended to identify that U.S. Patent Application No. 10/944,925 has now issued as U.S. Patent No. 7,970,849.

Claim Rejections – 35 U.S.C. 101

Claims 17-24 have been rejected under 35 U.S.C. 101 for being directed to non-statutory subject matter. As noted above, claims 17-24 have been amended inserting "non-transitory" in the preamble as suggested by the Examiner and in accordance with USPTO Office Notice 1351 OG 212. Applicant respectfully submits that claims 17-24 thus amended comply with 35 U.S.C. 101.

Claim Rejections – 35 U.S.C. 102

Claims 1-5, 9-13, and 17-21 have been rejected under 35 U.S.C. 102(e) as being anticipated by Appelman (U.S. Patent No. 7,181,497). Applicant respectfully traverses the rejections as follows.

Claim 1 recites in part:

> "...the incoming textual portion of each displayed incoming instant message being horizontally aligned at a same first horizontal position;

and

"...the outgoing textual portion of each displayed outgoing instant message being horizontally aligned at a same second horizontal position, the second horizontal position being different from the first horizontal position".

Applicant respectfully submits that Appelman does not teach or suggest displaying incoming and outgoing textual portions aligned at first and second horizontal positions wherein the first and second horizontal positions are different from each other, and thus Appelman does not anticipate claim 1.

Appelman may show, among other things, a conversation screen (see FIG. 18) that includes multiple outgoing (T) and multiple incoming (F) messages. However, in Appelman, the incoming and outgoing messages are all aligned at the same exact horizontal position. Applicant respectfully submits that this is the opposite of what is recited in claim 1. In fact, as clearly shown in FIG. 18, even the indentations used to separate the message information from the message itself are the same for both incoming and outgoing messages. Moreover, as shown in FIG. 20, even if the messages are offset due to differently sized user names (e.g., more1934 versus mjohnson1934), both the incoming and outgoing messages include the same offset and are therefore intentionally aligned at the same horizontal location. In other words, based on what is shown in Appelman, no two outgoing messages would be aligned at the same horizontal position which is different from two incoming message aligned at another horizontal position. Furthermore, and in any event each line of the conversation, shown in each and every screen illustrated by Appelman, aligns the text left-justified (i.e., the T and F which begin each line are always justified to the left edge of the screen). There is no teaching or suggestion of aligning incoming and outgoing textual portions at different horizontal alignments.

For at least the above reasons, Appelman does not teach each and every element of claim 1 and thus cannot anticipate. Similar arguments apply to claims 9, 17, and claims 2-5, 10-13, and 18-21 dependent on claims 1, 9, or 17.

Claim Rejections – 35 U.S.C. 103

Claims 6-8, 14-16, and 22-24 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Appelman in view of Yamada (U.S. Patent No. 6,889,063). Applicant respectfully traverses the rejections as follows.

Although Yamada may teach providing an alert after a predetermined amount of time has

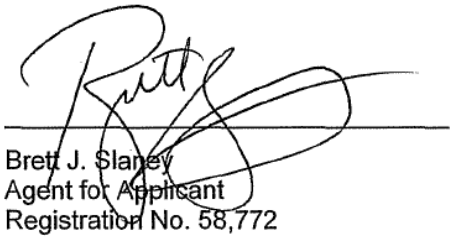
passed since an in-absence incoming call message has been displayed, Yamada fails to teach or suggest displaying incoming and outgoing textual portions aligned at first and second horizontal positions wherein the first and second horizontal positions are different from each other and thus cannot anticipate, and thus does not teach what is missing from Appelman. For at least this reason, Applicant respectfully submits that claims 6-8, 14-16, and 22-24 are patentable over Appelman in view of Yamada.

Summary

In view of the foregoing, Applicant respectfully submits that the present application is in condition for allowance.

Applicant therefore requests early reconsideration and allowance of the present application.

Respectfully submitted,



Brett J. Slaney
Agent for Applicant
Registration No. 58,772

Date: October 24, 2011

BLAKE, CASSELS & GRAYDON LLP
199 Bay Street
Suite 4000, Commerce Court West
Toronto ON M5L 1A9
Canada

Tel: 416-863-2518
BS/

Electronic Acknowledgement Receipt

EFS ID:	11252887
Application Number:	13111734
International Application Number:	
Confirmation Number:	6081
Title of Invention:	Handheld Electronic Device and Associated Method Providing Time Data in a Messaging Environment
First Named Inventor/Applicant Name:	Gerhard D. Klassen
Customer Number:	91704
Filer:	Brett Joseph Slaney/Judith Martin
Filer Authorized By:	Brett Joseph Slaney
Attorney Docket Number:	70314/00568
Receipt Date:	24-OCT-2011
Filing Date:	19-MAY-2011
Time Stamp:	17:02:55
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		11144-US-CNT3_OA-Response.pdf	853781 5b28b7611cebde0a51ce2b3dc137f54d32ac19cc	yes	10

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Specification	2	2
	Claims	3	7
	Applicant Arguments/Remarks Made in an Amendment	8	10
Warnings:			
Information:			
	Total Files Size (in bytes):	853781	

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
13/111,734	05/19/2011	

APPLICATION AS FILED – PART I

	(Column 1)	(Column 2)	SMALL ENTITY ☐		OR	OTHER THAN SMALL ENTITY	
FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)		RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

AMENDMENT	10/24/2011	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 24	Minus	** 24	= 0	X $ =		OR	X $60=	0
	Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $ =		OR	X $250=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/VERONICA DAY EVERETT/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08A (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Substitute for form 1449/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet 1 of 1

Complete if Known	
Application Number	13/111,734
Filing Date	May 19, 2011
First named Inventor	KLASSEN, Gerhard D.
Art Unit	2457
Examiner Name	LAI, Michael C.
Attorney Docket Number	70314/00568

U.S. PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Document Number Number-Kind Code[2] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document Country Code[3]	Number[4]	Kind-Code[5] (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear	T[6]
		WO	02/21413	A2	03-14-2002	ZAPLET, INC.		

Examiner Signature		Date Considered	

EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant. [1] **Applicants' unique citation designation number** (optional). [2] See Kinds Codes of USPTO Patent Documents at www.uspto gov or MPEP 901.04. [3] Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). [4] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [5] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. [6] Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 (1-800-786-9199) and select option 2.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 March 2002 (14.03.2002)

PCT

(10) International Publication Number
WO 02/21413 A2

(51) International Patent Classification[7]: G06F 17/60

(21) International Application Number: PCT/US01/42041

(22) International Filing Date:
5 September 2001 (05.09.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/230,037 5 September 2000 (05.09.2000) US

(71) Applicant: **ZAPLET, INC.** [US/US]; 3000 Bridge Parkway, Redwood Shores, CA 94065 (US).

(72) Inventors: **AXE, Brian**; 342 Liberty Street, San Francisco, CA 94114 (US). **EVANS, Steven**; 13300 Lennox Way, Los Altos Hills, CA 94022 (US). **HANSON, Michael**; 973 Oak Lane, Menlo Park, CA 94025 (US). **HOM, Wayne**; 1141 Wayne Way, San Mateo, CA 94403 (US). **HYDE, Timothy**; 670 Shotwell Street, San Francisco, CA 94110 (US). **KRUMPELSTAEDTER, John**; 10893 Sweet Oak Street, Cupertino, CA 95014 (US).

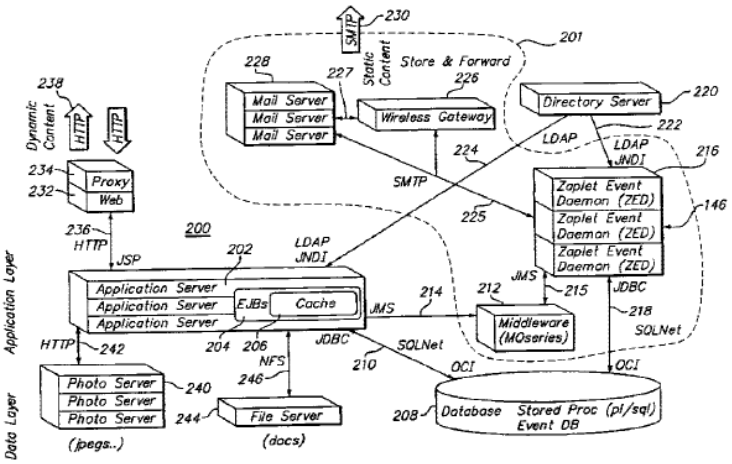
(74) Agents: **PALERMO, Christopher** et al.; Hickman Palermo Truong & Becker, LLP, 1600 Willow Street, San Jose, CA 95125 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: METHODS AND APPARATUS PROVIDING ELECTRONIC MESSAGES THAT ARE LINKED AND AGGREGATED



(57) Abstract: A method for associating related electronic messages in computer storage. A first transportable application is created and stored. User input requesting creation of a link from the first transportable application to another transportable application is received. User input that selects a second transportable application from among a plurality of previously created transportable applications is received. A link from the first transportable application to the second transportable application is created and stored. As a result, transportable application may be inter-related in complex message webs. The message webs may also be inter-related in message web rings. Links may be generated manually or automatically, based on context, workflow processes, or other known relationships among applications. Recipient lists and data may propagate among fields of linked transportable applications, directly or according to abstract business rules. Further, multiple-part electronic messages are disclosed. A graphical user interface of an electronic messaging system displays a message in the form of one or more header portions and one or more body portions. Each of the body portions has a selection region. While one body portion is visible at a given time, all the selection regions are continuously visible in the user interface to facilitate selection of any of the body portions at a particular time. Selecting a selection region of a non-displayed body portion causes a server to generate a refreshed user interface that includes the entirety of the selected body portion, and that hides the previously viewed body portion. As a result, a large amount of associated information may be combined in a single message in a way that is clearly organized and easily accessible.

WO 02/21413 A2

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report*

— *entirely in electronic form (except for this front page) and available upon request from the International Bureau*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

METHODS AND APPARATUS PROVIDING ELECTRONIC MESSAGES THAT ARE LINKED AND AGGREGATED

CROSS-REFERENCE TO RELATED APPLICATIONS

Domestic priority is claimed under 35 U.S.C. 119(e) from prior Provisional application Ser. No. 60/230,037, filed September 05, 2000, the entire contents of which are hereby incorporated by reference as if fully set forth herein.

FIELD OF INVENTION

The present invention generally relates to data processing in the fields of electronic messaging and collaborative communications. The invention relates more specifically to a collaborative communications system providing transportable applications, multiple-page electronic messages and electronic messages that are linked and aggregated.

BACKGROUND OF THE INVENTION

Despite more than fifty years of development, computer systems and computer programs remain difficult to use. In particular, individuals who are not well-versed in technology or computer programming encounter numerous problems in carrying out simple data processing tasks, such as collaborating on the form or content of a document, polling others to arrive at a decision, scheduling a meeting or activity, and other tasks. Several main approaches have been used in the past to facilitate these activities.

In a first approach, a pre-configured computer program or application is provided to all individuals who will participate in the data processing tasks. The individuals interact with the application to carry out the tasks. This approach requires little technical knowledge, but the approach is often too rigid and limits the individuals to using the particular forms, features and functions selected by the developers of the application. A related approach involves use of electronic mail ("e-mail") for distribution of information or for collaborative communications. E-mail is inefficient and often requires sending multiple messages before a task can be completed. Further, since the content of the messages is static, the content tends to become old and stale. The number and content of the messages are not easily organized.

In another prior approach, users construct their own custom computer program to perform exactly the desired task. This approach requires extensive technical knowledge that is beyond the capability of a typical user.

In a third approach, certain collaborative computing applications and environments have been provided. Applications such as Microsoft NetMeeting enable users to participate in online meetings, but have a significant limitation in that they are synchronous; a meeting participant must be online and present to participate. Applications such as Lotus Notes have enabled users to share documents, but creation and deployment of applications to carry out more sophisticated business tasks, such as automating personnel recruiting processes, is complicated and requires programming knowledge. Further, the collaborative processes that are supported by these applications "off the shelf" are limited and do not encompass many multi-step business processes.

A common disadvantage of all these approaches is that it is very hard for individual, non-technical users to create computer-based applications or systems that automate specific desired tasks and provide the applications or systems to others. Further, the approaches generally enable users to see only a static view of content that is under discussion or modification. Another disadvantage is that the approaches are typically synchronous; a user must be present during the collaboration and has no way to view actions or communications that have occurred before the user arrived or after the user leaves.

Another common disadvantage is that integrating the past approaches with existing or "legacy" databases and applications is extremely complicated, and requires programming knowledge. There is no easy way to enable such legacy systems to automatically interact with collaborative applications, without human intervention, to exchange data and carry out business processes. There is no simple means to enable the collaborative applications to receive signals from the external applications, take action based on the signals, or initiate new processes based on the signals.

Based on the foregoing, there is a clear need for an improved way to facilitate collaborative communication among individuals. In particular, there is a need for an approach that can enable new kinds of collaborative productivity applications, created by anyone and delivered using familiar communication mechanisms.

There is a related need for an approach that can facilitate distribution of secure, collaborative applications among separate but networked enterprises, so that users or applications of different enterprises can collaborate.

Computers have been used in messaging services since the 1960s, when suppliers of telex and telegraph services began deploying automated switching functions and providing complete computer-based messaging systems. An example of such a system is a teletypewriter system. Many of these systems used computers as data switches that

relayed messages without providing message composition capabilities or the ability to search through previously sent messages.

Development of interactive time-sharing computer systems resulted in creation of linking tools that enabled two users to interlock their terminals in order to see what the other was typing. Mailbox tools were developed to allow users to send short messages to other users on-line on the same system. For example, the interactive time-sharing system at MIT's Project MAC had both linking and mailbox facilities that supported text messages.

Construction of the Advanced Research Projects Agency Network (ARPANET) in 1969-1975 initiated extensive research in store-and-forward communication techniques. An outgrowth of this research was the development of electronic messaging facilities, including electronic mail. Such facilities supported transport of simple text messages.

In the early 1980s, international standards bodies began work on electronic messaging with the goal of ensuring interoperability among different systems. In 1984, CCITT adopted a standard for store-and-forward messaging services called X.400 Message Handling Services (MHS). The International Standards Organization (ISO) created a competing standard called Message Oriented Text Interchange Standard (MOTIS). Differences among the versions were largely overcome by 1988.

The period 1973 to 1989 saw development of several computer systems that featured graphical user interfaces rather than character displays, such as the Xerox Alto, the Apple Lisa and Macintosh, and personal computers using Microsoft Windows. During this period, users developed interest in the ability to send messages having elements other than text. Examples of such multimedia e-mail systems include Slate, from BBN Software Products, and the first release of Lotus Notes, from Lotus Development Corporation.

To support transmission of formatted images, sound, files, folders, and video attachments, in 1992 the Internet Engineering Task Force (IETF) created a multi-media messaging standard known as Multipurpose Internet Mail Extensions (MIME). Using MIME, users of different kinds of computers can exchange e-mail messages that include formatted images, sound, files, folders, and video attachments. Further, a single MIME e-mail can contain alternative representations of the same data. For example, there can be an attachment in text form followed by one containing bitmap page images of the same information.

Implementations of MIME generally displayed messages in a graphical user interface window that presented different attachments as icons embedded within a text

message. Other implementations, such as NeXTMail, from NeXT Computer, Inc., could display text, graphical elements, and attachment icons within the same message. A drawback of this implementation, however, was that to view different parts of a long message, a user was required to scroll down through a single contiguous message display in order to reach the desired part of the message.

Further, such systems provide no way to organize related elements of a message, other than manual cutting of one part of the message and pasting it into another upper or lower part of the same contiguous message. This procedure could quickly become unwieldy when such messages were exchanged among members of a group. As each member of the group added comments, graphics, or other attachments to the message, the message would become longer and longer, requiring extensive scrolling and searching to locate a desired part, or to correlate one comment of one user with a related comment of another user.

Other systems display multimedia attachments in a separate pane of a graphical user interface window that is displaying the message. For example, in Microsoft Outlook, when multimedia attachments are present, they are displayed at the bottom of the message window, divided from the message body by a separator bar. A disadvantage of this approach is that it is hard to correlate the attachment with a particular part of the message body.

In a more recent development, e-mail programs such as Microsoft Outlook have been linked to browser programs such as Microsoft Internet Explorer. Through such linkage, the e-mail program can receive an e-mail message formatted in Hypertext Markup Language (HTML) and display a rendered version of the HTML within an e-mail message display window. Using this approach, e-mail messages may contain text, graphics, and embedded hyperlinks to other content, multimedia resources, or applications. However, this approach has not cured the deficiency of prior approaches with respect to content organization and scrolling. It is still necessary to scroll through a long HTML e-mail to find desired content, and there is no simple way to correlate related content. Further, authoring HTML e-mail requires familiarity with complex HTML authoring software such as Microsoft FrontPage.

Based on the foregoing, there is a clear need in this field for a way to improve organization of content of e-mail messages.

There is a specific need for a way to organize related content portions of e-mail messages within the messages in a way that makes related content easy to find and display.

There is also a need for a way to create, deliver, receive, and display e-mail messages in which related content is correlated and accessed without scrolling or similar cumbersome operations.

E-mail, the World Wide Web, and instant messaging each have contributed to a revolution in the way people communicate. Each of these platforms offers a variety of benefits. For example, e-mail is available at the user's convenience; messages remain in an inbox until the user deletes them. Because e-mail arrives in the inbox, the user does not have to "go" anywhere to retrieve it. E-mail is useful for sending a note to a friend or addressing a large group. Because e-mail is individually addressable, a user can adapt the recipient list to the subject. Experts have estimated that users spend more time in e-mail than any other Internet application.

The World Wide Web ("Web") contains an abundance of information that is dynamically changing and ever growing. It is graphical and interactive, and offers the ability to conduct transactions. With relatively simple user interface operations, such as button clicks, the user can receive valuable information, purchase an item, or review work materials that have been placed on an internal website. Instant messaging offers the advantage of timeliness; the recipient knows that a newly received message is current, because the recipient is in a one-on-one communication session with the sender. This form of messaging has grown rapidly because people want information that is current.

However, e-mail, the web, and instant messaging have inherent limitations that restrict what users can do with them and how they communicate information to users. For example, e-mail today consists of static text that is neither dynamic nor interactive. E-mail messages are considered to be "dead on arrival" because they are current when sent, but are not current when read. For example, a user may not read an e-mail until hours after it was originally sent and there can be subsequent e-mails that supersede the original message, but the original outdated message remains in the inbox. There is no logical order to messages that are received in the inbox other than the time of arrival. Occasionally, the user can even receive an answer to a question before the question arrives, because of delays in server responses. The user is required to either scan all messages or sort them into some order to verify that the user has all needed information before taking action.

Unthreaded responses are another problem with e-mail. If a user sends a message to six people, and they all respond, the user receives six new messages in the inbox referring to one subject. The user is left to organize it all. Perhaps the most significant limitation of e-mail is the volume of messages that can flood a mailbox. When a user receives hundreds of e-mail messages a day, the sheer volume of e-mail is daunting.

Using the Web requires the user to proactively visit a location for information updates, and the user may need to visit multiple locations and pages before the user finds desired information. The Web is not personal or targeted, so it is not as effective as e-mail for communications. There are meeting sites on the Web, but users must seek out the destination, which requires them to remember URLs and passwords. There are also sites that offer group discussion and collaboration, but their groups are generally rigid and are based around a common interest.

Instant messaging allows for dynamic communication, but only if both participants are actively signed on and are available to receive messages. If one user is not able to respond, the link is dead and the user has lost the ability to instantly communicate. In addition, it's difficult to document "conversations" that occur through instant messaging; users lose the ability to have an audit trail or history.

Based on the foregoing, there is a need for a communication system that provides ways to share messages and information, ideally by providing the best features of e-mail, the Web, and instant messaging, while overcoming the limitations that each of these carries.

There is a particular need for methods and mechanisms for linking multiple messages together so that multiple message threads and sub-topics may be related and associated with a particular topic.

Users and applications may desire to link one message to another related message. However, past approaches to communication systems do not allow information, such as applications and electronic messages, to be related, threaded, managed and stored in an efficient, flexible, and simple way. Such approaches also do not provide a way to obtain a view of complex webs of messages.

There is also a need for methods and mechanisms that can aggregate related topical information within the context of a single message or web of messages.

SUMMARY OF THE INVENTION

The foregoing needs, and other needs that will become apparent from the following description, are achieved in the present invention, which comprises, in one aspect, a method and system for communicating and collaborating using transportable applications. Embodiments provide a breakthrough communication approach that improves collaboration among groups of individuals, within corporations, among customers, partners, and suppliers, and among distributed computer systems. In one approach, computer processes enable such users to create or host a new kind of

collaborative productivity application called a transportable application. Transportable applications as disclosed herein may be used, for example, to bring together people, systems and information needed to contribute, make decisions, and take action on collaborative business processes or projects. In one embodiment, when a transportable application is opened, it reveals a live, shared, structured workspace that is specific to a project or process. Unlike past approaches, the workspace is continually updated to reflect the latest input of any recipient or user.

The transportable applications can be created by anyone, including individuals, non-technical users within an enterprise, or automatically by other computer programs, applications, or programmatic mechanisms. Each transportable application may comprise one or more pages each having one or more building blocks that encapsulate processing functions and data. Each building block may comprise one or more static content regions, one or more dynamic content regions, and other information. Content for the dynamic content regions may be stored in a database of a server.

The transportable applications can be distributed to others using existing transport mechanisms. For example, in one specific approach, transportable applications can be deployed in a manner analogous to sending an electronic mail message. Recipients may be individual users, groups, or other programs or applications.

The transportable applications can be accessed programmatically, or by an e-mail in-box, or using a web browser. When a transportable application is accessed by a recipient, data for its dynamic content regions is automatically retrieved. Input created by a recipient is accepted by the server and updates the dynamic content region of the transportable application. If a recipient modifies data in the transportable application, the modifications or updates are stored asynchronously and dynamically with respect to the sending and receiving of the modifications or updates by the recipients. Thus, the dynamic content is always current, and whenever any recipient opens and views the transportable application, the content of the transportable application is retrieved and displayed in then-current form.

Unlike past approaches, which rely on distribution of static information, transportable applications as defined herein enable full interaction and navigation of dynamic content without the user leaving the transportable application. A shared workspace is created in which multiple persons or systems can interact within the same transportable application, and all responses are aggregated in one place. The content of the transportable application is current when read; the transportable application is constantly updated so users can always see the most current information and responses of

other group members. Transportable applications may be supported by related services, such as a secure mode that enables only authorized members to view and interact with the content, notifications that inform a user when an update to a transportable application has occurred, etc. Transportable applications may generate events that are acted upon by other transportable applications, and may act upon events that are received from external systems.

In a business enterprise environment, transportable applications and the systems and methods disclosed herein enable the enterprise to save time, reduce costs, and accomplish more. Companies can use transportable applications for simple shared projects to the complex and collaborative business processes required for global, multi-company projects, depending upon the need. Collaborative business processes may be rapidly automated and streamlined. Collaboration may be extended outside the corporate firewall by sending the transportable applications to others. The relevance and speed of collaborative application development may be improved. Costs of distributed team interaction may be reduced.

In one approach, the system features an integration framework comprising connectors that provide connectivity to and integrating with existing or "legacy" databases and applications. An author can create transportable applications that can retrieve data values from and set data values in such external applications and systems without programming knowledge, or even automatically from another program. As a result, legacy systems can automatically interact with collaborative applications, without human intervention, to exchange data and carry out business processes. A simple and efficient means is provided to enable the collaborative applications to receive signals from the external applications, take action based on the signals, or initiate new processes based on the signals.

According to another aspect, a method for associating related electronic messages in computer storage is disclosed. A first transportable application is created and stored. User input requesting creation of a link from the first transportable application to another transportable application is received. User input that selects a second transportable application from among a plurality of previously created transportable applications is received. A link from the first transportable application to the second transportable application is created and stored.

As a result, transportable application may be inter-related in complex message webs. The message webs may also be inter-related in message web rings. Links may be generated manually or automatically, based on context, workflow processes, or other

known relationships among applications. Recipient lists and data may propagate among fields of linked transportable applications, directly or according to abstract business rules.

According to another aspect, a method for processing a request to display an electronic message is disclosed. First message data is generated. The first message data defines at least a first message portion and one or more navigation regions for one or more other message portions of a multiple-part electronic message having a plurality of message portions. The first message data is provided to a client. A request is received from the client for a second portion of the electronic message selected from among the other message portions. Second message data is generated which, when processed at a user interface of the client, causes the client to display the second portion of the electronic message. The second message data is provided to the client.

Multiple-part electronic messages are disclosed. A graphical user interface of an electronic messaging system displays a message in the form of one or more header portions and one or more body portions. Each of the body portions has a selection region. While one body portion is visible at a given time, all the selection regions are continuously visible in the user interface to facilitate selection of any of the body portions at a particular time. Selecting a selection region of a non-displayed body portion causes a server to generate a refreshed user interface that includes the entirety of the selected body portion, and that hides the previously viewed body portion. As a result, a large amount of associated information may be combined in a single message in a way that is clearly organized and easily accessible.

Transportable applications are disclosed. The transportable applications may be linked into webs of associated messages. For example, multiple related discussions or applications may be associated and organized. The webs of messages may be linked into rings of related message webs. As a result, workflow processes and complex associated information may be accessed and analyzed in an ordered way.

In other aspects, the invention encompasses computer readable media, and systems configured to carry out the foregoing steps. Other features and advantages of the invention will be apparent from the following detailed description, the drawings, and from the claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

FIG. 1A is a block diagram providing a structural overview of a system context in which an embodiment may be used.

FIG. 1B is a block diagram that illustrates software elements of one embodiment of a service unit.

FIG. 1C is a block diagram of the system of FIG. 1A illustrating additional elements.

FIG. 1D is a block diagram illustrating an example of interaction of a container and its building blocks with system services.

FIG. 1E is a block diagram illustrating a high-level view of a process of creating and interacting with a transportable application.

FIG. 2A is a block diagram of system architecture that may be used to host the foregoing services and provide such services to clients.

FIG. 2B is a block diagram of an example software architecture of elements of the system of FIG. 2A.

FIG. 2C is a diagram of an example graphical appearance of a transportable application, according to one embodiment.

FIG. 2D is a block diagram of an alternate embodiment of a system architecture that features use of certain Microsoft® servers and systems.

FIG. 2E is a block diagram of an alternate embodiment of a system architecture that features use of certain Microsoft® servers and systems.

FIG. 2F is a block diagram of an alternate embodiment of a system architecture that features use of certain Microsoft® servers and systems.

FIG. 3 is a diagram of dependency relationships among interfaces that may be used, in one embodiment.

FIG. 4 is a block diagram of an example of a membership hierarchy.

FIG. 5 is a block diagram of objects and data communications paths involved in presentation of content associated with a building block.

FIG. 6 is a block diagram illustrating interaction of programmatic objects in one embodiment of a presentation model.

FIG. 7 is a flow diagram of one embodiment of a process of carrying out response-based notifications.

FIG. 8A is a diagram of an example graphical user interface screen display of the Escalation Exception Manager template.

FIG. 8B is a diagram of a screen display of an Addressing page that may be displayed, in one example embodiment.

FIG. 8C is a diagram of an example of the Message Security Options window that may be displayed, in one example embodiment.

FIG. 8D is a diagram of an example of the Page Security Options window that may be displayed, in one example embodiment.

FIG. 9A is a diagram of the Join Group screen that is displayed when the selected group is private.

FIG. 9B is a diagram of a My Groups screen that is generated and displayed when a user selects the My Groups link of the portal home page.

FIG. 9C is a diagram of a portion of the screen display of FIG. 9B.

FIG. 9D is a diagram of an Application Builder screen display that is displayed in response to selecting the Application Builder.

FIG. 9E is a diagram of a page navigation dialog that may be displayed, in one example embodiment.

FIG. 10A is a diagram of a Default Page Settings screen that may be used to change properties of a selected default page of a template.

FIG. 10B is a diagram of a Page Builder screen that may be displayed in response to selecting the Page Builder, in one example embodiment.

FIG. 10C is a diagram of a Survey Page Builder window that may be displayed, in one example embodiment.

FIG. 11A is a diagram of the graphical appearance of an Approval List building block as it appears in a transportable application page under development using the Page Builder.

FIG. 11B is a diagram of an example embodiment of a Discussion building block that may be used to captures a discussion between message participants.

FIG. 11C is a diagram of an example embodiment of a File Sharing building block, according to an example embodiment, which may be used to share an electronic file among a group of recipients for collaborative review or updating.

FIG. 11D is a diagram of an Image building block that may be provided, in one example embodiment.

FIG. 11E is a diagram of an Image Gallery building block that may be provided, in one example embodiment.

FIG. 11F is a diagram of an Information Fields building block that may be provided, in one example embodiment.

FIG. 11G is a diagram of an Inline Document building block that may be provided, in one example embodiment.

FIG. 11H is a diagram of an Inline Web Page building block that may be provided, in one example embodiment.

FIG. 12 is a diagram of an example transportable application that contains an inline Web page.

FIG. 13A is a diagram of an Invitation building block that may be provided, in one example embodiment.

FIG. 13B is a diagram of a Poll building block that may be provided, in one example embodiment.

FIG. 13C is a diagram of an Advanced Options window that may be displayed, in one example embodiment.

FIG. 13D is a diagram of a Schedule building block that may be provided, in one example embodiment.

FIG. 13E is a diagram of a Table building block that may be provided, in one example embodiment.

FIG. 14A is a diagram of a Choice building block that may be provided, in one example embodiment.

FIG. 14B is a diagram of a Free Text Response building block that may be provided, in one example embodiment.

FIG. 14C is a diagram of a Ratings building block that may be provided, in one example embodiment.

FIG. 15 is a block diagram that illustrates data and control flow among building blocks, data access objects, interface methods and the database.

FIG. 16 is a flow diagram illustrating a process of presenting building block information using the foregoing mechanisms.

FIG. 17 is a block diagram illustrating elements of an event handling system, in one example embodiment.

FIG. 18A is a block diagram of a first enterprise application integration approach that uses an asynchronous approach.

FIG. 18B is a block diagram of the system of FIG. 18A wherein a custom connector is used.

FIG. 18C is a block diagram of an application-server centric integration approach for providing a synchronous integration solution.

FIG. 18D is a block diagram of an enterprise application integration approach that provides synchronous integration through one or more synchronous protocols.

FIG. 18E is a block diagram of an enterprise application integration approach that uses event-based communication.

FIG. 19 is a block diagram that illustrates a computer system with which an embodiment may be implemented.

FIG. 20 is a block diagram illustrating a plurality of messages that are linked across different folders.

FIG. 21A is a diagram of a first embodiment of a graphical user interface display of a multiple-page electronic message.

FIG. 21B is a diagram of a second embodiment of a graphical user interface display of a multiple-page electronic message.

FIG. 22A is a diagram of a third embodiment of a graphical user interface display of a multiple-page electronic message.

FIG. 22B is a diagram of a fourth embodiment of a graphical user interface display of a multiple-page electronic message.

FIG. 22C is a diagram of a further embodiment of a graphical user interface display of a multiple-page electronic message that includes an indicator of additional pages.

FIG. 22D is a diagram of a further embodiment of a graphical user interface display of a multiple-page electronic message.

FIG. 23A is a flowchart of a process for linking messages, according to one embodiment.

FIG. 23B is a flowchart of a process of automatically linking messages in another embodiment.

FIG. 23C is a flow diagram of a process of automatically creating message links in response to a change in an object.

FIG. 23D is a flow diagram of a process of updating message links in response to changes in message content.

FIG. 23E is a flow diagram of a process of suggested message linking.

FIG. 24 is a flow diagram of a process of displaying HTML content in an e-mail client with browser navigation features.

FIG. 25A is a block diagram of a linked collection of related message webs, referred to herein as a message web ring.

FIG. 25B is a flow diagram of a process of creating a message web ring.

FIG. 26A is a block diagram illustrating messages in a message web having shared address lists.

FIG. 26B is a flow diagram of a process of generating a list of recipients of a transportable application.

FIG. 27A is a block diagram illustrating that the content of messages that are linked can be changed, with automatic propagation of changed content to linked messages.

FIG. 27B is a flow diagram of a process of updating data among linked messages in a message web.

FIG. 27C illustrates one application of the processes of FIG. 27A, FIG. 27B in which a linked collection of related messages are used to aggregate data from a child message up to a parent message.

FIG. 28A is a diagram of a graphical user interface display in which a transportable application includes a plurality of tabs that switch between the content of message web members.

FIG. 28B is a flow diagram of a process of adding tabs to a transportable application of the type shown in FIG. 28A.

FIG. 28C illustrates a notification message of a message web.

FIG. 29 illustrates a method for tracing deleted messages using links.

FIG. 30 illustrates a message web map that may be used in a graphical user interface of a messaging system that supports linked messages, in one example embodiment.

FIG. 31A, FIG. 31B, and FIG. 31C are diagrams of graphical user interface displays that illustrate an example of a recruiting process using a message web.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

Unless otherwise defined, all technical and scientific terms used herein have substantially the same meaning as commonly understood by one of ordinary skill in the art to which the invention pertains. ZAPLET is a trademark of Zaplet, Inc., Redwood Shores, California. Microsoft, Microsoft Outlook, Microsoft Active Directory, Microsoft Exchange Server, Microsoft SQL Server, Microsoft Internet Explorer, and Microsoft

Internet Information Server are trademarks of Microsoft Corporation, Redmond, Washington. All other trademarks mentioned are the property of their respective owners.

Embodiments are described herein in sections according to the following outline:

TRANSPORTABLE APPLICATION COMMUNICATIONS SYSTEM

System Structural Overview

Clients

Networks

Servers—Application Server, Proxy Server, Mail Server, Wireless Gateway, Directory Server, Middleware, File Server, Photo Server, Database Server, etc.

Support Applications—Java, Database, etc.

Transportable Applications—Structural and Functional Overview

Graphical Appearance of a Transportable Application—Generally

Receiving and Sending Transportable Applications

Containers

Root Container

Contained Containers

Implementation with Enterprise Java Beans

Templates

Pages and Building Blocks

Object Model

Membership

Building Blocks—Programmatic Aspects

Directory Integration—Global Object Identifiers

Presentation of Transportable Applications

E-mail Client Presentation

Personal Portal Presentation

Multiple-Page Transportable Applications

Programmatic Processes for Presentation

Authoring Transportable Applications

Application Editor

Page Editor

Building Blocks--Examples

Site Style Editor

Communications From Transportable Applications to Users ("Notifications")

Notifications Based on User Responses or Timed Events

Notifications Based on Rules and Attributes
Object Communications—Programmatic Methods
Object Communications—Event-Driven Methods (Event Handling System)
Database Schema
Application Programming Interface & Object Methods
Object Communications—External Systems
Enterprise Application Integration Using Connectors
Security Processes
Cluster-Specific Encryption and Request Routing
Sharing Transportable Application Data Among Multiple Sites
System Administration Processes
Using Transportable Applications in Business Processes and Workflow
MULTIPLE-PART ELECTRONIC MESSAGES
LINKING AND AGGREGATING MESSAGES
HARDWARE OVERVIEW
EXTENSIONS AND ALTERNATIVES

- - -

1.0 TRANSPORTABLE APPLICATION COMMUNICATIONS SYSTEM

According to an embodiment, a communications system, and data processing methods that it carries out, facilitate collaborative communication and work among a group of individuals. Embodiments feature structures and processes for creating, authoring, transporting, updating, and viewing certain transportable applications. In general, a transportable application is an interactive, dynamic, addressable computer software application that can be transported among systems. A transportable application comprises one or more pages. Each page comprises one or more building blocks, and can be designed for either specific or general uses. Each building block contains the user interface, data elements, and processing capability that enable specific functions within a page.

A transportable application can be transported among different computer systems, clients and servers using transport mechanisms such as e-mail and other data transport protocols. When a transportable application arrives at a destination computer system, a user of that system can open the transportable application, view content that is dynamically asynchronously updated from a server source, and interact with functional elements of the transportable application in order to carry out work or other useful tasks.

Transportable applications are a new way to communicate by turning an existing transport mechanism, such as e-mail, into a live, shared place. Transportable applications enable users to share documents, data, business applications, schedules and other information with a group of people or with other systems. They can be used to make decisions with friends, family, or co-workers, schedule meetings and events, gather group data such as contact information, collect opinions, send invitations and RSVPs, and other tasks. In one embodiment, transportable applications enable all recipients in a group to communicate using just one e-mail message. Because a single transportable application takes the place of a thread of emails, transportable applications reduce inbox clutter and eliminate confusing strings of messages.

Transportable applications as described herein are better than e-mail because they provide information that is fresh and up-to-date. In one embodiment, transportable applications arrive in the user's inbox like normal e-mail, but they are current when the user opens them; new information can be included between the time the original author sent the transportable application and the time you open it. The transportable applications are current when read, not current when sent, eliminating the problem of receiving messages that are obsolete upon arrival.

The transportable applications add organization to the normally undifferentiated pile of individual e-mail messages. For example, transportable applications provide a structured container in which related data is collected, thus freeing users of organizing multiple pieces of e-mail. Additionally, transportable applications show aggregate views of multiple pieces of e-mail, thus summarizing information for better decision making. Also, transportable applications save time by reducing the volume of e-mail. By consolidating many responses into a transportable application, there are fewer pieces of e-mail to read.

Transportable applications are better than using the Web as a communications medium, for numerous reasons. For example, transportable applications push individualized messages into a user's Inbox. Transportable applications do not require a user to remember to visit a web site to see if they have a message.

Moreover, transportable applications are better than instant messaging as a day-to-day communications platform because transportable applications overcome the difficulty in getting multiple people to collaborate in real time. Transportable applications are targeted for group communication and persistent knowledge acquisition. In contrast, instant messaging is not designed for group communications and persistent knowledge acquisition.

1.1 System Structural Overview

FIG. 1A is a block diagram providing a structural overview of a system context in which an embodiment may be used. In general, a client 102 is communicatively coupled through a network 108 to a service unit 105. The client 102 creates requests for transportable applications and dispatches the requests into network 108. The requests are received by service unit 105 and processed, resulting in creation of a response, which is sent back over network 108 to client 102.

Client 102 may be, for example, a workstation, personal computer, personal digital assistant, cellular telephone or other wireless processing device, or any other end station data processor. Client 102 also may be a programmatic process that forms a part of a larger computer system or application program. Network 108 is any data communication network that can carry data communications between client 102 and service unit 105. Network 108 may be a direct electronic connection of client 102 of service unit 105; a local area network; a wide area network; one or more internetworks; a combination of a an access network, a service provider network, and one or more internetworks; or any combination of the foregoing. Client 102 also may comprise the combination of any of the foregoing with a client software application that enables a user to view, create, edit, and manage transportable applications.

Service unit 105 is a facility that hosts hardware and software elements that provide the functions described herein, and may be a private business enterprise, a service provider that provides the functions described herein to others, e.g., on a licensed basis, subscription basis, etc., or any other entity that hosts appropriate service equipment and facilities.

Client 102 executes a browser 104 and a personal information manager 106. Browser 104 is a conventional Internet or World Wide Web browser, such as Microsoft Internet Explorer. Personal information manager 106 is one or more software elements that provide e-mail processing, among other functions. Optionally, personal information manager 106 also may provide other personal information management functions such as address book management, calendar management, notepads, task lists, etc. An example of a commercial product suitable for use as personal information manager 106 is Microsoft Outlook. In an alternative approach, client 102 may be an external computer system that receives transportable application programmatically, without using browser 104 or personal information manager 106.

FIG. 1B is a block diagram that illustrates software elements of one embodiment of service unit 105. In general, in one embodiment, service unit 105 hosts a multi-page

messaging service 110 that is communicatively coupled to an event processor 112, HTTP service 114, message service 116, database service 118, security service 120, device gateways 122, and applications 124. Such elements inter-operate to provide multiple-page messages and carry out the processes that are described further herein.

FIG. 1C is a block diagram of the system of FIG. 1A illustrating additional elements.

In the example embodiment of FIG. 1B, multi-page messaging service 110 comprises component/container processing logic 130, administrative services 132, policy management & reporting services 134, access control service 136, mail services 138, device services 140, offline message processing services 144, and event and notification services 146.

The component/container processing logic 130 creates and manages a plurality of message components and message containers using processes that are described herein. The administrative services 132 provide administrative functions, such as adding or deleting users, importing files of users, identifying an external directory server and associating it with the system, etc. The policy management & reporting services 134 facilitate creating and managing policies for acting on transportable applications, access controls, etc.

Access control service 136 is communicatively coupled to security service 120, which may include an authentication, authorization and access (AAA) server, such as a RADIUS server. Access control service 136 communicates with other elements of multi-page messaging service 110 to receive requests to determine whether a particular user is authenticated in the system. The access control service also can receive and process requests to determine whether an authenticated user is authorized to access a particular message or a particular page of a particular message.

Security service 120 provides authentication and protection of user names and passwords. In one embodiment, security service 120 also provides Secure Sockets Layer (SSL) communication facilities so that communications from clients 102 to service unit 105, and from service unit 105 to an external directory server or other secure element, can be encrypted and carried out in a secure manner. Security service 120 also may provide means for integration with third-party security systems. Using such integration, security service 120 may be extended to provide authentication, access control, privacy, non-repudiation, and data integrity services. Such third-party systems may be the Sun Java Cryptography Engine, systems from Entrust, Cylink, Cybersafe, Verisign, etc. Security service 120 may support encrypted communications using asymmetric encryption

algorithms such as RSA, DSS, D-H, etc.; symmetric encryption algorithms such as DES, triple-DES, DESX, IDEA, ES2, ES4, Blowfish, etc., message digest or hash algorithms such as SHA-1, SHA, MD5, MD2, RIPEMD, etc. Such encryption algorithms can be called programmatically from containers, building blocks, or system services that need to encrypt or decrypt data for communication, storage, or retrieval.

Mail services 138 is responsible for receiving data from other elements of the multi-page messaging service 110 and creating e-mail messages based on the received data. Mail services 138 also interfaces with message service 116 to dispatch and forward messages to users and clients. An example of a commercial product that may be used for message service 116 is Microsoft Exchange Server 2000.

In one specific embodiment, each of the foregoing services is implemented in the form of one or more Java class libraries. Thus, the services are provided by generic software modules that can be called by other software components of the system. The modules can incorporate third-party components, such as calculation engines, a workflow engine, etc.

FIG. 2A is a block diagram of system architecture that may be used to host the foregoing services and provide such services to clients.

One or more application servers 202 host containers in the form of Enterprise Java Beans 204 and store programmatic objects representing containers, building blocks, pages, and transportable applications in a cache 206. Application servers 202 execute code of the containers and building blocks and call servers and services in a services domain 201 to result in servicing client requests. In one embodiment, application servers 202 are implemented as a plurality of clusters of application servers, to provide load balancing and increased throughput. Inbound client requests are routed to a particular cluster using a routing mechanism that is described further herein.

Application servers 202 are communicatively coupled by link 210 to database server 208, which comprises one or more databases. The databases in database server 208 provide persistent storage of stored procedures, a repository of data and logic for containers, building blocks, and other system objects, and an event database that supports event-based messaging. Database 208 may be, for example, an Oracle database server, Microsoft SQL Server, etc. Link 210 may be implemented, in one specific embodiment, as JDBC calls from applications server 202 that traverse an SQL network using the SQLNet protocol and arrive at database server 208 in Oracle Command Interface (OCI) format. Alternatively, ODBC may be used, as in the case of SQL Server.

The database server 208 provides one source for content for building blocks and containers that application servers 202 may dynamically retrieve and present to clients 102 as part of processing transportable application and executing their building blocks. Other data sources may be used. In one embodiment, application servers 202 are communicatively coupled by an NFS link 246 to one or more file servers 244. This arrangement enables application servers 202 to obtain text files, word processing documents and other contents of a file system to include in the dynamic content of a building block or for other purposes.

One or more adapters 209 cooperate with a connection framework in application server 202 to communicate with one or more external systems or applications 205. Such external systems or applications include legacy applications, enterprise systems, etc., as described further herein.

In another embodiment, application servers 202 are communicatively coupled to one or more image servers 240 by an HTTP link 242. This arrangement enables application servers 202 to rapidly retrieve one or more graphical images, digital photos, and other image information to include in transportable applications when they are presented to clients.

Application servers 202 may obtain dynamic content from one or more external sources and may communicate with clients using connections to public networks. In one embodiment, application servers 202 are communicatively coupled by a first HTTP link 236 to an HTTP server 232 (Web server 232), proxy server 234, and second HTTP link 238. This arrangement enables application servers 202 to receive a large number of separate HTTP connections and requests from clients 102 over second HTTP link 238, which are received at proxy server 234 and passed to the application servers over the first HTTP link 236. When the application servers have prepared content for presentation to clients in response to client requests, Java Server Pages elements of the application servers send the content to HTTP server 232 over the first HTTP link 236. The HTTP server 232 then issues an appropriately encapsulated HTTP response to the clients over HTTP link 238, through proxy server 234.

Application servers 202 are also communicatively coupled by link 214 to middleware 212, which provides basic messaging infrastructure functions. In one specific embodiment, link 214 uses the Java Messaging System (JMS) protocol, which may also be used to transport data on link 215 from middleware 212 to one or more event daemons 216 of event service 146. The event service 146 is communicatively coupled by link 218 to database server 208. In one specific embodiment, link 218 is implemented by JDBC

calls from event service 146 that traverse the SQLNet and arrive at database server 208 in OCI format.

Directory information such as user names and group membership may be obtained from one or more directory servers 220 that are communicatively coupled to application servers 202 by link 224 and to event service 146 by link 222. In one embodiment, directory servers 220 conform to Lightweight Directory Access Protocol (LDAP) and communications on links 222, 224 are carried out using LDAP. Directory servers 220 need not be co-located with application server 202, and may be located in any location that is accessible using LDAP communications over a network or similar facility.

Application servers 202 communicate with e-mail clients using mail servers 228, and with wireless devices, such as wireless personal digital assistants and cellular phones, using a wireless gateway 226. In one embodiment, one or more mail servers 228 are communicatively coupled by an SMTP link 225 to event services 146, which is responsible to generate events that request mail transport. Event services 146 also are coupled by SMTP link 225 to wireless gateway 226, and the wireless gateway may be communicatively coupled to the mail servers by link 227. Mail servers 228 and wireless gateway 226 communicate with e-mail clients, other client devices using external networks that are reached using SMTP link 230.

FIG. 2B is a block diagram of an example software architecture of elements of the system of FIG. 2A.

The software elements of the system may be viewed as logically organized in a client tier 248, middle tier 252, and data tier 275. Client tier 248 comprises an e-mail client 249 that executes on a client computer, e.g., client 102, and a browser 250 that executes at the same client. The e-mail client 249 and browser 250 may interoperate with a shared DLL that provides HTML parsing functions, so that the e-mail client can display HTML e-mail messages.

In one embodiment, system services generally are provided by a plurality of servlet constructs that implement the services. For example, middle tier 252 comprises a mail server 254 that communicates with e-mail client 249 over a network link. The mail server is communicatively coupled to one or more mail queuing servlets 262 that are organized in a Web container 258 of program elements. The Web container 258 further comprises a mail formatter 260, which may be implemented in the form of one or more Java Server Pages. Creation and modification of dynamic content of a transportable application is generally performed by dynamic content creation servlets 264. A Web page

formatter 266, implemented in the form of one or more JSPs, is responsible for formatting Web pages or other HTML content that may form a part of a transportable application.

In one specific embodiment, middle tier 252 further comprises an EJB container 268 that comprises a plurality of program elements in the form of Enterprise Java Beans (EJBs). EJBs are Java® language classes that can interoperate with a plurality of servers within an enterprise network. Data access EJBs 270 are responsible for interfacing higher-layer elements with database 208 and one or more stored procedures 276 in the database. Core EJBs 272 are responsible for carrying out access control processes, managing event services, etc. Data and event integration EJBs 274 provide an interface to external systems such as a legacy database 278, an enterprise application integration program or bus 279, etc.

In an embodiment, these elements are implemented using Java-based technologies in conformance with the Java 2 Enterprise Edition (J2EE) specification for server-based applications. However, embodiments are not limited to this context.

FIG. 2D is a block diagram of an alternate embodiment of a system architecture that features use of certain Microsoft® servers and systems. In general, FIG. 2D depicts a client layer 2000 that communicate with servers 2002. Client layer 2000 comprises, in this example, a Microsoft Outlook client 2004. Servers 2002 comprise a Microsoft Exchange server 2008, an Active Directory server 2026, Web storage system 2020, application server 202, database server 208, and host integration server 2034.

In this configuration, Outlook client 2004 communicates over a mail application programming interface (MAPI) or remote procedure call (RPC) link 2006 with a MAPI processor 2012 of the Microsoft Exchange server 2008. The Microsoft Exchange server 2008 further comprises an SMTP agent 2010, OWA agent 2014, event handling system 2016, and Exchange OLE Database agent 2018. The SMTP agent 2010 can communicate with SMTP-compatible mail servers to enable the system to transport e-mail messages across a network. Event handling system 2016 receives events generated by application server 202 or its applications and processes them by, for example, storing information in mailbox store 2022 and public stores 2024 of Web storage system 2020. Database operations by or on behalf of application server 202 are facilitated by Exchange OLE Database agent 2018, which is communicatively coupled to Web storage system 2020.

Web storage system 2020 provides Exchange server 2008 with a unified storage name space for all messages, calendar information, documents, discussions, or other data. Its objects are accessible using HTTP, XML, COM, WebDAV, and file system calls. In one embodiment, transportable applications are stored in Web storage system 2020. Such

transportable applications, their document attachments, and their building blocks are addressable in Web storage system 2020 using URLs. Web storage system 2020 may also store information about a plurality of Exchange offline folders, facilitating synchronization of offline updates to transportable applications.

Active Directory server 2026 is communicatively coupled to Exchange server 2008 and stores directory information, such as definitions of users and groups, locations of network elements such as servers, hosts, workstations, and printers, etc. The Active Directory server 2026 is also accessible by application code 202A, which is hosted by application server 202, through a common data object (CDO) agent 2028 that is hosted by the application server. Active Directory is an example of an LDAP directory as described herein.

Application server 202 further comprises an Active Data Objects (ADO) process 2030 that is communicatively coupled to Exchange OLE Database agent 2018 and to database 208 and host integration server 2034. These communication paths enable functions of application code 202 to set and retrieve values of data objects that are stored in database 208, in external legacy systems that are accessed using host integration server 2034. Further, the application code 202 can set and retrieve values of data objects that are stored in Web storage system 2020 by appropriate messages to Exchange OLE Database agent 2018.

Application code 202A of application server 202 comprises sequences of computer program instructions that are configured to carry out the functions that are described herein.

In the configuration of FIG. 2D, transportable applications that are hosted by application server 202 may be created, viewed, and managed using Outlook client 2004. The Outlook client 2004 is an example of an HTML e-mail client that can display transportable application content in rich HTML form within an e-mail message window. The transportable applications may be transported over a network using Exchange server 2008. Active Directory server 2026 may serve as a source for user and group information. Building blocks may be implemented in the form of Web forms, which provide user interface front-ends to data and logic in the Web storage system 2020.

Exchange server 2008 includes instant messaging support with presence detection API's. Thus, application server 202 may determine, by an appropriate API call, if participants in transportable applications are actually online, and utilize instant messaging for communications, and record such sessions at user request. Transportable applications can include NetMeeting features for initiating and storing data conferences, such as

virtual white board, application sharing, video conferences, audio conferences, etc., through Exchange server 2008.

Exchange has threaded discussion capabilities that are programmatically available. In one embodiment, application server 202 uses the threaded discussion APIs to provide a threaded discussion capability, with presence detection, as a core building block.

Calendaring and contact information access through Exchange server 2008 allows for scheduling features, such as automatic meeting creation, available time lookup and resolution, meeting cancellations, etc., and more integrated addressing and address validation on the server side. Use of Exchange server 2008 also enables application server 202 to communicate transportable applications to PDA, mobile phone, and other non-PC devices, over wired and wireless.

A workflow engine of Exchange server 2008 has an accompanying graphical workflow designer that allows execution of logic and invocation of components in and outside of Exchange, via COM, that provide message routing, approval, and state transitions. Thus, transportable applications of application server 202 may participate and drive these workflows.

FIG. 2E is a block diagram of an alternate embodiment of a system architecture that features use of certain Microsoft® servers and systems. As in FIG. 2D, the architecture of FIG. 2E features an Outlook client 2004, Exchange server 2008, Active Directory server 2026, Web storage system 2020, application server 202, and host integration server 2034. The architecture of FIG. 2E further features an Internet Information Server (IIS) 2036 that is communicatively coupled to Exchange server 2008 via OWA agent 2014. In client layer 2000, an Internet Explorer browser 2005 is communicatively coupled, for example, over a network, to IIS 2036.

Browser 2005 and Outlook client 2004 may carry out HTML display functions using a shared DLL.

In this configuration, application server 202 may serve transportable applications to a user through browser 2005, Outlook client 2004, or both. For example, a user may use browser 2005 to open an HTTP connection to IIS 2036, which communicates HTTP requests and posts among application server 202 and the browser. Using this mechanism, a user may obtain the portal presentation described herein and may view, create, and manage transportable applications. Alternatively, as in FIG. 2D, the user may obtain a portal view, or view an individual transportable application, in an e-mail window that is generated by Outlook client 2004.

FIG. 2F is a block diagram of an alternate embodiment of a system architecture that features use of certain Microsoft® servers and systems. As in FIG. 2E, the architecture of FIG. 2F features an Outlook client 2004, Exchange server 2008, Active Directory server 2026, Web storage system 2020, application server 202, host integration server 2034, and an Internet Information Server (IIS) 2036 that is communicatively coupled to Exchange server 2008 via OWA agent 2014. In client layer 2000, an Internet Explorer browser 2005 is communicatively coupled, for example, over a network, to IIS 2036. Any other browser 2005A, such as Netscape Navigator, may participate in client layer 2000. Browser 2005 and Outlook client 2004 may carry out HTML display functions using a shared DLL.

The architecture of FIG. 2F further features a plurality of Active Server Page (ASP) constructs that are hosted by IIS 2036. The ASPs are callable from application code 202A, as indicated by link 2038, by passing URLs over a network to IIS 236. The URLs identify a particular ASP and provide one or more function parameters or data values.

In this configuration, certain transportable application functions that are defined herein are carried out by appropriately configured ASPs 2040 rather than by application code 202A. For example, any of the functions that are described herein as implemented using JSP constructs may be implemented, alternatively, using ASPs 2040 in a system configured as in FIG. 2F.

In still another alternative configuration, elements of the Microsoft ".NET" infrastructure are used. In this embodiment, the database server 208 is Microsoft SQL Server, but all other components are found within Windows 2000, Windows XP, and related server products of Microsoft. For example, ISAPI filters are used as a substitute for J2EE servlets as disclosed herein; Active Server Pages substitute for JSP's; Microsoft Message Queues (MSMQ) substitute for JMS; COM+ components substitute for EJB's; and ODBC is the database communication protocol rather than JDBC. Data access objects (DAOs) as disclosed herein may be implemented using a custom OLE database provider that is compatible with the Microsoft Active Data Object (ADO+) framework. COM+ access to external legacy applications may be provided as an alternative to JCA or connectors as disclosed herein.

As set forth in the foregoing, embodiments of transportable applications, an application server, and associated functions described herein are applicable equally to contexts that use Microsoft servers, products, and related elements, as well as Java

servers, open-source elements, servers or products of Sun Microsystems and its partners, etc.

1.2 Transportable Applications—Structural Overview

A system having the foregoing configuration may be used to create, send, respond to, interact with, and manage a plurality of transportable applications.

FIG. 1E is a block diagram illustrating a high-level view of a process of creating and interacting with a transportable application. In block 190, an author of a transportable application accesses an application authoring system. The application authoring system may comprise a transportable application editor that is provided by application server 200 and accessed through an enterprise network or over a public internetwork.

In block 191, the author creates, authors and edits a new transportable application using the editor, resulting in creating and storing the transportable application, as shown in block 192. Such authoring also includes identifying one or more recipients of the transportable application. In block 193, the transportable application is transported to all the recipients using a suitable transport mechanism. Transport mechanisms may include e-mail, wireless gateways, or other means.

When the transport mechanism is e-mail, in block 194, each recipient receives an e-mail message identifying the transportable application and appearing to contain the transportable application. Assume that a particular recipient opens the e-mail message that identifies the transportable application. In response, in block 195, the system dynamically determines what client e-mail program is in use, and determines what display capabilities it has with respect to fonts, colors, graphics, other types of information, etc. In block 196, the transportable application is displayed to the user using its e-mail client, based on the available display capabilities of the client. Further, any dynamic content associated with the transportable application is automatically retrieved, in its then-current form, from a database and displayed as part of the transportable application.

In block 197, the recipient responds to the transportable application in a manner analogous to replying to an e-mail message. Text comments or other changes to dynamic content of the transportable application may be entered in the application as it is displayed within a window of the e-mail program. Any such changes are updated at the database in dynamic content associated with the transportable application, as shown by block 198. As a result, any other recipient who subsequently views the same transportable application will receive a current view of the dynamic content. In this sense, the dynamic content and the transportable application is always current when read.

FIG. 1F is a block diagram illustrating details of a process of authoring a transportable application as in block 191 of FIG. 1E.

In block 191A, one or more pages for the transportable application are created or retrieved from storage. In block 191B, a template for the transportable application is created. In block 191C, a transportable application is created based on the template and/or the pages.

Each of blocks 191A, 191B, 191C may be carried out by a different individual. For example, one person can create a template and another person can created a transportable application based on the template. Recipients of the transportable application may, depending on the settings of the transportable application, change the structure of the transportable application. Alternatively, the same person can be responsible for all stages of the authoring process by creating templates, authoring new transportable applications, and participating in collaboration with others within the framework of a transportable application. Individual pages and templates may be shared with others.

1.2.1 Graphical Appearance of a Transportable Application—Generally

FIG. 2C is a diagram of an example graphical appearance of a transportable application, according to one embodiment. In this embodiment, a transportable application is displayed in a transportable application window 280 that comprises, in general, a command options bar 282, an action request or note 284, a header bar 286, and one or more pages 287A, 287B. Each of the pages 287A, 287B may comprise one or more building blocks 292, 294. Each building block contains the user interface, data elements, and processing capability that enable specific functions within a page. Navigation among pages is accomplished by selecting a navigation area 288 associated with a particular page, or by selecting a navigation direction icon 296.

In one specific embodiment, command options bar 282 includes a New Message button, Reply to Author button, Forward button, and Print View button. The New Message button is used to author a new transportable application message. A Reply to Author button is used to send a message directly to the author of the transportable application displayed in window 280. The Forward button is used to forward the transportable application to additional recipients. The Print View button is used to obtain a properly formatted printed version of the transportable application.

Header bar 286 provides addressing information identifying the author of the transportable application, its recipients, and its subject or title. In one embodiment, the specific names of recipients are hidden, and header bar 286 includes a View All

Recipients link 286a. Selecting the View All Recipients link 286a causes the system to display a list of all recipients.

One or more notifications may be associated with a transportable application. If a notification is present for a transportable application, a notification link 290 is displayed, and provides a hyperlink with which the user may obtain a view of the notifications. When a user receives a transportable application, a notification system enables the user to "opt-in" to individual transportable applications in which the user has an interest. In one approach, if dynamic content associated with the transportable application changes, a new copy of the transportable application is sent to the user. Alternatively, notifications may comprise re-bolding an identifier of the transportable application in the user's e-mail client. In this manner, the user can track changes without having to remember to continually go back and check on the state of the transportable application. For example, a user may read a transportable application, opt-in to the notification system for that transportable application, delete the transportable application from the user's e-mail Inbox, and wait for a notification when anything changes in the transportable application.

Each user is sent only a single copy of the transportable application, even if there are multiple responses to it. The system knows when the transportable application has been read by the user and can minimize the number of notifications sent. If the user has not opted-in to the notification system, the user can always open the transportable application or refresh the display of client 102, e.g., by selecting a Refresh button of the browser, to see the latest information.

A transportable application also can change state during its lifetime. For example, an author can close a transportable application; this notifies the recipients that comments or replies are no longer being accepted on the subject of the transportable application. The author of the transportable application can then communicate a decision or the results of the transportable application to all the recipients. Thus, the user no longer needs to spend time writing a response to an e-mail when its subject matter is essentially closed.

An Options link 291 provides a mechanism to display a list of actions that the user can take with the transportable application.

The action request or note 284 displays a brief urgent note or request from the author of the transportable application. Such notes provide a way to carry out a side conversation among recipients of the transportable application and the author or other recipients. Such notes may be stored persistently and generally are directed to one recipient. In one embodiment, a transportable application that is sent in the form of an HTML e-mail to a recipient may comprise a script, which is executed when the message

is opened by the recipient, that establishes an HTTP connection to application server 202, requests any changes to dynamic content of the transportable application, determines whether the recipient has received any notes, and displays the notes if any have been received. The notes may be displayed within the transportable application, as in FIG. 2C, or in a pop-up window that is displayed separately.

1.2.2 Receiving and Sending Transportable Applications

In general, interacting with a transportable application involves receiving a message that references or contains a transportable application, entering a response, and using one or more transportable application commands. A user may receive a transportable application as a message carried using any suitable transport mechanism. In this description, for purposes of illustrating a simple example, the transport mechanism is assumed to be e-mail. However, any other suitable transport mechanism may be used, such as wireless gateways, voice and other multimedia protocols, etc. Further, various e-mail protocols may be used, such as SMTP, MAPI, etc.

When a user receives a transportable application, the user receives a specially designated e-mail message in the in-box of the e-mail client of the user. To begin interacting with the transportable application, the user opens the message. In response, the contents of the transportable application are automatically dynamically updated from a server, such as application server 202 of FIG. 2A. The transportable application is also updated when a user submits a response.

Specific processes for interaction with a transportable application depend on what pages and building blocks are contained in the transportable application. Specific building blocks are described further herein. In general, interaction involves selecting a desired page using a navigation area 287B and locating a response button. Each building block has a response button such as "Enter Response," "Add Image," "Add Table Row," etc. The response button is selected. In response, the system generates and sends an Enter Response window to the client 102, which displays it as a new graphical user interface window. An appropriate response is entered.

When the response is complete, the user selects a Submit button to send the response information to the server. Alternatively, the user may select Cancel to exit without entering a response. In response to a Submit selection, information for the transportable application in a database associated with the server is updated to show the response. The foregoing steps may be repeated any number of times for any number of pages in the transportable application. When the user is finished entering responses, the user closes the message.

As noted above, notifications may be turned on and off by selecting notifications link 290. FIG. 7A is a diagram of a Set Notifications user interface window that is displayed when notifications link 290 is selected. Window 700 comprises an On radio button 702, Off radio button 704, and Customize radio button 706.

If the Customize radio button 706 is selected, in response, the server generates and sends to the client a Customize Notifications window. FIG. 7B is a diagram of a Customize Notifications window that may be displayed. The Customize Notifications window 708 displays one or more page check boxes 710 corresponding to pages of the transportable application. Selecting one of the check boxes 710 causes the system to generate a notification message only when a change occurs in the associated page. Window 708 also comprises an Overall Application check box 712 that triggers a notification when any change occurs in any part of the transportable application. Window 708 further comprises a Page Change checkbox 714 that triggers a notification when one or more pages are added to the transportable application or when its status changes. Window 708 also comprises a Set Frequency pull-down menu 716 that enables change notification messages to be sent at particular time intervals.

1.2.3 Containers

Each transportable application comprises one or more containers. Each container is implemented in the form of a programmatic object in an object-oriented programming environment. Each container may be a root container or an embedded container. Each transportable application has one root container, and one or more embedded containers.

The root container is responsible for interacting with transport services such as mail services 338, access and addressing services, and sending only new or changed information to and from its contained building blocks. Each root container stores a list of pointers or other references to contained child objects or building blocks. Thus, a tree walk of the pointers may be carried out to identify all building blocks or other child nodes of a particular container. A child, such as a building block or contained container, may have multiple parent containers; this mechanism enables, for example, a particular page to appear in more than one transportable application or in more than one template.

Each embedded container references and logically contains one or more other embedded containers or building blocks. Containers can call functions of the system services shown in FIG. 1A, FIG. 1B, and FIG. 1C, and thereby provide access to such services to the containers or building blocks that they contain. Further, containers coordinate presentation of content of contained building blocks. Containers can subscribe

to events relating to variable values that are published by contained building blocks and embedded containers. Containers also may publish self-defined events for the purpose of providing data or other information to other containers or building blocks, and containers may define global variable values that can be retrieved and modified by contained building blocks and embedded containers.

In one specific embodiment, each container is implemented in the form of an Enterprise Java Bean.

1.2.4 Templates

Transportable applications are created based on transportable application templates. To create a new transportable application, a user must first access a transportable application template. Templates may provide corporate processes, general-purpose applications, or other functions.

In one specific embodiment, corporate process templates include templates for recruiting management, collections, financial reporting, and similar corporate-level tasks. Examples of corporate process templates include:

Collections. Facilitates the collection of past due customer accounts. Multiple departments can participate in the process and upload relevant documents, read and add comments to a threaded discussion, and track the overall process using a shared timeline or task list.

Escalation Exception Manager. Helps resolve a customer problem by displaying a current view of the case, managing interactions, and facilitating decision-making. Supplements call center applications by extending collaboration to other departments.

Financial Reporting. Collections the quarter's performance highlights and manages the development of the earnings release and earnings call script.

Forecasting. Compiles forecast input from multiple sources. Participants can modify a document and then provide their approval or disapproval.

Lead Management. Distributes, compiles, and tracks lead data. Participants can enter data in the shared table, assess the opportunity, and close the lead.

Order Exception Management. Tracks and manages the resolution of a problem that occurs in fulfilling the order. Enables collaboration across the enterprise, partners and suppliers to define the exception and decide on a solution.

Product Configuration. Manages the product configuration process from initial product requirements to configuration and approval. Participants can modify documents and approve or disapprove the final versions.

Proposal Management. Displays a proposal for review and feedback and compiles sign-off by reviewers or stakeholders.

Recruiting Management. Organizes and streamlines the process of interviewing a candidate, from receiving the resume to making an offer.

In another specific embodiment, general purpose templates include:

Discussion. Initiate a group discussion. Try brainstorming ideas or resolving issues among a select group of recipients.

File approval. Request approval on files that require consensus. Participants either disapprove or approve and share comments.

File sharing. Distribute files for participants to review, discuss and collaborate on. A version control option allows participants to check files in and out.

Image. Display and discuss a single image from a computer or the World Wide Web.

Image Gallery. Display and discuss a plurality of images that participants or the user contribute to a shared stored area.

Image Poll. Gather feedback and reach consensus about images.

Inline document. Review and discuss an HTML document; the contents may be displayed within the transportable application page, so that participants do not have to open another window.

Interactive Web Page. Interact with and discuss a Web page. Participants can interact with the page's hyperlinks and fields and can navigate as they would on a Web browser.

Invitation. Invite colleagues to meetings and other events; display event details and collect RSVPs and comments from invitees.

Poll. Gather opinions and feedback; see voters' comments and a chart of the results.

Ratings. Gather feedback on up to 100 questions or statements. Participants can rate each on a scale and provide comments.

Schedule. Schedule a meeting or event by specifying dates and times and availability.

Table. Create a table to capture input from a team. Import a file to start the table, or create a table from scratch. Tables can be exported to a file for reuse.

The foregoing are merely examples, and templates providing other functions may be created or provided. Each template is created using an application editor or Application Builder, as further described herein. Updating a template using such an editor

or builder does not causes changes to transportable applications that have been instantiated from the template, however.

1.2.5 Pages and Building Blocks

Generally, building blocks of a transportable application define the specific appearance, content, and functions of the transportable application. Each building block comprises data elements, logical elements, and presentation elements, each of which may be implemented in the form of one or more classes, methods and interfaces in an object-oriented programming environment. Each building block is contained within a container, and communicates with other building blocks through the container. The data associated with data elements of the building blocks is stored in a persistent data store, e.g., a database server. Building blocks are implemented, in one embodiment, as one or more Java Server Pages.

FIG. 1D is a block diagram illustrating an example of interaction of a container and its building blocks with system services. For purposes of illustrating a simple example, FIG. 1D illustrates one instance of a container 180 having three instances of building blocks 182A, 182B, 182N. However, embodiments are not limited to this example context, and a practical system may have many thousands or millions of instances of building blocks and containers.

Client requests 163 arrive at a request dispatcher 165, which de-encapsulates each client requests and routes it to a particular service selected from among client services 174. Client services 174 may comprise a presentation manager 164, store/create manager 166, submit manager 168, and authentication manager 170. Client services 174 may call one or more system services 161 as needed to carry out specific functions. In the example of FIG. 1D, system services 161 comprise security manager 120, notification manager 148, mail services 138, directory manager 160, and system manager 162. In other examples and embodiments, system services 161 may comprise any of the services shown in FIG. 1B and FIG. 1C.

Programmatic components and interfaces 176 interact with the client services 174. Each container 180 and building block 182A, 182B, 182N implements such components and interfaces. For example, container 180 and building block 182A, 182B, 182N each implement a Presentable interface that comprises programmatic methods that determine how the container and building blocks are presented in a user interface. Container 180 implements a NodeObject interface that comprises methods that determine the object hierarchy of the container, i.e., which specific building blocks it contains. A further

description of such object interactions is provided herein in the section entitled Object Model.

In one specific embodiment, the data associated with data elements of the building blocks is stored in a persistent data store, e.g., a database server, that is dedicated to storage of system data and not generally exposed to external clients. The building blocks access services of the system programmatically through an associated container. In contrast, data for a Dynamic Web Page building block is stored in an external database, and accesses services of the system through a connector, using processes described herein in the section entitled "Enterprise Application Integration Using Connectors."

In this configuration, containers may be used to programmatically create transportable applications as described herein without use of an editor, builder, or other graphical user interface or user intervention. A program component may create or instantiate a container, add building blocks to it, and register the building blocks in the system. For example, in response to an event, an action can create a new transportable application and send it.

A description of examples of building blocks that may be provided in an embodiment is provided herein in the sections entitled "Application Editor," "Page Editor," and "Building Blocks—Examples."

1.2.6 Object Model

According to one embodiment, transportable applications and system services are implemented using programmatic objects that are created using an object-oriented development environment and executed in an object-oriented runtime environment. The programmatic objects are organized according to an abstract object model that comprises base interfaces, support classes, and component objects. A preferred embodiment is implemented using the Java environment, but this environment is not required, and program implementations in other object-oriented environments, or in procedural programming languages, threaded interpreted languages, etc., are possible.

In general, the base interfaces serve as APIs that can be called by other objects and services in the system. The intermediate support classes are classes with partial or full implementation of specific functionality, providing typical combinations of basic interfaces. The component objects are fully implemented classes for carrying out specific tasks.

Base interfaces, in one embodiment, relate to rendering and presentation of objects, data persistence, containment, event handling, service callbacks, core service management, and client service management.

FIG. 3A is a diagram of dependency relationships among interfaces that may be used, in one embodiment.

A Rendering/Presentation package provides interfaces for rendering objects. A Presentable interface 332 is used to render an object to a particular target client or device. A service or object that wishes to present the contents of an object creates and populates attribute values for information about a client request in a ClientContext data structure. This information may comprise a device identifier, rendering format, etc. An implementation of the Presentable interface uses the ClientContext data to create appropriate data for rendering.

A Data package deals with data persistence from different data sources. In one embodiment, a "DAO" interface 334 contains methods that allow creation of a data access object (DAO) from a data source, its maintenance and data storage.

A Hierarchy/Containment/Collection package deals with object hierarchy and containment processes, e.g., collecting child building blocks and containers, etc. In one specific embodiment, a CollectionNodeObject interface 336 provides hierarchy functionality for a derived object. This interface supports navigation across object trees and DAGs. The interface can be used to provide functionality that includes both object trees and object collections. An InheritanceCriteria interface 338 defines how default inheritance policies (e.g., security, notification) apply to a specific object hierarchy. Examples of these are access control inheritance for objects in a given hierarchy. The interface also covers whether the child object inherit state information from parent objects or not. For example, if a child container or building block object inherits state from a parent container object, and the parent container is closed, then all child containers will close. In contrast, state would not be inherited if deleting a group does not necessarily mean removing all members of the group, though it might be an option.

An event handling package deals with handling of events generated from different components in the system. In one specific embodiment, an EventHandler interface 340 provides a base event handler interface. Further description of event handling processes, in an example embodiment, is provided herein in the section entitled "Object Communications—Event-Driven Methods (Event Handling System)."

A service callback interface package contains definitions for callback interfaces that are used by service managers. The callback interfaces provide APIs that application

objects provide to services. Control managers, services and other components can use such interfaces to communicate with application components.

In one specific embodiment, an interface "ServiceInterface" 302 provides a base interface for all service interfaces. An interface "SecurityInterface" 304 provides a security service interface that can be used to query an object for access control permissions and other security related functionality. An interface "NotificationInterface" 306 provides interfaces that can be used to generate necessary notifications for a derived object. An interface "CommInterface" 308 is used for communication with users and other systems. An interface "MailableInterface" 310 is used to define how to make an object transportable using e-mail, and an object can query such interface to receive content and target mail addresses. An interface "LifecycleInterface" 312 covers the lifecycle management of a component, in the form of operations such as close a component, create a component, respond to a component, and add or delete members of groups and folders.

A package of core service managers contains a set of core service engines. In one embodiment, certain critical core service engines, e.g., a security service engine, are not extendable. They provide basic services that other components can use. Each of the service managers is implemented in standalone fashion and is not dependent on any functionality from other system components.

In one specific embodiment, a ServiceManager interface 314 provides a base interface for all other service managers. A SecurityManager interface 316 is a security provider for services such as authentication, access control and SSL, etc. A NotificationManager interface 318 is a notification provider. A CommunicationManager interface handles SNMP communications among system entities. A DirectoryManager interface provides LDAP directory functionality.

A package of client service managers provides interfaces that are related to platform services, and which rely on the components to provide certain functionality. The client service managers are analogous to brokers that act as liaisons between objects and services to achieve a particular end result, such as rendering an object in a user interface.

In one specific embodiment, a PresentationManager provides a broker that deals with object presentation in the system. Such objects may comprise: building blocks; container types such as container objects, folders, or groups; or meta objects such as authoring objects. A StoreCreateManager provides an interface for creating and storing objects. A SubmitManager provides an interface for submitting changes. An

AuthenticationManager provides functionality to authenticate users and applications that wants to Zaplet platform services, in communication with the SecurityManager.

A plurality of support classes are provided for the foregoing interfaces; the support classes generally comprise abstract classes with partial interface implementations. The support classes encapsulate a list of common interfaces across different platform components. Thus, the support classes enable objects to reuse common functionally across multiple components. In one specific embodiment, support classes comprise a MailableInterface, SystemService, Container, and BuildingBlock.

Component objects of the system are fully implemented objects. Components are composed from a single support class and one or more base interfaces classes. In some cases, a component needs functionality from multiple support classes. In this case, needed interfaces are implemented at the component level by delegating functionality to embedded support objects.

In one embodiment, a BuildingBlock component object represents an atomic displayable component that can be extended. Examples of building blocks include a poll building block, schedule building block, dynamic Web page building block, document upload building block, etc. FIG. 3 illustrates a schedule building block interface 320 and a poll building block interface 322, as examples. Each is constructed based on a building block interface 324.

An ApplicationContainer component object 326 represents a transportable application in the system, and may reference and logically include one or more building block objects. A FolderObject component object 328 represents a document folder in a manner analogous to file system folders in an operating system. The FolderObject component object provides a mechanism to organize different resources, such as folders, files, task templates, etc. A GroupObject component object 330 provides a mechanism to organize a set of resources and users as a group. A MetaObject component object 332 represents meta information for objects that can be authored in the system, and comprises a container that includes a representation for authoring control flow.

FIG. 3B is a diagram of a containment hierarchy for objects that define a transportable application. In general, an Application Container object 326 (FIG. 3A) represents a transportable application and contains one or more building blocks and optionally other containers. For example, a containment hierarchy 350 (FIG. 3B) may comprise a first ApplicationContainer 326A that comprises first and second building blocks 320A, 320B and a second ApplicationContainer 326B. The second ApplicationContainer 326B further contains additional building blocks 320C, 320D.

Each ApplicationContainer provides services to its contained Building Blocks. When a Container contains other Containers, the hierarchy is represented as a tree structure, as in FIG. 3B. A Container acts as a coordinator for its contained building blocks and interacts with them in order to store user inputs and responses and also for getting the presentation data of the building blocks. Similarly, a Container interacts with its contained Containers in order to propagate certain actions such as closing a transportable application. When a transportable application is closed, all its contained transportable applications are also closed.

Once a Container has been created, new building blocks and containers can be added to it. A Container provides necessary APIs for adding the building blocks and containers and also for interacting with them. Table 1 presents an example class definition of a container.

TABLE 1—CONTAINER CLASS

```
public class ZapletContainer{

     void ZapletContainer(ZapletRequest request);

     void          addBuildingBlock(ZapletRequest
request);
     void          addChildZaplet(ZapletRequest
request);

Vector          getChildren();
ZapletData      getZapletData();
Zaplet               getParent();
Zaplet               getChild(ObjectId zapletId);
Zaplet               getRoot();

     void          handleResponse(ZapletRequest
request,
ObjectId userId);
     void          handleEditResponse(ZapletRequest
request,
```

```
      ObjectId userId);
      void            handleEditCreate(ZapletRequest request,
      ObjectId userId);
           PresenterData   getDisplayData(ZapletRequest
request);
      PresenterData   getSubmitData(ZapletRequest request);

      void            sendReminder(String reminderMessage);
      void            forward(String forwardMessage, String
toList);
      void            close(String closeMessage);
      }
```

Using the API defined by Table 1, an *addBuildingBlock* method is invoked to add a building block to a transportable application. An *addChildZaplet* method is invoked to create a child transportable application. A *getChildren* method returns the child transportable applications for a given transportable application. This method returns the transportable applications that are immediate children of a given transportable application. A *getZapletData* method is invoked to retrieve data relating to a transportable application.

A *getParent* method returns the parent transportable application of a given transportable application. A *getChild* method returns the specified child transportable application. A *getRoot* method returns the root transportable application in the hierarchy.

A *handleResponse* method is invoked to handle a user response to a transportable application. An *editResponse* method is invoked to edit an already submitted response, provided the response can be edited. An *editDefinition* method is invoked to edit the definition of a transportable application. A *getDisplayData* is invoked to get the presentation data for the transportable application. A *getSubmitData* is invoked to get necessary data to prompt the user to submit a response to a transportable application.

A *sendReminder* method is invoked to send a reminder message to the participants of the transportable application. A *forward* method is used to forward a Zaplet to a set of recipients specified by the 'toList' parameter. A *close* method is used to close the Zaplet for any further interactions.

1.2.7 Membership

Users of service unit 305 are organized in one or more groups. Each group is represented by a programmatic object in a membership hierarchy. Each group may have one or more child groups or users. Group and user information may be imported from an external directory service, such as an LDAP directory, or may be defined in the system database. Groups may be collections of individuals in a department, project team, or other group with a common goal. Groups provide an archive for the shared knowledge of people who work together. Groups also provide convenient distribution lists for addressing transportable applications.

Membership concepts are programmatically represented by folder objects and group objects. A membership group is a collection of other membership subgroups and users. A folder is a collection of other subfolders and meta-objects. By definition, a group is also a user, and a folder is also a meta-object. Each object in a membership hierarchy comprises a reference to an underlying object or user. For example, when a particular task is saved in two different folders, two references to the same task (one for each folder) are stored in the hierarchy.

FIG. 4 is a block diagram of an example of a membership hierarchy. An "Everyone" group 402, represented by a group object, contains an Engineering group 404, users 406A, 406B, a transportable application folder 410, and a meta-object folder 412. The transportable application folder 410 contains transportable applications 410A, 410B. The meta-object folder 412 contains tasks 412A, 412B. Engineering group 404 contains users 406C, 406C. The example illustrated in FIG. 4 may be modeled using the NodeObject interface, CollectionNodeObject interface, UserObject interface, FolderObject interface, MetaObject interface to represent one or more authoring tasks or templates, and GroupObject interface.

1.2.8 Building Blocks—Programmatic Aspects

Building blocks are functional units of a transportable application. Each building block encapsulates business logic, data and presentation. A BuildingBlock object is always be contained in an ApplicationContainer object. Server processes provide services to building blocks. An ApplicationContainer invokes various methods on a BuildingBlock object to process various user actions such as creating a transportable application, responding to a transportable application or editing a transportable application and so on. A BuildingBlock can also handle asynchronous events.

Each building block implements the BuildingBlock interface of FIG. 3, which defines methods for storing data for building blocks, storing responses for the building blocks and also for providing the necessary data for presentation. The presenter data that

is returned from a building block can be presented in different formats, as described below. In one specific embodiment, the BuildingBlock interface is defined as set forth in Table 2.

TABLE 2—BUILDING BLOCK INTERFACE EXAMPLE

```
public interface BuildingBlock{
void            handleCreate(ZapletRequest request,
                             BuildingBlockDAO bbDAO);
void            editCreate(ZapletRequest request,
ObjectId userId);
     void            handleResponse(ZapletRequest
request,
ObjectId userId);
     void            editResponse(ZapletRequest request,
ObjectId userId);
void            handleEvent(EventData eventData);
     PresenterData   getDisplayData(ZapletRequest
request);
PresenterData   getEditResponseData(ZapletRequest
request);
}
```

In this embodiment, a *handleCreate* method is invoked to handle creation of the building block data in the database 208. An *editCreate* method is invoked to edit building block data in the database 208. A *handleResponse* method is invoked to create a user response in the database 208. An *editResponse* method is invoked to handle editing of an already created user response. A *handleEvent* method is invoked to handle an event by the building block.

A *getDisplayData* method is invoked when the building block data needs to be displayed in a transportable application. A *getEditResponseData* method is invoked when the response data needs to be displayed in a submit dialog box for editing.

In one embodiment, a BuildingBlockMetaData interface specifies methods on a building block meta-data object. Such methods are responsible for providing the necessary information in order to construct an authoring snippet or a response dialog snippet, related to the building block. In one specific embodiment, the BuildingBlockMetaData interface is defined as set forth in Table 3.

TABLE 3—BUILDING BLOCK METADATA INTERFACE EXAMPLE

```
public interface BuildingBlockMetaData{
     Vector          getAuthoringParams();
     Vector          getResponseParams();
     Vector          getConditions();
     Vector          getActions();
}
```

In this embodiment, a *getAuthoringParams* method returns a vector of authoring parameters, which can be used by a tool to construct an authoring snippet for the building block. Each element of the vector is of BuildingBlockParam type. A *getResponseParams* method returns a vector of user response parameters, which can be used by a tool to construct a user response dialog box for the building block. Each element of the vector is of BuildingBlockParam type. A *getConditions* method returns a vector of conditions that a building block can evaluate, which can be used by a tool to define a set of conditions for the building block. Each element of the vector is of ZapletCondition type. A *getActions* method returns a vector of actions that a building block can perform, which can be used by a tool to specify a set of actions for the building block. Each element of the vector is of ZapletAction type.

When a Building Block is invoked to either to create its data in the database or store user responses or retrieve its data from the database, it uses a Data Access Component (DAC) to access the database. The DAC comprises programmatic objects that enable building blocks to access data in database 208 such that developers of the building blocks do not need to know the specific schema of the database, and such that the building blocks are not affected by any future changes in the schema. Further, the DAC provides, in effect, an API that enables a developer to store building block data in database 208 without knowing the schema.

In one specific embodiment, the DAC comprises data access objects and data objects. Data access objects serve as a channel to write and read data from the database 208. Data access objects encapsulate knowledge about objects and tables in the database 208 that are acted upon. Data access objects provide methods to accomplish data persistence and data retrieval queries. Data access objects can carry out access control checks on the data that is being accessed, in cooperation with the security services described herein, to ensure that a particular user or building block is authorized to retrieve selected data.

FIG. 15 is a block diagram that illustrates data and control flow among building blocks, data access objects, interface methods and the database. One or more building blocks 292, 294 communicate with data access objects 1500. Each building block 292, 294 is associated with a corresponding building block data access object 1502 that stores metadata and data used in program logic relating to the building block. Such metadata may include a database identifier, building block identifier, transportable application identifier, etc. The data used in program logic may include scalar values that determine behavior of specific building blocks. For example, for a Poll building block, the data used in program logic may indicate a chart type value that specifies what kind of chart is used for displaying poll results.

Data other than building block data may be stored in database 208. Accordingly, other types of data access objects are provided based on the kind of data that is stored. In one embodiment, data access objects comprise choice data access objects 1506, response data access objects 1508, etc. Each data access object encapsulates information about the underlying database tables by working with interface objects. For example, response data access objects provide a generic public interface that allows building blocks to store any responses from the recipients of a transportable application in the database. An example of such a response is a recipient's response to a poll question.

Each building block data access object provides access to tables of database 208 through interface objects. In one embodiment, the interface objects include a QuestionInsert object 1504, QuestionXMLInsert object 1510, QuestionResult or ResponseInsert object 1514, and QuestionXMLResult or ResponseXMLInsert object 1516. The interface objects communicate with database 208 through one or more stored procedures 1512. In one embodiment, the stored procedures act, in part, to convert user data received from the interface objects to XML format for storage in database 208; metadata that is not received from a user-created building block, or otherwise internally generated, is stored in conventional database columns. A pre-defined method may be used to store any building block-specific programmatic object as a string value in the database. For example, in one embodiment, the method setFieldValue(fieldName, value) is used to store a Java object as an XML string in the database.

A retrieval method is provided to facilitate retrieval of a data object that corresponds to a building block data access object. Data objects are created when data is retrieved from the database 208. Thus, data objects facilitate the flow of results from the database resulting from any query that is executed by a building block or any other component such as a presenter. Each data object is a wrapper around result objects such

as the QuestionResult object, and QuestionXML result object. Each type of data access object has a corresponding data object to capture results from the database tables. A pre-defined method may be used to retrieve an object that was stored in the database using the corresponding data access object. For example, in one embodiment, the method getFieldValue(fieldName, value) is used to retrieve an object that was stored in the database using the corresponding data access object. Any information that was stored in XML form by a data access object can be retrieved from the database, parsed by an XML parser, and then the query results can be returned on demand. Thus, a data object is a result object that any component in the system can retrieve by providing appropriate input parameters.

Each building block data access object has a corresponding building block data object that maps to the result object QuestionXMLResult. Each data access object provides a reference to a corresponding result object in the database or a cache thereof.

Using this mechanism, building block authors and developers can focus on business logic and program logic that is implemented in a building block rather than on the database schema, because the DAC separates such logic from the database schema by interposing an abstract layer. Accordingly, the system becomes more extensible because creating new building blocks is simplified. Further, dependencies between building blocks and the database are removed. The building blocks are insulated from any major changes that may occur in the database schema.

Graphical user interface elements associated with groups, and processes for joining and using groups from the user perspective, are described herein in the section entitled Personal Portal Presentation.

1.2.9 Directory Integration—Global Object Identifiers

In one embodiment, each transportable application is identified by a global identifier. A transportable application is indirectly transported to each named recipient by sending a message, such as an electronic mail message, to each recipient that contains a reference to the transportable application in the form of a URL that includes the global identifier. When the user opens the message, the URL is invoked, causing the user's browser to retrieve the dynamic content and other elements of the transportable application.

In one approach for implementing the foregoing, the global identifier has the following format: <AppserverURL>/Z?m=<globalId>

where "<AppserverURL>" is the URL of an application server, such as "http://www.zaplet.com," and "<globalID>" is the global identifier. In one specific

embodiment, the global identifier value has the format DbId_ObjTyp_ObjId, where "DbId" is a value that uniquely identifies a cluster. The DbId value may numeric or may be a string, such as a URL of an application server. When a numeric value is used, then a mapping of numeric values to URLs is stored in a directory. Using a URL requires no such mapping, but increases the length of the global identifier value. Length may be a limiting factor in certain encryption schemes, such as TwoFish. The ObjTyp and ObjId values refer to the type and the local identifier of the object in the cluster. The entire global identifier value is encrypted using a site-specific seed, as described further herein.

In an alternative implementation approach, directory-based access to processing clusters and other objects is used rather than direct reference to clusters as provided in the above approach. In a directory-based approach, objects are referenced using distinguished names that are registered in a directory. The directory may be an X.500 directory, LDAP directory, or similar repository that is associated with database 208. For example, Microsoft Active Directory may be used.

In this approach, each processing cluster is uniquely identified using a distinguished name ("DN"). The DN maps, in a directory, to a URL or other specific identifier of the cluster. There could be additional information in the directory that could be referenced. As defined in the X.500 standard, DNs are elements of a hierarchical naming system in which each DN has <name,value> attributes that are used to traverse a directory hierarchy or tree. For example, the DN "C=US, O=FireDrop, OU=Engineering, HN=alpha" identifies a path in a directory hierarchy to the entry for machine alpha in the Engineering dept at FireDrop which is in the US. That directory entry may contain the URL of an application server that is on this machine. A DN can be transformed into a numeric value using ASN.1 encoding.

Each cluster may be associated with a different enterprise, institution, or other organization. For example, clusters may be located at different competitive parties. Clusters that cooperate with each other need to have a trust relationship, which may be established through a third party security organization. Based on the trust relationship, each party that owns or operates a cluster can set up information for its peers in its own directory that is then used to construct a global identifier value. Since each directory structure is arbitrary and may be determined by the organization, the global identifier for an object also is site dependent. Thus, there may be many valid global identifiers for a single object. However, a benefit of the DN approach is that the DN is an indirect reference to the site information. One site's DN cannot be meaningfully applied to another site, thereby preventing improper access to a cluster from a user associated with a

different site. Furthermore, directory level authentication must be successful in order to retrieve site information.

1.3 Presentation of Transportable Applications

1.3.1 E-mail Client Presentation

In one embodiment, client 102 executes a client application program that facilitates authoring, viewing, interacting with, and managing one or more transportable applications. In this description, such a program is termed a "transportable application client" or, where it is clear from the context, "client." Authoring refers, for example, to creating a transportable application; creating a transportable application based on a template; integrating address information from an address book with a transportable application; converting an existing transportable application to a discussion-type transportable application; creating one or more transportable applications that implement a business process; and similar tasks. Viewing refers, for example, to viewing a transportable application; viewing one or more transportable applications that implement a business process; and viewing transportable applications when the client 102 is disconnected from a network ("off-line viewing"); and similar tasks. Interacting refers, for example, to interacting with a transportable application; interacting with one or more transportable applications that implement a business process; interacting with transportable applications when the client 102 is disconnected from a network; viewing one or more notifications; and similar tasks. Management refers, for example, to moving transportable applications into and out of folders and groups, and similar tasks.

In one approach, the transportable application client interacts with one or more electronic mail client programs ("e-mail clients"). The e-mail clients may comprise Microsoft Outlook 98, Outlook 2000, Outlook Express, Lotus Notes, Eudora, Netscape Messenger, etc. Typically one of the foregoing is installed in client 102 and the transportable application client interacts with it using shared dynamic linked libraries (DLLs), public APIs, or other programmatic communications mechanisms. The transportable application client also interacts with a network browser such as Microsoft Internet Explorer, Netscape Navigator, etc., as well as with the operating system of client 102, which may be Microsoft Windows 95, Windows 98, Windows NT, Windows 2000, etc.

Specific functions of an embodiment of a transportable application client are now described.

In one specific embodiment, a client is provided with an automatic installer that can copy components of the client from an installation media to a client computer system

(e.g., client 102), and integrate such components with the operating system, browser and e-mail application. The installation media may be a CD-ROM, file server, Internet download, etc. Alternatively, users may receive an HTML email containing a signed ActiveX control, or equivalent program element, that initiates installation of the client. Installation may involve storing configuration data in a registry of the operating system of client 102.

The client is provided with an automatic start configuration so that the client initiates execution when a client computer, e.g., client 102, starts operation. Thereafter, the client may be stopped and restarted repeatedly by the user. After startup, the client generates an indicator icon that is displayed in an operating system "tray" location. When started, the client initially enters an inactive state. The client activates in response to a log-on process carried out by the user. When the client is activated, the indicator icon changes appearance to indicate that the client is in the active state. To deactivate the client, the user may either log off or stop the client, at which time the indicator icon reverts back to its Started state.

In the log-on process, the client prompts the user for a user identifier and password and then authenticates the identifier and password against the authentication database. The client provides an option to save the password for the user in an encrypted format; userids and passwords are saved as a value pair. Further, the client provides an option to log on automatically when the client is started based on a saved password. Using this option, the last user who logged off is logged on at startup, provided a saved password is available.

When a user receives a notification in responses to receiving a new transportable application, an updated transportable application or changes to the status of a transportable application, the indicator icon changes appearance to indicate that a pending notification exists. Optionally, under control of the client, client 102 may generate a sound or play an audio file to announce the receipt of a notification.

When a user hovers a mouse or similar pointing device over the transportable application indicator icon, the client displays the number of new and updated transportable applications. Selecting the indicator icon causes the client to display the title of each new or updated transportable application and enables the user to select a transportable application for viewing.

Also in one embodiment, the client notifies the user when new or updated software components become available. For example, the client indicates the availability of software updates with a flashing indicator icon. When requested by the user, the

installer function of the client downloads and installs the applicable new or updated components.

In another feature, the client enables the viewing of transportable applications off-line by creating a snapshot of transportable application dynamic data in a local cache. When the transportable application is viewed off-line, the client redirects and resolves data references using the local cache. In one configuration, the client maintains a persistent connection to the application server 202 and uses unused bandwidth to download data for the transportable application to the local cache. Only incremental data is downloaded. The download interval, download data size and cache size are configurable.

A user may use the e-mail client executed by client 102 to respond to transportable applications. In this configuration, the client overrides the Reply, Reply to All, and Forward buttons or functions of the e-mail client. In addition, each transportable application may be addressed using the e-mail client. For example, recipients may be selected using a pre-configured search order of various information sources, such as a corporate directory, personal contacts, aliases, and groups. When a transportable application is sent, recipient names are resolved to an address that can be processed by an e-mail server or a similar transport mechanism. The client may provide its own address book and the user may add recipient names to it.

The client can indicate changes in status of a transportable application and updates in the e-mail client. A previously read transportable application may be marked as unread if new responses have been posted. Such notifications should only affect email found in the user's Inbox. If the email has been moved to another folder, these notifications do not occur.

The client communicates with application server 202 to determine if a notification should be sent to the user. The client notifies the user when new or updated templates for transportable applications become available. When requested by the user, the installer downloads installs only the applicable templates.

1.3.2 Personal Portal Presentation

According to one embodiment, transportable applications are accessible through a personal portal. Generally, a personal portal is represented by a graphical user interface window that contains one or more hyperlinks for accessing system functions. The personal portal provides membership services, including defining groups and managing group membership. The personal portal displays organized lists of transportable applications that the user of the portal has received, organized according to group and

individual user. Thus, the portal provides an organized entry point through which a user may create, organize and send transportable applications and access tools and services for doing so. The portal provides secure access to transportable applications through user authentication processes.

In one embodiment, the personal portal is defined as and implemented using a container and a set of building blocks. Thus, each personal portal is itself a transportable application like others in the system. In this embodiment a user receives the personal portal through a transport mechanism, such as HTML-enabled e-mail, for example. The user opens the message and receives a view of an initial page ("home page") of the portal by reading the transportable application. In this alternative, selecting links in the home page of the portal causes the system to generate new screen displays and carry out other functions within the e-mail window. Thus, the user may connect to the system and interact with all its functions without launching a browser or other separate application.

In this embodiment, the portal comprises a transportable application having two or more Listing building blocks, by default. The first listing building block is an In-Box that lists all transportable applications that have been received by the user who is viewing the portal. The second listing building block comprises a list of links to authoring functions, to assist users in rapidly accessing functions of the application editor and page editor disclosed herein. In an enterprise having a plurality of departments, there is a portal transportable application for each department; each such application comprises a page for each user group in the department. A page for all users is provided. Group moderators may add building blocks to the group pages. Groups may be nested within other groups.

Alternatively, the portal is one or more software applications, represented by a plurality of graphical user interface windows, which are served or operated by a service unit, such as service unit 105. In general, a portal is associated with a business enterprise that owns, operates or hosts system 200. The portal is accessed using a browser executed at a client computer.

In each embodiment, the portal provides access to an Application Builder application and a plurality of templates for creating transportable applications. The portal provides each user with a personal archive of transportable applications that they have sent and received, and provides users with access to groups that provide a repository of shared knowledge regarding transportable applications. In one specific embodiment, the portal home page comprises the following links: Group Directory; My Groups; Personal Messages; Search; Builder; Preferences.

To join a group, a user selects the Group Directory link. A group directory screen appears that lists groups alphabetically by name, shows a description of each group, shows how many messages regarding transportable applications were sent to a group in the past seven days, and shows the group membership status value. If the group is open to everyone, the status value is Join Group. If the group is open by invitation only, the status value is Private. If the current user is a member of a group in the list, the status value is Member. The user then selects either the Name value or the Status value, both of which are hyperlinks, of the group that the user wishes to join.

In response, a Join Group screen is displayed. The form of the Join Group screen varies depending on whether the selected group is open to everyone or private. If the group is open to everyone, then the Join Group screen comprises a Join button and a Cancel button; a user may join the group by selecting the Join button. FIG. 9A is a diagram of the Join Group screen that is displayed when the selected group is private. In this configuration, Join Group screen 900 includes a group name 902, group description 904, a private group label 906, an email link 908, and a Close button 909. To attempt to join a private group, the user may select email link 908. In response, an email message is automatically sent to the moderator or "owner" of the group to request access for the user. Alternatively, the user may select the Close button 909 to close screen 900.

FIG. 9B is a diagram of a My Groups screen that is generated and displayed when a user selects the My Groups link of the portal home page. My Groups screen 910 comprises function links 912, an Application Shortcuts area 914, an Updated Messages area 916, a My Group Activity area 918, and a New Groups area 920. In the function links 912, a Group Directory link 922 displays a list and description of all groups. A Personal Messages link 924 accesses all transportable applications sent to of from the user, including group transportable applications and transportable applications that the user has created for personal use.

The Application Shortcuts area 914 provides rapid access to certain applications. The Updated Messages area 916 provides a list of updated transportable applications for which the user has turned notifications on. The My Group Activity area 918 provides a list of group transportable applications that have been received in the last seven days. The New Groups area 920 provides a list of recently added groups.

The function links 912 further comprise a My Groups link 934 that provides access to detailed functions pertaining to groups of which the user is a member. One or more group links 934A are displayed in a hierarchical tree in association with the My Groups link 934. A user may select one of the group links 934A to carry out specific

operations relating to a group. In one specific embodiment, the group operations include: invite a member; membership list; unsubscribe; contract moderator; group description; and view recent group activity.

In the invite member operation, a user can invite another user to join the selected group. An Invite Member screen is displayed. The user enters one or more addresses of users or groups to invite to join the selected group. The user also enters a short introductory message. In response, the system generates and sends a message to each of the named users inviting them to join the selected group, and including the introductory message. If the currently selected group is a private group, the user must send an e-mail to the group moderator to request additional members.

In the membership list operation, a user can view the group membership list. A membership list screen appears, listing members by email address, name and the date on which they joined the group. In the unsubscribe operation, the user can leave or "unsubscribe" from the group. In the contract moderator operation, the user can send an e-mail to the group moderator. In the group description operation, the user can view the description of the group. In the view recent group activity operation, the user can view the ten most recently updated transportable applications for the group.

FIG. 9C is a diagram of a portion of the screen display of FIG. 9B. FIG. 9C illustrates function links 912 when a user has selected one group name link 934A. In response, an Applications link 936 and an Archive link 938 are displayed in a position hierarchically subordinate to the group name link 934A. A member of a group can use any transportable applications created for that group, or create new transportable applications for it. To access the applications, the user selects the Applications link 936. In response, a list of links to applications is displayed in the My Groups screen 910. The user may launch an application by selecting the associated link. To create new transportable applications for the group, the user uses the Application Builder, as described further herein.

The user may also display all transportable applications sent to the group in the group archive by selecting the Archive link 938. In one specific embodiment, the group archive lists all transportable applications sent for the group. The user may filter the list of transportable applications by application type using a drop-down menu. The user may sort the archive list by Author, Subject, application type, date created, or date last modified.

The Personal Messages link 924 provides access to transportable applications that have been sent to the user, and to transportable applications that the user has created for

personal use. To access such applications, the user selects the Personal Messages link 924, and then selects an Applications link that is displayed. The Search link 926 provides access to a search engine that can search the full text of all transportable applications by keywords.

The Preferences link 930 is used to change a user's profile of preferences, to thereby tailor the way that the user interacts with groups and transportable applications. Changes to preferences may be made by the user at any time. Through a Preferences dialog, the user may make changes to a user Profile, Address Book, Group Preferences, and a client or Plug-in. To facilitate such changes, the Preferences dialog includes a Profile link, Address Book link, Group Preferences link, and a Plug-in link.

Using the Profile link, the user can change information in the user profile, such as name, password, postal code, time zone, message format, and whether message recipients are automatically added to the Address Book. The message format profile value enables the user to obtain a preferred user experience using the e-mail client. For example, if the user's e-mail client does not accept HTML, the transportable applications received by the user may appear to be "broken." The user can change the format in which transportable applications are delivered to the user. In one specific embodiment, available formats include Plain Text and HTML.

The Address Book is a personalized database of contact persons for sending transportable applications. In one embodiment, selecting the Address Book causes the system to display a list of available address books (Personal, Company, etc.). The user can select one of the available address books and search for contact data within it. The user may add or modify entries or important entries from other programs.

The Group Preferences link may be used to control how incoming transportable applications from groups are delivered. In one embodiment, the user may select a Web Only option, with which transportable applications are delivered to the portal only, or an Email & Web option, which provides delivery of transportable applications both to the portal and the e-mail in-box of the user.

The Plug-in link provides access to a client software element that can execute on a user client, e.g., client 102. By selecting the Plug-in link, the user can install, uninstall, and view the status of the Plug-in. In operation, the Plug-in automatically informs the user when updated transportable applications have been sent, using a distinctive icon that is shown in a designated portion of the graphical user interface. For example, in a Windows system, the distinctive icon is shown in the system tray. Also, in another embodiment, if the client computer is configured to access messages when disconnected from the

network, the user can use the client to synchronize transportable applications for offline viewing and replying. Thus Plug-in may be implemented in the form of C++ code that is optimized to the Windows environment. Additionally, in an embodiment, the Plug-in adds new buttons to the toolbar of the user's messaging client. For example, a New Appmail button is provided in an e-mail client so that the user may initiate a process of creating a transportable application from within the e-mail client. The user may also create a new e-mail message and send it in the form of a transportable application by selecting a Send As Appmail button.

In certain embodiments, the Group Directory, My Groups, and Group Archive functions each are implemented as separate transportable applications having containers and listing building blocks.

To facilitate the foregoing operations, portal user roles and group user roles are defined in the security framework that is described herein. In one specific embodiment, a portal Administrator acts as portal editor, has all Coordinator permissions, has Owner control for all groups in the portal, can assign individuals to roles, and can create new roles. A Coordinator can create a group, delete a group, and exercise all User permissions. A User can view the group directory, join a group if it is public, and use the Application Builder and Page Builder that are linked to the portal.

1.3.4 Programmatic Processes for Presentation

FIG. 5 is a block diagram of objects and data communications paths involved in presentation of content associated with a building block. In general, a transportable application container object 502 references and contains a building block 504.

When the building block 504 is invoked by the container object 502 to either to create data in the database 208 or store user responses or retrieve its data from the database, the building block uses Data Access Components (DAC) to access the database. In order to enable presentation of its information, the building block returns a Presenter Data Object (PDO). Each PDO contains all data of a building block that can possibly be displayed on any device in any form, and all associated interaction pathways. A building block presenter object (termed a "presenter" herein) uses the PDO in order to do the actual presentation. For example, HTML presenter 506 can present the data in the PDO in HTML format. Alternatively, a WML presenter 508 can present the data from the PDO in WML format to a wireless device gateway. This mechanism allows presentation of the same PDO in different formats such as HTML, WML, XML, etc.

Presenter data objects as disclosed herein enable reuse of presenters, separate responsibilities of functional elements of the system, enable independent development of

building blocks and presenters thereby enhancing modularity and simplifying development, and provide flexibility to handle custom as well as generic presenters. The presenter data objects provide an abstract interface that mediates communications of building blocks and presenters.

FIG. 6 is a block diagram illustrating interaction of programmatic objects in one embodiment of a presentation model. In this embodiment, each Building Block 182C, 182D produces a specific PDO 602A, 602B that contains all the possible data of that Building Block.

Two types of presenters can be used by a Building Block for presentation: Generic Presenters, and Custom Presenters. Generic Presenters work with any PDOs that implement a certain interface that they require. A Custom Presenter has detailed knowledge about the PDO, and can work with only that specific PDO. A PDO that does not implement a certain interface known by a Generic Presenter can still use that presenter by means of a Converter that maps the specific PDO into an interface that is understood by the Generic Presenter. In the embodiment of FIG. 6, generic presenters include a table presenter 606B that can present a table of data, and line presenter 606D that can present a line graph, etc. Custom presenters include a poll presenter 606A that can present data for a user poll, and a schedule presenter 606C that can present a calendar or schedule.

Each PDO encapsulates data objects that represent the building block data, provides methods to access all the possible building block data, and may implement formatting interfaces like table interface, chart interface etc.

Presenters are responsible for rendering the data, and may use generic rendering utilities. Generic presenters work with a basic PDO that implements a certain interface. A custom presenter may directly work with a specific PDO. Each presenter implements a BuildingBlockPresenter interface.

In one alternative, a building block may pass a PDO to a converter 604A, 604B that converts data of the PDO to a different format that is understood by an interface of the custom presenter. For example, in situations where a specific PDO needs to work with a generic presenter, an object that implements an interface known to the generic presenter is constructed. Such converter objects map a specific PDO to this known interface. Each converter works with a PDO and a corresponding presenter. In one embodiment, a converter implements a ConvertInterface that produces an object that implements the interface known to the generic presenter.

Table 4 provides an example of a code implementation of the objects of FIG. 6.

TABLE 4—CODE FOR PRESENTER ABSTRACT LAYER ELEMENTS

```
public class PollPDO {
        //data
        public int getTotalVotes()
        {
        }

        //get details about each voter.
        public String getVoterEmail( int i)
        {
        }

        public String getVoterName( int i)
        {
        }

        public String getLastChoice( String email )
        {
        }

        // details about the choices.
        public Vector getChoiceLabels()
        {
        }

        // return a vector of Integers that represent choice
totals.
        public Vector getChoiceTotals()
        {
        }

}
```

```
    public class PollPresenter implements
BuildingBlockPresenter
    {
        PollPDO pdo = null;
        public PollPresenter( Object pdo )
        {
            this.pdo = (PollPDO)pdo;

        }

        public void PresentLive( Writer out,
HttpServletRequest req )
        {
            // give it to some utility or draw it inline.
            chartUtils.drawPie(out, getChoiceLabels(),
getChoiceTotals() );
        }

        // other methods in BuildingBlockPresenter.
    }

    interface TableInterface
    {
            int getNumRows();
            int getNumColumns();
            String getColumnName(int);
            Class getColumnClass(int);
            Object getValueAt(int, int);
            // possibly more methods
    }

    public interface ConvertInterface
    {
        public Object convert(Object sourcepdo);
    }
```

```
    // presents a table representation of the poll.
    public class PollTableConverter implements
TableInterface, ConvertInterface
    {
        PollPDO pdo = null;
        String columnNames = {"Voter Name", "Voter Email",
"Choice" };
        public Object convert( Object pdo)
        {
           this.pdo = pdo;
           // do some other housekeeping.

           return this;

        }

        Object getValueAt( int i, int j )
        {
            if ( j == 0 )
            {
                return pdo.getVoterName(i);
            }
            else if ( j == 1 )
            {
                return pdo.getVoterEmail(i);
            }
            else if ( j == 2 )
            {
                return pdo.getVoterChoice(i);
            }
            else
                return null;
        }
        int getNumRows()
        {
            return pdo.getTotalVotes();
```

```
        }

        int getNumColumns()
        {
            return 3;  // oh no, hard coding
        }

        int getColumnName( int i)
        {
            return columnNames[i];
        }

        Class getColumnClass( int i)
        {
            // everything is a string. Also need to catch
exceptions.
            return Class.forName("java.lang.String");
        }
    }
```

Using these structures, when a building block needs to be presented, control logic of application server 202 requests the building block to construct the building block specific PDO. The control logic also contains knowledge about the specific presenter to be used to display the building block e.g. whether to use a custom presenter or a generic presenter. Such information is obtained at the time that a building block is authored.

In case of a custom presenter, control logic of application server 202 instantiates the presenter with the specific PDO obtained from the building block. In the case of a generic presenter, the control logic uses a converter, if necessary, to convert the PDO into a format known to the generic presenter, before constructing the generic presenter. It then passes the presenter to a Java Server Page (JSP) that controls overall presentation of the Zaplet. The JSP uses BuildingBlockPresenter interface of the presenter to render the data. Table 5 presents a pseudocode implementation of the foregoing process.

TABLE 5—PSEUDOCODE FOR PRESENTATION PROCESS

```
//iterate through different bldg blocks
```

```
    for ( int i=0; i < numBB; i++ )
    {
        Object pollpdo = bb.getPresenterData(request);
        BuildingBlockPresenter pr = null;
        String presenterName = getPresenterName(bb);
        String converterName = getConverterName(bb,
presenterName);
        if ( converterName != null )
        {
            ConvertInterface i =
ConverterFactory.createConverter(converterName);
            Object tablepdo = i.convert(pollpdo);
            pr =
PresenterFactory.createPresenter(presenterName, tablepdo);
        }
        else
            pr =
PresenterFactory.createPresenter(presenterName, pollpdo);

        presenterVector.add(pr);
    }

        forwardRequestToSomeJSP(presenterVector);
```

A building block may use more than one presenter. For example, displaying results from a poll building block may use a pie chart presenter or a table presenter. To facilitate this possibility, an independently authorable component is provided, comprising a combination of a building block and the set of presenters corresponding to the actions supported by the building block. Thus, a poll building block with a pie chart presenter for display comprises a first independently authorable component, and a poll building block with a bar graph presenter for display comprises a first independently authorable component.

When development of a new building block is complete, it is registered in the system. Each independently authorable component is registered in the system separately. Registration involves providing the system with metadata about the new building block,

such as its name, actions it carries out, events that it generates, and types of devices to which it can present data. When a building block is registered, it is displayed to and can be selected by authors of pages and transportable applications when they use the application editor and page editor as described herein.

FIG. 16 is a flow diagram illustrating a process of presenting building block information using the foregoing mechanisms. In block 1602, a request to present a view is received at the application server. The request may comprise an HTTL request or an internal programmatic function call. In block 1604, the request is forwarded to a container that contains the building block associated with the view. In block 1606, the container forwards the request to either an individual building block, or to all the building blocks contained in the container, based on the type of container.

In block 1608, the container invokes an appropriate building block method for the action. In block 1610, a presentation data object is received from the building block method. In block 1610, based on information collected at registration time, the application server forwards the presentation data object to the presenter that was registered for the current action. The system then invokes a presentation method on the presenter.

Actions, in this context, may comprise standard user actions or building block specific actions. Standard user actions are internally known actions that trigger a container to call on all its contained building blocks with a request to participate in such actions. For example, when a transportable application is to be displayed in a browser of client 102, the container requests each of its building blocks for a presentation data object for a "get display data" action. All building blocks that wish to participate in such action return a non-null presentation data object, which are given to the corresponding presenters for rendering.

Building block specific actions, such as a sort action relating to table data in a table building block, originate form the presentation of standard user actions. For example, a sort action originates when a user selects a link in a table building block data that has been presented. The target URL for the action contains enough information for the system to route the request to a specific building block. The building block then responds with a presentation data object that is given to the appropriate presenter.

1.4 Authoring Transportable Applications

FIG. 7C is a flow diagram illustrating a high-level view of a process of authoring a transportable application.

Optionally, in block 720, a user logs on to a portal, such as a portal of the type described herein in the section entitled "Personal Portal Presentation." The portal home page provides a plurality of menu options in a navigation bar. In block 722, a New Message option is selected from the navigation bar. Alternatively, rather than carrying out the steps of block 720, block 722, a user can select a New Message option from within any existing transportable application. In response, the system displays a list of template folders. In block 724, the user navigates to a template folder or subfolder that contains a desired template. The user then selects a desired template, as in block 726. In one embodiment, an Application Shortcuts list displays frequently-used templates, and a user may select a template directly from the Application Shortcuts list rather than navigating to a folder.

For purposes of illustrating an example, assume that a user selects a Starter Set folder of templates, a Corporate Process sub-folder of templates, and an Escalation Exception Manager template. In response, the template is displayed, as shown by block 728.

FIG. 8A is a diagram of an example graphical user interface screen display of the Escalation Exception Manager template. In general, screen display 800 comprises a Customer Case Summary page 802A, Manage Case page 802B, and Addressing page 802C. Customer Case Summary page 802A comprises a message header 804, page header 806, attachments area 808, and discussion area 801.

The user enters information in the message header 804. In this example, information is required in none of the fields of the message header, but such information may be required in other templates. The user also enters information in the page header 806, including Page Title and Introduction. The user then enters values for the rest of the fields that are required, attaching files and adding descriptions as needed. Depending on the building blocks in each page, the user may have to upload files of the appropriate type, upload pictures, specify URLs for Web pages, and other tasks.

When all such information is entered, the user selects the Next button 808 to advance in the authoring process. The system displays the next page 802B, 802C and receives further input until the template is completed. The user may select a Preview Page button 810 to receive a view of what the transportable application will look like to recipients.

When the last page of a template is reached, the page displays an Add a New Page button and an Address and Send button. The user may add optional pages to the

transportable application by selecting the Add New Page button. The optional steps of removing and adding pages are represented by block 732 of FIG. 7C.

Referring now to FIG. 7D, in block 734, the transportable application is addressed. The addressing operation may be initiated by selecting the Address and Send button. In response, the system displays Addressing page 802C. FIG. 8B is a diagram of a screen display of an Addressing page that may be displayed, in one example embodiment. In this example, Addressing page 802C comprises an address field 822 that can receive one or more individual e-mail addresses in a comma-separated list. Alternatively, the user may select an address book link 824 and select the recipient e-mail addresses from a stored list of addresses. Additionally or alternatively, the user may select one or more groups of recipients from a list 826 of groups. Selecting a named group in list 826, followed by selecting an Add button 830, causes the system to copy the group name to a Send To list 828 that is used when the transportable application is sent.

Also as part of addressing, the user enters subject text in a Subject field 832.

Optionally, the user may apply access controls or other security controls to the message or its pages by selecting a Message Security Options link 834 and Page Security Options link 836. If the Message Security Options link 834 is selected, then in response, the system displays a Message Security Options window. FIG. 8C is a diagram of an example of the Message Security Options window that may be displayed, in one example embodiment. Message Security Options window 840 comprises a Share Author Rights pull-down 842; by selecting a group from the pull-down menu, the author of the transportable application can share author access rights with a named groups.

In this context, "author rights" refers to a set of privileges for working with transportable application that are reserved to the author of the transportable application. Such author rights may include the ability to send a reminder message, close the transportable application, hide or show pages of the transportable application, or add a page to the transportable application.

To send a reminder message, the author selects a Send A Reminder option from among the message options. The author indicates the recipient choice for the reminder, and enters reminder text. In response, the system will re-send the transportable application to the recipients with new comments at the top.

Closing a transportable application disallows any further additions to the transportable application or to its dynamic content. To close a transportable application, the author selects a Close This Message option from among the message options. The author enters any desired closing message text, such as an explanation of why the

transportable application is closed. The transportable application may be re-sent to all recipients with the closing message text at the top, or the closing text may be displayed in the body of the transportable application when existing recipients re-open it.

To hide or show a page, the author selects a Hide/Show Pages option from among the message options. The Hide/Show Pages option enables the author to hide a page that is old or no longer relevant without having to permanently delete it.

The user also may select a sending mode from within a Sending Mode area 844 by selecting one of a plurality of radio buttons 846, 848, 849 that are respectively associated with a Send Unsecured mode, Send Secured mode, and Send Secured with Restricted Forwarding mode. In Send Unsecured mode, no special access controls are applied to the message. Recipients can read and forward the message without first logging into the portal. In Send Secured mode, recipients must log in to the portal before they can view the message. Recipients can forward the message to others. All recipients must be registered in the portal and logged in before they can read the message. In effect, Send Secured requires a recipient to overcome the barrier of user authentication before the user can read the message. In Send Secured with Restricted Forwarding mode, recipients must log-in to the portal to view a message, and only the author can forward the transportable application.

Referring again to FIG. 8B, if the Page Security Options link 836 is selected, then in response, the system displays a Page Security Options window. FIG. 8D is a diagram of an example of the Page Security Options window that may be displayed, in one example embodiment. Page Security Options window 850 comprises one or more access lists 852 that are displayed in association with radio buttons 856, 858. Each radio button is associated with a particular page of the transportable application, as indicated by a page title header 854. In this case, the page title is Discussion, as indicated by page title 854A. Thus, each named page may have different security options. The access lists 852 may be broad or specific. For example, an Everyone access list 856 provides all current and forwarded recipients with access to the Discussion page. A Specific Recipients list 857 enables only selected users and groups to access the page, if such users or groups are named as part of list 857 and radio button 858 is selected.

Selecting a Submit button 859 sends the selected security options to the server.

Referring again to FIG. 7D, In block 736, the user sends the transportable application by selecting the Send Now button 838.

1.4.1 Application Editor

According to one embodiment, an application editor or application builder is provided for assembling pages into a template for authoring and sending transportable applications. Using the application builder, a user may build as many transportable applications as the user needs, for many different purposes.

Referring again to FIG. 9B, to launch the application builder, a user logs into a portal and receives screen display 910. The user selects the Builder link 928 from among function links 912. In response, the system displays a screen with a link to the Application Builder and the Page Builder (page editor), which is described in the next section. The user selects the Application Builder.

FIG. 9D is a diagram of an Application Builder screen display that is displayed in response to selecting the Application Builder. Screen display 950 comprises a Name field 954 and a Description field 956 in a header area 952. The user enters a description for a new transportable application template. The name appears in the application selection dialog box and helps users select the appropriate application.

The user also enters data in an Introduction Settings area 958, which comprises a plurality of custom field label fields 960 and corresponding Required check boxes 962. The user enters names for any custom fields that authors can fill out when creating a new transportable application based on the template. The corresponding check box 962 is selected if the user wishes to require authors to fill out the field before they can send a new transportable application based on the template.

The Application Builder screen can also receive data specifying the name of a status field in a Name of Status Field 964, and a list of corresponding settings in a Possible Settings field 966. The list comprises one or more possible settings that the status field can assume for specific transportable applications that are constructed based on the template. Example status settings include "New," "Open," "Closed," etc., and may be customized by the author of the template as appropriate for the template.

The user may also add one or more default pages to the template by selecting an Add Page link 968A in Default Pages area 968. Default pages form part of any transportable application that is build and sent based on the application template. Similarly, the user may add one or more optional pages to the template by selecting an Add Page link 969A from an Optional Pages area 969. The optional pages are pages that the author and recipients can choose from when adding a page to the transportable application. In response to selecting one of the Add Page links 968A, 969A, the system

displays a page navigation dialog in which the user may navigate to a page folder and select a page to add.

FIG. 9E is a diagram of a page navigation dialog that may be displayed, in one example embodiment. Dialog 970 comprises a Page Folder pull-down list 972 that may be used to select a folder of pages. In the example of FIG. 9E, the Personal folder is selected. Selecting a navigation button 974 causes the system to display the parent folder of the current folder. When a folder is selected using pull-down list 972, the system displays a list 973 of pages that are in the selected folder. A user may create a new page by selecting a Create New Page link 978. In the example shown, a page named NEW PAGE is selected, and metadata describing the selected page is displayed in a description area 979. Upon navigation to the desired page, the user may add it to the current application template by selecting an Apply link 977.

Selecting a new folder button 975 enables creation of a new folder. In response, the system displays a dialog that prompts the user to enter a name and description for the new folder. The user can create any number of sub-level folders. For example, the user can create a folder named New Pages Folder, open it, and create more folders named Home and Work. The user may move within the folder tree by selecting a folder name to view its contents, then selecting the navigation button 974.

FIG. 10A is a diagram of a Default Page Settings screen that may be used to change properties of a selected default page of a template. In one specific embodiment, a user may use Default Page Settings screen 1000 to change a page name, specify whether an included page is required, and enable updates to a page. Default Page Settings screen 1000 comprises a Page Title field 1004 that can receive a new name for the page. Default Page Settings screen 1000 also comprises an Enable Updates checkbox. When the Enable Updates checkbox is checked, then changes that are made to the page by a page author are made to the template as well, so that any transportable application that is created thereafter also will reflect such changes.

Default Page Settings screen 1000 also comprises a Require This Page checkbox 1008. If it is checked, then the associated page is required in any transportable application that is built based on the associated template, and cannot be deleted.

Pages added to the Optional Pages list are available to authors of transportable applications if such authors wish to add a page. Optional pages appear in the Add Page screen and are not included automatically, but can be added as needed. The user may specify whether authors of transportable applications may add pages to a transportable application that is based on the template from outside the Optional Pages list. In one

embodiment, such specification is provided by checking an "Allow authors to add Page from outside this list" checkbox in an Optional Pages screen.

After one or more pages are added, the user may save the application template, and optionally change its name. The user may modify the application template at any time by retrieving it and modifying any of the foregoing properties in the manner described above.

1.4.2 Page Editor (Page Builder)

A page editor may be provided for assembling building blocks into pages that can be added to transportable application templates, and then to transportable applications. In one specific embodiment, a Page Builder allows the user to assemble different building blocks into a page. Which building blocks are added depends on the function that the page author wants the page to serve.

Referring again to FIG. 9B, to launch the application builder, a user logs into a portal and receives screen display 910. The user selects the Builder link 928 from among function links 912. In response, the system displays a screen with a link to the Application Builder and the Page Builder (page editor). The user selects the Page Builder.

FIG. 10B is a diagram of a Page Builder screen that may be displayed in response to selecting the Page Builder, in one example embodiment. Page Builder screen 1010 comprises a Name field 1012 that displays a temporary name of the page under development, such as "New Page." To create a new page, the user enters a description of the new page in a Description field 1014 and introduction text in Introduction field 1016. The description value appears in the Page Selection dialog box and helps users select the appropriate page, and the introduction text appears at the top of the page to introduce users to the page. Alternatively, to create a new page, the user may select the Open button from among function buttons 1019, view a page browser showing available pages and folders of pages, select Create A New Page from the page browser, and then enter the description and introduction value.

The user may then add one or more building blocks to the page, or simply save the page in its current form. To save the page, the user selects the Save As button from among function buttons 1019, navigates to a folder in which the user wishes to save the page, enters a name for the page, and selects a Save button. The user may save the page in this manner at any time, and can discard the changes made to the page at any time by selecting the New button from among function buttons 1019.

In one embodiment, launching the Page Builder causes the system to retrieve and display a default page template that comprises two default building blocks. In one specific

embodiment, the default building blocks are a discussion building blocks and information fields building block. From the default page, new building blocks may be added, and existing building blocks may moved or deleted.

To add building blocks, the Insert Building Block link 1018 is selected. In response, the system displays a list of building blocks. The list comprises, for each building block, a name value, description value, and graphical representation of the building block. In one specific example embodiment, the list identifies the following available building blocks:

Approval list. Allows a list of participants to respond with approvals for documents, shared files or the content of a message. May be combined with an Inline Document or Image Gallery building block to gather sign-offs on final drafts.

Discussion. Captures the discussion between message participants, eliminating back-and-forth e-mails. May be added to a page with a File Sharing or Interactive Web Page building block to discuss a document's content or Web site.

File Sharing. In addition to distributing files for review, the File Sharing building block enables versioning, to allow participants to automatically share changes with each other. The File Sharing building block may be added to any transportable application page to allow participants to add relevant files at any time.

Information Fields. Enables creating named fields to build forms to display names, dates, locations, etc. May be combined with other building blocks to capture information about the state of a project, names of participants, address information, or outstanding issues.

Inline Document. Participants can review the full content of an HTML document without opening a second application. May be combined with, for example, a Poll, Approval List, or Discussion building block to allow participants to share opinions of the document.

Inline Web Page. Displays a fully interactive Web page within a page of a transportable application, including browser-like navigation controls. May be combined with, for example, a Poll or Ratings building block to capture and share responses among participants without launching a separate browser.

Invitation. Displays the details of an event, including title, description, time, and location. May be combined with a Poll building block to allow participants to respond and add comments.

Image. Displays a graphical image on the page. May be combined with, for example, a Poll, Approval List, Ratings, or Discussion building block, in the same page, to allow participants to submit their opinions of the image.

Image Gallery. Displays a plurality of graphical images, e.g., in "thumbnail" format, within a page. All images may be sent in a transportable application at the same time, or participants may add their own images at any time. May be combined with, for example, an Approval List or Ratings building block to collect opinions on photos or artwork.

Poll. Participants vote for one or more of a plurality of listed options, and the results are displayed in a pie chart, bar chart, etc. May be combined with, for example, an Image Gallery, File Sharing, or Schedule building block to help make business decisions.

Ratings. Respondents share their ratings with each other on one or more questions on a discrete scale, e.g., ratings from 1 to 10. May be used to collect feedback or gauge group members' opinions.

Schedule. Participants specify their availability for one or more proposed dates or times. The Schedule building block can export data to programs that support a calendar data interchange format. May be used, for example, to determine the best time for a meeting or event.

Table. Compile data from multiple individuals into a tabular format; participants can add to or edit the information. The table building block can import or export data, for example, data files in delimited file formats. May be combined with other building blocks to launch a project, prioritize items, or share contact information.

Listing. Retrieve various types of data from the database and display it in a list format, using HTML or any other desired presentation format.

Further description about building blocks is provided elsewhere herein. To add a building block to a page, the user selects the desired building block by selecting its name from the list or selecting the corresponding graphic image. The list closes, and the selected building block is added to the page, which expands to accommodate the building block. The Insert Building Block link 1018 is re-displayed, and the user may add one or more additional building blocks by selecting the link 1018 again. In one embodiment, the Insert Building Block link 1018 is re-displayed both above and below each added building block, so that the relative order of building blocks in a page may be specified by selecting the link 1018 in the correct ordinal position. Thus, the user may place building blocks above, below, or between other building blocks that have been previously added by selecting the link 1018 at the appropriate position.

The user may also control the visibility of building blocks to users who receive a transportable application that contains them by choosing to hide or show particular building blocks. When a building block is marked as hidden, recipients of a transportable application that contains a page with that building block do not see the building block. Users may similarly add new pages, move, delete, show or hide existing pages. Applying the hide option enables an author to create a complex template that can be used in many different situations, and selectively reveal desired building blocks as appropriate for the particular situation.

FIG. 11A is a diagram of the graphical appearance of an Approval List building block as it appears in a transportable application page under development using the Page Builder. Approval List building block 1100, and all other building blocks, comprises a Move Up link 1102, Move Down link 1104, and Delete link 1106. A user may delete the building block from the page under development by selecting the Delete link 1106. In response, the building block is logically removed from the page, and the page is re-displayed without the building block.

The order in which building blocks appear in the Page Builder is the order in which authors and recipients of transportable applications based on that page will see the building blocks. The user may select links 1102, 1104 to change the relative order of the building block by moving it before or after another building block. In either case, the page is re-displayed with the building block in its new position.

In one embodiment, a Survey Page Builder is provided to enable users to assemble specific building blocks into pages that carry out surveys of recipients. Each survey page comprises one or more building blocks selected from a Choice building block, Free Text Response building block, and Ratings building block. A survey page may be added to any transportable application or template.

To create a survey page, a user obtains a portal view (FIG. 9B) and selects Builder link 928. The Builder window appears, and the user selects a Survey Page Builder link within it. In response, a Survey Page Builder window opens. FIG. 10C is a diagram of a Survey Page Builder window that may be displayed, in one example embodiment. In this embodiment, the Survey Page Builder window 1020 comprises a Name field 1022 that displays a default value of "New Page," a Description field 1024 that may receive a text description of the page, and an Introduction field 1026 that may receive a text introduction for the page that is displayed to recipients at the top of the page.

Survey Page Builder window 1020 further comprises one or more survey options. In one embodiment, a "Make recipient responses anonymous" link 1028 is provided.

When it is selected, responses to the survey are stored in anonymous form. An "Allow recipients to change responses until the survey is closed" link 1030 may be provided. When selected, the system allows recipients of the page to change their responses at any time. When it is not selected, a response of a recipient may not be changed after it is submitted.

The user may add one or more survey building blocks by selecting the Insert Building Block link 1018 at a point in the page where the user wishes to have a building block appear. In response, a list of available building blocks is displayed. In one embodiment, the list comprises:

Choice. Respondents vote for one or more of the listed options; results are displayed, e.g., in a pie chart, bar chart, etc., and are visible only to the author of the transportable application.

Free Text Response. Respondents enter text responses to a question into a single line text box, or a larger comment box. Responses are gathered in a table and are visible only to the author of the transportable application.

Ratings. Respondents provide their ratings on a list of questions or items according to a discrete scale, e.g., 1 to 10. The results are summarized in a graphical format that is visible only to the author of the transportable application.

The user may select one of the building blocks by selecting its name or icon. The user completes the building block with any required information; fields for completion by recipients are left blank. The user may re-order the building blocks or delete one or more building blocks; the page is then saved. The page is added to a template for a transportable application as described herein.

1.4.3 Building Blocks—Examples

A description of specific building blocks is now provided. The building blocks described herein are merely examples, and other building blocks, fewer or more building blocks may be provided to carry out different functions or to configure different transportable applications. For convenience and in order to illustrate clear examples, the building blocks are described in part in terms of graphical appearance, functions and features; however, in an implementation, each building block comprises a plurality of executable computer program modules that are defined by class files and other computer program source statements. Thus, the functions and appearance of each building block herein are determined by and implemented in the form of computer code appropriate to carry out the functions and provide the appearance that are described herein.

Referring again to FIG. 11A, the graphical appearance of an Approval List building block 1100 is shown. The Approval list building block 1100 allows a list of participants to respond with approvals for documents, shared files or the content of a message, and may be combined with an Inline Document or Image Gallery building block to gather sign-offs on final drafts. Approval list building block 1100 comprises an Instructions field 1108 that may receive text instructions that are displayed to recipients of a page that contains the building block 1100. Approval list building block 1100 further comprises a list 1110 of one or more user names 1112 corresponding to recipients of the building block and corresponding response indications 1114. The user names 1112 are dynamically updated according to the recipients of the building block 1100. Approval list building block 1100 further comprises an Include linked comment check box 1116. If checked, the system will display comments from participants in the transportable application. The comments are dynamically updated as they are added.

FIG. 11B is a diagram of an example embodiment of a Discussion building block that may be used to captures a discussion between message participants. Discussion building block 1120 comprises a dynamically updated list 1122 of one or more user names 1124, associated comment text 1126, and a time stamp value 1128 indicating the time when the associated comment was added. When recipients receive a transportable application that contains a Discussion building block 1120, one or more of the recipients may select the building block and add a comment to the topic under discussion. The list 1122 is dynamically updated with the newly added comment so that the next recipient who opens the transportable application will see all previously added comments.

FIG. 11C is a diagram of an example embodiment of a File Sharing building block, according to an example embodiment, which may be used to share an electronic file among a group of recipients for collaborative review or updating. In this embodiment, File Sharing building block 1130 comprises a Description field 1132 in which the author of a template that contains the building block may enter description text. An Enable File Versioning check box 1134 is provided. When it is checked, the system requires each user to check out the associated file before it can be viewed or modified. The file is checked in after modification.

File Sharing building block 1130 also comprises an Attach Files link 1136. The author of the application or template that includes building block 1130 may select link 1136 to specify a file that is shared among recipients of an application that contains the page. In addition, the Attach Files link 1136 may be used by recipients of the transportable application to add files for sharing, discussion or collaboration. Thus, the

one or more files that are shared using building block 1130 may be specified by recipients of a transportable application that contains a page that contains the building block 1130, rather than specified by an author of a transportable application that contains the page or a template that contains such a page.

In one embodiment, when a user is interacting with a transportable application having a page that contains File sharing building block 1130, and the user elects to upload a new file for sharing or discussion, the system prompts the user to decide whether versioning should be enabled for the document. A versioning value is stored as a file attribute data value in the database in association with the file. In another embodiment, the Enable File Versioning check box 1134 is displayed in association with every file link for each file that has been uploaded. Thus, a user who is the author of the transportable application or otherwise acting in an owner role may elect to apply versioning at any time.

FIG. 11D is a diagram of an Image building block that may be provided, in one example embodiment. An Image building block 1140 may be used to display a graphical image on a page of a transportable application. In one embodiment, Image building block 1140 comprises an image 1142, Attach Image button 1144, Image Name field 1146, and Image Description field 1148. After adding the building block 1140 to the page, the author selects the Attach Image button 114 to attach an image to the building block. In response, the system displays a file browse dialog that enables the author to navigate to a desired image, e.g., in a filesystem or folder of the author's computer, server or network, and select the image. The system uploads the selected image to application server 202 and stores it in database 208 in association with the page under construction. The user may also add a name value in Image Name field 1146, and a description value in Image Description field 1148.

FIG. 11E is a diagram of an Image Gallery building block that may be provided, in one example embodiment. An Image Gallery building block 1150 may be used to display a plurality of graphical images within a page. In this embodiment, Image Gallery building block 1150 comprises an image worksheet 1152, Name field 1154, Add Image function button 1156, Add Multiple Images button 1158, and checkbox 1159. To add one or more images to the image gallery, the author selects either the Add Image function button 1156 or the Add Multiple Images button 1158. In response, the system displays a file browse dialog that enables the author to navigate to a desired image, e.g., in a filesystem or folder of the author's computer, server or network, and select the image. The system uploads the selected image to application server 202 and stores it in database

208 in association with the page under construction. The user may also add a name value in Name field 1154.

All images may be sent in a transportable application at the same time. Alternatively, if the checkbox 1159 is selected by the author, then participants who receive a transportable application that contains building block 1150 may add their own images at any time. Conversely, if checkbox 1159 is not checked, then participants may not add additional images to the gallery.

The Image building block and Image Gallery building block are supported by appropriate error logic that generates one or more errors if exceptional conditions occur when images are uploaded by the author. Examples of exceptional conditions include: a file is too large; network connection failure; incorrect image format; incorrect image width; etc.

FIG. 11F is a diagram of an Information Fields building block that may be provided, in one example embodiment. In this embodiment, Information Fields building block 1160 comprises a list 1161 of one or more field names 1162 each having an associated field value 1164. The author of a page that contains the building block 1160 provides values for field names 1162; the associated field values are left blank and are added by participants who receive the page. Additional pairs of field names and values may be added to the building block by the author by selecting an Add Row button 1166. Accordingly, an author may create named fields to build forms to display names, dates, locations, etc., and a transportable application can capture information about the state of a project, names of participants, address information, or outstanding issues.

In one embodiment, each field of the Information Fields building block may have an abstract data type associated with it using the data typing mechanisms described herein with respect to the Table building block.

FIG. 11G is a diagram of an Inline Document building block that may be provided, in one example embodiment. In this embodiment, Inline Document building block 1170 comprises an Upload File link 1172. To place a document within a page of a transportable application, to enable participants to review the content of the document without opening a second application, an author of a page that contains building block 1170 selects link 1172. In response, the system displays a file browse dialog that enables the author to navigate to a desired file, e.g., in a filesystem or folder of the author's computer, server or network, and select the file. The system uploads the selected file to application server 202 and stores it in database 208 in association with the page under

construction. The name of each file that has been uploaded is displayed in the building block, e.g., in an area above the link 1172.

In one embodiment, building block 1170 accepts one file for upload. In this embodiment, multiple documents may be displayed in a transportable application by adding another page to the application, or by including multiple building blocks 1170 within the same page. Alternatively, multiple files may be included.

FIG. 11H is a diagram of an Inline Web Page building block that may be provided, in one example embodiment. In this embodiment, building block 1180 displays a fully interactive Web page within a page of a transportable application, including browser-like navigation controls. The author selects a location field 1182 and provides a URL or other location identifier of the Web page for display.

In one embodiment, the URL may be changed at any time by recipients of pages that contain the building block. An administrative console function may be provided to enable establishing filters for building blocks.

FIG. 12 is a diagram of an example transportable application that contains an inline Web page. In this example embodiment, a transportable application 1200 is shown in a view from a graphical user interface of an e-mail client program 1210. Transportable application 1200 comprises a header 1208 and an inline Web page building block 1212. The building block 1212 includes a navigation header 1201 and a Web page image area 1202 that displays the inline Web page. Navigation header 1201 comprises a Back button 1204, Forward button 1205, and Home button 1206. Navigation header 1201 is displayed automatically by the system when the transportable application 1200 is presented. Thus, the navigation header 1201 is system-generated and does not form part of the Web page in image area 1202. The Web page in image area 1202 is displayed in the same form that it would have if viewed using a browser in conventional manner. Back button 1204, Forward button 1205, and Home button 1206 each comprise links to JavaScript code that carries out the functions of the button. A specific embodiment of such code is described further herein with reference to FIG. 24.

FIG. 13A is a diagram of an Invitation building block that may be provided, in one example embodiment. In this embodiment, Invitation building block 1300 comprises the following data entry fields, which receive text information provided by the author of a page that includes the building block: Event Title field 1304, which receives a title of the event for which the invitation is issued; Description field 1306, which receives a brief description of the event; Details field 1308, which receives text providing detailed information about the event; Date field 1310, which receives a value identifying the date

of the event; Duration field 1312, which receives a value identifying the time period of the event; Location field 1314, which receives a value specifying where the event will occur; Address field 1316, which receives a value providing an address for the event; and a deadline field 1318, which provides a deadline date for responding to the invitation.

The value in the deadline field 1318 may be used, programmatically, to automatically trigger closing a transportable application or page that includes the Invitation building block 1300.

An author of a page that contains building block 1300 may select a style for the graphical appearance of the building block using a Select Invitation Style link 1302. In response, the system displays a list of one or more pre-defined styles for invitations. The author selects one of the styles, fills in the foregoing event data, and saves the building block.

FIG. 13B is a diagram of a Poll building block that may be provided, in one example embodiment. In this embodiment Poll building block 1320 comprises a Question field 1322 in which the user enters the specific question that recipients are requested to answer when the building block is in use. The author also enters one or more answer choices in answer fields 1324; in responding to the poll question, recipients are required to select one of the answer choices.

As recipients interact with the building block, the author may view a graphic image of the results in graph or chart form. The author may select the specific form of the results by selecting one of a plurality of radio buttons 1326, each of which is associated with a different presentation format such as pie chart, bar chart, etc. The author may enable recipients to attach comments to their responses by selecting Comment check box 1328.

Poll building block 1320 further comprises an Advanced Options link 1329 that enables the author to set other functional behavior of the building block. In response to selecting link 1329, the system displays an Advanced Options window. FIG. 13C is a diagram of an Advanced Options window that may be displayed, in one example embodiment. In this embodiment, the author may use Advanced Options window 1330 to select whether recipients of a poll may vote for one or more of the answer choices 1324 of FIG. 13B, by selecting a value from a pull-down menu 1332. The author may instruct the system to include an additional answer choice of "Other" in the list, and allow recipients to write in an answer, by selecting check box 1334.

In one embodiment, if an "Other" choice is allowed, and recipients write in answers, the written-in answers are automatically added to the choice list. The written-in

answers are dynamically updated to the database. Thus, any later recipients who open the same transportable application see the written-in answers as additional poll choices.

One or more Vote Options check boxes 1336 may be provided. In one embodiment, the author may instruct the system to allow recipients and participants in the poll to change a vote by selecting a first check box. The author may instruct the system to allow participants to vote anonymously by selecting a second check box. The author may specify when the poll results are available for viewing by the participants by selecting one of a plurality of radio buttons 1338. In one embodiment, the author may specify that the poll results are available to participants always, after a participant has voted, after the poll closes, or never. In one embodiment, the author may require all ratings to be submitted on an anonymous basis ("forced anonymity"), by selecting an option or check box when the Poll building block is added to an application.

FIG. 13D is a diagram of a Schedule building block that may be provided, in one example embodiment. Participants specify their availability for one or more proposed dates or times in a schedule table. Using Schedule Type radio buttons 1342, the author may specify whether the schedule is structured or free form. In a structured schedule, for each time alternative, the author of a transportable application that includes the Schedule building block 1340 specifies the exact date, time and duration of each time alternative. In a free-form schedule, the author can enter any desired information to describe each alternative.

A Location field 1344 receives a value identifying a proposed location for the scheduled event. A list 1346 of a plurality of time alternatives or choices 1348 is provided. Each choice 1348 has an associated date field 1350, time field 1352, and duration field 1354. Each choice represents a possible best time for a meeting or event, for example. In one embodiment, a maximum of five choices are provided in the building block, as in FIG. 13D, and an option check box 1356 enables the author to specify whether recipients can propose additional choices. If option check box 1356 is not checked, then a recipient of building block 1340 in a transportable application is required to select one of the five choices as best for that individual's schedule.

In one embodiment, the Schedule building block lists at least one schedule response for all recipients of a transportable application that contains the building block. Further, a blank row is displayed in the schedule table for all recipients who have not responded. Thus, even if a recipient has not responded to the transportable application, that recipient is listed in the schedule. This enables all recipients to see who has provided an availability response and who is yet to respond. Programmatically, the Schedule

building block may use a getRecipientResult() method to retrieve a list of recipients of the transportable application. The list of recipients is compared to a second list of all recipients who are known to have responded to the transportable application. The difference in the lists provides those recipients who should be listed in the schedule table with blank entries.

FIG. 13E is a diagram of a Table building block that may be provided, in one example embodiment. A table building block 1360 may be used to compile data from multiple individuals into a tabular format; participants can add to or edit the information. The Table building block can import or export data, for example, data files in delimited file formats. In the embodiment of FIG. 13E, a data table 1364 is defined by a name field 1362, one or more rows 1374 and a plurality of columns 1370. Each column 1370 has an associated column name 1372. Each row and column has an associated radio button which, if selected by the author of the page that contains the building block, makes that row or column appear in the table when the page is displayed in a transportable application.

The author may modify the size and appearance of the table using buttons 1368. In one specific embodiment, buttons 1368 comprise an insert button, delete button, and properties button. A user selects a particular row or column by selecting its associated radio button, and then selects one of the buttons 1368. In response, if the insert button is selected, the system will insert a row or column adjacent to the selected row or column; delete the selected row or column; or enable the user to modify properties of the selected row or column. In one embodiment, such properties include column width, text alignment (left, right, center), cell format, etc. In still another embodiment, the requested changes may be applied to all rows and columns in the table by selecting a separate check box.

In an alternative embodiment, selecting the insert button causes the system to prompt the user to enter a numeric value representing a number of rows or column to insert, thereby enabling inserting multiple rows or columns in a single operation.

In another embodiment, table columns may be formatted according to one of a plurality of abstract data types, e.g., date, numbers, currency, time, etc. The data types may also be enumerated data types, such as a priority type (high, medium, low); an answer type (yes or no), a status type (open, closed, pending, resolved); an approval type (approved, do not approve, undecided); or a custom enumerated type that is user-defined. To apply a data type, the author selects the Properties button from among buttons 1368 and then selects a data type from among a list of available data types. The selected data type is stored in the database as an attribute value for the associated column of the table.

If the author adds rows to the table, then columns in the row inherit the data type attribute values. When a recipient adds data to a cell, type checking is carried out to ensure that the value that was entered matches the type of the cell; if a type match occurs, an error is reported.

In addition, data values for table cells may be imported from a data file by selecting an Import link 1366. For example, data values may be imported from a spreadsheet file, delimited file, comma-separated file, tab-separated file, or the equivalent.

Use of enumerated data types enables users to construct Table building blocks that effectively track progress on action items relating to a project. For example, a group of users who are collaborating on a particular project can create a transportable application with a Table building block that identifies tasks, a party responsible for each task, and a status value taken from an enumerated data type. As users complete assigned tasks, they update the Table building block status value as appropriate. Any user who views the transportable application sees the then-current status values for all tasks.

FIG. 14A is a diagram of a Choice building block that may be provided, in one example embodiment. Using a Choice building block, respondents vote for one or more of the listed options; results are displayed, e.g., in a pie chart, bar chart, etc., and are visible only to the author of the transportable application. In this embodiment Choice building block 1420 comprises a Question field 1422 in which the user enters the specific question that recipients are requested to answer when the building block is in use. The author also enters one or more answer choices in answer fields 1424; in responding to the question, recipients are required to select one of the answer choices provided in the fields.

As recipients interact with the building block, the author may view a graphic image of the choice responses in graph or chart form. The author may select the specific form of the results by selecting one of a plurality of radio buttons 1426, each of which is associated with a different presentation format such as pie chart, bar chart, etc. In one embodiment, the author also may use a pull-down menu 1432 to select whether recipients of a poll may vote for one or more of the answer choices 1424 by selecting a value from a pull-down menu 1432. The author may instruct the system to include an additional answer choice of "Other" in the list, and allow recipients to write in an answer, by selecting check box 1434.

FIG. 14B is a diagram of a Free Text Response building block that may be provided, in one example embodiment. In this embodiment, respondents enter text responses to a question into a text box or comment field. The author specifies a question

for recipients to answer by entering it in question field 1432. Responses are gathered either in a single-line text box or multi-line comment area according to a selection by the author of a corresponding radio button 1434. In an embodiment, a text editor is integrated into question field 1432 so that respondents may use standard text editing keyboard commands and cursor operations as they enter and revise text. In another embodiment, question field 1432 is integrated with a browser DLL, or similar rendering library. In this configuration, respondents may enter HTML tags in the question field, which are stored in association with the building block, and rendered and displayed in HTML format when a response in the question field is displayed to other recipients.

FIG. 14C is a diagram of a Ratings building block that may be provided, in one example embodiment. In this embodiment, Ratings building block 1440 comprises an Instructions field 1442 in which the author enters text instructions to recipients about how to complete ratings of criteria. The author may define a rating scale by selecting a maximum value 1444. For example, if the author wants criteria rated by participants on a scale from "1" to "5," then the author selects "5" as maximum value 1444. In addition to the numeric rating values, the author may permit a rating of "not applicable" or "N/A" by selecting check box 1446. One or more corresponding rating labels may be provided in rating fields 1448. One or more criteria to receive ratings are provided in item fields 1450. Additional item fields may be added, or item fields may be removed, from among item fields 1450 by selecting an Add Entries link or Remove Entries link 1452.

In one embodiment, the author may require all ratings to be submitted on an anonymous basis ("forced anonymity"), by selecting an option or check box when the Ratings building block is added to an application.

A Listing building block is provided to enable a transportable application to retrieve various types of data from the database and display it in a list format, using HTML or any other desired presentation format.

In one specific embodiment, the Listing building block is implemented as an abstract class that can be subclassed to present any result set of database data that can be described by a programmer or developer. For example, the Listing building block may be used to generate lists of groups, folders, messages, and transportable applications for use in the Group Directory, Group Messages Inbox, Group Application Templates, and Personal Messages Inbox that are displayed through the portal and other mechanisms as described herein.

In this embodiment, the Listing building block implements a Question interface to issue database queries, and a Presenter interface to carry out data presentation. The

Listing building block also generates a Create event, Read event, and Close event to the event handling system when it is created, read, or closed, respectively. The Listing building block contains no actual data of its own; rather, all data it displays is metadata about other objects, such as transportable applications, users, groups, or templates. Thus, it need not generate a Response event.

A Group Directory building block may be implemented as a subclass of the Listing building block. The Group Directory building block displays a list of all the Groups available, including group name, description, and status. The Group Directory building block may receive a selection of a group from user input. In response, the Group Directory building block determines whether the user is authorized to access the selected group, and then generates a Status value as output. The Status value may be Join Group, Member, or Private. Join Group means that the user is not a member of the group and the group is not private, i.e., the user may join the group. Member means that the user is already a member of the group. Private means that the user is not a member and group joining is restricted.

A Group Messages Inbox building block also may be implemented as a subclass of the Listing building block. The Group Messages Inbox building block displays a list of all transportable applications sent to a group including author, subject, type, date created, and date last modified. The Group Messages Inbox building block may be used in two different instances to provide a list of current group messages as part of a group home page, and to provide a group archive list.

A Group Applications Templates building block also may be implemented as a subclass of the Listing building block. The Group Applications Templates building block displays a list of selected application templates for a particular group, organized by template name and date created. The selected application templates may all those that are in that group's application directory, for example. There may be an instance of the Group Applications Templates building block for each group.

A User Messages Inbox building block may be may be implemented as a subclass of the Listing building block and generates a list of all transportable applications sent by and received by a user, except that received group messages are not displayed. The list comprises, for each transportable application in the list, a status value, attachment status value, author name, subject, and date last modified. The status value indicates whether the associated transportable application is new or unread, updated or changed, or read. The attachment status value indicates whether the associated transportable application has one or more files attached, or no attachments. The User Messages Inbox building block

provides graphical controls to create folders, move transportable applications into folders, or to delete them.

1.4.4 Site Style Editor

According to one embodiment, the Style Editor is configured to enable a portal administrator to create one or more Site-specific Styles. In this context, a "site" is an installation of system 200 at a particular enterprise or organization, such as a corporation. The Site Style Editor displays only those properties of transportable applications and templates that are relevant to creating styles for a particular site. Such properties may include site snippets (including the header and navigation), color scheme, branding and corporate-color images, and fonts to be used throughout the site.

In one approach to implementing the foregoing, a Style Properties table of database 208 includes a column that identifies whether a style property is a general property or a site-specific property. The Style Editor is configured to allow setting a mode, e.g., generic or site-specific, when a user invokes or enters the Editor. When the Style Editor is in a site-specific mode, it shows only properties relevant to that mode.

A Site Style Preview Tool is provided to enable a user, who is creating a Site Style, to preview the appearance of the site so that other users of the system will not be disrupted during development time. Only the Site Style creator can see the site using the Site Style being created. In one embodiment, Site Style development and previewing is carried out on a separate application server to minimize performance issues. Further, in one approach, the preview of the site is displayed in a frameset of a browser, enabling a user to freely browse the site.

When one or more styles have been created, users can select a style at the time that a transportable application is authored. For example, at authoring time, when a transportable application is created or a task is added, a user can either select an existing Style through the Style Picker or create a new Style during authoring and automatically apply that Style to the transportable application that is being authored.

In one approach, a Style Editor is provided having an action bar with a plurality of graphical user interface buttons that trigger different actions and behavior. In one specific embodiment, a Clear button resets Style properties. A Save button saves the Style. If it already exists in the database, it is overwritten without asking the user for a new name. If it does not exist, then the user is prompted to provide a name and description. A Save As button saves the Style after first prompting the user for a name and description. An Open button loads a Style into the Editor from the user's personal collection, or from a Group or Global folder.

A Preview button displays a preview of the current Style. The style is previewed in association with the type of transportable application that the user is in the process of authoring. An Apply and Exit button is also provided. When selected, if the Style has not been saved, then the user is prompted to provide a name and description of the Style, and it is saved. The Style is automatically applied to the then-current transportable application under development, and the Editor is closed. Cancel and Exit buttons may also be provided to enable termination of style editing and exiting the application.

The Style Editor also may be configured to save and load styles to and from groups, global areas, or folders of groups or global areas. This enables groups of users to export styles to such areas for the purpose of sharing them. Use of folders enables users to organize styles. Folders and style references may be created as node objects.

1.5 Communications From Transportable Applications to Users ("Notifications")

1.5.1 Notifications Based on User Responses or Timed Events

In an embodiment, application server 200 includes processes that can selectively deliver notification messages to users based on actions carried out by transportable applications. In general, a notification is a message, sent by electronic mail or any other transport mechanism, to a recipient of a transportable application separate from the application as a result of an action taken by that recipient or by another recipient of the application. For example, assume that a transportable application includes a discussion building block in which a group of participants are discussing a particular topic. If a participant enables notifications for that application, the system sends that participant a notification message each time that another participant updates the application with a new comment.

In one embodiment, each participant or user affirmatively requests notifications in an "opt-in" process. Alternatively, however, notifications may be used to accomplish more complicated business processes or workflows. Further, in one embodiment, notifications are associated with transportable applications as a whole; alternatively, each building block may independently generate a notification based on an action of a participant using that particular building block. In still another alternative, notifications are issued based on rules when conditions specified in the rules are satisfied by user actions.

In this description, participants or users may assume one or more roles. An individual acting in an Owner /Moderator role is a person who initiates or starts a process. The Owner owns the process and is accountable for all its functions. In the corporate