BLANK ROME LLP
COUNSELORS AT LAW

Phone: *(302) 425-6408*
Fax: *(302) 428-5106*
Email: *Caponi@BlankRome.com*

July 9, 2009

BY HAND & ELECTRONIC FILING

The Hon. Leonard P. Stark
J. Caleb Boggs Federal Building
U.S. District Court for the District of Delaware
844 N. King Street, Unit 26, Room 6100
Wilmington, DE 19801-3556

Re: ***Leader Technologies, Inc. v. Facebook, Inc.*, C.A. No. 08-862-JJF**

Dear Judge Stark:

This is a patent infringement action brought by Plaintiff Leader Technologies, Inc. ("LTI"), in which LTI asserts that Defendant Facebook, Inc. ("Facebook") infringes certain claims of U.S. Patent No. 7,319,761 ("'761 patent"). Facebook has denied that it infringes and has asserted various defenses, including that the '761 patent is invalid in light of the prior art.

Facebook respectfully submits this letter brief to address LTI's withholding of information responsive to Facebook Interrogatory No. 10, which seeks LTI's contentions regarding whether an earlier-filed provisional patent application filed by LTI's inventors discloses the elements claimed in the '761 patent. This information, which is exclusively in LTI's possession, is critical to Facebook's invalidity defense because it bears on the effective filing date of the '761 patent for determining what constitutes "prior art" that may be used to invalidate it. LTI has refused to provide a firm answer, despite having all the knowledge and information to do so.

In response to Facebook Interrogatory No. 10, LTI initially responded that the "entirety" of its six-page provisional application disclosed the "entirety" of its 60-page later-filed application, providing no further explanation. After Facebook filed a motion to compel on this issue, LTI retreated and supplemented its response, but provided only "non-limiting" and "non-exhaustive" "examples" in its response. LTI's supplemental response does not cure the deficiencies of its initial response, leaves uncertainty and is not an actual answer. Facebook respectfully requests that the Court require LTI to provide a complete response to Interrogatory

No. 10.

A. LTI's Six-Page Provisional Application and 60-Page Utility Application

On December 11, 2002, LTI's named inventors filed U.S. Provisional Patent Application No. 60/432,255 (the "Provisional Application"). *See* Ex. C. The Provisional Application contained no figures or claims and consisted of only six pages of text and an attachment with some additional text and several pages of source code. *Id.* Twelve months later, on December 10, 2003, the inventors filed U.S. Patent Application Ser. No. 10/732,744 (the "'744 Application"). The later-filed '744 Application ultimately led to the issuance of the patent-in-suit. *See* D.I. 1 at Ex. A.

The '744 Application claims priority to the earlier-filed Provisional Application and shares its title, but the similarities end there. The later-filed application contains substantial additions, triples the page count and adds 21 figures and 44 claims. Most importantly, the later-filed '744 Application, for the first time, disclosed new subject matter absent from the Provisional Application, including, for example, "tracking," "tracking components," "workspaces," "dynamic association" and "user-defined data," which are all limitations incorporated into the claims of the patent-in-suit at issue here.

B. Facebook's Discovery and LTI's Responses

In order to determine the exact universe of potentially invalidating prior art, Facebook needs to understand which filing date LTI will be held to using in this case. To that end, Facebook propounded Interrogatory No. 10 asking:

> For each claim of the '761 patent that you contend is entitled to the benefit of the filing date of U.S. Provisional Application No. 60/432,255 (December 11, 2002) ("'255 Application"), identify with particularity all portions of the '255 Application that you contend disclose each element of each such claim.

Ex. A at 2. This is a standard interrogatory in any patent infringement case. As noted by the Federal Circuit: "It is elementary patent law that a patent application is entitled to the benefit of the filing date of an earlier filed application only if the disclosure of the earlier application provides support for the claims of the later application, as required by 35 U.S.C. § 112." *PowerOasis, Inc. v. T-Mobile USA, Inc.*, 522 F.3d 1299, 1306 (Fed. Cir. 2008) (citation omitted).

All of the information LTI needs to provide a full response is contained in the four corners of its Provisional Application and is, therefore, entirely within its possession. No external discovery is required for LTI to provide a complete response. LTI initially responded to Interrogatory No. 10 by simply stating the Provisional Application "in its entirety provides support for the subject matter of the claims," without further explanation. Ex. A at 2. Facebook filed a motion to compel after LTI refused to supplement its response. *See* D.I. 55. Faced with Facebook's motion, LTI changed its mind and stated that it would supplement its response.

LTI's served its supplemental response to Interrogatory No. 10 on June 8. The supplemental response purports to identify portions of the Provisional Application, but the entire response is qualified with the statement that LTI has provided only "non-limiting" and "non-exhaustive" "examples" of its contentions. *See* Ex. A at 3. An additional conference and further correspondence failed to resolve the issue. Ex. B.

C. LTI Should be Ordered to Provide a Complete Response

The parties disagree about whether the patent-in-suit is entitled to the priority date of the Provisional Application or the '744 Application. This issue will turn on whether the content of LTI's Provisional Application discloses the subject matter of the claims of the '761 patent. Facebook is entitled to obtain all of LTI's contentions on this issue.

LTI, as the alleged patent owner, bears the burden of establishing that its claims are entitled to the benefit of the filing date of the Provisional Application. *See PowerOasis, Inc.*, 522 F.3d at 1303-07. To meet its burden, LTI must show that the Provisional Application discloses each and every aspect of the claims of the patent as issued and in a manner that enables a person reasonably skilled in the art to make and use it without undue experimentation. *See* 35 U.S.C. § 119(e)(1); 35 U.S.C. § 112 ¶ 1. In other words, LTI can successfully claim priority to the Provisional Application only if it can show that the earlier-filed application disclosed all of the elements later claimed in the patent-in-suit. *See PowerOasis, Inc.*, 522 F.3d at 1306.

As noted, LTI has only provided purported "non-limiting" and "non-exhaustive" "examples" of how the Provisional Application allegedly discloses the elements of the asserted claims. *See* Ex. A at 3. Its supplemental response does not answer the interrogatory or provide the information that Facebook is entitled to receive. *See, e.g., Herdlein Techs., Inc. v. Century Contractors, Inc.*, 147 F.R.D. 103, 106-07 (W.D.N.C. 1993) (party failed to properly respond to interrogatory requiring identification of "all" documentary evidence when it only provided examples).

BLANK ROME LLP
COUNSELORS AT LAW

LTI's agreement to provide only "non-limiting" and "non-exhaustive" "examples" of its contentions unfairly prejudices Facebook. The obvious motive behind LTI's equivocations is so it can later rely on portions of the Provisional Application that it has not identified in its response to Interrogatory No. 10. This defeats the whole purpose of Interrogatory No. 10 and discovery generally – to narrow issues and "to avoid 'surprise' or 'trial by ambush.'" *American Stock Exchange, LLC v. Mopex, Inc.*, 215 F.R.D. 87, 93 (S.D.N.Y. 2002). Again, all of the information LTI would need to provide a complete response is, and has always been, in its possession.

Accordingly, Facebook respectfully requests that the Court compel LTI to respond fully and unequivocally, or to state that its June 8 supplemental response sets forth all of its contentions.

Respectfully submitted,

Steven L. Caponi
I.D. No. 3484

SLC:pfc
Enclosures

cc: Philip A. Rovner, Esquire – via e-service (w/o enc.)
Paul J. Andre, Esquire – via e-service (w/o enc.)
Lisa Kobialka, Esquire – via e-service (w/o enc.)

EXHIBIT A

IN THE UNITED STATES DISTRICT COURT

FOR THE DISTRICT OF DELAWARE

LEADER TECHNOLOGIES, INC., a Delaware corporation,	)	
	)	Civil Action No. 08-862-JJF
Plaintiff-Counterdefendant,	)	
v.	)	
FACEBOOK, INC., a Delaware corporation,	)	
Defendant-Counterclaimant	)	

LEADER TECHNOLOGIES, INC.'S FIRST SUPPLEMENTAL RESPONSE TO FACEBOOK, INC.'S INTERROGATORY NO. 10

Pursuant to Fed. R. Civ. P. 33, Plaintiff Leader Technologies, Inc. ("Leader") hereby submits the following supplemental response to Defendant Facebook, Inc.'s ("Facebook") Interrogatory No. 10. Leader expressly incorporates by reference each and every general and specific objection and the prior responses to this interrogatory.

Leader makes this supplemental response herein ("Supplemental Response") based solely on its current knowledge, understanding, and belief as to the facts and information available to it as of the date of the Supplemental Response. Additional discovery and investigation may lead to additions to, changes in, or modifications of this Supplemental Response. This Supplemental Response, therefore, being given without prejudice to Leader's right to further supplement this Supplemental Response pursuant to Fed. R. Civ. P. 26(e), or to provide subsequently discovered information and to introduce such subsequently discovered information at the time of any trial or proceeding in this action.

SUPPLEMENTAL RESPONSE TO INTERROGATORY NO. 10

Interrogatory No. 10

For each claim of the '761 patent that you contend is entitled to the benefit of the filing date of U.S. Provisional Application No. 60/432,255 (December 11, 2002) ("'255 Application"), identify with particularity all portions of the '255 Application that you contend disclose each element of each such claim.

Original Response to Interrogatory No. 10

Leader objects to this Interrogatory to the extent that it is vague and ambiguous. Leader also objects to this Interrogatory to the extent that it is overly broad, unduly burdensome, and oppressive. Leader objects to this Interrogatory to the extent it is premature because Facebook has not yet produced documents or responded to written discovery in the litigation. Leader objects to this Interrogatory to the extent it is premature, as the Court has not yet construed claim terms of the '761 Patent. Leader further objects to this Interrogatory to the extent to the extent that it seeks the disclosure of proprietary or confidential business information or trade secrets. Leader further objects to this Interrogatory to the extent that it calls for a legal conclusion and expert testimony. Leader further objects to this Interrogatory to the extent that it seeks information that is protected by the attorney-client privilege, the work product doctrine, and/or any other applicable privilege.

Subject to and without waiving its objections, Leader responds that each claim of the '761 Patent is entitled to a priority date of no later than December 11, 2001 on the basis of Provisional Application No. 60/432,255 ("'255 Application"). The '255 Application in its entirety provides support for the subject matter of the claims.

First Supplemental Response to Interrogatory No. 10:

Leader incorporates by reference the general and specific objections and response to this Interrogatory set forth in its March 20th Responses. Subject to and without waiving the foregoing objections, Leader responds that each claim of the '761 Patent is entitled to a priority date of no later than December 11, 2002 on the basis of Provisional Application No. 60/432,255 ("'255

Application"). The '255 Application in its entirety provides support for the subject matter of the claims.

The following chart provides non-exhaustive and non-limiting examples of specific portions of the '255 Application that disclose the asserted claims of the '761 patent.

'761 Patent Claims	**Provisional Application[1]**
1. A computer-implemented network-based system that facilitates management of data, comprising:	"This invention relates to management and storage of electronic information." [Specification, Pg. 1, ¶ 1.][2]
a computer-implemented context component of the network-based system for capturing context information associated with the user-defined data created by user interaction of a user in a first context of the network-based system, the context component dynamically storing the context information in metadata associated with the user-defined data, the user-defined data and metadata stored on a storage component of the network-based system; and	"As users create and change their context, the files and application follow, dynamically capturing those shifts in context." [Specification, Pg. 5, ¶ 23.][2] "action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown

[1] Specification: refers to the specification filed with the '255 Application. The Specification includes the Field of Invention, Background of Invention, Summary of Invention, Description of Embodiments, Example, and Board Module.

Source Code: refers to the source code filed with the '255 Application. In order to facilitate the mapping of specific lines of code to claim elements, this portion of the '255 Application has been reformatted as a separate Adobe Acrobat document with page and line numbers and is attached hereto as Exhibit 1. The references in the chart are to those page and line numbers.

[2] Please note: This citation only provides a non-exhaustive and non-limiting example. Other citations are available as would be understood by one ordinarily skilled in the art. As Leader's investigation of the '255 Application is ongoing, this citation will be supplemented as appropriate.

'761 Patent Claims	Provisional Application[1]
	=newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown); action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true); action.addActionListener(AddWebRelationshipActionListener.GLOBAL); action.addInterfaceListener(AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]
a computer-implemented tracking component of the network-based system for tracking a change of the user from the first context to a second context of the network-based system and dynamically updating the stored metadata based on the change, wherein the user access the data from the second context.	"Content is preferably associated with a routing algorithm referred to herein as a webslice. Thus the content has an intelligent quality whereby upon a change of structure of the web, the content knows which board or boards it should be on both before and after the change in structure." [Specification, Pg. 6, ¶ 4.][2] "Specify the web that that this webslice is taken from...private void setTraversalCode (int code) { this.traversal = TraversalFactory.getTraversal (code)" [Source code, Pgs. 9-10.][2] "Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content

'761 Patent Claims	Provisional Application[1]
	as part of the change in structure." [Specification, Pg. 7.][2] "action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown); action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true); action.addActionListener(AddWebRelationshipActionListener.GLOBAL); action.addInterfaceListener(AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]
2. The system of **claim 1**, the context component is associated with a workspace, which is a collection of data and application functionality related to the user-defined data.	"public void setTraversal ..."[Source code, Pg. 9, ll. 49 - 50.][2]
4. The system of **claim 1**, the context information includes a relationship between the	"'board' Module. 'WEB VERSION 1' WORKING DESCRIPTION...The

'761 Patent Claims	Provisional Application[1]
user and at least one of an application, application data, and user environment.	implications of this difference on the automation of workflow processes are profound." [Specification, Pgs. 8-10.][2]
5. The system of **claim 1**, the context component captures context information of the first context and context information related to at least one other context.	"Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2]
7. The system of **claim 1**, wherein data created in the first context is associated with data created in the second context.	"Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2]
8. The system of **claim 1**, the context information is tagged to the user-defined data via the metadata when the user-defined data is created.	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID,

'761 Patent Claims	**Provisional Application**[1]

	"ChildBoard",null,
	requestState.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	action=newInterfaceAction "addRelationship",
	"AddRelationship",toolCode,true);
	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]

9. A computer-implemented method of managing data, comprising computer-executable acts of:	"This invention relates to management and storage of electronic information." [Specification, Pg. 1, ¶ 1.][2]

creating data within a user environment of a web-based computing platform via user interaction with the user environment by a user using an application, the data in the form of at least files and documents;	"action.addActionListener
	(RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener
	(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener
	(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub);
	//Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship");
	SingleSelectGroupKeyField boardDropDown=newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null,
	requeststate.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	boardDropDown=new
	BoardKeyField(CHILD_BOARD_FIELD_ID,
	"ChildBoard",null,
	requestState.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	action=newInterfaceAction "addRelationship",
	"AddRelationship",toolCode,true);

'761 Patent Claims	Provisional Application[1]

	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]

dynamically associating metadata with the data, the data and metadata stored on a storage component of the web-based computing platform, the metadata includes information related to the user, the data, the application, and the user environment;	"Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2]
	
	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub);
	//Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship");
	SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null,
	requeststate.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null,
	requestState.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true);

'761 Patent Claims	Provisional Application[1]

	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]

tracking movement of the user from the user environment of the web-based computing platform to a second user environment of the web-based computing platform; and	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub);
	//Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship");
	SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null,
	requeststate.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null,
	requestState.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true);
	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]

'761 Patent Claims	Provisional Application[1]
dynamically updating the stored metadata with an association of the data, the application, and the second user environment wherein the user employs at least one of the application and the data from the second environment.	"Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2] "action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown); action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true); action.addActionListener(AddWebRelationshipActionListener.GLOBAL); action.addInterfaceListener(AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener(AddInterface

'761 Patent Claims	Provisional Application[1]
	Listener.GLOBAL); sub.addAction (action); page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]

10. The method of **claim 9**, further comprising capturing context information of the user.	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub);
	//Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship");
	SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null,
	requeststate.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null,
	requestState.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true);
	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]

11. The method of **claim 9**, further comprising indexing content of the user environment such that a plurality of users can access the content	"Specify the web that that this webslice is taken from...private void setTraversalCode (int code) { this.traversal =

'761 Patent Claims	**Provisional Application**[1]
from an associated plurality of user environments.	TraversalFactory.getTraversal (code)" [Source code, Pgs. 9-10.][2]
12. The method of **claim 9**, the least one of the data and the application is associated automatically with the second user environment.	"Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2]
13. The method of **claim 9**, further comprising accessing the user environment and the second user environment using a browser.	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown); action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true); action.addActionListener(AddWebRelationshipActionListener.GLOBAL); action.addInterfaceListener(AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL); sub.addAction (action);

'761 Patent Claims	**Provisional Application**[1]
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]
14. The method of **claim 9**, further comprising communicating with the user environment using a TCP/IP communication protocol.	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown); action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true); action.addActionListener(AddWebRelationshipActionListener.GLOBAL); action.addInterfaceListener(AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]
15. The method of **claim 9**, further comprising locating the user environment from a remote location using a URL address.	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL);

'761 Patent Claims	**Provisional Application**[1]

	sub.addAction (action);
	page.add (sub);
	//Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship");
	SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null,
	requeststate.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null,
	requestState.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true);
	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]

16. The method of **claim 9**, further comprising accessing the user environment via a portable wireless device.	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub);
	//Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship");
	SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null,
	requeststate.getCurrentUser( ).getId( ));

'761 Patent Claims	Provisional Application[1]

	sub.add (boardDropDown);
	boardDropDown=new
	BoardKeyField(CHILD_BOARD_FIELD_ID,
	"ChildBoard",null,
	requestState.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	action=newInterfaceAction "addRelationship",
	"AddRelationship",toolCode,true);
	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]

21. A computer-readable medium for storing computer-executable instructions for a method of managing data, the method comprising:	"This invention relates to management and storage of electronic information." [Specification, Pg. 1, ¶ 1.][2]
creating data related to user interaction of a user within a user workspace of a web-based computing platform using an application;	"As users create and change their context, the files and application follow, dynamically capturing those shifts in context." [Specification, Pg. 5, ¶ 23.][2] "action.addActionListener..."[Source code, Pg. 6, l. 31 and l. 49; p. 6, ll. 30 - 52.][2]
dynamically associating metadata with the data, the data and metadata stored on the web-based computing platform, the metadata includes information related to the user of the user workspace, to the data, to the application and to the user workspace;	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship");

'761 Patent Claims	Provisional Application[1]
	SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown); action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true); action.addActionListener(AddWebRelationshipActionListener.GLOBAL); action.addInterfaceListener(AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]
tracking movement of the user from the user workspace to a second user workspace of the web-based computing platform;	"Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2] "action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship");

'761 Patent Claims	Provisional Application[1]
	SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIE LD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown); action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true); action.addActionListener(AddWebRelationshi pActionListener.GLOBAL); action.addInterfaceListener(AddInterfaceListe ner.GLOBAL); action.setErrorInterfaceListener(AddInterface Listener.GLOBAL); sub.addAction (action); page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]
dynamically associating the data and the application with the second user workspace in the metadata such that the user employs the application and data from the second user workspace; and	"action.addActionListener (RemoveWebRelationshipActionListener.GLO BAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubFor m","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIE LD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( ));

'761 Patent Claims	Provisional Application[1]

	sub.add (boardDropDown);
	action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true);
	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]
	
	"Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2]

indexing the data created in the user workspace such that a plurality of different users can access the data via the metadata from a corresponding plurality of different user workspaces.	"Specify the web that that this webslice is taken from...private void setTraversalCode (int code) { this.traversal = TraversalFactory.getTraversal (code)" [Source code, Pgs. 9-10.][2]
23. A computer-implemented system that facilitates management of data, comprising:	"This invention relates to management and storage of electronic information." [Specification, Pg. 1, ¶ 1.][2]
a computer-implemented context component of a web-based server for defining a first user workspace of the web-based server, assigning one or more applications to the first user workspace, capturing context data associated with user interaction of a user while in the first user workspace, and for dynamically storing the context data as metadata on a storage component of the web-based server, which	"As users create and change their context, the files and application follow, dynamically capturing those shifts in context." [Specification, Pg. 5, ¶ 23.][2] "action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL);

'761 Patent Claims	Provisional Application[1]
metadata is dynamically associated with data created in the first user workspace; and	action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown); action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true); action.addActionListener(AddWebRelationshipActionListener.GLOBAL); action.addInterfaceListener(AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]
a computer-implemented tracking component of the web-based server for tracking change information associated with a change in access of the user from the first user workspace to a second user workspace, and dynamically storing the change information on the storage component as part of the metadata, wherein the user accesses the data from the second user workspace.	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIE

'761 Patent Claims	Provisional Application[1]
	LD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown); action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true); action.addActionListener(AddWebRelationshipActionListener.GLOBAL); action.addInterfaceListener(AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2] "Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2]
24. The system of **claim 23**, wherein the tracking component automatically creates the metadata when the user accesses the first user workspace.	"Content is preferably associated with a routing algorithm referred to herein as a webslice. Thus the content has an intelligent quality whereby upon a change of structure of the web, the content knows which board or boards it should be on both before and after the change in structure." [Specification, Pg. 6, ¶ 4.][2] "Specify the web that that this webslice is taken from...private void setTraversalCode (int code) { this.traversal = TraversalFactory.getTraversal (code)" [Source

'761 Patent Claims	**Provisional Application**[1]
	code, Pgs. 9-10.][2]
25. The system of **claim 23**, wherein the context component captures relationship data associated with a relationship between the first user workspace and at least one other user workspace.	"Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2]
26. The system of **claim 23**, wherein an application associated with the first user workspace is automatically accessible via the second user workspace when the user moves from the first user workspace to the second user workspace.	"Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pg. 7.][2]
29. The system of **claim 23**, wherein when the data created in the first user workspace is accessed from the second user workspace, in response to which the context component adds information to the metadata about the second user workspace.	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL); action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL); sub.addAction (action); page.add (sub); //Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship"); SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null, requeststate.getCurrentUser( ).getId( )); sub.add (boardDropDown); boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null, requestState.getCurrentUser( ).getId( )); sub.add (boardDropDown);

'761 Patent Claims	**Provisional Application**[1]

	action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true);
	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]

31. The system of claim **23**, wherein the storage component stores the data and the metadata according to at least one of a relational and an object storage methodology.	"action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);action.addInterfaceListener (AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL);
	sub.addAction (action);
	page.add (sub);
	//Add new Relationships sub-form sub = new ConcreteSubForm("createRelationshipsSubForm","CreateNewRelationship");
	SingleSelectGroupKeyField boardDropDown =newBoardKeyField(PARENT_BOARD_FIELD_ID,"ParentBoard",null,
	requeststate.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	boardDropDown=new BoardKeyField(CHILD_BOARD_FIELD_ID, "ChildBoard",null,
	requestState.getCurrentUser( ).getId( ));
	sub.add (boardDropDown);
	action=newInterfaceAction "addRelationship", "AddRelationship",toolCode,true);
	action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
	action.addInterfaceListener(AddInterfaceListener.GLOBAL);
	action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
	sub.addAction (action);

'761 Patent Claims	Provisional Application[1]
	page.add (sub) ;" [Source code, Pg. 6, ll. 30 - 52.][2]
32. The system of **claim 23**, wherein storing of the metadata in the storage component in association with data facilitates many-to-many functionality of the data via the metadata.	"get Board ..." [Source code, Pg. 8, ll. 52 - 53.][2]
33. The system of **claim 23**, wherein the first user workspace provides access to at least one communications tool, which includes e-mail, voicemail, fax, teleconferencing, instant message, chat, contacts, calendar, task, notes, news, ideas, vote, web and video conferencing, and document sharing functionality.	"S[t]ill another object of the invention is to provide a communication tool that integrates two or more different communication applications such as telephony, unified messaging, decision support, document management, portals, chat, collaboration, search, vote, relationship management, calendar, personal information management, profiling, directory management, executive information systems, dashboards, cockpits, tasking, meeting, conferencing, etc. into a common application." [Specification, Page 4, ¶ 17.][2]
34. The system of **claim 23**, wherein one or more applications include file storage pointers that are dynamic and associated with the first user workspace.	"In preferred embodiments, webs may be utilized to maintain the location of content within a complex and changing set of boards...Alternatively, the loc[a]tion of content may be determined by detecting changes in structure, detecting the temporary location of the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure." [Specification, Pgs. 6-7.][2]

Leader's investigation of this matter is continuing and its response to this Interrogatory will be supplemented as additional information becomes known to it.

POTTER ANDERSON & CORROON LLP

By: 

Philip A. Rovner (#3215)
Hercules Plaza
P.O. Box 951
Wilmington, DE 19899
(302) 984-6000
provner@potteranderson.com

Attorneys for Plaintiff
Leader Technologies, Inc.

OF COUNSEL:

Paul J. André
Lisa Kobialka
Meghan A. Wharton
King & Spalding, LLP
333 Twin Dolphin Drive
Suite 400
Redwood Shores, California 94065-6109
(650) 590-7100

Dated: June 8, 2009
919712

VERIFICATION

I, Michael T. McKibben, for and on behalf of Plaintiff Leader Technologies, Inc. ("Leader"), hereby certify that:

I am the Chairman and Founder of Leader and am duly authorized to execute this verification on behalf of Leader.

I have read Leader's First Supplemental Response to Facebook's Interrogatory No. 10. ("Supplemental Response"), and I am familiar with the factual statements contained therein.

To the best of my knowledge, information and belief formed after a reasonable inquiry under the circumstances, the statements of fact made in Leader's Supplemental Response are true and well-grounded in fact.

Although the truth of all said Supplemental Response may not be known to me personally, they are based in whole or in part on information received from others or derived from corporate records.

I declare under penalty of perjury under the Laws of the United States that the above statements are true and correct.

June 8, 2009
Date

Michael T. McKibben
Michael T. McKibben

EXHIBIT 1

Looking at the code for Web (my comments in [ ] 's) :

```
package com.leader.osapplication.board;

import java.util.*;
import com.leader.util.*;
import com.leader. debug.*;
import com.leader.persist.*;
import com.leader.persist.vbsf.*;
import com.leader. osapplication.*;
import com.leader. osapplication.field.*;
import com.leader.osapplication.util.*;
import com.leader. osapplication.actions.*;
import corn.leader. osapplication.framework.*;
import corn.leader. osapplication.exception.*;
import corn.leader. osapplication.interfaces.*;
import com.leader. osapplication.sessionstate.*;

/**
    * A collections of boards with connected relationships tying them
together.
    * The stereotypical example is an org chart in a company where each
person is
    * a node on the web.
    *
    * @author Jeff R. Lamb
    * @author Betsy Foote
    * @author Eric Rosenberg
    */
public class Web extends Content {

    public static final String RELATIONSHIPS LIST FIELD ID =
"existingRelationshipaList";
    public static final String CHILD_BOARD_FIELD_ID = "childBoard";
    public static final String PARENT BOARD FIELD ID "parentBoard";

[These are the relationships that make up the web. If a board
participates in any relationship in this collection, then they are part
of this web]

    private Collection relationships =
CollectionFactory.getPersistenceCapableCollection ( ) ;

[Webs are named to allow them to be easy to work with for the users]

    private String name;

/**
    * VBSF required no argument constructor.
    */
private Web( ) {
    super ( );
}

/**
```

```
        * Constructor
        * @param name the name to give this Web
        */
    public Web(String name) {
        this ( );
        this.name = name;
        }

        / /CI
        public ContentInterface newContent (Map pairs, RequestState
    requestState) throws LeaderException {
            return new Web(TextField.convert ("name", pairs));
        }

        / /CI
        public void setCurrentValues(Map pairs, Requeststate requestState){
            if (pairs.containsKey("webNameTextField") ) {
                setName ((String) pairs.get ("webNameTextField") ) ;
            }
        }

        / /CI
        public String getValidForAddErrorMessage( ) {
            String errorMessage = null;
            if(getName( ) == null || "".equals(getName( ).trim( ))) {
                errorMessage = "You must designate a name for your Web.";
            }
            return errorMessage;
        }

        / /CI
        public int getContentToolCode ( ) {
            return LeaderConstants.BOARD_WEB_TOOL;
        }

        /**SE*/
        public String getName( ) {
            return name;
        }

        /**SE*/
        public void setName(String name){
            this.name = name;
        }

        / *-*
        * Add a WebRelationship to the Web.
        * @param relationship The relationship to add.

            */
        public void addWebRelationship (WebRelationship relationship) {
            if(relationship !=null){
                relationships.add (relationship) ;
            }
```

```
    }

    /**
        * Remove a WebRelationship from the Web.
        * @param relationship The relationship to remove.
        * /
    public void removeWebRelationship(Webaelationship relationship) {
        if(relationship != null) {
            relationships.remove (relationship);
        }
    }

    /**
        * Remove a WebRelationship from the Web.
        * @param relationshipId The object id of the relationship to remove.
        */
    public void removeWebRelationship(Long relationshipId) {
        if(relationshipId != null) {
            Iterator iterator = relationships.iterator( ) ;
            while (iterator.hasNext( )) {
                WebRelationship relationship =
(WebRelationship) iterator.next ( );
                if(relationshipId.equals(relationship.getId( ))) {
                    removeWebRelationship (relationship);
                }
            }
        }
    }

    /**
        * Get all the WebRelationships on this Web. If there are no
relationships,
        * return a 0 length array.
        * @return WebRelationship array.
        */
    private WebRelationship [ ] getWebRelatibnships ( ) {
        return (WebRelationship [ ] ) new ArrayList (relationships) .toArray(new
WebRelationship[relationships.size ( ) ] ); //WebRelationship
[ ] ) relationships.toArray(new WebRelationship[reiationships.size( ) ] );
    }

    /**
        * Determine whether a given board is in this web.
        * @param board Board we want to check on.
        * @return boolean True if board is in this web, false otherwise.
        */
    public boolean contains(Board board) {
        List websoards = getBoardsList ( ) ;
        return webBoards.contains (board);
    }

    /**
```

```
        * Get all the board included in this Web. If there are no
relationships,
        * and hence no boards, return an empty List.
        * @return Board[ ] Array of boards in this Web.
        */
    public List getBoardsList ( ) {
        List boardList = new ArrayList ( ) ;
        WebRelationship [ ] relations = getWebRelationships( ) ;
        for (int i=0; i < relations.length; i++) {
            Board parent = relations[i].getParent( );
            Board child = relations[i].getChild( );
            if (!boardList.contains (parent)) boardList.add(parent);
            if (!boardList.contains(child)) boardList.add(child);
        }
        return boardList;
    }

    /**
        * Get all the Children of a Board on this Web.
        * @param board the board to find children of.
        * @return Set of children Boards. 0 size set if board parameter is
null
        * or when there are no children.
        */
    public Set getChildren(Board board) {
        Set childrenSet = new HashSet ( );
        if(board == null) {
            return cbildrenSet;
        }
        Iterator allRelationships = relationships.iterator( );
        while (allRelationships.hasNext ( ) ) {
            WebRelationship relationship =
(WebRelationship) allRelationships .next ( );
            if (relationship.getParent ( ) .getId ( ) .equals (board.getId ( ) ) ) {
                childrenSet.add (relationship.getChild( ) ) ;
            }
        }
        return childrenSet;
    }

    /**
        * Get all the Parents of a Board on this Web.
        * @param board the board to find parents of.
        * @return Set of parent Boards. 0 size set if board parameter is
null
        * or when there are no parents.
        */
    public Set getParents (Board board) {
        Set parentsSet = new HashSet( );
        if (board == null)
            return parents Set;
        }
```

```
            Iterator allRelationships = relationships.iterator ( );
            while (allRelationships.hasNext ( ) ) {
            -   WebRelationship relationship =
    (WebRelationship) allRelationships.next ( ) ;
5               if (relatiorishiip.getChild( ) .getId( ).equals(board.getId( ) ) ) {
                    parentsSet.add(relationship.getParent ( ) };
                }
            }
            return parentsSet;
10      }

        /**
            * Get all the Peers (all children of all parents of the board).
            * @param board the board to find siblings of.
15          * @return Set of Boards. 0 size set if board parameter is null
            * or when there are no peers.
            */
        public Set getPeers(Board board) {
            Set childrenOfParents = new Hashset( );
20          if(board == null){
                return childrenOfParents;
            }
            Set parentBoards = getParents (board);
            Iterator parentBoardsIterator = parentBoards.iterator( );
25          while (parentBoardsIterator.hasNext ( ) } {
                Set children = getChildren((Board)parentBoardsIterator.next ( ) ) ;
                childrenOfParents.addAll (children);
            }
            childrenOf Parents.remove (board);
30          return childrenOfParents;
        }

        //CI
        public Field[ ] getDisplayFields(RequestState requestState) throws
35  LeaderException {
            List fields = new ArrayList ( );
            TextField textField = new TextField ("name",getName( ), "Web Name");
            textField.setLinkText(" (Edit)");
            textField.setUrlId(LeaderConstants.BOARD_WEB_TOOL,"""+getId( ) );
40          FieldUtilities .jmakeFieldAToolActivator (textField, requestState,
    this, getContentToolCode ( ) , getContentToolCode( ) );
            fields.add(textField);
            Field[ ] dateFields = DateField.getComponentFields (new
    DateTimeField(getLastModified( ) ) ) ;
45          dateFields[0].setTitle("Last Modified Date");
            fields.add(dateFields[0]);
            fields.add(dateFields[1]);
            return (Field[ ]) fields.toArray(new Field{fields.size( ) ]};
        }
50
        //CI
        public String getDisplayName( ) }
```

```
        return "Web" ;
    }

    //CI
    public Form getForm(Requeststate requestState,int displayCode,int
toolCode) {
        Debug.println ("Web.getForm: for " + this, Debug.DEBUG);
        Form form = new ConcreteForm ("webForm", "General Web Attributes");
        int pageIndex = 0;
        int selectedIndex = requestState.getMultiPageIndex( );
        toolCode = getContentToolCode ( );

        //Web name sub-form.
        Page page = new ConcretePage ("createWebPage", pageIndex,
    selectedIndex);
            SubForm sub = new ConcreteSubForm ("webNameSubForm","Web name");
            sub.add(new TextField("webNameTextField", (getName( ) != null ?
    getName( ) : ""), "Web name", true));
        page.add (sub) ;

        //Existing relationships sub—form.
        sub = new ConcreteSubForm ("existingWebRelationshipsSubForm",
"Existing Web Relationships");
        sub.add(getWebRelationshipsListField(requestState.getPairsMap ( ) ) );

        InterfaceActionaction = new
InterfaceAction ("removeRelationship", "Remove Relationship", toolCode,
true);

action.addActionListener (RemoveWebRelationshipActionListener.GLOBAL);
        action.addInterfaceListener (AddInterfaceListener.GLOBAL);
        action.setErrorInterfaceListener (AddInterfaceListener.GLOBAL);
        sub.addAction (action);
        page.add (sub) ;

        //Add new Relationships sub-form
        sub = new ConcreteSubForm("createRelationshipsSubForm", "Create New
Relationship");
        SingleSelectGroupKeyField boardDropDown = new
BoardKeyField (PARENT_BOARD_FIELD_ID, "Parent Board", null,
requeststate.getCurrentUser( ).getId( ) );
        sub.add (boardDropDown);
        boardDropDown = new BoardKeyField(CHILD_BOARD_FIELD_ID, "Child
Board", null, requestState.getCurrentUser( ).getId( ) );
        sub.add (boardDropDown);
        action = new InterfaceAction ("addRelationship", "Add
Relationship", toolCode,true);
        action.addActionListener (AddWebRelationshipActionListener.GLOBAL);
        action.addInterfaceListener (AddInterfaceListener.GLOBAL);
        action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL) ;
        sub.addAction (action);
        page.add (sub) ;
```

```
            form.add (page);
            return form;
        }

        /**VBSF*/
        private Collection getRelationshiopsCollection( ) {
            return relationships;
        }

        /**VBSF*/
        private void setRelationshipsCollection(Collection collection){
            -this.relationships = collection;
        }

        /**
            * Return a Field representing a list view of the web relationships
    in this
            * web. This is used by the getForm method, and by the
    MyContext Interface.
            * @param pairs SE
            * @return a Field
            */
        public Field getWebRelationshipsListField(Map pairs) {
            Iterator iterator = relationships.iterator( );
            List displayFieldsList = new ArrayList( );
            Long[ ] keys = new Long[relationships.size( ) ];
            for(int i=0; iterator.hasNext ( ); i++) {
                WebRelationship relationship = (WebRelationship) iterator .next( );
                keys[i] = relationship.getId( );
                displayFieldsList.add(relationship.getDisplayFields( ) );
            }
            Long[ ] selectedKeys =
    MultiSelectListKeyField.convert (RELATIONSHIPS_LIST_FIELD_ID, pairs);
        Field[ ] [ ] displayFields = (Field[ ] [ ] ) displayFieldsList.toArray(new
    Field[relationships.size( ) ] [0] );
        MultiSelectListKeyField relationshipsList = new
    MultiSelectListrCeyField (RELATIONSHIPS_LIST_FIELD_ID, keys, "Existing
    Web Relationships", selectedKeys, displayFields) ;
            return relationshipstist;
        }
    }

    [END Web.java]

    Looking at the code for WebSlice.java:

    package com.leader.osapplication.board;

    import com.leader.osapplication.framework.*;
    import com.leader.osapplication.*;
    import com.leader.osapplication.util.*;
    import com.leader.osapplication.exception.*;
```

```
import com.leader.osapplication.sessionstate.*;
import com.leader.debug.*;
import java.util.*;

/**
    * A collection of enough information to isolate a set of boards from
the set
    * of all boards. This is typically codified as a Web to use, a
starting board
    * and a Traversal. The Traversal is then used to travel across the Web
from
    * the starting board and return a list of Boards.
    *
    * @author Jeff R. Lamb
    * @author Eric Rosenberg
    */
public class Webslice extends AbstractPersistedObject{

    private Web web;

    private Board board;
    private Traversal traversal;

    /**VBSF*/
    private WebSlice ( ) {
        super ( ) ;
    }

    /**
        * Constructor
        * @param webToUse which Web is this WebSlice a slice of
        * @param boardToUse when you start moving around the Web, where do
you
        * start from?
        * @param traversalToUse what traversal (strategy) should be used to
        * move around the Web to carve out this WebSlice
        */
    public Webslice (Web webToUse, Board boardToUse, Traversal
traversalToUse) {
        this ( ) ;
        setWeb (webToUse) ;
        setBoard (boardToUse) ;
        setTraversal (traversalToUse) ;
    }

    /**
        * Return the boards that are currently part of this webslice. This
can
        * change as the web that the webslice lies on is edited.
        * @return the boards that are a member of the slice
        */
    public Board [ ] getBoards * ) {
        return getTraversal ( ) .getBoards(web, board) ;
```

```
    }

    /**
        * Specify the web that that this webslice is taken from.
        * @param webToUse the web to use if coming up with the set of boards
the
        * web slice represents
        */
    public void setWeb(Web webToUse){ this.web = webToUse; }

    /**
        * Get the web that the webslice is taken from.
        * @return web that the web slice is a part of
        */
    public Web getweb ( ) {  return this.web;}

    /**
        * Specify the board that is the starting point for this webslice
        * @param boardToUse the board that is the starting point for the
webslice
        * @throws IllegalArgumentException if boardToUse is not in this web
        */
    public void setBoard (Board boardToUse){
        / / These null checks are to bypass the 'contains' check when VBSF
may
        / / be using this method with a null value or before setting web.
        if (boardToUse = = null | | web = = null | | web.contains(boardToUse) ) {
            this.board = boardToUse;
        }
        else{

            / / throw an IllegalArgumentException if boardToUse is NOT in
            / / webToUse.
            throw new IllegalArgumentException("The starting Board of a
WebSlice must be part of the Web.");
        }
    }

    /**
        * Get the board that is the starting point for the webslice
        * @return board that is the starting point for the webslice
        */
    public Board getBoard ( ) { return this.board;}

    /**
        * Specify the traversal used to get the boards for this webslice
        * @param traversalToUse SE
        */
    public void setTraversal (Traversal traversalToUse) {
        this.traversal = traversalToUse;
    }
```

```
        /**
            * Get the traversal used to get the boards for this webslice
            * @return traversal used to get the boards for this webslice
            */
5       public Traversal getTraversal ( ) { return this.traversal; }

        /**VBSF*/
        private int getTraversalCode( ) { return
    TraversalFactory.getCode (traversal); }
10
        /**VBSF*/
        private void setTraversalCode (int code) { this.traversal =
    TraversalFactory.getTraversal (code) ; }

15  }
```

IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

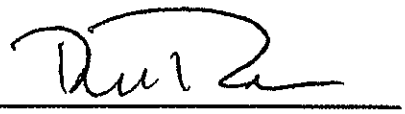
CERTIFICATE OF SERVICE

I, Philip A. Rovner, hereby certify that on June 8, 2009, true and correct copies of the within document were served on the following counsel of record, at the addresses and in the manner indicated:

BY EMAIL AND FIRST CLASS MAIL

Thomas P. Preston, Esq.
Steven L. Caponi, Esq.
Blank Rome LLP
1201 Market Street
Wilmington, DE 19801
Preston-T@blankrome.com
caponi@blankrome.com

Heidi L. Keefe, Esq.
Mark R. Weinstein, Esq.
Craig W. Clark, Esq.
Melissa H. Keyes, Esq.
White & Case LLP
3000 El Camino Real
5 Palo Alto Square, 9th Floor
Palo Alto, CA 94306
hkeefe@whitecase.com; mweinstein@whitecase.com
cclark@whitecase.com; mkeyes@whitecase.com



Philip A. Rovner (#3215)
Potter Anderson & Corroon LLP
Hercules Plaza
P. O. Box 951
Wilmington, DE 19899
(302) 984-6000
provner@potteranderson.com

EXHIBIT B

Clark, Craig W.

From: Young, Rowena [RowenaYoung@KSLAW.com]
Sent: Sunday, June 28, 2009 2:21 PM
To: Clark, Craig W.
Cc: Kobialka, Lisa; Andre, Paul; Hannah, James; Rovner, Philip A.; Keefe, Heidi L.; Weinstein, Mark R.; caponi@blankrome.com; Kastens, Kristopher
Subject: RE: LTI v. Facebook

Dear Craig,

You again misrepresent Leader's position concerning it's detailed response to Interrogatory No. 10. You have our position as set forth in prior correspondence.

Best regards,
Rowena

From: Clark, Craig W. [mailto:cclark@paloalto.whitecase.com]
Sent: Friday, June 26, 2009 4:33 PM
To: Young, Rowena
Cc: Kobialka, Lisa; Andre, Paul; Hannah, James; Rovner, Philip A.; Keefe, Heidi L.; Weinstein, Mark R.; caponi@blankrome.com; Kastens, Kristopher
Subject: RE: LTI v. Facebook

Rowena,

Your position that (1) LTI is not withholding responsive information, but that (2) LTI has not identified all portions of the provisional application that it contends disclose the elements of the claims in the patent, is inconsistent. LTI is required to produce all responsive information in its possession in response to this proper discovery or to provide some principled basis for withholding it. LTI has admitted that it has additional information uniquely in its possession but has provided no basis for withholding that information. As I explained during our call, Rule 33 requires LTI's complete response and Rule 37 treats an incomplete response as a failure to respond. *See* Fed. R. Civ. P. 37(a)(3).

If you would like to discuss the issue further, we can do so on our Monday call. Otherwise, we plan to contact the Court for resolution.

Thank you,

Craig W. Clark | WHITE & CASE LLP
3000 El Camino Real | 5 Palo Alto Square | 9th Floor | Palo Alto, CA | 94306
t + 1 650 213 0307 | f + 1 650 213 8158 | cclark@whitecase.com

---

From: Young, Rowena [mailto:RowenaYoung@KSLAW.com]
Sent: Friday, June 19, 2009 3:56 PM
To: Clark, Craig W.
Cc: Kobialka, Lisa; Andre, Paul; Hannah, James; Rovner, Philip A.; Keefe, Heidi L.; Weinstein, Mark R.; caponi@blankrome.com; Kastens, Kristopher
Subject: RE: LTI v. Facebook

Craig,

Your email mischaracterizes a number of Leader's positions with respect to the June 17 meet and confer and we do not believe that such posturing is helpful in resolving the discovery issues facing the parties. However, Leader remains committed to resolving these matters civilly, expeditiously and without the intervention of the Court if possible.

That being said, our position regarding Facebook's interrogatory 10 is that Leader's supplemental response satisfies any requirements placed on it under the applicable rules or case law. While our position remains that the '255 provisional application as a whole provides the necessary support for the subject matter of the claims of the '761 patent, our supplemental response included a detailed claim chart that provided specific examples of where such support can be found in the provisional application on an element-by-element basis for each asserted claim. Therefore, our supplemental response provides detailed support that is more than sufficient to establish the '761 patent's claim of priority to the provisional application and Facebook can move forward with its prior art search. During the meet and confer we stated that we would consider further supplementation if Facebook could provide a reasonable explanation as to why any additional supplementation is necessary at this time. Facebook could provide no other reason for why additional supplementation was required other than because Facebook asked for it. If Facebook can cite to any case law supporting its position that Leader is required to provide additional detail in its response to interrogatory 10, we will certainly consider it.

Leader did not indicate that it "intends to cite unidentified portions of the provisional application in support [of] its priority claim later in the litigation," rather, we stated that Leader may supplement its interrogatory response as a result of future events, such as claim construction. However, and as we repeatedly stated, Leader's response to date is more than adequate to support the claims under what we believe to be any reasonable interpretation.

Regarding Facebook's request for Leader to specify detailed support for unasserted claims, we stated we would certainly supplement our response to interrogatory 10 if any of these claims are later asserted. However, the priority date of unasserted claims is not relevant. Again, if Facebook can cite to any case law supporting its contention that the priority date of unasserted claims is relevant to any issue in this matter, we will certainly consider it.

Accordingly, your assertion that Leader is "withholding responsive information" is unsupported and without merit.

If you have further support for any of your positions please provide them to us and we will consider them. If you would like to discuss these matters further, please let me know. Otherwise, please feel free to make appropriate arrangements to contact the Court and the parties can proceed per the June 12 Order Regarding Discovery Matters.

Best regards,

Rowena

From: Clark, Craig W. [mailto:cclark@paloalto.whitecase.com]
Sent: Thursday, June 18, 2009 4:51 PM
To: Young, Rowena
Cc: Kobialka, Lisa; Andre, Paul; Hannah, James; Rovner, Philip A.; Keefe, Heidi L.; Weinstein, Mark R.; caponi@blankrome.com
Subject: LTI v. Facebook

Rowena,

Following up on our June 17, 2009 call, I understand that LTI's supplemental response to Facebook Interrogatory No. 10 is not complete and fails to cite all portions of the provisional application that LTI contends disclose each limitation of each claim of the later-filed application. I also understand

that LTI contends that its supplement, which provides only "non-limiting" and "non-exhaustive" exemplary citation to the provisional application, is sufficient, in LTI's view, and will not be further supplemented at this time. You also expressed that LTI intends to cite unidentified portions of the provisional application in support its priority claim later in the litigation, for example, during claim construction. Finally, you took the position that LTI need not provide support for the unasserted claims of the patent despite LTI's position that it may later assert such claims.

As I explained yesterday, Facebook disagrees. Facebook Interrogatory No. 10 calls for LTI to provide a complete response specifically identifying each portion of the provisional application that LTI contends discloses each limitations of each claim LTI contends is entitled to the priority date of the provisional application. Facebook disagrees that LTI has established--or that it can establish--such priority. However, that disagreement is not germane to this discovery dispute. At this stage of the case, Facebook is entitled to all facts in LTI's possession responsive to proper discovery, like Facebook Interrogatory No. 10. By your admission, LTI is withholding responsive information.

Please advise whether LTI will reconsider its position or whether we will have to further involve the Court.

Sincerely,

Craig W. Clark | WHITE & CASE LLP
3000 El Camino Real | 5 Palo Alto Square | 9th Floor | Palo Alto, CA | 94306
t + 1 650 213 0307 | f + 1 650 213 8158 | cclark@whitecase.com

EXHIBIT C

A/PROV

PTO/SB/16 (10-01)
Approved for use through 10/31/2002 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

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53 (c).

Express Mail Label No.

INVENTOR(S)		
Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
Michael T. Jeff R.	McKibben Lamb	Westerville, Ohio Westerville, Ohio

☐ *Additional inventors are being named on the ___ separately numbered sheets attached hereto*

TITLE OF THE INVENTION (500 characters max)

METHOD FOR DYNAMIC ASSOCIATION OF ELECTRONICALLY STORED INFORMATION WITH ITERATIVE WORKFLOW CHANGES

Direct all correspondence to: **CORRESPONDENCE ADDRESS**

☒ Customer Number 25534 → *Place Customer Number Bar Code Label here*

OR *Type Customer Number here*

☐ Firm or Individual Name					
Address					
Address					
City		State		ZIP	
Country		Telephone		Fax	

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒ Specification *Number of Pages* 18 ☐ CD(s), Number

☐ Drawing(s) *Number of Sheets* ☐ Other (specify)

☐ Application Data Sheet. See 37 CFR 1.76

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT

☒ Applicant claims small entity status. See 37 CFR 1.27.

☒ A check or money order is enclosed to cover the filing fees

FILING FEE AMOUNT ($): 80.00

☐ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number:

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

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government

☒ No

☐ Yes, the name of the U S Government agency and the Government contract number are _____

Respectfully submitted,

SIGNATURE Frederick N. Samuels

Date 12/11/2002

REGISTRATION NO. *(if appropriate)* 34715

Docket Number: 547.0003P

TYPED or PRINTED NAME Frederick N. Samuels

TELEPHONE 202-331-8777

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

This collection of information is required by 37 CFR 1.51. The information is used by the public to file (and by the PTO to process) a provisional application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 8 hours to complete, including gathering, preparing, and submitting the complete provisional application to the PTO. 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, Washington, D.C., 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Box Provisional Application, Assistant Commissioner for Patents, Washington, D.C. 20231.

[redacted]

METHOD FOR DYNAMIC ASSOCIATION OF ELECTRONICALLY STORED INFORMATION WITH ITERATIVE WORKFLOW CHANGES

I. Field of the Invention

[0001] This invention relates to management and storage of electronic information. More particularly, this invention relates to new structures and methods for creating relationships between users, applications, files and folders.

II. Background of the Invention

[0002] Digital communications solutions are presently supplied to users in ways that are completely divorced from their business context. A particular item of communication provides little or no inherent understanding of how that communication furthers the purpose and intent of the group or enterprise. In other words, an email inbox collects emails about all topics, business and personal. The email application itself is not discerning about topic, priority or context beyond perhaps rudimentary "message filters" that will look for certain key words or people then place those items in target folders. Generally, it simply presents a sequential list of messages received. Similarly, a fax machine receives fax pages in sequence. A fax machine is not discerning about topic, priority or context. It simply outputs fax pages. Once received, it remains the task of the recipient to sort, categorize and organize these items of communication in ways most meaningful to that person. The organization task generally occurs outside the context of the particular communications tool itself.

[0003] Typical methods for organization of communications are limited and fragmented. For example, for an email, the recipient may either leave all email in the inbox or move it to another electronic folder. For a fax, the recipient is likely to place that fax in a file folder that is identified by project name or name of recipient. These

typical methods of organizing communications are wholly inadequate for a number of reasons:

[0004] 1. **Organization** – the recipient is left to do all the work of organization and categorization of the communications rather than having the systems themselves doing that work for them, automatically.

[0005] 2. **Leadership** – the linkage between business strategy and an individual act of communication is non-existent.

[0006] 3. **Categorization** – the items themselves rarely apply to only one topic of interest. As such, under current systems, the items would need to be manually stored in multiple locations (either electronic or "brick and mortar" folders). For example, a fax letter to a sales manager may contain information about contact addresses, market intelligence data, specific product requests, and financial accounting.

[0007] 4. **Knowledge Sharing** – items often relate to organizational issues for which one or more work groups need access; access that is denied when the recipient "buries" that item in his/her personal filing system, electronic or otherwise.

[0008] 5. **Context** – prior art communications tools do not know the business and/or personal context(s) within which files are created and used. For example, a person may create three files in a word processor, one relating to sales, the second relating to operations and the third relating to his son's football team. However, the word processor itself has no way of knowing to automatically store those three files in at least three different places.

[0009] 6. **Security & Privacy** – the applications and their file storage methods are generally insecure; they do not conform to a single, dependable security model.

[redacted]

[0010] Known software applications create and store files outside of a contextual framework. For example, when a user creates a Microsoft Word (*.doc) file in Microsoft Word 2000, the user must select a single folder within which to store that file. The file may be stored in an existing folder or the user may create a new folder to receive the file. This file management method is known as Lightweight Directory Application Protocol (LDAP). LDAP borrowed the physical world paper file management scheme where a machine/application creates files, stores those files in individual folders and stores those folders in cabinets. Under this scheme, context is completely independent of the application. File context is limited to the decision made by the user about which folder the file should be stored. The user decision does not adequately represent reflect the true context of the file given that the file may contain information that could reasonable be stored in multiple folders.

[0011] Another limitation of LDAP is that little or no information is contained within the file about the user and the context and circumstances of the user at the time the file was created. Current processes designed to add context to files such as the "meta-data" tagging approach, involve having a knowledge officer view files after they have been stored and create meta-data tags with additional key words associated with the file for search purposes.

[0012] Notwithstanding the usefulness of the above-described methods, a need still exists for a communications tool that associates files generated by applications with individuals, groups and topical context.

[redacted]

III. Summary of the Invention

[0013] It is an object of the invention to provide a communication tool that seamlessly facilitates, collects, compiles and distributes communication data.

[0014] It is a further object of the invention to provide a communication tool that links communication data to enterprise leadership priorities.

[0015] It is another objective of the invention to provide a communication tool that performs communications tasks while simultaneously reminding the user of his/her individual work priorities.

[0016] It is still a further object of the invention to provide a communication tool that automatically stores contextual information relating to an item of communication and utilizes that contextual in performance of communication tasks.

[0017] Sill another object of the invention is to provide a communication tool that integrates two or more different communication applications such as telephony, unified messaging, decision support, document management, portals, chat, collaboration, search, vote, relationship management, calendar, personal information management, profiling, directory management, executive information systems, dashboards, cockpits, tasking, meeting, conferencing, etc. into a common application.

[0018] Still a further object of the invention is to provide a structure for defining relationships between complex collections of data.

[0019] Yet another object of the invention is to provide a process for automating workflow between multiple entities.

[0020]

[0021] Given the following enabling description, the invention should become evident to a person of ordinary skill in the art.

IV. Description of the Embodiments

[0022] In the past, intuitive, dynamic, changeable workflow processes have proved to be too dynamic and expensive for automation. The present invention utilizes "boards" and "webs" to automate workflow processes and define relationships between data and applications. As users create and change their contexts, the files and applications automatically follow, dynamically capturing those shifts in context.

[0023] As used herein, a "board" is defined as a collection of data and application functionality related to a user-defined topic. For example, a user defined topic may be a department of a company or a project that involves the company. In the case of a project, the board preferably includes all of the data relating to that project including email, tasks, calendar events, ideas, discussions, meetings, phone calls, files, contact records, people, etc. Data and applications may be grouped in a board based on the identity of the tag.

[0024] As used herein, the term "web" refers to a collection of interelated boards. Boards in a web may have, for example, a parent-child relationship. A given board may have more than one parent and may have more than one child. A board may not be its own child or its own parent. However, boards may have various relationships to each other. For example, a board may be part of a circular relationship of any complexity such as the following: A is parent to B; B is parent to C and C is parent to A.

[0025] In accordance with the invention, webs may be used to maintain the location of content within a complex and changing set of boards and support automation of the

workflow process. Automation of the workflow process may shown by the following example.

Example

The workflow process to be automated is A→B→C. Three different people are assigned to each item. Therefore A(1,2,3) → B(4,5,6) → C(7,8,9). The workflow change desired in this example is A → B/C → C.

In the known environment, LDAP, it is necessary for the automation sequence to predetermine how work data flows from A to B and C. Then, the automation module for inputs to D must be spelled out and rewritten to consolidate split input from B and C. As such, the automation support for this workflow change will always lag behind the ability of the people involved to start working with the new workflow assumptions.

In contrast, in accordance with the present invention, webs and boards are preferably the context for applications, files and folders. Hence, the workflow process may be readily reorganized by making a change to one or more of the webs and boards.

In preferred embodiments, webs may be utilized to maintain the location of content within a complex and changing set of boards. Content is preferably associated with a routing algorithm referred to herein as a webslice. Thus the content has an intelligent quality whereby upon a change of structure of the web, the content knows which board or boards it should be on both before and after the change of structure. In keeping with a preferred aspect of the invention, the location of the content may be

determined at dynamically at run using the routing algorithm. Alternatively, the loction of content may be determined by detecting changes in structure, detecting the temporary location fo the content on the boards in the routing algorithm before and after the change and adjusting the location of the affected content as part of the change in structure.

ATTACHMENT 2

"board" Module

"WEB VERSION 1" WORKING DESCRIPTION

Webs are collections of boards and a collection of parent-child relationships between those boards. Boards in a web may have more than one parent and my have more than one child. A board may not be its own child (and thus may not be its own parent), but may participate in a circular relationship of any complexity (A is parent to B. B is parent to C. C is parent to A).
WebSlices are a way of representing an algorithm that's ultimate output is a set of boards. A webslice consists of a Web, a starting board, and a traversal (of arbitary complexity). Take for example a web of boards a b and c where b and c are children of a. A webslice that referenced this board, started at a and used a traversal of "all cihldren" would return b and c. If the smae traversal on the same web had started at b, the empty set would be the result.
Webs can be utilized to maintain the location of content within a complex and changing set of boards. If content has a webslice associated with it, then any change of structure in the web would still result in the content (with the webslice) knowing what boards it should be on both before and after the change of structure. Actually effecting this change of location can be done by allowing the "location" to be determined dynamically at run time using the webslice or can be accomplished by detecting changes in structure, detecting the (temporary) location of the content on the boards in the slice before and after the change and adjusting the location of the affected content as part of the change in web structure.

CIAP also facilitates a new business workflow process. Workflow automation is currently a site-specific effort. The workflow between A to B to C must be clearly specified in all its variables prior to automation. Automation fixes this workflow in code. Changes to the workflow require manual changes to the code. Predictable, repeatable, transactional and hierarchical workflow processes are best suited to this approach. LDAP and hierarchical storage models work best in this environment. Multiple applications work independently of the storage, generating and reporting data to and from the storage model.

Intuitive, dynamic, changeable workflow processes have proved too dynamic and expensive for automation. CIAP changes that. CIAP is key off users and context, not off of applications and files. As users create and change their contexts, the files and applications automatically follow, dynamically capturing those shifts of context.

Professional services consulting is currently held hostage by a cumbersome, expensive, time-consuming and often dehumanizing process known as "change management." The modus operandi of these firms is to for the implementation of that firm's change model. These models have a variety of names: Balanced Scorecard, Critical Success Factors, Vital Signs, etc. These models are often intended to replace traditional "command and control" models. Generally this is an either/or process. This change in the workflow practices in a company is time consuming. Generally these new processes begin a spate of new automation projects to support these changes. However, as any professional services person knows, the automation, like the change process itself, is iterative. Typically 50% of the changes initially championed will not work. Then 25% of the secondary changes will not work. Then, 12.5 of the third round of changes will not work... and so on. As a consequence, automation always lags behind, many times in terms of years.

CIAP allows professional services providers to support IT automation professionals with an approach to automation support of workflow changes that changes and adapts as the organization learns with little to no change to the underlying IT architecture.

To use a simple example, A → B → C is the workflow process we want to automate. We assign 3 different people to each item, Therefore A(1,2,3) → B(4,5,6) → C(7,8,9).

LDAP Implementation

Persons (1,2,3,4,5,6,7,8,9) → Applications → **A**files, **B**files, **C**files → **A**folders, **B**folders, **C**folders.

Now let's say a workflow change is proposed to look like this: A → B/C → D. In an LDAP environment, before the people involved have any automation support for this change, the automation sequence *pre-determine* how work data flows from A to B & C. Then, the automation module for inputs to D must be *spelled out and rewritten* to consolidate split input from B & C. In other words, the automation support for this change will always lag behind the ability of the people involved to start working with the new workflow assumptions. LDAP structure forces a regimented, minimalistic approach to the automation of workflow processes.

CIAP Implementation

Persons (1,2,3,4,5,6,7,8,9) → Web → **A**board,**B**board,**C**board (incl. Applications, Files, Folders)

Now let's say the workflow changes to A → B/C → D. In a CIAP environment a simple adjustment is made to the webs & boards table and the entire workflow process is reorganized with all the relevant data files appropriate reorganized and available. This should always be the first step in the change process. The first step in the change process should always be the instantaneous reorganization of the people and topic associations along with the communications tools. At this stage in the change, no predictable, repeatable, transactional or hierarchical process can be established. That can only come with time and consistency. Some processes must remain flexible, unpredictable, yet they

[redacted]

are processes nonetheless. CIAP allows for the simultaneous automation of repeatable and dynamic processes.

In CIAP, the People, Webs and Boards become the automatic context for Applications, Files and Folders. In LDAP the Applications, Files and Folders have *no* inherent relationship to the People or their Context. The implications of this difference on the automation of workflow process are profound.

Looking at the code for Web (my comments in []'s):

```
package com.leader.osapplication.board;

import java.util.*;
import com.leader.util.*;
import com.leader.debug.*;
import com.leader.persist.*;
import com.leader.persist.vbsf.*;
import com.leader.osapplication.*;
import com.leader.osapplication.field.*;
import com.leader.osapplication.util.*;
import com.leader.osapplication.actions.*;
import com.leader.osapplication.framework.*;
import com.leader.osapplication.exception.*;
import com.leader.osapplication.interfaces.*;
import com.leader.osapplication.sessionstate.*;

/**
 * A collections of boards with connected relationships tying them
together.
 * The stereotypical example is an org chart in a company where each
person is
 * a node on the web.
 *
 * @author Jeff R. Lamb
 * @author Betsy Foote
 * @author Eric Rosenberg
 */
public class Web extends Content {

  public static final String RELATIONSHIPS_LIST_FIELD_ID =
"existingRelationshipsList";
  public static final String CHILD_BOARD_FIELD_ID = "childBoard";
  public static final String PARENT_BOARD_FIELD_ID = "parentBoard";

[These are the relationships that make up the web. If a board
participates in any relationship in this collection, then they are part
of this web]

  private Collection relationships =
CollectionFactory.getPersistenceCapableCollection();

[Webs are named to allow them to be easy to work with for the users]

  private String name;
```

[redacted]

```
/**
 * VBSF required no argument constructor.
 */
private Web(){
  super();
}

/**
 * Constructor
 * @param name the name to give this Web
 */
public Web(String name){
  this();
  this.name = name;
}

//CI
public ContentInterface newContent(Map pairs, RequestState
requestState) throws LeaderException {
  return new Web(TextField.convert("name",pairs));
}

//CI
public void setCurrentValues(Map pairs, RequestState requestState){
  if (pairs.containsKey("webNameTextField")){
    setName((String)pairs.get("webNameTextField"));
  }
}

//CI
public String getValidForAddErrorMessage(){
  String errorMessage = null;
  if(getName() == null || "".equals(getName().trim())){
    errorMessage = "You must designate a name for your Web.";
  }
  return errorMessage;
}

//CI
public int getContentToolCode(){
  return LeaderConstants.BOARD_WEB_TOOL;
}

/**SE*/
public String getName(){
  return name;
}

/**SE*/
public void setName(String name){
  this.name = name;
}

/**
 * Add a WebRelationship to the Web.
 * @param relationship The relationship to add.
```

[redacted]

```
     */
    public void addWebRelationship(WebRelationship relationship){
      if(relationship != null){
        relationships.add(relationship);
      }
    }

    /**
     * Remove a WebRelationship from the Web.
     * @param relationship The relationship to remove.
     */
    public void removeWebRelationship(WebRelationship relationship){
      if(relationship != null){
        relationships.remove(relationship);
      }
    }

    /**
     * Remove a WebRelationship from the Web.
     * @param relationshipId The object id of the relationship to remove.
     */
    public void removeWebRelationship(Long relationshipId){
      if(relationshipId != null){
        Iterator iterator = relationships.iterator();
        while(iterator.hasNext()){
          WebRelationship relationship =
(WebRelationship)iterator.next();
          if(relationshipId.equals(relationship.getId())){
            removeWebRelationship(relationship);
          }
        }
      }
    }

    /**
     * Get all the WebRelationships on this Web.  If there are no
relationships,
     * return a 0 length array.
     * @return WebRelationship array.
     */
    private WebRelationship[] getWebRelationships(){
      return (WebRelationship [])new ArrayList(relationships).toArray(new
WebRelationship[relationships.size()]); //WebRelationship
[])relationships.toArray(new WebRelationship[relationships.size()]);
    }

    /**
     * Determine whether a given board is in this web.
     * @param board Board we want to check on.
     * @return boolean True if board is in this web, false otherwise.
     */
    public boolean contains(Board board){
      List webBoards = getBoardsList();
      return webBoards.contains(board);
    }

    /**
```

[redacted]

```
    * Get all the board included in this Web.  If there are no
relationships,
    * and hence no boards, return an empty List.
    * @return Board[] Array of boards in this Web.
    */
   public List getBoardsList(){
     List boardList = new ArrayList();
     WebRelationship[] relations = getWebRelationships();
     for (int i=0; i < relations.length; i++){
       Board parent = relations[i].getParent();
       Board child = relations[i].getChild();
       if (!boardList.contains(parent)) boardList.add(parent);
       if (!boardList.contains(child)) boardList.add(child);
     }
     return boardList;
   }

   /**
    * Get all the Children of a Board on this Web.
    * @param board the board to find children of.
    * @return Set of children Boards.  0 size set if board parameter is
null
    * or when there are no children.
    */
   public Set getChildren(Board board){
     Set childrenSet = new HashSet();
     if(board == null){
       return childrenSet;
     }
     Iterator allRelationships = relationships.iterator();
     while (allRelationships.hasNext()){
       WebRelationship relationship =
(WebRelationship)allRelationships.next();
       if (relationship.getParent().getId().equals(board.getId())){
         childrenSet.add(relationship.getChild());
       }
     }
     return childrenSet;
   }

   /**
    * Get all the Parents of a Board on this Web.
    * @param board the board to find parents of.
    * @return Set of parent Boards.  0 size set if board parameter is
null
    * or when there are no parents.
    */
   public Set getParents(Board board){
     Set parentsSet = new HashSet();
     if(board == null){
       return parentsSet;
     }
     Iterator allRelationships = relationships.iterator();
     while (allRelationships.hasNext()){
       WebRelationship relationship =
(WebRelationship)allRelationships.next();
       if (relationship.getChild().getId().equals(board.getId())){
```

[redacted]

```
            parentsSet.add(relationship.getParent());
          }
        }
        return parentsSet;
      }

      /**
       * Get all the Peers (all children of all parents of the board).
       * @param board the board to find siblings of.
       * @return Set of Boards.  0 size set if board parameter is null
       * or when there are no peers.
       */
      public Set getPeers(Board board){
        Set childrenOfParents = new HashSet();
        if(board == null){
          return childrenOfParents;
        }
        Set parentBoards = getParents(board);
        Iterator parentBoardsIterator = parentBoards.iterator();
        while(parentBoardsIterator.hasNext()){
          Set children = getChildren((Board)parentBoardsIterator.next());
          childrenOfParents.addAll(children);
        }
        childrenOfParents.remove(board);
        return childrenOfParents;
      }

      //CI
      public Field[] getDisplayFields(RequestState requestState) throws
    LeaderException{
        List fields = new ArrayList();
        TextField textField = new TextField("name",getName(), "Web Name");
        textField.setLinkText("(Edit)");
        textField.setUrlId(LeaderConstants.BOARD_WEB_TOOL,""+getId());
        FieldUtilities.makeFieldAToolActivator(textField, requestState,
    this, getContentToolCode(),getContentToolCode());
        fields.add(textField);
        Field[] dateFields = DateField.getComponentFields(new
    DateTimeField(getLastModified()));
        dateFields[0].setTitle("Last Modified Date");
        fields.add(dateFields[0]);
        fields.add(dateFields[1]);
        return (Field[])fields.toArray(new Field[fields.size()]);
      }

      //CI
      public String getDisplayName(){
        return "Web";
      }

      //CI
      public Form getForm(RequestState requestState,int displayCode,int
    toolCode) {
        Debug.println("Web.getForm: for " + this, Debug.DEBUG);
        Form form = new ConcreteForm("webForm", "General Web Attributes");
        int pageIndex = 0;
        int selectedIndex = requestState.getMultiPageIndex();
```

[redacted]

```
        toolCode = getContentToolCode();

        //Web name sub-form.
        Page page = new ConcretePage("createWebPage", pageIndex,
    selectedIndex);
        SubForm sub = new ConcreteSubForm("webNameSubForm","Web name");
        sub.add(new TextField("webNameTextField", (getName() != null ?
    getName() : ""), "Web name", true));
        page.add(sub);

        //Existing relationships sub-form.
        sub = new ConcreteSubForm("existingWebRelationshipsSubForm",
    "Existing Web Relationships");
        sub.add(getWebRelationshipsListField(requestState.getPairsMap()));

        InterfaceAction action = new
    InterfaceAction("removeRelationship","Remove Relationship",toolCode,
    true);

    action.addActionListener(RemoveWebRelationshipActionListener.GLOBAL);
        action.addInterfaceListener(AddInterfaceListener.GLOBAL);
        action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
        sub.addAction(action);
        page.add(sub);

        //Add new Relationships sub-form
        sub = new ConcreteSubForm("createRelationshipsSubForm", "Create New
    Relationship");
        SingleSelectGroupKeyField boardDropDown = new
    BoardKeyField(PARENT_BOARD_FIELD_ID, "Parent Board", null,
    requestState.getCurrentUser().getId());
        sub.add(boardDropDown);
        boardDropDown = new BoardKeyField(CHILD_BOARD_FIELD_ID, "Child
    Board", null, requestState.getCurrentUser().getId());
        sub.add(boardDropDown);
        action = new InterfaceAction("addRelationship","Add
    Relationship",toolCode,true);
        action.addActionListener(AddWebRelationshipActionListener.GLOBAL);
        action.addInterfaceListener(AddInterfaceListener.GLOBAL);
        action.setErrorInterfaceListener(AddInterfaceListener.GLOBAL);
        sub.addAction(action);
        page.add(sub);

        form.add(page);
        return form;
      }

      /**VBSF*/
      private Collection getRelationshipsCollection(){
        return relationships;
      }

      /**VBSF*/
      private void setRelationshipsCollection(Collection collection){
        this.relationships = collection;
      }
```

[redacted]

```
  /**
   * Return a Field representing a list view of the web relationships
in this
   * web. This is used by the getForm method, and by the
MyContextInterface.
   * @param pairs SE
   * @return a Field
   */
  public Field getWebRelationshipsListField(Map pairs){
    Iterator iterator = relationships.iterator();
    List displayFieldsList = new ArrayList();
    Long[] keys = new Long[relationships.size()];
    for(int i=0; iterator.hasNext(); i++){
      WebRelationship relationship = (WebRelationship)iterator.next();
      keys[i] = relationship.getId();
      displayFieldsList.add(relationship.getDisplayFields());
    }
    Long[] selectedKeys =
MultiSelectListKeyField.convert(RELATIONSHIPS_LIST_FIELD_ID, pairs);
    Field[][] displayFields = (Field[][])displayFieldsList.toArray(new
Field[relationships.size()][0]);
    MultiSelectListKeyField relationshipsList = new
MultiSelectListKeyField(RELATIONSHIPS_LIST_FIELD_ID, keys, "Existing
Web Relationships", selectedKeys, displayFields);
    return relationshipsList;
  }
}
```

[END Web.java]

Looking at the code for WebSlice.java:

```
package com.leader.osapplication.board;

import com.leader.osapplication.framework.*;
import com.leader.osapplication.*;
import com.leader.osapplication.util.*;
import com.leader.osapplication.exception.*;
import com.leader.osapplication.sessionstate.*;
import com.leader.debug.*;
import java.util.*;

/**
 * A collection of enough information to isolate a set of boards from
the set
 * of all boards. This is typically codified as a Web to use, a
starting board
 * and a Traversal. The Traversal is then used to travel across the Web
from
 * the starting board and return a list of Boards.
 *
 * @author Jeff R. Lamb
 * @author Eric Rosenberg
 */
public class WebSlice extends AbstractPersistedObject{

  private Web web;
```

[redacted]

```
  private Board board;
  private Traversal traversal;

  /**VBSF*/
  private WebSlice(){
    super();
  }

  /**
   * Constructor
   * @param webToUse which Web is this WebSlice a slice of
   * @param boardToUse when you start moving around the Web, where do
you
   * start from?
   * @param traversalToUse what traversal (strategy) should be used to
   * move around the Web to carve out this WebSlice
   */
  public WebSlice(Web webToUse, Board boardToUse, Traversal
traversalToUse){
    this();
    setWeb(webToUse);
    setBoard(boardToUse);
    setTraversal(traversalToUse);
  }

  /**
   * Return the boards that are currently part of this webslice. This
can
   * change as the web that the webslice lies on is edited.
   * @return the boards that are a member of the slice
   */
  public Board[] getBoards(){
    return getTraversal().getBoards(web, board);
  }

  /**
   * Specify the web that that this webslice is taken from.
   * @param webToUse the web to use if coming up with the set of boards
the
   * web slice represents
   */
  public void setWeb(Web webToUse){ this.web = webToUse; }

  /**
   * Get the web that the webslice is taken from.
   * @return web that the web slice is a part of
   */
  public Web getWeb(){ return this.web;}

  /**
   * Specify the board that is the starting point for this webslice
   * @param boardToUse the board that is the starting point for the
webslice
   * @throws IllegalArgumentException if boardToUse is not in this web
   */
  public void setBoard(Board boardToUse){
```

[redacted]

```
    // These null checks are to bypass the 'contains' check when VBSF
may
    // be using this method with a null value or before setting web.
    if (boardToUse == null || web == null || web.contains(boardToUse)){
      this.board = boardToUse;
    }
    else{

      // throw an IllegalArgumentException if boardToUse is NOT in
      // webToUse.
      throw new IllegalArgumentException("The starting Board of a
WebSlice must be part of the Web.");
    }
  }

  /**
   * Get the board that is the starting point for the webslice
   * @return board that is the starting point for the webslice
   */
  public Board getBoard(){ return this.board;}

  /**
   * Specify the traversal used to get the boards for this webslice
   * @param traversalToUse SE
   */
  public void setTraversal(Traversal traversalToUse){
    this.traversal = traversalToUse;
  }

  /**
   * Get the traversal used to get the boards for this webslice
   * @return traversal used to get the boards for this webslice
   */
  public Traversal getTraversal(){ return this.traversal; }

  /**VBSF*/
  private int getTraversalCode() { return
TraversalFactory.getCode(traversal); }

  /**VBSF*/
  private void setTraversalCode(int code){ this.traversal =
TraversalFactory.getTraversal(code);}

}
```

[redacted]