

AO 120 (Rev. 08/10)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK
---	---



In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas, Marshall Division on the following

Trademarks or Patents. (the patent action involves 35 U.S.C. § 292.):

DOCKET NO. TBD	DATE FILED 7/18/2018	U.S. DISTRICT COURT Eastern District of Texas, Marshall Division
PLAINTIFF SAS Institute Inc.		DEFENDANT World Programming Limited, MineQuest Business Analytics, LLC, MineQuest LLC, Angoss Software Corp., Luminex Software, Inc., Yum! Brands, Inc., Shaw Industries Group, Inc., and Hitachi Vantara Corp.
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 7,170,519	1/30/2007	SAS Institute Inc.
2 7,447,686	11/4/2008	SAS Institute Inc.
3 8,498,996	7/30/2013	SAS Institute Inc.
4		
5		

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY <input type="checkbox"/> Amendment <input type="checkbox"/> Answer <input type="checkbox"/> Cross Bill <input type="checkbox"/> Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE
-------	-------------------	------

Copy 1—Upon initiation of action, mail this copy to Director . Copy 3—Upon termination of action, mail this copy to Director
 Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 7,447,686 B2
APPLICATION NO. : 10/303106
DATED : November 4, 2008
INVENTOR(S) : Levine

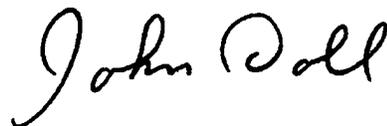
Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 10, line 43, delete "textualization:" and insert -- textualization; --.

Signed and Sealed this

Twenty-fourth Day of February, 2009



JOHN DOLL
Acting Director of the United States Patent and Trademark Office

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Inventor(s): Levine

Serial No.: 10/303,106

Patent No.: 7,447,686

Filed: November 22, 2002

Issued: November 4, 2008

For: Computer-Implemented System And Method For Handling Database
Statements

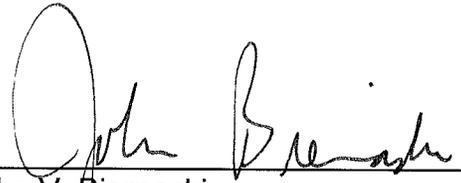
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

**NOTIFICATION OF ERROR IN PRINTING PATENT
CERTIFICATE OF CORRECTION REQUESTED
UNDER 37 CFR § 1.323**

In proofreading the above-referenced patent, it has been noted that the following error occurred in the printing thereof. A Certificate of Correction is therefore requested. (See enclosed Certificate of Correction)

No fees are deemed to be due in connection with the issuance of the Certificate of Correction as all errors are printing errors of the United States Patent and Trademark Office. In the event, however, that fees are due, please charge any fees required by this request to Jones Day's Deposit Account No. 501432, reference 343355600054.

Respectfully submitted,



John V. Biernacki
Registration No. 40,511
Jones Day
North Point
901 Lakeside Avenue
Cleveland, OH 44114
(216) 586-7747

Date: _____

1/14/2009

**UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION**Page 1 of 1

PATENT NO. : 7,447,686

APPLICATION NO.: 10/303,106

ISSUE DATE : November 4, 2008

INVENTOR(S) : Levine

It is certified that an error appears or errors appear in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In column 10, line 43, delete "textualization:" and insert -- textualization; --.

MAILING ADDRESS OF SENDER (Please do not use customer number below):

John V. Biernacki
Jones Day
North Point
901 Lakeside Avenue
Cleveland, Ohio 44114

This collection of information is required by 37 CFR 1.322, 1.323, and 1.324. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1.0 hour to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Attention Certificate of Corrections Branch, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

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

Electronic Acknowledgement Receipt

EFS ID:	4612518
Application Number:	10303106
International Application Number:	
Confirmation Number:	2037
Title of Invention:	COMPUTER-IMPLEMENTED SYSTEM AND METHOD FOR HANDLING DATABASE STATEMENTS
First Named Inventor/Applicant Name:	Frederick J. Levine
Correspondence Address:	John V. Biernacki Jones, Day, Reavis & Pogue North Point 901 Lakeside Avenue Cleveland OH 44114 US 2165863939 -
Filer:	Stephen D. Scanlon/John V. Biernacki
Filer Authorized By:	Stephen D. Scanlon
Attorney Docket Number:	343355600054
Receipt Date:	14-JAN-2009
Filing Date:	22-NOV-2002
Time Stamp:	15:48:55
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Request for Certificate of Correction	DOC254.pdf	69847 5bcd125937d1b6f1c1a0fb68881763749baa89611	no	3

Warnings:

Information:

Total Files Size (in bytes): 69847

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

New Applications Under 35 U.S.C. 111

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

National Stage of an International Application under 35 U.S.C. 371

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

New International Application Filed with the USPTO as a Receiving Office

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



APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/303,106	11/04/2008	7447686	343355600054	2037

7590 10/15/2008

John V. Biernacki
Jones, Day, Reavis & Pogue
North Point
901 Lakeside Avenue
Cleveland, OH 44114

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b) (application filed on or after May 29, 2000)

The Patent Term Adjustment is 921 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (<http://pair.uspto.gov>).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site <http://pair.uspto.gov> for additional applicants):

Frederick J. Levine, Durham, NC;

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **Mail** **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
 or **Fax** **(571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

7590 06/30/2008

John V. Biernacki
 Jones, Day, Reavis & Pogue
 North Point
 901 Lakeside Avenue
 Cleveland, OH 44114

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

_____ (Depositor's name)
_____ (Signature)
_____ (Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/303,106	11/22/2002	Frederick J. Levine	343355600054	2037

TITLE OF INVENTION: COMPUTER-IMPLEMENTED SYSTEM AND METHOD FOR HANDLING DATABASE STATEMENTS

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	\$1440	\$300	\$0	\$1740	09/30/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
BETIT, JACOB F	2164	707-100000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363). <input type="checkbox"/> Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached. <input type="checkbox"/> "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a Customer Number is required.	2. For printing on the patent front page, list (1) the names of up to 3 registered patent attorneys or agents OR, alternatively, (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.	1 <u>Jones Day</u> 2 _____ 3 _____
---	---	--

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)
 PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.
 (A) NAME OF ASSIGNEE **SAS Institute Inc.**
 (B) RESIDENCE: (CITY and STATE OR COUNTRY) **Cary, North Carolina**

Please check the appropriate assignee category or categories (will not be printed on the patent) : Individual Corporation or other private group entity Government

4a. The following fee(s) are submitted: <input checked="" type="checkbox"/> Issue Fee <input checked="" type="checkbox"/> Publication Fee (No small entity discount permitted) <input type="checkbox"/> Advance Order - # of Copies _____	4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above) <input type="checkbox"/> A check is enclosed. <input type="checkbox"/> Payment by credit card. Form PTO-2038 is attached. <input checked="" type="checkbox"/> The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number <u>501432</u> (enclose an extra copy of this form).
--	--

5. Change in Entity Status (from status indicated above)
 a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature *John V. Biernacki* Date September 30, 2008
 Typed or printed name John V. Biernacki Registration No. 40,511

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

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.

Electronic Patent Application Fee Transmittal

Application Number:	10303106
Filing Date:	22-Nov-2002
Title of Invention:	COMPUTER-IMPLEMENTED SYSTEM AND METHOD FOR HANDLING DATABASE STATEMENTS
First Named Inventor/Applicant Name:	Frederick J. Levine
Filer:	Stephen D. Scanlon/John V. Biernacki
Attorney Docket Number:	343355600054

Filed as Large Entity

Utility under 35 USC 111(a) Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	1501	1	1440	1440
Publ. Fee- early, voluntary, or normal	1504	1	300	300

Description	Fee Code	Quantity	Amount	Sub-Total in USD(\$)
Extension-of-Time:				
Miscellaneous:				
Total in USD (\$)				1740

Electronic Acknowledgement Receipt

EFS ID:	4029831
Application Number:	10303106
International Application Number:	
Confirmation Number:	2037
Title of Invention:	COMPUTER-IMPLEMENTED SYSTEM AND METHOD FOR HANDLING DATABASE STATEMENTS
First Named Inventor/Applicant Name:	Frederick J. Levine
Correspondence Address:	John V. Biernacki Jones, Day, Reavis & Pogue North Point 901 Lakeside Avenue Cleveland OH 44114 US 2165863939 -
Filer:	Stephen D. Scanlon/John V. Biernacki
Filer Authorized By:	Stephen D. Scanlon
Attorney Docket Number:	343355600054
Receipt Date:	30-SEP-2008
Filing Date:	22-NOV-2002
Time Stamp:	11:07:22
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	\$1740

RAM confirmation Number	629
Deposit Account	501432
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

- Charge any Additional Fees required under 37 C.F.R. Section 1.16 (National application filing, search, and examination fees)
- Charge any Additional Fees required under 37 C.F.R. Section 1.17 (Patent application and reexamination processing fees)
- Charge any Additional Fees required under 37 C.F.R. Section 1.19 (Document supply fees)
- Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)
- Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	DOC084.pdf	96246 5b32f3135b1acfa3e543b69a6d45fc3e26df38f5	no	1

Warnings:

Information:

2	Fee Worksheet (PTO-06)	fee-info.pdf	31959 f2cf5d5fe359612eb4ebfd80519c97f6b0810e15	no	2
---	------------------------	--------------	---	----	---

Warnings:

Information:

Total Files Size (in bytes):	128205
-------------------------------------	--------

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

New Applications Under 35 U.S.C. 111

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

National Stage of an International Application under 35 U.S.C. 371

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

New International Application Filed with the USPTO as a Receiving Office

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



UNITED STATES PATENT AND TRADEMARK OFFICE

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



Bib Data Sheet

CONFIRMATION NO. 2037

Table with 5 columns: SERIAL NUMBER (10/303,106), FILING OR 371(c) DATE (11/22/2002), CLASS (707), GROUP ART UNIT (2164), ATTORNEY DOCKET NO. (343355600054)

APPLICANTS
Frederick J. Levine, Durham, NC;
** CONTINUING DATA *****
** FOREIGN APPLICATIONS *****
IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 01/08/2003

Table with 6 columns: Foreign Priority claimed, 35 USC 119 (a-d) conditions met, STATE OR COUNTRY (NC), SHEETS DRAWING (19), TOTAL CLAIMS (62), INDEPENDENT CLAIMS (7)

ADDRESS
John V. Biernacki
Jones, Day, Reavis & Pogue
North Point
901 Lakeside Avenue
Cleveland, OH44114

TITLE
COMPUTER-IMPLEMENTED SYSTEM AND METHOD FOR HANDLING DATABASE STATEMENTS

Table with 2 columns: FILING FEE RECEIVED (2232) and FEES: Authority has been given in Paper No. ... to charge/credit DEPOSIT ACCOUNT No. ... for following: (List of fee options: All Fees, 1.16 Fees (Filing), 1.17 Fees (Processing Ext. of time), 1.18 Fees (Issue), Other, Credit)



NOTICE OF ALLOWANCE AND FEE(S) DUE

7590 06/30/2008

John V. Biernacki
Jones, Day, Reavis & Pogue
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER	
BETTIT, JACOB F	
ART UNIT	PAPER NUMBER
2164	
DATE MAILED: 06/30/2008	

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/303,106 11/22/2002 Frederick J. Levine 343355600054 2037

TITLE OF INVENTION: COMPUTER-IMPLEMENTED SYSTEM AND METHOD FOR HANDLING DATABASE STATEMENTS

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
-------------	--------------	---------------	---------------------	----------------------	------------------	----------

nonprovisional NO \$1440 \$300 \$0 \$1740 09/30/2008

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **Mail** **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
 or **Fax** **(571)-273-2885**

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

7590 06/30/2008

John V. Biernacki
 Jones, Day, Reavis & Pogue
 North Point
 901 Lakeside Avenue
 Cleveland, OH 44114

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/303,106 11/22/2002 Frederick J. Levine 343355600054 2037

TITLE OF INVENTION: COMPUTER-IMPLEMENTED SYSTEM AND METHOD FOR HANDLING DATABASE STATEMENTS

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
-------------	--------------	---------------	---------------------	----------------------	------------------	----------

nonprovisional NO \$1440 \$300 \$0 \$1740 09/30/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
----------	----------	----------------

BETIT, JACOB F 2164 707-100000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

- Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
- "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. Use of a **Customer Number is required.**

2. For printing on the patent front page, list

- (1) the names of up to 3 registered patent attorneys or agents OR, alternatively, 1 _____
- (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed. 2 _____
- 3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : Individual Corporation or other private group entity Government

4a. The following fee(s) are submitted:

- Issue Fee
- Publication Fee (No small entity discount permitted)
- Advance Order - # of Copies _____

4b. Payment of Fee(s): (Please first reapply any previously paid issue fee shown above)

- A check is enclosed.
- Payment by credit card. Form PTO-2038 is attached.
- The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number _____ (enclose an extra copy of this form).

5. Change in Entity Status (from status indicated above)

- a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27. b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

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.



UNITED STATES PATENT AND TRADEMARK OFFICE

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

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
10/303,106 11/22/2002 Frederick J. Levine 343355600054 2037

7590 06/30/2008

John V. Biernacki
Jones, Day, Reavis & Pogue
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER

BETT, JACOB F

ART UNIT PAPER NUMBER

2164

DATE MAILED: 06/30/2008

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 729 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 729 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

Notice of Allowability

Application No.	Applicant(s)	
10/303,106	LEVINE, FREDERICK J.	
Examiner	Art Unit	
Jacob F. Bétit	2164	

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

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

- 1. This communication is responsive to the amendment dated 15 October 2007 and the interview held 11 January 2008.
- 2. The allowed claim(s) is/are 1-44 and 65-70.
- 3. Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) All b) Some* c) None of the:
 - 1. Certified copies of the priority documents have been received.
 - 2. Certified copies of the priority documents have been received in Application No. _____.
 - 3. Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.

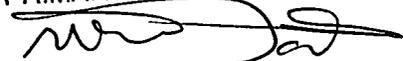
Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application. **THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.**

- 4. A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
 - 5. CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) hereto or 2) to Paper No./Mail Date _____.
 - (b) including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.
- Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
- 6. DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

- 1. Notice of References Cited (PTO-892)
- 2. Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3. Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
- 4. Examiner's Comment Regarding Requirement for Deposit of Biological Material
- 5. Notice of Informal Patent Application
- 6. Interview Summary (PTO-413), Paper No./Mail Date 20080111.
- 7. Examiner's Amendment/Comment
- 8. Examiner's Statement of Reasons for Allowance
- 9. Other _____.

NEVEEN ABEL-JALIL
PRIMARY EXAMINER



EXAMINER'S AMENDMENT

1. An extension of time under 37 CFR 1.136(a) is required in order to make an examiner's amendment which places this application in condition for allowance. During a telephone conversation conducted on 11 January 2008, John V. Biernacki requested an extension of time for 3 MONTH(S) and authorized the Director to charge Deposit Account No. 501432 the required fee of \$1020 for this extension and authorized the following examiner's amendment. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

2. The application has been amended as follows:

Please replace the listing of claims with the following:

1. (CURRENTLY AMENDED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format;

generating a second fourth-generation language database statement for use that is used within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement

wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

using a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein a first software driver textualizes through a second method a logical piece into fourth-generation database language text that is compatible with the second database system's query language format;

wherein a second software driver textualizes through a third method a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;

wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query language format and that is not compatible with the first database system's query language format.

2. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the database statement functional difference specifies at least a portion of a statement format that is compatible with the second database system's query language format and that is incompatible with the first database system's query language format

wherein the tree contains a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be used in generating the second database statement.

3. (PREVIOUSLY PRESENTED) The method of claim 1 wherein object-oriented techniques are used to access the database functional language difference data.

4. (ORIGINAL) The method of claim 3 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.
5. (ORIGINAL) The method of claim 4 wherein a logical piece is a phrase logical piece.
6. (ORIGINAL) The method of claim 4 wherein a logical piece is an identifier logical piece.
7. (CURRENTLY AMENDED) The method of claim 4 wherein an SQL component object defaults to a default native SQL textualization method ~~for use in~~ when generating the second database statement.
8. (PREVIOUSLY PRESENTED) The method of claim 7 wherein an SQL component object comprises an override to account for functional differences between the first and second database systems' query language formats.
9. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise a phrase component object.
10. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise an identifier component object.
11. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise an expression component object.

12. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise a parent component object.

13. (ORIGINAL) The method of claim 1 further comprising the steps of:

identifying, for the first database statement, query language parts that are common between the first database system's query language format and the second database system's query language format; and

generating the second database statement based upon the identified common query language parts.

14. (ORIGINAL) The method of claim 13 wherein the language parts are common based upon a predetermined standardized query language format.

15. (ORIGINAL) The method of claim 14 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.

16. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the database functional language difference data facilitates the generation of the second database statement by specifying common language parts between the first and second database system's language formats.

17. (ORIGINAL) The method of claim 16 wherein the language parts are common based upon a predetermined standardized query language format.

18. (ORIGINAL) The method of claim 17 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.

19. (ORIGINAL) The method of claim 1 wherein the second database system is a different type of database system than the first database system.

20. (CURRENTLY AMENDED) The method of claim 1 wherein ~~generating~~ the second database statement ~~provides the ability to manipulate~~ data within the second database system from the first database system.

21. (ORIGINAL) The method of claim 1 wherein generating automatically the second database statement provides the ability to transparently manipulate data within the second database system from the first database system.

22. (CURRENTLY AMENDED) The method of claim 1 wherein the generated second database statement is provided to the second database system ~~for execution and~~ executed by the second database system.

23. (ORIGINAL) The method of claim 1 wherein the generated second database statement is in a format such that the second database statement is directly executable by the second database system.

24. (ORIGINAL) The method of claim 1 wherein the second database statement is a functional equivalent of the first database statement but for differences between the first and second database systems' query language formats.

25. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language formats are based upon a predetermined standardized query language format.

26. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the first and second database systems' query language formats are based upon a standardized fourth-generation structured query language (SQL) version.

27. (ORIGINAL) The method of claim 26 wherein the first database system's query language format utilizes a superset of the SQL standard.

28. (ORIGINAL) The method of claim 26 wherein the second database system's query language format utilizes a superset of the SQL standard.

29. (CURRENTLY AMENDED) The method of claim 1 wherein the first and second database systems' query language formats specify different formats for a preselected query-related function, wherein the first database statement is formatted in the first database system's query language format to perform the query-related function,

wherein the second database statement is generated based upon the database functional difference data ~~so as to be~~ and is formatted in the second database system's query language format, wherein the generated second database statement is executable within the second database system ~~so as to perform~~ and performs the query-related function within the second database system.

30. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the tree is an SQL tree that is used to generate the second database statement, wherein the SQL tree contains data that represents the syntax of the first database statement.

31. (ORIGINAL) The method of claim 30 wherein the SQL tree contains metadata related to the first database statement.

32. (ORIGINAL) The method of claim 31 wherein the first database statement is parsed into logical text pieces which are stored in the SQL tree.

33. (ORIGINAL) The method of claim 32 wherein the second database statement takes into account any second database system-specific query language syntax.

34. (PREVIOUSLY PRESENTED) The method of claim 1 further comprising the step of:

generating a third database statement for use within a third database system, wherein the third database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the third database statement is compatible with the third database system's query language format.

35. (PREVIOUSLY PRESENTED) The method of claim 34 further comprising the step of:

generating a fourth database statement for use within a fourth database system, wherein the fourth database statement is generated based upon the first database statement and

upon the accessed database functional language difference data, wherein the fourth database statement is compatible with the fourth database system's query language format.

36. (ORIGINAL) The method of claim 1 wherein the first and second database systems are relational database management systems.

37. (ORIGINAL) The method of claim 1 wherein the first database system comprises a data mining application.

38. (ORIGINAL) The method of claim 37 wherein the second database system comprises a relational database management system.

39. (ORIGINAL) The method of claim 1 wherein the first database system comprises a relational database management system.

40. (ORIGINAL) The method of claim 39 wherein the second database system comprises a data mining application.

41. (ORIGINAL) The method of claim 1 wherein the first database system comprises an enterprise resource planning system.

42. (ORIGINAL) The method of claim 41 wherein the second database system comprises an enterprise resource planning system.

43. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language format includes format specifications for insert, select, update, and delete database commands.

44. (CURRENTLY AMENDED) Computer software stored on a computer readable media, the computer software comprising program code for ~~carrying~~ carrying out a method according to claim 1.

45-64 (CANCELLED)

65. (CURRENTLY AMENDED) The method of claim 64 1 wherein the first software driver's details of textualization into a different fourth-generation database language is hidden within the first software driver.

66. (CURRENTLY AMENDED) The method of claim 64 1 wherein the switching of the association includes switching pointing of the first method to the second method for the first software driver.

67. (CURRENTLY AMENDED) The method of claim 64 1 wherein the plurality of component software objects includes a phrase component software object, an identifier component software object, and an expression component software object.

68. (PREVIOUSLY PRESENTED) The method of claim 67 wherein the phrase component software object handles textualization of database phrases;

wherein the identifier component software object handles textualization of entities referenced in a database;

wherein the expression component software object handles textualization of expressions.

69. (PREVIOUSLY PRESENTED) The method of claim 68 wherein the phrase component software object handles textualization of database WHERE phrases;

wherein the identifier component software object handles textualization of column names referenced in a database.

70. (CURRENTLY AMENDED) The method of claim 69 wherein the first database statement contains an expression which contains a phrase which contains an identifier;

wherein the expression component software object processing the expression contained in the first database statement;

wherein the expression component software object invokes the phrase component software object ~~in order to process~~ that processes the phrase contained in the first database statement;

wherein the phrase component software object invokes the identifier component software object ~~in order to process~~ that processes the identifier contained in the first database statement.

Application/Control Number:
10/303,106
Art Unit: 2164

Page 13

3. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob F. Bétit whose telephone number is (571) 272-4075. The examiner can normally be reached on Monday through Friday 9:30 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Rones can be reached on (571) 272-4085. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

jfb
11 Jan 2008


NEVEEN ABEL-JALIL
PRIMARY EXAMINER

Interview Summary	Application No. 10/303,106	Applicant(s) LEVINE, FREDERICK J.	
	Examiner Jacob F. Bétit	Art Unit 2164	

All participants (applicant, applicant's representative, PTO personnel):

- (1) Jacob F. Bétit (3) _____
(2) John V. Biernacki (4) _____

Date of Interview: 11 January 2007.

Type: a) Telephonic b) Video Conference
c) Personal [copy given to: 1) applicant 2) applicant's representative]

Exhibit shown or demonstration conducted: d) Yes e) No.
If Yes, brief description: _____.

Claim(s) discussed: 1, 7, 20, 22, 29 and 44-60.

Identification of prior art discussed: N/A.

Agreement with respect to the claims f) was reached. g) was not reached. h) N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: The attorney authorized the examiner to cancel all non-allowed claims and to clarify the language in several of the allowed claims. The attorney authorized an extension of time so that an examiner's amendment could be made.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.


Examiner's signature, if required

Index of Claims 	Application/Control No. 10303106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.
	Examiner Betit, Jacob F	Art Unit 2164

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

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

CLAIM		DATE								
Final	Original	02/06/2006	08/06/2006	07/06/2007	01/11/2008					
1	1	✓	✓	✓	=					
2	2	✓	✓	✓	=					
3	3	✓	✓	✓	=					
4	4	✓	✓	✓	=					
5	5	✓	✓	✓	=					
6	6	✓	✓	✓	=					
7	7	✓	✓	✓	=					
8	8	✓	✓	✓	=					
9	9	✓	✓	✓	=					
10	10	✓	✓	✓	=					
11	11	✓	✓	✓	=					
12	12	✓	✓	✓	=					
13	13	✓	✓	✓	=					
14	14	✓	✓	✓	=					
15	15	✓	✓	✓	=					
16	16	✓	✓	✓	=					
17	17	✓	✓	✓	=					
18	18	✓	✓	✓	=					
19	19	✓	✓	✓	=					
20	20	✓	✓	✓	=					
21	21	✓	✓	✓	=					
22	22	✓	✓	✓	=					
23	23	✓	✓	✓	=					
24	24	✓	✓	✓	=					
25	25	✓	✓	✓	=					
26	26	✓	✓	✓	=					
27	27	✓	✓	✓	=					
28	28	✓	✓	✓	=					
29	29	✓	✓	✓	=					
30	30	✓	✓	✓	=					
31	31	✓	✓	✓	=					
32	32	✓	✓	✓	=					
33	33	✓	✓	✓	=					
34	34	✓	✓	✓	=					
35	35	✓	✓	✓	=					
36	36	✓	✓	✓	=					

Index of Claims 	Application/Control No. 10303106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.
	Examiner Betit, Jacob F	Art Unit 2164

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

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

CLAIM		DATE							
Final	Original	02/06/2006	08/06/2006	07/06/2007	01/11/2008				
37	37	✓	✓	✓	=				
38	38	✓	✓	✓	=				
39	39	✓	✓	✓	=				
40	40	✓	✓	✓	=				
41	41	✓	✓	✓	=				
42	42	✓	✓	✓	=				
43	43	✓	✓	✓	=				
50	44	✓	✓	✓	=				
	45	✓	✓	✓	-				
	46	✓	✓	✓	-				
	47	✓	✓	✓	-				
	48	✓	✓	✓	-				
	49	✓	✓	✓	-				
	50	✓	✓	✓	-				
	51	✓	✓	✓	-				
	52	✓	✓	✓	-				
	53	✓	✓	✓	-				
	54	✓	✓	✓	-				
	55	✓	✓	✓	-				
	56	✓	✓	✓	-				
	57	✓	✓	✓	-				
	58	✓	✓	✓	-				
	59	✓	✓	✓	-				
	60	✓	✓	✓	-				
	61	✓	✓	✓	-				
	62	✓	✓	✓	-				
	63			✓	-				
	64			O	-				
44	65			O	=				
45	66			O	=				
46	67			O	=				
47	68			O	=				
48	69			O	=				
49	70			O	=				

Search Notes 	Application/Control No. 10303106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.
	Examiner Betit, Jacob F	Art Unit 2164

SEARCHED			
Class	Subclass	Date	Examiner
707	3, 4, 100	1/11/08	jfb

SEARCH NOTES		
Search Notes	Date	Examiner
searched EAST enclosed	02/06/2006	JFB
updated prior search	8/6/06	JFB
spoke to Pr Sam Rimell about 112 first issue	7/27/06	JFB
spoke to Sam Rimell, Neveen Abel-Jalil, and Charles Ronos about claims 64-70	7/6/07	JFB

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner
707	3, 4, 100	1/11/08	jfb
	inventor name search	1/11/08	jfb

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	91	((convert\$4 translat\$4) with sql).ab.	US-PGPUB; USPAT; EPO	OR	ON	2007/07/06 08:02
S2	11	((convert\$4 translat\$4) with sql).ab. and (hierarch\$4 tree) with (translat\$4 convert\$4)	US-PGPUB; USPAT; EPO	OR	ON	2008/01/17 18:01
S3	8	((convert\$4 translat\$4) with sql).ab. and (hierarch\$4 tree) with (translat\$4 convert\$4) not xml	US-PGPUB; USPAT; EPO	OR	ON	2007/07/06 08:06
S4	16204	((707/3) or (707/4) or (707/100)).CCLS.	US-PGPUB; USPAT; EPO	OR	OFF	2008/01/11 16:02
S5	2	sql with convert\$4 and sql with (part pieces) with parse	US-PGPUB; USPAT; EPO	OR	ON	2008/01/11 16:01
S6	0	S4 and textualize same parse and (sql fourth adj generation adj language)	US-PGPUB; USPAT; EPO	OR	ON	2008/01/11 16:03
S7	4408	S4 and (sql fourth adj generation adj language)	US-PGPUB; USPAT; EPO	OR	ON	2008/01/11 16:03
S8	0	S7 and convert\$4 with (query statement) same pars\$4 same textualiz\$4	US-PGPUB; USPAT; EPO	OR	ON	2008/01/11 16:04



UNITED STATES PATENT AND TRADEMARK OFFICE

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

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
10/303,106 11/22/2002 Frederick J. Levine 343355600054 2037

7590 10/29/2007
John V. Biernacki
Jones, Day, Reavis & Pogue
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER

BETIT, JACOB F

ART UNIT PAPER NUMBER

2164

MAIL DATE DELIVERY MODE

10/29/2007

PAPER

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

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

50

Advisory Action Before the Filing of an Appeal Brief	Application No. 10/303,106	Applicant(s) LEVINE, FREDERICK J.	
	Examiner Jacob F. Bétit	Art Unit 2164	

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

THE REPLY FILED 15 October 2007 FAILS TO PLACE THIS APPLICATION IN CONDITION FOR ALLOWANCE.

1. The reply was filed after a final rejection, but prior to or on the same day as filing a Notice of Appeal. To avoid abandonment of this application, applicant must timely file one of the following replies: (1) an amendment, affidavit, or other evidence, which places the application in condition for allowance; (2) a Notice of Appeal (with appeal fee) in compliance with 37 CFR 41.31; or (3) a Request for Continued Examination (RCE) in compliance with 37 CFR 1.114. The reply must be filed within one of the following time periods:
- a) The period for reply expires 3 months from the mailing date of the final rejection.
- b) The period for reply expires on: (1) the mailing date of this Advisory Action, or (2) the date set forth in the final rejection, whichever is later. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of the final rejection.
- Examiner Note: If box 1 is checked, check either box (a) or (b). ONLY CHECK BOX (b) WHEN THE FIRST REPLY WAS FILED WITHIN TWO MONTHS OF THE FINAL REJECTION. See MPEP 706.07(f).

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

NOTICE OF APPEAL

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

AMENDMENTS

3. The proposed amendment(s) filed after a final rejection, but prior to the date of filing a brief, will not be entered because
- (a) They raise new issues that would require further consideration and/or search (see NOTE below);
- (b) They raise the issue of new matter (see NOTE below);
- (c) They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal; and/or
- (d) They present additional claims without canceling a corresponding number of finally rejected claims.

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

4. The amendments are not in compliance with 37 CFR 1.121. See attached Notice of Non-Compliant Amendment (PTOL-324).
5. Applicant's reply has overcome the following rejection(s): _____.
6. Newly proposed or amended claim(s) 1-44 and 65-70 would be allowable if submitted in a separate, timely filed amendment canceling the non-allowable claim(s).
7. For purposes of appeal, the proposed amendment(s): a) will not be entered, or b) will be entered and an explanation of how the new or amended claims would be rejected is provided below or appended.
- The status of the claim(s) is (or will be) as follows:
- Claim(s) allowed: _____.
- Claim(s) objected to: 64-70.
- Claim(s) rejected: 1-63.
- Claim(s) withdrawn from consideration: _____.

AFFIDAVIT OR OTHER EVIDENCE

8. The affidavit or other evidence filed after a final action, but before or on the date of filing a Notice of Appeal will not be entered because applicant failed to provide a showing of good and sufficient reasons why the affidavit or other evidence is necessary and was not earlier presented. See 37 CFR 1.116(e).
9. The affidavit or other evidence filed after the date of filing a Notice of Appeal, but prior to the date of filing a brief, will not be entered because the affidavit or other evidence failed to overcome all rejections under appeal and/or appellant fails to provide a showing a good and sufficient reasons why it is necessary and was not earlier presented. See 37 CFR 41.33(d)(1).
10. The affidavit or other evidence is entered. An explanation of the status of the claims after entry is below or attached.

REQUEST FOR RECONSIDERATION/OTHER

11. The request for reconsideration has been considered but does NOT place the application in condition for allowance because: _____.
12. Note the attached Information Disclosure Statement(s). (PTO/SB/08) Paper No(s). _____
13. Other: _____.

Continuation of 3. NOTE:

The amendments to claims 45, 46, 49, 50, and 60 require further consideration since the scope of these claims is not identical to that of previously objected to claims 64-70. These claims include separate embodiments and use different language as compared to the combination of independent claim 1 and dependent claim 64.

It is also noted that independent claim 46 contains two periods which is not in accordance with proper claim structure. (See MPEP 608.01(m)).



CHARLES RONES
SUPERVISORY PATENT EXAMINER

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTOR(S) : F. Levine
TITLE : Computer-Implemented System And Method For
Handling Database Statements
SERIAL NO. : 10/303,106
FILING DATE : 11/22/2002
GROUP ART UNIT : 2164
EXAMINER : J. Betit
ATTORNEY DOCKET NO. : 343355600054

Mail Stop Amendment AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSIVE AMENDMENT

Please consider the following amendments and remarks.

DO NOT ENTER
JB
25 Oct 2007

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTOR(S) : F. Levine
TITLE : Computer-Implemented System And Method For
Handling Database Statements
SERIAL NO. : 10/303,106
FILING DATE : 11/22/2002
GROUP ART UNIT : 2164
EXAMINER : J. Betit
ATTORNEY DOCKET NO. : 343355600054

Mail Stop Amendment AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

RESPONSIVE AMENDMENT

Please consider the following amendments and remarks.

CLAIMS

1. (CURRENTLY AMENDED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format;

generating a second fourth-generation language database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement

wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

using a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein a first software driver textualizes through a second method a logical piece into fourth-generation database language text that is compatible with the second database system's query language format;

wherein a second software driver textualizes through a third method a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;

wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query language format and that is not compatible with the first database system's query language format.

2. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the database statement functional difference specifies at least a portion of a statement format that is

compatible with the second database system's query language format and that is incompatible with the first database system's query language format

wherein the tree contains a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be used in generating the second database statement.

3. (PREVIOUSLY PRESENTED) The method of claim 1 wherein object-oriented techniques are used to access the database functional language difference data.
4. (ORIGINAL) The method of claim 3 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.
5. (ORIGINAL) The method of claim 4 wherein a logical piece is a phrase logical piece.
6. (ORIGINAL) The method of claim 4 wherein a logical piece is an identifier logical piece.
7. (ORIGINAL) The method of claim 4 wherein an SQL component object defaults to a default native SQL textualization method for use in generating the second database statement.
8. (PREVIOUSLY PRESENTED) The method of claim 7 wherein an SQL component object comprises an override to account for functional differences between the first and second database systems' query language formats.

9. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise a phrase component object.
10. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise an identifier component object.
11. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise an expression component object.
12. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise a parent component object.
13. (ORIGINAL) The method of claim 1 further comprising the steps of:
 - identifying, for the first database statement, query language parts that are common between the first database system's query language format and the second database system's query language format; and
 - generating the second database statement based upon the identified common query language parts.
14. (ORIGINAL) The method of claim 13 wherein the language parts are common based upon a predetermined standardized query language format.
15. (ORIGINAL) The method of claim 14 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.
16. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the database functional language difference data facilitates the generation of the second

database statement by specifying common language parts between the first and second database system's language formats.

17. (ORIGINAL) The method of claim 16 wherein the language parts are common based upon a predetermined standardized query language format.

18. (ORIGINAL) The method of claim 17 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.

19. (ORIGINAL) The method of claim 1 wherein the second database system is a different type of database system than the first database system.

20. (ORIGINAL) The method of claim 1 wherein generating the second database statement provides the ability to manipulate data within the second database system from the first database system.

21. (ORIGINAL) The method of claim 1 wherein generating automatically the second database statement provides the ability to transparently manipulate data within the second database system from the first database system.

22. (ORIGINAL) The method of claim 1 wherein the generated second database statement is provided to the second database system for execution by the second database system.

23. (ORIGINAL) The method of claim 1 wherein the generated second database statement is in a format such that the second database statement is directly executable by the second database system.

24. (ORIGINAL) The method of claim 1 wherein the second database statement is a functional equivalent of the first database statement but for differences between the first and second database systems' query language formats.

25. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language formats are based upon a predetermined standardized query language format.

26. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the first and second database systems' query language formats are based upon a standardized fourth-generation structured query language (SQL) version.

27. (ORIGINAL) The method of claim 26 wherein the first database system's query language format utilizes a superset of the SQL standard.

28. (ORIGINAL) The method of claim 26 wherein the second database system's query language format utilizes a superset of the SQL standard.

29. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the first and second database systems' query language formats specify different formats for a preselected query-related function, wherein the first database statement is formatted in the first database system's query language format to perform the query-related function, wherein the second database statement is generated based upon the database functional difference data so as to be formatted in the second database system's query language format, wherein the generated second database statement is executable

within the second database system so as to perform the query-related function within the second database system.

30. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the tree is an SQL tree that is used to generate the second database statement, wherein the SQL tree contains data that represents the syntax of the first database statement.

31. (ORIGINAL) The method of claim 30 wherein the SQL tree contains metadata related to the first database statement.

32. (ORIGINAL) The method of claim 31 wherein the first database statement is parsed into logical text pieces which are stored in the SQL tree.

33. (ORIGINAL) The method of claim 32 wherein the second database statement takes into account any second database system-specific query language syntax.

34. (PREVIOUSLY PRESENTED) The method of claim 1 further comprising the step of:

generating a third database statement for use within a third database system, wherein the third database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the third database statement is compatible with the third database system's query language format.

35. (PREVIOUSLY PRESENTED) The method of claim 34 further comprising the step of:

generating a fourth database statement for use within a fourth database system, wherein the fourth database statement is generated based upon the first database statement and

upon the accessed database functional language difference data, wherein the fourth database statement is compatible with the fourth database system's query language format.

36. (ORIGINAL) The method of claim 1 wherein the first and second database systems are relational database management systems.

37. (ORIGINAL) The method of claim 1 wherein the first database system comprises a data mining application.

38. (ORIGINAL) The method of claim 37 wherein the second database system comprises a relational database management system.

39. (ORIGINAL) The method of claim 1 wherein the first database system comprises a relational database management system.

40. (ORIGINAL) The method of claim 39 wherein the second database system comprises a data mining application.

41. (ORIGINAL) The method of claim 1 wherein the first database system comprises an enterprise resource planning system.

42. (ORIGINAL) The method of claim 41 wherein the second database system comprises an enterprise resource planning system.

43. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language format includes format specifications for insert, select, update, and delete database commands.

44. (ORIGINAL) Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 1.

45. (CURRENTLY AMENDED) A computer-implemented system that includes a processor for handling a first database fourth-generation language query that is formatted in a first query format and is executable by a first database system, comprising:

a data store to store tree-structured data that is representative of syntax and metadata of the first database fourth-generation language query;

wherein the tree contains logical pieces parsed from the first database fourth-generation language query;

a data structure for storing query specific data that indicates at least one query functional language difference from the first query format, wherein the query functional language difference is a query syntax difference; and

a textualization module having a data access connection to the tree-structured data and the data structure, wherein the textualization module generates a database specific query based upon the tree-structured data and the query specific data, wherein the database specific query accounts for the difference from the first query format so that the database specific query may be executed by a different type of database system

wherein the textualization module generates fourth generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

wherein the textualization module includes a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein the textualization module includes a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein a first software driver textualizes through a second method a logical piece into fourth-generation database language text that is compatible with a second database system's query language format;

wherein a second software driver textualizes through a third method a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

a processor for switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;

wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query

language format and that is not compatible with the first database system's query language format.

46. (CURRENTLY AMENDED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and

generating a second fourth-generation language database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

using a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein a first software driver textualizes through a second method a logical piece into fourth-generation database language text that is compatible with the second database system's query language format;

wherein a second software driver textualizes through a third method a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;

wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query language format and that is not compatible with the first database system's query language format.

47. (ORIGINAL) The method of claim 46 wherein the language parts are determined to be common based upon a predetermined standardized query language format.

48. (ORIGINAL) The method of claim 47 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.

49. (CURRENTLY AMENDED) A computer-implemented system that includes a processor for handling a database statement from a first database system, comprising:

means for receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and

means for generating a second database fourth-generation language statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement;

wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

means for using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

means for using a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein a first software driver textualizes through a second method a logical piece into fourth-generation database language text that is compatible with the second database system's query language format;

wherein a second software driver textualizes through a third method a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

means for switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;

wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query

language format and that is not compatible with the first database system's query language format.

50. (CURRENTLY AMENDED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call textualization specific data,

wherein the generated program call is in a format that is compatible with the second database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call

wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

using a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein a first software driver textualizes through a second method a logical piece into fourth-generation database language text that is compatible with the second database system's query language format;

wherein a second software driver textualizes through a third method a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;

wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query language format and that is not compatible with the first database system's query language format.

51. (ORIGINAL) The method of claim 50 wherein the program call textualization specific data is used to generate application program interface (API) calls from the first database statement.

52. (ORIGINAL) The method of claim 50 wherein object-oriented techniques are used to access the program call textualization specific data.
53. (ORIGINAL) The method of claim 52 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.
54. (ORIGINAL) The method of claim 53 wherein a logical piece is a phrase logical piece.
55. (ORIGINAL) The method of claim 53 wherein a logical piece is an identifier logical piece.
56. (ORIGINAL) The method of claim 50 wherein the first database system's query language format is based upon a standardized structured query language (SQL) version.
57. (PREVIOUSLY PRESENTED) The method of claim 50 further comprising the steps of:
- accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format; and
 - generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein

the second database statement is compatible with the second database system's query language format.

58. (PREVIOUSLY PRESENTED) The method of claim 50 further comprising the steps of:

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format; and

generating a second database statement for use within a third database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the third database system's query language format.

59. (ORIGINAL) Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 50.

60. (CURRENTLY AMENDED) A computer-implemented system that includes a processor for handling a database statement from a first database system, comprising the steps of:

means for receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

means for generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call textualization specific data,

wherein the generated program call is in a format that is compatible with the second database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call for use within the second database system;

wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

means for using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

means for using a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein a first software driver textualizes through a second method a logical piece into fourth-generation database language text that is compatible with the second database system's query language format;

wherein a second software driver textualizes through a third method a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

means for switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;

wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query language format and that is not compatible with the first database system's query language format.

61- 64 (CANCELLED)

65. (PREVIOUSLY PRESENTED) The method of claim 64 1 wherein the first software driver's details of textualization into a different fourth-generation database language is hidden within the first software driver.

66. (PREVIOUSLY PRESENTED) The method of claim 64 1 wherein the switching of the association includes switching pointing of the first method to the second method for the first software driver.

67. (PREVIOUSLY PRESENTED) The method of claim 64 1 wherein the plurality of component software objects includes a phrase component software object, an identifier component software object, and an expression component software object.

68. (PREVIOUSLY PRESENTED) The method of claim 67 wherein the phrase component software object handles textualization of database phrases;

wherein the identifier component software object handles textualization of entities referenced in a database;

wherein the expression component software object handles textualization of expressions.

69. (PREVIOUSLY PRESENTED) The method of claim 68 wherein the phrase component software object handles textualization of database WHERE phrases;

wherein the identifier component software object handles textualization of column names referenced in a database.

70. (PREVIOUSLY PRESENTED) The method of claim 69 wherein the first database statement contains an expression which contains a phrase which contains an identifier;

wherein the expression component software object processing the expression contained in the first database statement;

wherein the expression component software object invokes the phrase component software object in order to process the phrase contained in the first database statement;

wherein the phrase component software object invokes the identifier component software object in order to process the identifier contained in the first database statement.

REMARKS

Claims 1-70 are pending in the application. Claims 1, 45, 46, 49, 50, 60 and 61 are independent claims. Claims 1-63 stand rejected by the examiner. Claims 64-70 have been noted as allowable if rewritten so as not to depend from a rejected base claim. Claims 61-64 are cancelled by this amendment. Assignee traverses the instant claim rejections.

Regarding independent claim 1, although assignee disagrees with examiner's 35 U.S.C. § 103(a) rejection of claim 1 based upon Bodamer and Chow and reserves the right to further argue this in the future through a continuation application, in order to expedite prosecution, assignee has elected to incorporate the entirety of allowable claim 64 into the body of claim 1. It is respectfully submitted that in light of this amendment, claim 1 is now in condition for allowance, and it is respectfully requested that the 35 U.S.C. § 103(a) rejection of independent claim 1 be withdrawn.

Similar amendments have been made to independent claims 45, 46, 49, 50, and 60. In light of these amendments, it is respectfully submitted that these claims are allowable for similar reasoning as offered for claim 1. Therefore, it is respectfully requested that the rejection of these independent claims under 35 U.S.C. § 103(a) be withdrawn.

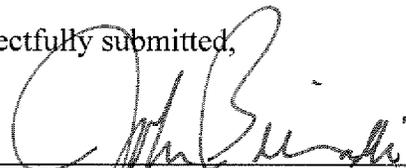
Applicant at this time has not submitted any arguments in support of the patentability of the dependent claims. It is believed that independent claims 1, 45, 49, 50, and 60 are now in condition for allowance such that all of the dependent claims which depend either directly or indirectly therefrom are also in condition for allowance.

CONCLUSION

For the foregoing reasons, assignee respectfully submits that the pending claims are allowable. Therefore, the examiner is respectfully requested to pass this case to issue.

Respectfully submitted,

By: _____



John V. Biernacki
Reg. No. 40,511
JONES DAY
North Point
901 Lakeside Avenue
Cleveland, Ohio 44114
(216) 586-3939

Electronic Acknowledgement Receipt

EFS ID:	2318310
Application Number:	10303106
International Application Number:	
Confirmation Number:	2037
Title of Invention:	Computer-implemented system and method for handling database statements
First Named Inventor/Applicant Name:	Frederick J. Levine
Correspondence Address:	John V. Biernacki Jones, Day, Reavis & Pogue North Point 901 Lakeside Avenue Cleveland OH 44114 US 2165790212 -
Filer:	Stephen D. Scanlon/John V. Biernacki
Filer Authorized By:	Stephen D. Scanlon
Attorney Docket Number:	343355600054
Receipt Date:	15-OCT-2007
Filing Date:	22-NOV-2002
Time Stamp:	11:05:07
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no
------------------------	----

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1		NETSC086.PDF	739889 b17cb4291a53199a66a9bad612b19c1bef1107ac	yes	25

Multipart Description/PDF files in .zip description			
Document Description	Start	End	
Amendment After Final	1	1	
Claims	2	23	
Applicant Arguments/Remarks Made in an Amendment	24	25	

Warnings:

Information:

Total Files Size (in bytes):	739889
-------------------------------------	--------

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

New Applications Under 35 U.S.C. 111

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

National Stage of an International Application under 35 U.S.C. 371

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

New International Application Filed with the USPTO as a Receiving Office

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

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 10/303,106	Filing Date 11/22/2002	<input type="checkbox"/> To be Mailed
---	---	----------------------------------	---------------------------------------

APPLICATION AS FILED – PART I			OTHER THAN SMALL ENTITY			
	(Column 1)	(Column 2)	SMALL ENTITY <input type="checkbox"/>	OR		
FOR	NUMBER FILED	NUMBER EXTRA	RATE (\$)	FEE (\$)	RATE (\$)	FEE (\$)
<input checked="" type="checkbox"/> BASIC FEE <small>(37 CFR 1.16(a), (b), or (c))</small>	N/A	N/A	N/A		N/A	740
<input type="checkbox"/> SEARCH FEE <small>(37 CFR 1.16(k), (l), or (m))</small>	N/A	N/A	N/A		N/A	
<input type="checkbox"/> EXAMINATION FEE <small>(37 CFR 1.16(o), (p), or (q))</small>	N/A	N/A	N/A		N/A	
TOTAL CLAIMS <small>(37 CFR 1.16(i))</small>	62 minus 20 =	* 42	X \$ =		X \$18 =	756
INDEPENDENT CLAIMS <small>(37 CFR 1.16(h))</small>	7 minus 3 =	* 4	X \$ =		X \$84 =	336
<input type="checkbox"/> APPLICATION SIZE FEE <small>(37 CFR 1.16(s))</small>	If the specification and drawings exceed 100 sheets of paper, the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).					
<input type="checkbox"/> MULTIPLE DEPENDENT CLAIM PRESENT <small>(37 CFR 1.16(j))</small>						
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL		TOTAL	1832

APPLICATION AS AMENDED – PART II					OTHER THAN SMALL ENTITY			
	(Column 1)	(Column 2)	(Column 3)					
AMENDMENT	10/15/2007	CLAIMS REMAINING AFTER AMENDMENT	HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE (\$)	ADDITIONAL FEE (\$)	RATE (\$)	ADDITIONAL FEE (\$)
	Total (37 CFR 1.16(i))	* 58	Minus ** 62	= 0	X \$ =		OR X \$50=	0
	Independent (37 CFR 1.16(h))	* 6	Minus *** 7	= 0	X \$ =		OR X \$210=	0
	<input type="checkbox"/> Application Size Fee (37 CFR 1.16(s))							
	<input type="checkbox"/> FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							
					TOTAL ADD'L FEE		OR TOTAL ADD'L FEE	0

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

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
 ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
 *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
 The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
 Frederick E. Briscoe

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

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

Application Number 	Application/Control No. 10/303,106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.
	Examiner Jacob F. Bétit	Art Unit 2164



UNITED STATES PATENT AND TRADEMARK OFFICE

mn

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/303,106	11/22/2002	Frederick J. Levine	343355600054	2037

7590 07/13/2007
 John V. Biernacki
 Jones, Day, Reavis & Pogue
 North Point
 901 Lakeside Avenue
 Cleveland, OH 44114

EXAMINER

BETIT, JACOB F

ART UNIT	PAPER NUMBER
2164	

MAIL DATE	DELIVERY MODE
07/13/2007	PAPER

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

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

DETAILED ACTION

Remarks

1. In response to communications filed on 12 April 2007, claims 45, 49, 60, and 62 have been amended and claims 63-70 have been added per the applicant's request. Claims 1-70 are presently pending in the application.

Claim Objections

2. Claim 62 objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. "A claim in dependent form shall be construed to incorporate by reference all the limitations of the claim to which it refers" (see 35 U.S.C. §112 fourth paragraph). Because claim 62 specifies for an "switch" that could eliminate steps from claim 61 it does not incorporate all the limitations of 61 under all circumstances. Claim 62 makes optional (on condition of a switch) steps that were required in claim 61. The step "generating a second database fourth-generation language statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional difference data" is a required step in claim 61. It is improper for claim 62 to contain a switch that can eliminate this step.

Claim Rejections - 35 USC § 103

Art Unit: 2164

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

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

4. Claims 1-36, 38-39, and 43-63 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bodamer et al. (U.S. patent No. 6,041,344) in view of Chow et al. (U.S. patent No. 6,941,298 B2).

As to claim 1, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

accessing database functional language difference data, wherein the database language difference data indicates a format that contains at least one database statement difference from the first database system's query language format (see column 7, line 18 through column 8, line 37); and

generating a second fourth-generation language database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format (see column 8, line 38-67);

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 2, Bodamer et al. as modified teaches wherein the database statement functional difference specifies at least a portion of a statement format that is compatible with the second database system's query language format and that is incompatible with the first database system's query language format (see Bodamer et al., column 7, line 18 through column 8, line 37)

wherein the tree contains a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be used in generating the second database statement (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

As to claim 3, Bodamer et al. as modified teaches wherein object-oriented techniques are used to access the database functional language difference data (see Bodamer et al., column 5, lines 7-54).

Art Unit: 2164

As to claim 4, Bodamer et al. as modified teaches wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement (see Bodamer et al., column 7, lines 43-67).

As to claim 5, Bodamer et al. as modified teaches wherein a logical piece is a phrase logical piece (see Bodamer et al., column 8, lines 47-67).

As to claim 6, Bodamer et al. as modified teaches wherein a logical piece is an identifier logical piece (see Bodamer et al., column 8, lines 47-67).

As to claim 7, Bodamer et al. as modified teaches wherein an SQL component object defaults to a default native SQL textualization method for use in generating the second database statement (see Bodamer et al., column 7, lines 43-67).

As to claim 8, Bodamer et al. teaches wherein an SQL component object comprises an override to account for functional differences between the first and second database systems' query language formats (see Bodamer et al., column 7, lines 43-67)

As to claim 9, Bodamer et al. as modified teaches wherein SQL component objects comprise s phrase component object (see Bodamer et al., column 8, lines 47-67).

Art Unit: 2164

As to claim 10, Bodamer et al. as modified teaches wherein SQL component objects comprise an identifier component object (see Bodamer et al., column 8, lines 47-67).

As to claim 11, Bodamer et al. as modified teaches wherein SQL component objects comprise an expression component object (see Bodamer et al., column 8, lines 47-67).

As to claim 12, Bodamer et al. as modified teaches wherein SQL component objects comprise a parent component object (see Bodamer et al., column 8, lines 47-67).

As to claim 13, Bodamer et al. as modified teaches further comprising the steps of: identifying, for the first database statement, query language parts that are common between the first database system's query language format and the second database system's query language format; and generating the second database statement based upon the identified common query language parts (see Bodamer et al., column 7, lines 43-67).

As to claim 14, Bodamer et al. as modified teaches wherein the language parts are common based upon a predetermined standardized query language format (see Bodamer et al., column 7, lines 43-67).

As to claim 15, Bodamer et al. as modified teaches wherein the standardized query language format is based upon a standardized structured query language (SQL) version (see Bodamer et al., column 7, lines 43-67).

As to claim 16, Bodamer et al. as modified teaches wherein the database functional language difference data facilitates the generation of the second database statement by specifying common language parts between the first and second database system's language formats (see Bodamer et al., column 7, lines 43-67).

As to claim 17, Bodamer et al. as modified teaches wherein the language parts are common based upon a predetermined standardized query language format (see Bodamer et al., column 7, lines 18-67).

As to claim 18, Bodamer et al. as modified teaches wherein the standardized query language format is based upon a standardized structured query language (SQL) version (see Bodamer et al., column 7, lines 18-67).

As to claim 19, Bodamer et al. as modified teaches wherein the second database system is a different type of database system than the first database system (see Bodamer et al., column 7, lines 18-67).

As to claim 20, Bodamer et al. as modified teaches wherein generating the second database statement provides the ability to manipulate data within the second database system from the first database system (see Bodamer et al., column 4, line 40 through column 5, line 20).

Art Unit: 2164

As to claim 21, Bodamer et al. as modified teaches wherein generating automatically the second database statement provides the ability to transparently manipulate data within the second database system from the first database system (see Bodamer et al., column 8, lines 38-46).

As to claim 22, Bodamer et al. as modified teaches wherein the generated second database statement is provided to the second database system for execution by the second database system (see Bodamer et al., column 8, lines 38-46).

As to claim 23, Bodamer et al. as modified teaches wherein the generated second database statement is in a format such that the second database statement is directly executable by the second database system (see Bodamer et al., column 8, lines 47-67).

As to claim 24, Bodamer et al. as modified teaches wherein the second database statement is a functional equivalent of the first database statement but for differences between the first and second database systems' query language formats (see Bodamer et al., column 7, lines 9-17).

As to claim 25, Bodamer et al. as modified teaches wherein the first and second database systems' query language formats are based upon a predetermined standardized query language format (see Bodamer et al., column 7, lines 43-67).

Art Unit: 2164

As to claim 26, Bodamer et al. as modified teaches wherein the first and second database systems' query language formats are based upon a standardized [~~fourth-generation~~] structured query language (SQL) version (see Bodamer et al., column 7, lines 43-67).

As to claim 27, Bodamer et al. as modified teaches wherein the first database system's query language format utilizes a superset of the SQL standard (see Bodamer et al., column 7, lines 25-30).

As to claim 28, Bodamer et al. as modified teaches wherein the second database system's query language format utilizes a superset of the SQL standard (see Bodamer et al., column 7, lines 30-35).

As to claim 29, Bodamer et al. as modified teaches wherein the first and second database systems' query language formats specify different formats for a preselected query-related function, wherein the first database statement is formatted in the first database system's query language format to perform the query-related function, wherein the second database statement is generated based upon the database functional difference data so as to be formatted in the second database system's query language format, wherein the generated second database statement is executable within the second database system so as to perform the query-related function within the second database system (see Bodamer et al., column 7, line 18 through column 8, line 37).

Art Unit: 2164

As to claim 30, Bodamer et al. as modified teaches wherein the tree is an SQL tree that is used to generate the second database statement, wherein the SQL tree contains data that represents the syntax of the first database statement (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

As to claim 31, Bodamer et al. as modified teaches wherein the SQL tree contains metadata related to the first database statement (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

As to claim 32, Bodamer et al. as modified teaches wherein the first database statement is parsed into logical text pieces which are stored in the SQL tree (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

As to claim 33, Bodamer et al. as modified teaches wherein the second database statement takes into account any second database system-specific query language syntax (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

As to claim 34, Bodamer et al. as modified teaches further comprising the step of:
generating a third database statement for use within a third database system, wherein the third database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the third database statement is

Art Unit: 2164

compatible with the third database system's query language format (see Bodamer et al., column 4, line 59 through column 5, line 20 and see column 7, line 18 through column 8, line 37).

As to claim 35, Bodamer et al. as modified teaches further comprising the step of:
generating a fourth database statement for use within a fourth database system, wherein the fourth database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the fourth database statement is compatible with the fourth database system's query language format (see Bodamer et al., column 4, line 59 through column 5, line 20 and see column 7, line 18 through column 8, line 37).

As to claim 36, Bodamer et al. as modified teaches wherein the first and second database systems are relational database management systems (see Bodamer et al., column 1, lines 44-52).

As to claim 39, Bodamer et al. as modified teaches wherein the first database system comprises a relational database management system (see Bodamer et al., column 1, lines 44-52).

As to claim 38, Bodamer et al. as modified teaches wherein the second database system comprises a relational database management system (see Bodamer et al., column 8, lines 10-46).

As to claim 43, Bodamer et al. as modified teaches wherein the first and second database systems' query language format includes format specifications for insert, select, update, and delete database commands (see Bodamer et al., column 16, line 64 through column 17, line 3).

As to claim 44, Bodamer et al. as modified teaches computer software stored on a computer readable media, the computer software comprising program code (see Bodamer et al., column 3, line 65 through column 4, line 38) for carrying out a method according to claim 1 (for the rejection of the limitations of claim 1, the applicant is directed to the rejection of claim 1 above).

As to claim 45, Bodamer et al. teaches a computer-implemented system that includes a processor, for handling a first database fourth-generation language query that is formatted in a first query format and is executable by a first database system, comprising:

a data structure for storing query specific data that indicates at least one query functional language difference from the first query format, wherein the query functional language difference is a query syntax difference (see column 7, line 18 through column 8, line 37); and

a textualization module having a data access connection to data and the data structure, wherein the textualization module generates a database specific query based on data and the query specific data, wherein the database specific query accounts for the difference from the first query format so that the database specific query may be executed by a different type of database system (see column 8, lines 38-67).

Bodamer et al. does not teach

a. a data store to store tree-structured data that is representative of syntax and metadata of the first database fourth-generation language query;

Art Unit: 2164

b. a textualization module having a data access connection to the tree-structured data and the data structure, wherein the textualization module generates a database specific query based upon the tree-structured data and the query specific data.

Chow et al. teaches a. and b., see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 46, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format (see column 7, line 18 through column 8, line 37); and

generating a second forth generation lanuage database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format (see column 8, lines 38-67).

Art Unit: 2164

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 47 and 48, see the rejection of claim 14 and 15 respectively.

As to claim 49, Bodamer et al. teaches a computer-implemented system that includes a processor for handling a database statement from a first database system, comprising:

means for receiving a first database fourth generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

means for identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format (see column 7, line 18 through column 8, line 37); and

means for generating a second database fourth-generation language statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database

Art Unit: 2164

statement is compatible with the second database system's query language format (see column 8, lines 38-67);

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 50, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system (see column 7, line 18 through column 8, line 37); and

generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call

Art Unit: 2164

textualization specific data, wherein the generated program call is in a format that is compatible with the second database system (see column 8, lines 38-67);

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 51, Bodamer et al. as modified teaches wherein the program call textualization specific data is used to generate application program interface (API) calls from the first database statement (see Bodamer et al., column 7, lines 18-35).

As to claims 52, 53, 54, and 55; see the rejections of claims 3, 4, 5, and 6 respectively.

As to claim 56, Bodamer et al. as modified teaches wherein the first database system's query language format is based upon a standardized structured query language (SQL) version (see Bodamer et al., column 4, lines 40-58).

As to claim 57, Bodamer et al. as modified teaches further comprising the steps of:

Art Unit: 2164

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format (see Bodamer et al., column 5, line 63 through column 6, line 15); and

generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format (see Bodamer et al., column 8, lines 38-67).

As to claim 58, Bodamer et al. as modified teaches further comprising the steps of:

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format (see Bodamer et al., column 5, line 63 through column 6, line 15); and

generating a second database statement for use within a third database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the third database system's query language format (see Bodamer et al., column 8, lines 38-67 and see figure 2A, reference number 300).

As to claim 59, see the rejection of claim 44 above.

As to claim 60, Bodamer et al. teaches a computer-implemented system that includes a processor for handling a database statement from a first database system, comprising the steps of:

means for receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

means for accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system (see column 7, line 18 through column 8, line 37); and

means for generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call textualization specific data, wherein the generated program call is in a format that is compatible with the second database system (see column 8, lines 38-67);

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of the metadata associated with the first database system is used in generating the program call for use within the second database system.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of

Art Unit: 2164

Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 61, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

parsing the first database statement to obtain first query metadata of the first database statement; using the obtained first query metadata to access database language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format (see column 7, line 18 through column 8, line 37);

generating a second database fourth-generation language statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format (see column 8, lines 38-67);

receiving another database statement from the first database system, wherein the additional database statement from the first database system is formatted according to the first database system's query language format (see figure 2A, reference number 300 and see column 4, lines 40-58);

Art Unit: 2164

parsing the additional database statement to obtain second query metadata of the additional database statement; using the second query metadata to generate a program call to a third database system which utilizes a different query language format than the first database system (see column 7, line 18 through column 8, line 37); and

issuing the program call to the third database system to access data contained in the third database system (see column 8, lines 38-67).

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 62, Bodamer et al. as modified teaches further comprising the steps of:
using an object-oriented component for generating database access instructions to the second database system (see Bodamer et al., column 5, lines 7-20);

determining whether to use a switch, wherein the switch indicates for the object-oriented component to use the accessed database functional language difference data to generate a program call to the second database system (see Bodamer et al., column 5, line 63 through column 6, line 15); and

Art Unit: 2164

issuing the program call to the second database system to access data contained in the second database system (see Bodamer et al., column 8, lines 38-67),

wherein if the switch did not exist, then the object-oriented component textualizes a database statement based upon the first query metadata and which is executable within the second database system (this limitation is optionally recited and does not properly depend from 61 because it acts as though to exclude limitations of that claim).

As to claim 63, Bodamer et al. as modified teaches wherein the tree contains logical pieces parsed from the first fourth-generation language database statement (see Chow et al., column 6, line 65 through column 7, line 14);

using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating forth-generation database language text (see Chow et al., column 5, line 54 through column 6, line 34);

wherein a first component software object is associated with a first logical piece contained in the tree; wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree (see Chow et al., column 6, line 65 through column 7, line 14).

5. Claims 37, 40-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bodamer et al. (U.S. patent No. 6,041,344) in view Chow et al. (U.S. patent No. 6,941,298 B2)

Art Unit: 2164

as applied to claims 1-36, 38-39, and 43-63 above and in further view of the applicant's admitted prior art (see MPEP 2144.04 C.).

As to claim 37, Bodamer et al. as modified, does not teach wherein the first database system comprises a data mining application.

The applicant has admitted that that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because many databases include data mining applications so that they can acquire new data and so they can more easily search the data they currently contain.

As to claim 40, Bodamer et al. as modified, does not teach wherein the second database system comprises a data mining application.

The applicant has admitted that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because many databases include data mining applications so that they can acquire new data and so they can more easily search the data they currently contain.

As to claim 41, Bodamer et al. as modified, does not teach wherein the first database system comprises an enterprise resource planning system.

The applicant has admitted that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because this would allow a company to plan appropriate resources for different projects being performed within a company which is a common use for a database in an corporate environment.

As to claim 42, Bodamer et al. as modified, does not teach wherein the second database system comprises an enterprise resource planning system.

The applicant has admitted that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because this would allow a company to plan appropriate resources for different projects being performed within a company which is a common use for a database in an corporate environment.

Allowable Subject Matter

6. Claims 64-70 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Response to Arguments

7. Applicant's arguments filed 6 July 2007 have been fully considered but they are not persuasive.

In response to the applicant's arguments that the cited passage does not teach "generating a second fourth-generation language database statement...", the arguments have been considered, but are not deemed persuasive. Although the cited section does disclose translating the statement to adapt for difference in schema, the cited section also discloses translating the statement between different overall language formats. "In an example of a data dictionary translation from and Oracle server to a Sybase server, the heterogeneous services modules 211

Art Unit: 2164

and 311' convert the client statement..." See column 9, lines 47-49. It is clear from column 7, lines \diamond where it states "an SQL statement translated by SQL services module 210b can be an argument for the Oracle-specific call 'opiosq,' which is then mapped onto the generic API 212 as 'parse'. Although Bodamer does not go into great detail about the translations, Bodamer does discuss four different kinds of translations including ones that read on the translations discussed in the applicant's claim 1 and those that the applicant characterizes being the only ones Bodamer discusses. The applicant has been improperly ignoring three of the translations that are discussed in the Bodamer reference and focusing on the fourth in both the remarks and in the interviews that were given.

In response to the applicant's arguments that "Bodamer is lacking in any disclosure regarding the use of SQL language functional differences being specified and used to translate from a first database statement into a second database statement", the arguments have been considered, but are not deemed persuasive. The applicant is directed to the second type of translation. This appears in column 7, lines 43-67.

A second type of translation relates to SQL statements, which are highly structured. Hence, although a SQL statement in the format of the local server 202 may not be acceptable to the foreign database system 208, the structured nature of a SQL statement enable s the heterogeneous services module 311 to convert the SQL statement of the local server 202 to a format of the foreign database system 208... a generic function such as "parse" for a certain SQL statement can be passed to the foreign database system 208 using the modules 311 in the local server 202 by translating the SQL statement from the native ... format to the format of the foreign database system... For example, an SQL statement translated by the SQL services module 210b can be an argument of the Oracle-specific call "opiosq," which is then mapped onto the generic API 212 as "parse." ... The driver 214 then can map the generic function onto the foreign database system API while including the translated SQL statement as the argument.

Art Unit: 2164

The applicant is further referred to column 9: “[O]ne agent 300a may interact with a Sybase database system, and another agent 300b may interact with an Informix database system, ect. Each agent includes the foundation services 204’ and a conversion module 211.” Also in column 11 lines 6-10: “The SQL translations (SQL x) services model 224b provides SQL translation information on how to transform SQL functions in the local server202 onto the functions of the target systems208 for the SQL services module 210b and the stored procedures module 210c”.

Conclusion

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

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

Art Unit: 2164

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob F. B  tit whose telephone number is (571) 272-4075. The examiner can normally be reached on Monday through Friday 9:30 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Rones can be reached on (571) 272-4085. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

jfb
6 Jul 2007


SAM RIMELL
PRIMARY EXAMINER

Index of Claims 	Application/Control No. 10303106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.
	Examiner Betit, Jacob F	Art Unit 2164

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

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

CLAIM		DATE									
Final	Original	02/06/2006	08/06/2006	07/06/2007							
	37	✓	✓	✓							
	38	✓	✓	✓							
	39	✓	✓	✓							
	40	✓	✓	✓							
	41	✓	✓	✓							
	42	✓	✓	✓							
	43	✓	✓	✓							
	44	✓	✓	✓							
	45	✓	✓	✓							
	46	✓	✓	✓							
	47	✓	✓	✓							
	48	✓	✓	✓							
	49	✓	✓	✓							
	50	✓	✓	✓							
	51	✓	✓	✓							
	52	✓	✓	✓							
	53	✓	✓	✓							
	54	✓	✓	✓							
	55	✓	✓	✓							
	56	✓	✓	✓							
	57	✓	✓	✓							
	58	✓	✓	✓							
	59	✓	✓	✓							
	60	✓	✓	✓							
	61	✓	✓	✓							
	62	✓	✓	✓							
	63			✓							
	64			O							
	65			O							
	66			O							
	67			O							
	68			O							
	69			O							
	70			O							

Index of Claims 	Application/Control No. 10303106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.
	Examiner Betit, Jacob F	Art Unit 2164

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

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

CLAIM		DATE							
Final	Original	02/06/2006	08/06/2006	07/06/2007					
	1	✓	✓	✓					
	2	✓	✓	✓					
	3	✓	✓	✓					
	4	✓	✓	✓					
	5	✓	✓	✓					
	6	✓	✓	✓					
	7	✓	✓	✓					
	8	✓	✓	✓					
	9	✓	✓	✓					
	10	✓	✓	✓					
	11	✓	✓	✓					
	12	✓	✓	✓					
	13	✓	✓	✓					
	14	✓	✓	✓					
	15	✓	✓	✓					
	16	✓	✓	✓					
	17	✓	✓	✓					
	18	✓	✓	✓					
	19	✓	✓	✓					
	20	✓	✓	✓					
	21	✓	✓	✓					
	22	✓	✓	✓					
	23	✓	✓	✓					
	24	✓	✓	✓					
	25	✓	✓	✓					
	26	✓	✓	✓					
	27	✓	✓	✓					
	28	✓	✓	✓					
	29	✓	✓	✓					
	30	✓	✓	✓					
	31	✓	✓	✓					
	32	✓	✓	✓					
	33	✓	✓	✓					
	34	✓	✓	✓					
	35	✓	✓	✓					
	36	✓	✓	✓					

Search Notes 	Application/Control No. 10303106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.
	Examiner Betit, Jacob F	Art Unit 2164

SEARCHED			
Class	Subclass	Date	Examiner

SEARCH NOTES		
Search Notes	Date	Examiner
searched EAST enclosed	02/06/2006	JFB
updated prior search	8/6/06	JFB
spoke to Pr Sam Rimell about 112 first issue	7/27/06	JFB
spoke to Sam Rimell, Neveen Abel-Jalil, and Charles Rones about claims 64-70	7/6/07	JFB

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	91	((convert\$4 translat\$4) with sql).ab.	US-PGPUB; USPAT; EPO	OR	ON	2007/07/06 08:02
S2	11	((convert\$4 translat\$4) with sql).ab. and (hierarch\$4 tree) with (translat\$4 convert\$4)	US-PGPUB; USPAT; EPO	OR	ON	2007/07/06 08:06
S3	8	((convert\$4 translat\$4) with sql).ab. and (hierarch\$4 tree) with (translat\$4 convert\$4) not xml	US-PGPUB; USPAT; EPO	OR	ON	2007/07/06 08:06

IFW 2/64



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTOR(S) : F. Levine

TITLE : Computer-Implemented System And Method For Handling Database Statements

SERIAL NO. : 10/303,106

FILING DATE : 11/22/2002

GROUP ART UNIT : 2164

EXAMINER : J. Betit

ATTORNEY DOCKET NO. : 343355600054

Mail Stop Amendment
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited today with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on April 9, 2007

By: Kathleen Kopczynski

RESPONSE TO NOTICE OF NON-COMPLIANT AMENDMENT

This is in response to the notice of non-compliant amendment mailed March 8, 2007 that maintained that the response failed to address at least the rejections of claims 37,40-42 under 35 U.S.C. § 103(a). This response addresses and traverses herein the rejections of claims 37, 40-42.

CLAIMS

1. (PREVIOUSLY PRESENTED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format;

generating a second fourth-generation language database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

2. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the database statement functional difference specifies at least a portion of a statement format that is compatible with the second database system's query language format and that is incompatible with the first database system's query language format

wherein the tree contains a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be used in generating the second database statement.

3. (PREVIOUSLY PRESENTED) The method of claim 1 wherein object-oriented techniques are used to access the database functional language difference data.
4. (ORIGINAL) The method of claim 3 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.
5. (ORIGINAL) The method of claim 4 wherein a logical piece is a phrase logical piece.
6. (ORIGINAL) The method of claim 4 wherein a logical piece is an identifier logical piece.
7. (ORIGINAL) The method of claim 4 wherein an SQL component object defaults to a default native SQL textualization method for use in generating the second database statement.
8. (PREVIOUSLY PRESENTED) The method of claim 7 wherein an SQL component object comprises an override to account for functional differences between the first and second database systems' query language formats.
9. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise a phrase component object.

10. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise an identifier component object.
11. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise an expression component object.
12. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise a parent component object.
13. (ORIGINAL) The method of claim 1 further comprising the steps of:
 - identifying, for the first database statement, query language parts that are common between the first database system's query language format and the second database system's query language format; and
 - generating the second database statement based upon the identified common query language parts.
14. (ORIGINAL) The method of claim 13 wherein the language parts are common based upon a predetermined standardized query language format.
15. (ORIGINAL) The method of claim 14 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.
16. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the database functional language difference data facilitates the generation of the second database statement by specifying common language parts between the first and second database system's language formats.

17. (ORIGINAL) The method of claim 16 wherein the language parts are common based upon a predetermined standardized query language format.
18. (ORIGINAL) The method of claim 17 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.
19. (ORIGINAL) The method of claim 1 wherein the second database system is a different type of database system than the first database system.
20. (ORIGINAL) The method of claim 1 wherein generating the second database statement provides the ability to manipulate data within the second database system from the first database system.
21. (ORIGINAL) The method of claim 1 wherein generating automatically the second database statement provides the ability to transparently manipulate data within the second database system from the first database system.
22. (ORIGINAL) The method of claim 1 wherein the generated second database statement is provided to the second database system for execution by the second database system.
23. (ORIGINAL) The method of claim 1 wherein the generated second database statement is in a format such that the second database statement is directly executable by the second database system.

24. (ORIGINAL) The method of claim 1 wherein the second database statement is a functional equivalent of the first database statement but for differences between the first and second database systems' query language formats.

25. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language formats are based upon a predetermined standardized query language format.

26. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the first and second database systems' query language formats are based upon a standardized fourth-generation structured query language (SQL) version.

27. (ORIGINAL) The method of claim 26 wherein the first database system's query language format utilizes a superset of the SQL standard.

28. (ORIGINAL) The method of claim 26 wherein the second database system's query language format utilizes a superset of the SQL standard.

29. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the first and second database systems' query language formats specify different formats for a preselected query-related function, wherein the first database statement is formatted in the first database system's query language format to perform the query-related function, wherein the second database statement is generated based upon the database functional difference data so as to be formatted in the second database system's query language format, wherein the generated second database statement is executable

within the second database system so as to perform the query-related function within the second database system.

30. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the tree is an SQL tree that is used to generate the second database statement, wherein the SQL tree contains data that represents the syntax of the first database statement.

31. (ORIGINAL) The method of claim 30 wherein the SQL tree contains metadata related to the first database statement.

32. (ORIGINAL) The method of claim 31 wherein the first database statement is parsed into logical text pieces which are stored in the SQL tree.

33. (ORIGINAL) The method of claim 32 wherein the second database statement takes into account any second database system-specific query language syntax.

34. (PREVIOUSLY PRESENTED) The method of claim 1 further comprising the step of:

generating a third database statement for use within a third database system, wherein the third database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the third database statement is compatible with the third database system's query language format.

35. (PREVIOUSLY PRESENTED) The method of claim 34 further comprising the step of:

generating a fourth database statement for use within a fourth database system, wherein the fourth database statement is generated based upon the first database statement and

upon the accessed database functional language difference data, wherein the fourth database statement is compatible with the fourth database system's query language format.

36. (ORIGINAL) The method of claim 1 wherein the first and second database systems are relational database management systems.

37. (ORIGINAL) The method of claim 1 wherein the first database system comprises a data mining application.

38. (ORIGINAL) The method of claim 37 wherein the second database system comprises a relational database management system.

39. (ORIGINAL) The method of claim 1 wherein the first database system comprises a relational database management system.

40. (ORIGINAL) The method of claim 39 wherein the second database system comprises a data mining application.

41. (ORIGINAL) The method of claim 1 wherein the first database system comprises an enterprise resource planning system.

42. (ORIGINAL) The method of claim 41 wherein the second database system comprises an enterprise resource planning system.

43. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language format includes format specifications for insert, select, update, and delete database commands.

44. (ORIGINAL) Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 1.

45. (CURRENTLY AMENDED) A computer-implemented system that includes a processor for handling a first database fourth-generation language query that is formatted in a first query format and is executable by a first database system, comprising:

a data store to store tree-structured data that is representative of syntax and metadata of the first database fourth-generation language query;

a data structure for storing query specific data that indicates at least one query functional language difference from the first query format, wherein the query functional language difference is a query syntax difference; and

a textualization module having a data access connection to the tree-structured data and the data structure, wherein the textualization module generates a database specific query based upon the tree-structured data and the query specific data, wherein the database specific query accounts for the difference from the first query format so that the database specific query may be executed by a different type of database system.

46. (PREVIOUSLY PRESENTED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

- receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;
- identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and
- generating a second fourth-generation language database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

47. (ORIGINAL) The method of claim 46 wherein the language parts are determined to be common based upon a predetermined standardized query language format.

48. (ORIGINAL) The method of claim 47 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.

49. (CURRENTLY AMENDED) A computer-implemented system that includes a processor for handling a database statement from a first database system, comprising:

means for receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and

means for generating a second database fourth-generation language statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

50. (PREVIOUSLY PRESENTED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call

textualization specific data, wherein the generated program call is in a format that is compatible with the second database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call.

51. (ORIGINAL) The method of claim 50 wherein the program call textualization specific data is used to generate application program interface (API) calls from the first database statement.

52. (ORIGINAL) The method of claim 50 wherein object-oriented techniques are used to access the program call textualization specific data.

53. (ORIGINAL) The method of claim 52 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.

54. (ORIGINAL) The method of claim 53 wherein a logical piece is a phrase logical piece.

55. (ORIGINAL) The method of claim 53 wherein a logical piece is an identifier logical piece.

56. (ORIGINAL) The method of claim 50 wherein the first database system's query language format is based upon a standardized structured query language (SQL) version.

57. (PREVIOUSLY PRESENTED) The method of claim 50 further comprising the steps of:

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format; and

generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format.

58. (PREVIOUSLY PRESENTED) The method of claim 50 further comprising the steps of:

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format; and

generating a second database statement for use within a third database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the third database system's query language format.

59. (ORIGINAL) Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 50.

60. (CURRENTLY AMENDED) A computer-implemented system that includes a processor for handling a database statement from a first database system, comprising the steps of:

means for receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

means for generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the

program call textualization specific data, wherein the generated program call is in a format that is compatible with the second database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call for use within the second database system.

61. (PREVIOUSLY PRESENTED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

parsing the first database statement to obtain first query metadata of the first database statement;

using the obtained first query metadata to access database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format;

generating a second database fourth-generation language statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional

language difference data, wherein the second database statement is compatible with the second database system's query language format;

receiving another database statement from the first database system, wherein the additional database statement from the first database system is formatted according to the first database system's query language format;

parsing the additional database statement to obtain second query metadata of the additional database statement;

using the second query metadata to generate a program call to a third database system which utilizes a different query language format than the first database system; and

issuing the program call to the third database system to access data contained in the third database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

62. (CURRENTLY AMENDED) The method of claim 61 further comprising the steps of:

using an object-oriented component ~~means~~ for generating database access instructions to the second database system;

determining whether to use ~~an override~~ a switch, wherein the ~~override~~ switch indicates for the object-oriented component ~~means~~ to use the accessed database

functional language difference data ~~first query metadata~~ to generate a program call to the second database system; and

issuing the program call to the second database system to access data contained in the second database system,

wherein if the override switch did not exist, then the object-oriented component ~~means~~ textualizes a database statement based upon the first query metadata and which is executable within the second database system.

63. (NEW) The method of claim 1 wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree.

64. (NEW) The method of claim 1 wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

using a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein a first software driver textualizes through a second method a logical piece into fourth-generation database language text that is compatible with the second database system's query language format;

wherein a second software driver textualizes through a third method a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;

wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query

language format and that is not compatible with the first database system's query language format.

65. (NEW) The method of claim 64 wherein the first software driver's details of textualization into a different fourth-generation database language is hidden within the first software driver.

66. (NEW) The method of claim 64 wherein the switching of the association includes switching pointing of the first method to the second method for the first software driver.

67. (NEW) The method of claim 64 wherein the plurality of component software objects includes a phrase component software object, an identifier component software object, and an expression component software object.

68. (NEW) The method of claim 67 wherein the phrase component software object handles textualization of database phrases;

 wherein the identifier component software object handles textualization of entities referenced in a database;

 wherein the expression component software object handles textualization of expressions.

69. (NEW) The method of claim 68 wherein the phrase component software object handles textualization of database WHERE phrases;

wherein the identifier component software object handles textualization of column names referenced in a database.

70. (NEW) The method of claim 69 wherein the first database statement contains an expression which contains a phrase which contains an identifier;

wherein the expression component software object processing the expression contained in the first database statement;

wherein the expression component software object invokes the phrase component software object in order to process the phrase contained in the first database statement;

wherein the phrase component software object invokes the identifier component software object in order to process the identifier contained in the first database statement.

REMARKS

Claims 1-62 are pending in the application. Claims 1, 45, 46, 49, 50, 60 and 61 are independent claims. Claims 63-70 have been added. Claim 62 is objected to. Claims 1-62 stand rejected by the examiner. Assignee traverses the instant claim rejections and objections.

Examiner's Interview

Assignee's representative would like to thank Examiner Betit and his Supervisor for the courtesies extended to assignee's representatives, Timothy Wilson, Gary Kuhn, Fred Levine, and John Biernacki, during the telephone interview on November 29, 2006. The interview discussed the cited reference Bodamer et al. (USPN 6,041,344) in view of claim 1. More specifically, the interview discussed Bodamer with respect to the office action's statements regarding "generating a second database statement..." step of claim 1. The interview discussed the term "Fourth-Generation language" as used within claim 1 and that SQL is an example of a Fourth-Generation language. The interview also discussed the cited reference Chow et al. (USPN 6,941,298). The remarks and the amendments contained herein summarize the interview.

Claim Objections

Claim 62 was objected to under 37 CFR 1.75(c) as being of improper dependent form for failing to further limit the subject matter of a previous claim. Assignee respectfully disagrees with the objection, but has amended claim 62 to remove reference to an override and instead recite determining whether to use a switch. Because of the

amendment to dependent claim 62, assignee respectfully requests that the objection to claim 62 be removed and that this application proceed to issuance.

Claim Rejections – 35 USC § 112

Claims 1-62 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, the office action maintained that the term "fourth-generation language" is not disclosed anywhere in the specification. Assignee respectfully traverses this rejection. As discussed in the interview, a non-limiting example of a fourth-generation language is SQL which is discussed in the specification. Accordingly, assignee respectfully submits that this rejection has been traversed and this case should proceed to issuance.

Claims 45, 49, 60 and 62 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, the office action maintained with respect to claims 45, 49, and 60 that these claims recite the limitation "[a] computer-implemented system" in their preambles. The office action maintained that it was not clear how the "system" differs from the computer, and if it differs from the computer it is not clear how one would differentiate between the computer and the system; and it is also not clear how the system would be implemented into the computer and not become a functional part of it. Assignee respectfully disagrees with the instant rejection, but to expedite prosecution of this application, assignee has amended claims 45, 49, and 60 based upon a suggestion made by the examiner during the interview.

Accordingly, assignee respectfully submits that these claims are in a condition for allowance and the application proceed to issuance.

With respect to claim 62, the office action maintained that the word "means" is preceded by the words "object-oriented component" in an attempt to use a "means" clause to recite a claim element as a means for performing a specified function, and since no function is specified by the word(s) preceding "means," it is impossible to determine the equivalents of the element, as required by 35 U.S.C. § 112, sixth paragraph. Assignee respectfully disagrees, but in order to expedite prosecution of this case, assignee has removed the word "means" from claim 62. Accordingly, assignee respectful submits that this claim is in a condition for allowance and the application should proceed to issuance.

Claim Rejections – 35 USC §§ 102 and 103

Claims 1-36, 38-39, and 43-62 stand rejection under 35 USC § 103(a) as being unpatentable over Bodamer et al. (U.S. Patent No. 6,041,344) in view of Chow et al. (U.S. Patent No. 6,941,298). Claims 37, 40-42 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Bodamer et al. (U.S. Patent No. 6,041,344) in view of Chow et al. (U.S. Patent No. 6,941,298) and in further view of the examiner's official notice. Assignee traverses these rejections.

Claim 1 is directed to a computer-implemented method for handling a first database statement from a first database system. The database statement is a fourth-generation language database statement formatted according to a language format used by the first database system. Database language difference data is accessed so that a second fourth-generation language database statement may be generated which is

operational within a different type of database system. As part of the process in converting the first fourth-generation language database statement to the second fourth-generation language database statement, a tree (that is representative of the syntax of the database language used within the first database system and of the metadata associated with the first database system) is used in generating the second fourth-generation language database statement.

The Bodamer reference does not disclose the limitations of claim 1. Bodamer appears to devote most of its disclosure to translations other than the database statement conversion that is the subject matter of claim 1, such as to a data dictionary translation. For example, the office action uses an excerpt from Bodamer (i.e., column 8, lines 38-67 of Bodamer) as the basis for anticipating the “generating a second fourth-generation language database statement ...” step of claim 1. However this passage from Bodamer is unrelated to a database statement conversion and instead is related to a different type of translation (i.e., the fourth type of translation discussed in Bodamer, namely the data dictionary translation).

This fourth type of translation is directed to handling database schema differences that might occur between two databases. This is significantly different than the database statement translation that is being performed in claim 1. Claim 1 looks at functional database language differences in order to generate a second database functional database statement. In contrast, the data dictionary translation of Bodamer examines two different databases’ schemata to determine schemata differences – that is, a data dictionary translation is performed in Bodamer because a “foreign database system 208, however, may include similar metadata that is organized differently”; see Bodamer at column 8,

lines 21-23.). As recognized by Bodamer itself (in establishing four categories of translations; see Bodamer at column 7, lines 18-20), performing database functional statement translation is different than performing data dictionary schemata translations. The assignee respectfully requests that the examiner cite to a passage within Bodamer that is discussing database statement translation and not to the other different types of translations. Because Bodamer does not disclose the limitations of claim 1, claim 1 cannot be anticipated by Bodamer and thus claim 1 is allowable.

Claim 1 also recites the use of database functional language difference data in order to generate the second SQL database statement. Claim 1's use of functional language differences to generate the second SQL database statement is advantageous, such as if there are only a few functional language differences between the first and second SQL environments, then only a few functional language differences have to be specified in order for the translation to occur. It is noted that Bodamer is lacking in any disclosure regarding the use of SQL language functional differences being specified and used to translate from a first database statement into a second database statement.

The other independent claims are allowable for similar reasons. For example, the office action uses passages from Bodamer that are discussing data dictionary translations and not the translations recited in these claims. Accordingly, Bodamer cannot anticipate these claims, and they should proceed to issuance.

Because the independent claims are allowable, the dependent claims are allowable as well and should proceed to issuance. More specifically with respect to claims 37, 40-42 which stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Bodamer et al. (U.S. Patent No. 6,041,344) in view of Chow et al. (U.S. Patent No. 6,941,298) and in

further view of the examiner's official notice, these claims (which depend from claim 1) are allowable, *inter alia*, at least for the reason that claim 1 is allowable. Accordingly, assignee respectfully submits that claims 37, 40-42 are allowable and should proceed to issuance.

Assignee has added claims 63-70 that depend directly or indirectly from claim 1. Assignee respectfully submits that none of the cited references disclose the limitations of these claims. For example, claim 63 recites that the tree contains logical pieces parsed from the first fourth-generation language database statement, and a plurality of component software objects are used to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text. A first component software object is associated with a first logical piece contained in the tree, and the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree. Because claim 63 contains additional patentable subject matter, claim 63 is allowable should proceed to issuance.

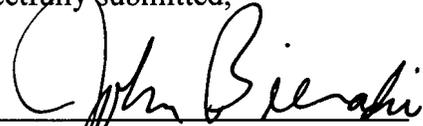
[Continued on next page]

CONCLUSION

For the foregoing reasons, Assignee respectfully submits that the pending claims are allowable. Therefore, the examiner is respectfully requested to pass this case to issue.

Respectfully submitted,

By: _____



John V. Biernacki
Reg. No. 40,511
JONES DAY
North Point
901 Lakeside Avenue
Cleveland, Ohio 44114
(216) 586-3939



UNITED STATES PATENT AND TRADEMARK OFFICE

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

AA

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/303,106	11/22/2002	Frederick J. Levine	343355600054	2037

7590 03/08/2007
 John V. Biernacki
 Jones, Day, Reavis & Pogue
 North Point
 901 Lakeside Avenue
 Cleveland, OH 44114

EXAMINER

BETIT, JACOB F

ART UNIT	PAPER NUMBER
2164	

2164

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
30 DAYS	03/08/2007	PAPER

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

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

**Notice of Non-Compliant
Amendment (37 CFR 1.121)**

Application No.	Applicant(s)	
10/303,106	LEVINE, FREDERICK J.	
Examiner	Art Unit	
Jacob F. Betit	2164	

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

The amendment document filed on 18 December 2006 is considered non-compliant because it has failed to meet the requirements of 37 CFR 1.121 or 1.4. In order for the amendment document to be compliant, correction of the following item(s) is required.

THE FOLLOWING MARKED (X) ITEM(S) CAUSE THE AMENDMENT DOCUMENT TO BE NON-COMPLIANT:

- 1. Amendments to the specification:
 - A. Amended paragraph(s) do not include markings.
 - B. New paragraph(s) should not be underlined.
 - C. Other _____.
- 2. Abstract:
 - A. Not presented on a separate sheet. 37 CFR 1.72.
 - B. Other _____.
- 3. Amendments to the drawings:
 - A. The drawings are not properly identified in the top margin as "Replacement Sheet," "New Sheet," or "Annotated Sheet" as required by 37 CFR 1.121(d).
 - B. The practice of submitting proposed drawing correction has been eliminated. Replacement drawings showing amended figures, without markings, in compliance with 37 CFR 1.84 are required.
 - C. Other _____.
- 4. Amendments to the claims:
 - A. A complete listing of all of the claims is not present.
 - B. The listing of claims does not include the text of all pending claims (including withdrawn claims)
 - C. Each claim has not been provided with the proper status identifier, and as such, the individual status of each claim cannot be identified. Note: the status of every claim must be indicated after its claim number by using one of the following status identifiers: (Original), (Currently amended), (Canceled), (Previously presented), (New), (Not entered), (Withdrawn) and (Withdrawn-currently amended).
 - D. The claims of this amendment paper have not been presented in ascending numerical order.
 - E. Other: _____.
- 5. Other (e.g., the amendment is unsigned or not signed in accordance with 37 CFR 1.4):
See Continuation Sheet

For further explanation of the amendment format required by 37 CFR 1.121, see MPEP § 714.

TIME PERIODS FOR FILING A REPLY TO THIS NOTICE:

- 1. Applicant is given **no new time period** if the non-compliant amendment is an after-final amendment or an amendment filed after allowance. If applicant wishes to resubmit the non-compliant after-final amendment with corrections, the **entire corrected amendment** must be resubmitted.
- 2. Applicant is given **one month**, or thirty (30) days, whichever is longer, from the mail date of this notice to supply the correction, if the non-compliant amendment is one of the following: a preliminary amendment, a non-final amendment (including a submission for a request for continued examination (RCE) under 37 CFR 1.114), a supplemental amendment filed within a suspension period under 37 CFR 1.103(a) or (c), and an amendment filed in response to a *Quayle* action. If any of above boxes 1. to 4. are checked, the correction required is only the **corrected section** of the non-compliant amendment in compliance with 37 CFR 1.121.

Extensions of time are available under 37 CFR 1.136(a) only if the non-compliant amendment is a non-final amendment or an amendment filed in response to a *Quayle* action.

Failure to timely respond to this notice will result in:

- Abandonment** of the application if the non-compliant amendment is a non-final amendment or an amendment filed in response to a *Quayle* action; or
- Non-entry** of the amendment if the non-compliant amendment is a preliminary amendment or supplemental amendment.

SAM RIMELL
PRIMARY EXAMINER

Legal Instruments Examiner (LIE), if applicable

Telephone No.

Continuation of 5 Other: The response fails to comply with 37 CFR §1.111(b) because it does not address every rejection given in the office action dated 10 August 2006. The response fails to address at least the rejections of claims 37, 40-42 under 35 USC 103(a). A reply must be reduced to writing which distinctly and specifically points out the supposed errors in the examiner's action and must reply to every ground of objection and rejection in the prior Office action. The reply must present arguments pointing out the specific distinctions believed to render the claims, including any newly presented claims, patentable over any applied references.



UNITED STATES PATENT AND TRADEMARK OFFICE

99

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/303,106	11/22/2002	Frédéric J. Levine	343355600054	2037

7590 12/18/2006
 John V. Biernacki
 Jones, Day, Reavis & Pogue
 North Point
 901 Lakeside Avenue
 Cleveland, OH 44114

EXAMINER

BETIT, JACOB F

ART UNIT	PAPER NUMBER
2164	

MAIL DATE	DELIVERY MODE
12/18/2006	PAPER

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

Interview Summary	Application No. 10/303,106	Applicant(s) LEVINE, FREDERICK J.	
	Examiner Jacob F. Betit	Art Unit 2164	

All participants (applicant, applicant's representative, PTO personnel):

- (1) Jacob F. Betit. (3) Fredrick J. Levine.
(2) Sam Rimell. (4) John V. Bernacki.

Date of Interview: 29 November 2006.

Type: a) Telephonic b) Video Conference
c) Personal [copy given to: 1) applicant 2) applicant's representative]

Exhibit shown or demonstration conducted: d) Yes e) No.
If Yes, brief description: _____.

Claim(s) discussed: 1 and 45.

Identification of prior art discussed: Bodamer et al. (US Pat. No. 6,041,344) and Chow et al. (US Pat. No. 6,941,298 B2).

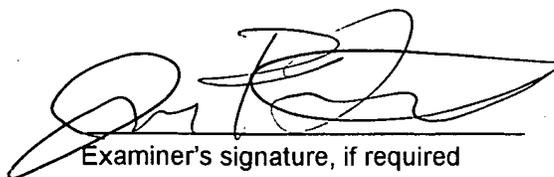
Agreement with respect to the claims f) was reached. g) was not reached. h) N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: See Continuation Sheet.

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.


Examiner's signature, if required

Continuation of Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments:

The examiner stated that the 112 first paragraph rejection would be withdrawn if it was conceded that SQL was equivalent to fourth-generation language. It was agreed that these were equivalents. The prior art was discussed. It was recommended that more detail of the particular methods disclosed in the specification be added to the claims in the form of dependents. This way the examiner can look at several of these methods when searching the prior art. It was recommended that the system claims be amended to include a processor in order to overcome the 35 USC §112 second paragraph rejection that was given on these claims.



Handwritten mark or signature in the top right corner.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTOR(S) : F. Levine

TITLE : Computer-Implemented System And Method For Handling Database Statements

SERIAL NO. : 10/303,106

FILING DATE : 11/22/2002

GROUP ART UNIT : 2164

EXAMINER : J. Betit

ATTORNEY DOCKET NO. : 343355600054

Mail Stop Amendment
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited today with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on December 8, 2006

By: *Kathleen Koppe*

RESPONSIVE AMENDMENT

Please consider the following amendments and remarks.

12/12/2006 JBALINAN 00000050 501432 10303106
 02 FC:1202 400.00 DA

CLAIMS

1. (PREVIOUSLY PRESENTED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format;

generating a second fourth-generation language database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

2. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the database statement functional difference specifies at least a portion of a statement format that is compatible with the second database system's query language format and that is incompatible with the first database system's query language format

wherein the tree contains a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be used in generating the second database statement.

3. (PREVIOUSLY PRESENTED) The method of claim 1 wherein object-oriented techniques are used to access the database functional language difference data.
4. (ORIGINAL) The method of claim 3 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.
5. (ORIGINAL) The method of claim 4 wherein a logical piece is a phrase logical piece.
6. (ORIGINAL) The method of claim 4 wherein a logical piece is an identifier logical piece.
7. (ORIGINAL) The method of claim 4 wherein an SQL component object defaults to a default native SQL textualization method for use in generating the second database statement.
8. (PREVIOUSLY PRESENTED) The method of claim 7 wherein an SQL component object comprises an override to account for functional differences between the first and second database systems' query language formats.
9. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise a phrase component object.

10. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise an identifier component object.
11. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise an expression component object.
12. (PREVIOUSLY PRESENTED) The method of claim 8 wherein SQL component objects comprise a parent component object.
13. (ORIGINAL) The method of claim 1 further comprising the steps of:
 - identifying, for the first database statement, query language parts that are common between the first database system's query language format and the second database system's query language format; and
 - generating the second database statement based upon the identified common query language parts.
14. (ORIGINAL) The method of claim 13 wherein the language parts are common based upon a predetermined standardized query language format.
15. (ORIGINAL) The method of claim 14 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.
16. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the database functional language difference data facilitates the generation of the second database statement by specifying common language parts between the first and second database system's language formats.

17. (ORIGINAL) The method of claim 16 wherein the language parts are common based upon a predetermined standardized query language format.
18. (ORIGINAL) The method of claim 17 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.
19. (ORIGINAL) The method of claim 1 wherein the second database system is a different type of database system than the first database system.
20. (ORIGINAL) The method of claim 1 wherein generating the second database statement provides the ability to manipulate data within the second database system from the first database system.
21. (ORIGINAL) The method of claim 1 wherein generating automatically the second database statement provides the ability to transparently manipulate data within the second database system from the first database system.
22. (ORIGINAL) The method of claim 1 wherein the generated second database statement is provided to the second database system for execution by the second database system.
23. (ORIGINAL) The method of claim 1 wherein the generated second database statement is in a format such that the second database statement is directly executable by the second database system.

24. (ORIGINAL) The method of claim 1 wherein the second database statement is a functional equivalent of the first database statement but for differences between the first and second database systems' query language formats.

25. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language formats are based upon a predetermined standardized query language format.

26. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the first and second database systems' query language formats are based upon a standardized fourth-generation structured query language (SQL) version.

27. (ORIGINAL) The method of claim 26 wherein the first database system's query language format utilizes a superset of the SQL standard.

28. (ORIGINAL) The method of claim 26 wherein the second database system's query language format utilizes a superset of the SQL standard.

29. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the first and second database systems' query language formats specify different formats for a preselected query-related function, wherein the first database statement is formatted in the first database system's query language format to perform the query-related function, wherein the second database statement is generated based upon the database functional difference data so as to be formatted in the second database system's query language format, wherein the generated second database statement is executable

within the second database system so as to perform the query-related function within the second database system.

30. (PREVIOUSLY PRESENTED) The method of claim 1 wherein the tree is an SQL tree that is used to generate the second database statement, wherein the SQL tree contains data that represents the syntax of the first database statement.

31. (ORIGINAL) The method of claim 30 wherein the SQL tree contains metadata related to the first database statement.

32. (ORIGINAL) The method of claim 31 wherein the first database statement is parsed into logical text pieces which are stored in the SQL tree.

33. (ORIGINAL) The method of claim 32 wherein the second database statement takes into account any second database system-specific query language syntax.

34. (PREVIOUSLY PRESENTED) The method of claim 1 further comprising the step of:

generating a third database statement for use within a third database system, wherein the third database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the third database statement is compatible with the third database system's query language format.

35. (PREVIOUSLY PRESENTED) The method of claim 34 further comprising the step of:

generating a fourth database statement for use within a fourth database system, wherein the fourth database statement is generated based upon the first database statement and

upon the accessed database functional language difference data, wherein the fourth database statement is compatible with the fourth database system's query language format.

36. (ORIGINAL) The method of claim 1 wherein the first and second database systems are relational database management systems.

37. (ORIGINAL) The method of claim 1 wherein the first database system comprises a data mining application.

38. (ORIGINAL) The method of claim 37 wherein the second database system comprises a relational database management system.

39. (ORIGINAL) The method of claim 1 wherein the first database system comprises a relational database management system.

40. (ORIGINAL) The method of claim 39 wherein the second database system comprises a data mining application.

41. (ORIGINAL) The method of claim 1 wherein the first database system comprises an enterprise resource planning system.

42. (ORIGINAL) The method of claim 41 wherein the second database system comprises an enterprise resource planning system.

43. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language format includes format specifications for insert, select, update, and delete database commands.

44. (ORIGINAL) Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 1.

45. (CURRENTLY AMENDED) A computer-implemented system that includes a processor for handling a first database fourth-generation language query that is formatted in a first query format and is executable by a first database system, comprising:

a data store to store tree-structured data that is representative of syntax and metadata of the first database fourth-generation language query;

a data structure for storing query specific data that indicates at least one query functional language difference from the first query format, wherein the query functional language difference is a query syntax difference; and

a textualization module having a data access connection to the tree-structured data and the data structure, wherein the textualization module generates a database specific query based upon the tree-structured data and the query specific data, wherein the database specific query accounts for the difference from the first query format so that the database specific query may be executed by a different type of database system.

46. (PREVIOUSLY PRESENTED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

- receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;
- identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and
- generating a second fourth-generation language database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

47. (ORIGINAL) The method of claim 46 wherein the language parts are determined to be common based upon a predetermined standardized query language format.

48. (ORIGINAL) The method of claim 47 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.

49. (CURRENTLY AMENDED) A computer-implemented system that includes a processor for handling a database statement from a first database system, comprising:

means for receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and

means for generating a second database fourth-generation language statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

50. (PREVIOUSLY PRESENTED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call

textualization specific data, wherein the generated program call is in a format that is compatible with the second database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call.

51. (ORIGINAL) The method of claim 50 wherein the program call textualization specific data is used to generate application program interface (API) calls from the first database statement.

52. (ORIGINAL) The method of claim 50 wherein object-oriented techniques are used to access the program call textualization specific data.

53. (ORIGINAL) The method of claim 52 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.

54. (ORIGINAL) The method of claim 53 wherein a logical piece is a phrase logical piece.

55. (ORIGINAL) The method of claim 53 wherein a logical piece is an identifier logical piece.

56. (ORIGINAL) The method of claim 50 wherein the first database system's query language format is based upon a standardized structured query language (SQL) version.

57. (PREVIOUSLY PRESENTED) The method of claim 50 further comprising the steps of:

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format; and

generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format.

58. (PREVIOUSLY PRESENTED) The method of claim 50 further comprising the steps of:

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format; and

generating a second database statement for use within a third database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the third database system's query language format.

59. (ORIGINAL) Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 50.

60. (CURRENTLY AMENDED) A computer-implemented system that includes a processor for handling a database statement from a first database system, comprising the steps of:

means for receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

means for generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the

program call textualization specific data, wherein the generated program call is in a format that is compatible with the second database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call for use within the second database system.

61. (PREVIOUSLY PRESENTED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

parsing the first database statement to obtain first query metadata of the first database statement;

using the obtained first query metadata to access database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format;

generating a second database fourth-generation language statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional

language difference data, wherein the second database statement is compatible with the second database system's query language format;

receiving another database statement from the first database system, wherein the additional database statement from the first database system is formatted according to the first database system's query language format;

parsing the additional database statement to obtain second query metadata of the additional database statement;

using the second query metadata to generate a program call to a third database system which utilizes a different query language format than the first database system; and

issuing the program call to the third database system to access data contained in the third database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

62. (CURRENTLY AMENDED) The method of claim 61 further comprising the steps of:

using an object-oriented component ~~means~~ for generating database access instructions to the second database system;

determining whether to use ~~an override~~ a switch, wherein the ~~override~~ switch indicates for the object-oriented component ~~means~~ to use the accessed database

functional language difference data first query metadata to generate a program call to the second database system; and

issuing the program call to the second database system to access data contained in the second database system,

wherein if the ~~override-switch~~ override-switch did not exist, then the object-oriented component ~~means-textualizes~~ textualizes a database statement based upon the first query metadata and which is executable within the second database system.

63. (NEW) The method of claim 1 wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree.

64. (NEW) The method of claim 1 wherein the tree contains logical pieces parsed from the first fourth-generation language database statement;

using a plurality of component software objects to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text;

wherein a first component software object is associated with a first logical piece contained in the tree;

wherein the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree;

using a plurality of software drivers to textualize logical pieces into fourth-generation database language text;

wherein a first software driver textualizes through a second method a logical piece into fourth-generation database language text that is compatible with the second database system's query language format;

wherein a second software driver textualizes through a third method a logical piece into fourth-generation database language text that is compatible with a third database system's query language format;

switching association of the first component software object from the first method to the second method for fourth-generation database language textualization;

wherein because of the switching of the association of the first component software object, the first component software object textualizes fourth-generation database language text that is compatible with the second database system's query

language format and that is not compatible with the first database system's query language format.

65. (NEW) The method of claim 64 wherein the first software driver's details of textualization into a different fourth-generation database language is hidden within the first software driver.

66. (NEW) The method of claim 64 wherein the switching of the association includes switching pointing of the first method to the second method for the first software driver.

67. (NEW) The method of claim 64 wherein the plurality of component software objects includes a phrase component software object, an identifier component software object, and an expression component software object.

68. (NEW) The method of claim 67 wherein the phrase component software object handles textualization of database phrases;
wherein the identifier component software object handles textualization of entities referenced in a database;

wherein the expression component software object handles textualization of expressions.

69. (NEW) The method of claim 68 wherein the phrase component software object handles textualization of database WHERE phrases;

wherein the identifier component software object handles textualization of column names referenced in a database.

70. (NEW) The method of claim 69 wherein the first database statement contains an expression which contains a phrase which contains an identifier;

wherein the expression component software object processing the expression contained in the first database statement;

wherein the expression component software object invokes the phrase component software object in order to process the phrase contained in the first database statement;

wherein the phrase component software object invokes the identifier component software object in order to process the identifier contained in the first database statement.

REMARKS

Claims 1-62 are pending in the application. Claims 1, 45, 46, 49, 50, 60 and 61 are independent claims. Claims 63-70 have been added. Claim 62 is objected to. Claims 1-62 stand rejected by the examiner. Assignee traverses the instant claim rejections and objections.

Examiner's Interview

Assignee's representative would like to thank Examiner Betit and his Supervisor for the courtesies extended to assignee's representatives, Timothy Wilson, Gary Kuhn, Fred Levine, and John Biernacki, during the telephone interview on November 29, 2006. The interview discussed the cited reference Bodamer et al. (USPN 6,041,344) in view of claim 1. More specifically, the interview discussed Bodamer with respect to the office action's statements regarding "generating a second database statement..." step of claim 1. The interview discussed the term "Fourth-Generation language" as used within claim 1 and that SQL is an example of a Fourth-Generation language. The interview also discussed the cited reference Chow et al. (USPN 6,941,298). The remarks and the amendments contained herein summarize the interview.

Claim Objections

Claim 62 was objected to under 37 CFR 1.75(c) as being of improper dependent form for failing to further limit the subject matter of a previous claim. Assignee respectfully disagrees with the objection, but has amended claim 62 to remove reference to an override and instead recite determining whether to use a switch. Because of the

amendment to dependent claim 62, assignee respectfully requests that the objection to claim 62 be removed and that this application proceed to issuance.

Claim Rejections – 35 USC § 112

Claims 1-62 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, the office action maintained that the term "fourth-generation language" is not disclosed anywhere in the specification. Assignee respectfully traverses this rejection. As discussed in the interview, a non-limiting example of a fourth-generation language is SQL which is discussed in the specification. Accordingly, assignee respectfully submits that this rejection has been traversed and this case should proceed to issuance.

Claims 45, 49, 60 and 62 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. More specifically, the office action maintained with respect to claims 45, 49, and 60 that these claims recite the limitation "[a] computer-implemented system" in their preambles. The office action maintained that it was not clear how the "system" differs from the computer, and if it differs from the computer it is not clear how one would differentiate between the computer and the system; and it is also not clear how the system would be implemented into the computer and not become a functional part of it. Assignee respectfully disagrees with the instant rejection, but to expedite prosecution of this application, assignee has amended claims 45, 49, and 60 based upon a suggestion made by the examiner during the interview.

Accordingly, assignee respectfully submits that these claims are in a condition for allowance and the application proceed to issuance.

With respect to claim 62, the office action maintained that the word "means" is preceded by the words "object-oriented component" in an attempt to use a "means" clause to recite a claim element as a means for performing a specified function, and since no function is specified by the word(s) preceding "means," it is impossible to determine the equivalents of the element, as required by 35 U.S.C. § 112, sixth paragraph. Assignee respectfully disagrees, but in order to expedite prosecution of this case, assignee has removed the word "means" from claim 62. Accordingly, assignee respectful submits that this claim is in a condition for allowance and the application should proceed to issuance.

Claim Rejections – 35 USC §§ 102 and 103

Claims 1-36, 38-39, and 43-62 stand rejection under 35 USC § 103(a) as being unpatentable over by Bodamer et al. (U.S. Patent No. 6,041,344) in view of Chow et al. (U.S. Patent No. 6,941,298).

Claim 1 is directed to a computer-implemented method for handling a first database statement from a first database system. The database statement is a fourth-generation language database statement formatted according to a language format used by the first database system. Database language difference data is accessed so that a second fourth-generation language database statement may be generated which is operational within a different type of database system. As part of the process in converting the first fourth-generation language database statement to the second fourth-generation language database statement, a tree (that is representative of the syntax of the

database language used within the first database system and of the metadata associated with the first database system) is used in generating the second fourth-generation language database statement.

The Bodamer reference does not disclose the limitations of claim 1. Bodamer appears to devote most of its disclosure to translations other than the database statement conversion that is the subject matter of claim 1, such as to a data dictionary translation. For example, the office action uses an excerpt from Bodamer (i.e., column 8, lines 38-67 of Bodamer) as the basis for anticipating the “generating a second fourth-generation language database statement ...” step of claim 1. However this passage from Bodamer is unrelated to a database statement conversion and instead is related to a different type of translation (i.e., the fourth type of translation discussed in Bodamer, namely the data dictionary translation).

This fourth type of translation is directed to handling database schema differences that might occur between two databases. This is significantly different than the database statement translation that is being performed in claim 1. Claim 1 looks at functional database language differences in order to generate a second database functional database statement. In contrast, the data dictionary translation of Bodamer examines two different databases’ schemata to determine schemata differences – that is, a data dictionary translation is performed in Bodamer because a “foreign database system 208, however, may include similar metadata that is organized differently”; see Bodamer at column 8, lines 21-23.). As recognized by Bodamer itself (in establishing four categories of translations; see Bodamer at column 7, lines 18-20), performing database functional statement translation is different than performing data dictionary schemata translations.

The assignee respectfully requests that the examiner cite to a passage within Bodamer that is discussing database statement translation and not to the other different types of translations. Because Bodamer does not disclose the limitations of claim 1, claim 1 cannot be anticipated by Bodamer and thus claim 1 is allowable.

Claim 1 also recites the use of database functional language difference data in order to generate the second SQL database statement. Claim 1's use of functional language differences to generate the second SQL database statement is advantageous, such as if there are only a few functional language differences between the first and second SQL environments, then only a few functional language differences have to be specified in order for the translation to occur. It is noted that Bodamer is lacking in any disclosure regarding the use of SQL language functional differences being specified and used to translate from a first database statement into a second database statement.

The other independent claims are allowable for similar reasons. For example, the office action uses passages from Bodamer that are discussing data dictionary translations and not the translations recited in these claims. Accordingly, Bodamer cannot anticipate these claims, and they should proceed to issuance.

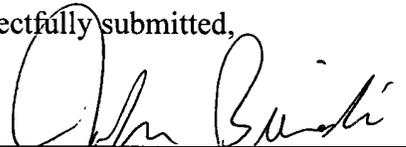
Assignee has added claims 63-70 that depend directly or indirectly from claim 1. Assignee respectfully submits that none of the cited references disclose the limitations of these claims. For example, claim 63 recites that the tree contains logical pieces parsed from the first fourth-generation language database statement, and a plurality of component software objects are used to textualize the logical pieces contained in the tree, wherein textualizing a logical piece includes generating fourth-generation database language text. A first component software object is associated with a first logical piece

contained in the tree, and the first component software object is associated with a first method to textualize, into fourth-generation database language text, the first component software object's associated logical piece that is contained in the tree. Because claim 63 contains additional patentable subject matter, claim 63 is allowable should proceed to issuance.

CONCLUSION

For the foregoing reasons, Assignee respectfully submits that the pending claims are allowable. Therefore, the examiner is respectfully requested to pass this case to issue.

Respectfully submitted,

By: 

John V. Biernacki
Reg. No. 40,511
JONES DAY
North Point
901 Lakeside Avenue
Cleveland, Ohio 44114
(216) 586-3939



PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) FY 2006 <i>(Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).)</i>		Docket Number (Optional) 343355600054
Application Number 10/303,106	Filed November 22, 2002	
For Computer-Implemented System And Method For Handling Database Statements		
Art Unit 2164	Examiner Jacob F. Betit	

This is a request under the provisions of 37 CFR 1.136(a) to extend the period for filing a reply in the above identified application.

The requested extension and fee are as follows (check time period desired and enter the appropriate fee below):

	Fee	Small Entity Fee	
<input checked="" type="checkbox"/> One month (37 CFR 1.17(a)(1))	\$120	\$60	\$ <u>120.00</u>
<input type="checkbox"/> Two months (37 CFR 1.17(a)(2))	\$450	\$225	\$ _____
<input type="checkbox"/> Three months (37 CFR 1.17(a)(3))	\$1020	\$510	\$ _____
<input type="checkbox"/> Four months (37 CFR 1.17(a)(4))	\$1590	\$795	\$ _____
<input type="checkbox"/> Five months (37 CFR 1.17(a)(5))	\$2160	\$1080	\$ _____

- Applicant claims small entity status. See 37 CFR 1.27.
- A check in the amount of the fee is enclosed.
- Payment by credit card. Form PTO-2038 is attached.
- The Director has already been authorized to charge fees in this application to a Deposit Account.
- The Director is hereby authorized to charge any fees which may be required, or credit any overpayment, to Deposit Account Number 501432. I have enclosed a duplicate copy of this sheet.

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

- I am the
- applicant/inventor.
 - assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed (Form PTO/SB/96).
 - attorney or agent of record. Registration Number 40,511
 - attorney or agent under 37 CFR 1.34.
Registration number if acting under 37 CFR 1.34 _____



 Signature
John V. Biernacki

 Typed or printed name

December 8, 2006

 Date
(216) 586-7747

 Telephone Number

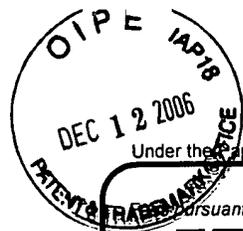
NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.136(a). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 6 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

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

12/12/2006 JRALINAN 00000050 501432 120.00 DA 01 FC:1251



Effective on 12/08/2004.
 Pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).
FEE TRANSMITTAL
For FY 2005

Complete if Known	
Application Number	10/303,106
Filing Date	November 22, 2002
First Named Inventor	Levine
Examiner Name	Jacob F. Betit
Art Unit	2164
Attorney Docket No.	343355600054

Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT (\$) 520.00

METHOD OF PAYMENT (check all that apply)

Check Credit Card Money Order None Other (please identify): _____

Deposit Account Deposit Account Number: 501432 Deposit Account Name: Jones Day

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

Charge fee(s) indicated below Charge fee(s) indicated below, except for the filing fee

Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17 Credit any overpayments

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

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

Application Type	FILING FEES		SEARCH FEES		EXAMINATION FEES		Fees Paid (\$)
	Fee (\$)	Small Entity Fee (\$)	Fee (\$)	Small Entity Fee (\$)	Fee (\$)	Small Entity Fee (\$)	
Utility	300	150	500	250	200	100	_____
Design	200	100	100	50	130	65	_____
Plant	200	100	300	150	160	80	_____
Reissue	300	150	500	250	600	300	_____
Provisional	200	100	0	0	0	0	_____

2. EXCESS CLAIM FEES

Fee Description	Fee (\$)	Small Entity Fee (\$)
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	200	100
Multiple dependent claims	360	180

Total Claims - 20 or HP = 8 x 50 = 400.00

HP = highest number of total claims paid for, if greater than 20.

Indep. Claims - 3 or HP = _____ x _____ = _____

HP = highest number of independent claims paid for, if greater than 3.

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is \$250 (\$125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets	Extra Sheets	Number of each additional 50 or fraction thereof	Fee (\$)	Fee Paid (\$)
_____ - 100 = _____	_____ / 50 = _____	(round up to a whole number) x _____	_____	_____

4. OTHER FEE(S)

Description	Fees Paid (\$)
Non-English Specification, \$130 fee (no small entity discount)	_____
Other (e.g., late filing surcharge): <u>Petition for Extension of Time</u>	<u>120.00</u>

SUBMITTED BY

Signature		Registration No. (Attorney/Agent) 40,511	Telephone (216) 586-3939
Name (Print/Type)	John V. Biernacki		Date December 8, 2006

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

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



UNITED STATES PATENT AND TRADEMARK OFFICE

Tr

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/303,106	11/22/2002	Frederick J. Levine	343355600054	2037

7590 08/10/2006
 John V. Biernacki
 Jones, Day, Reavis & Pogue
 North Point
 901 Lakeside Avenue
 Cleveland, OH 44114

EXAMINER

BETIT, JACOB F

ART UNIT	PAPER NUMBER
2164	

2164

DATE MAILED: 08/10/2006

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

Office Action Summary	Application No. 10/303,106	Applicant(s) LEVINE, FREDERICK J.	
	Examiner Jacob F. Betit	Art Unit 2164	

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

Period for Reply

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

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

Status

- 1) Responsive to communication(s) filed on 12 May 2006.
- 2a) This action is **FINAL**. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-62 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-62 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

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


SAM RIMELL
PRIMARY EXAMINER

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

Remarks

1. In response to communications filed on 12 May 2006, claims 1-3, 8-12, 16, 26, 29-30, 34-35, 45-46, 49, 50, 57-58, 60-62 have been amended per the applicant's request. Claims 1-62 are presently pending in the application.

Claim Objections

2. Claim 62 objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. "A claim in dependent form shall be construed to incorporate by reference all the limitations of the claim to which it refers" (see 35 U.S.C. §112 fourth paragraph). Because claim 62 specifies for an "override mechanism" that could eliminate steps from claim 61 it does not incorporate all the limitations of 61 under all circumstances. Claim 62 makes optional (on condition of an override) steps that were required in claim 61. MPEP 2106 II. C. states that optionally recited limitations are not given patentable weight. Therefore claim 62 is not in proper dependent form.

Claim Rejections - 35 USC § 112

3. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

4. Claims 1-62 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. "Fourth-Generation language" is not disclosed anywhere in the specification.

Claims 2-44 are rejected for being dependent on rejected claim 1.

Claims 47-48 are rejected for being dependent on rejected claim 46.

Claims 51-59 are rejected for being dependent on rejected claim 50.

Claim 62 is rejected for being dependent on rejected claim 61.

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claims 45, 49, 60 and 62 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

7. Claims 45, 49, and 60 recite the limitation "[a] computer-implemented system" in their preambles. It is not clear how the "system" differs from the computer, and if it differs from the computer it is not clear how one would differentiate between the computer and the system. It is also not clear how the system would be implemented into the computer and not become a functional part of it.

Art Unit: 2164

8. Claim 45 recites the limitation “a computer-implemented system for handling a first database [~~fourth generation language~~] query that is formatted in a first query format and is executable by a first database system”. It is not clear from this limitation if the computer-implemented system is the same as the “first database system”; and if it is not, it is not clear the difference between “handling” a query and executing a query.

9. Regarding claim 62, the word "means" is preceded by the word(s) "object-oriented component" in an attempt to use a "means" clause to recite a claim element as a means for performing a specified function. However, since no function is specified by the word(s) preceding "means," it is impossible to determine the equivalents of the element, as required by 35 U.S.C. 112, sixth paragraph. See *Ex parte Klumb*, 159 USPQ 694 (Bd. App. 1967).

10. Appropriate corrections are required.

Claim Rejections - 35 USC § 103

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

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

12. Claims 1-36, 38-39, and 43-62 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bodamer et al. (U.S. patent No. 6,041,344) in view of Chow et al. (U.S. patent No. 6,941,298 B2).

As to claim 1, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first [~~fourth-generation language~~] database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

accessing database functional language difference data, wherein the database language difference data indicates a format that contains at least one database statement difference from the first database system's query language format (see column 7, line 18 through column 8, line 37); and

generating a second [~~fourth-generation language~~] database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format (see column 8, line 38-67);

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 2, Bodamer et al. as modified teaches wherein the database statement functional difference specifies at least a portion of a statement format that is compatible with the second database system's query language format and that is incompatible with the first database system's query language format (see Bodamer et al., column 7, line 18 through column 8, line 37)

wherein the tree contains a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be used in generating the second database statement (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

As to claim 3, Bodamer et al. as modified teaches wherein object-oriented techniques are used to access the database functional language difference data (see Bodamer et al., column 5, lines 7-54).

As to claim 4, Bodamer et al. as modified teaches wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement (see Bodamer et al., column 7, lines 43-67).

As to claim 5, Bodamer et al. as modified teaches wherein a logical piece is a phrase logical piece (see Bodamer et al., column 8, lines 47-67).

As to claim 6, Bodamer et al. as modified teaches wherein a logical piece is an identifier logical piece (see Bodamer et al., column 8, lines 47-67).

As to claim 7, Bodamer et al. as modified teaches wherein an SQL component object defaults to a default native SQL textualization method for use in generating the second database statement (see Bodamer et al., column 7, lines 43-67).

As to claim 8, Bodamer et al. teaches wherein an SQL component object comprises an override to account for functional differences between the first and second database systems' query language formats (see Bodamer et al., column 7, lines 43-67)

As to claim 9, Bodamer et al. as modified teaches wherein SQL component objects comprise s phrase component object (see Bodamer et al., column 8, lines 47-67).

As to claim 10, Bodamer et al. as modified teaches wherein SQL component objects comprise an identifier component object (see Bodamer et al., column 8, lines 47-67).

As to claim 11, Bodamer et al. as modified teaches wherein SQL component objects comprise an expression component object (see Bodamer et al., column 8, lines 47-67).

As to claim 12, Bodamer et al. as modified teaches wherein SQL component objects comprise a parent component object (see Bodamer et al., column 8, lines 47-67).

As to claim 13, Bodamer et al. as modified teaches further comprising the steps of: identifying, for the first database statement, query language parts that are common between the first database system's query language format and the second database system's query language format; and generating the second database statement based upon the identified common query language parts (see Bodamer et al., column 7, lines 43-67).

As to claim 14, Bodamer et al. as modified teaches wherein the language parts are common based upon a predetermined standardized query language format (see Bodamer et al., column 7, lines 43-67).

As to claim 15, Bodamer et al. as modified teaches wherein the standardized query language format is based upon a standardized structured query language (SQL) version (see Bodamer et al., column 7, lines 43-67).

As to claim 16, Bodamer et al. as modified teaches wherein the database functional language difference data facilitates the generation of the second database statement by specifying common language parts between the first and second database system's language formats (see Bodamer et al., column 7, lines 43-67).

As to claim 17, Bodamer et al. as modified teaches wherein the language parts are common based upon a predetermined standardized query language format (see Bodamer et al., column 7, lines 18-67).

As to claim 18, Bodamer et al. as modified teaches wherein the standardized query language format is based upon a standardized structured query language (SQL) version (see Bodamer et al., column 7, lines 18-67).

As to claim 19, Bodamer et al. as modified teaches wherein the second database system is a different type of database system than the first database system (see Bodamer et al., column 7, lines 18-67).

As to claim 20, Bodamer et al. as modified teaches wherein generating the second database statement provides the ability to manipulate data within the second database system from the first database system (see Bodamer et al., column 4, line 40 through column 5, line 20).

As to claim 21, Bodamer et al. as modified teaches wherein generating automatically the second database statement provides the ability to transparently manipulate data within the second database system from the first database system (see Bodamer et al., column 8, lines 38-46).

Art Unit: 2164

As to claim 22, Bodamer et al. as modified teaches wherein the generated second database statement is provided to the second database system for execution by the second database system (see Bodamer et al., column 8, lines 38-46).

As to claim 23, Bodamer et al. as modified teaches wherein the generated second database statement is in a format such that the second database statement is directly executable by the second database system (see Bodamer et al., column 8, lines 47-67).

As to claim 24, Bodamer et al. as modified teaches wherein the second database statement is a functional equivalent of the first database statement but for differences between the first and second database systems' query language formats (see Bodamer et al., column 7, lines 9-17).

As to claim 25, Bodamer et al. as modified teaches wherein the first and second database systems' query language formats are based upon a predetermined standardized query language format (see Bodamer et al., column 7, lines 43-67).

As to claim 26, Bodamer et al. as modified teaches wherein the first and second database systems' query language formats are based upon a standardized [~~fourth-generation~~] structured query language (SQL) version (see Bodamer et al., column 7, lines 43-67).

Art Unit: 2164

As to claim 27, Bodamer et al. as modified teaches wherein the first database system's query language format utilizes a superset of the SQL standard (see Bodamer et al., column 7, lines 25-30).

As to claim 28, Bodamer et al. as modified teaches wherein the second database system's query language format utilizes a superset of the SQL standard (see Bodamer et al., column 7, lines 30-35).

As to claim 29, Bodamer et al. as modified teaches wherein the first and second database systems' query language formats specify different formats for a preselected query-related function, wherein the first database statement is formatted in the first database system's query language format to perform the query-related function, wherein the second database statement is generated based upon the database functional difference data so as to be formatted in the second database system's query language format, wherein the generated second database statement is executable within the second database system so as to perform the query-related function within the second database system (see Bodamer et al., column 7, line 18 through column 8, line 37).

As to claim 30, Bodamer et al. as modified teaches wherein the tree is an SQL tree that is used to generate the second database statement, wherein the SQL tree contains data that represents the syntax of the first database statement (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

Art Unit: 2164

As to claim 31, Bodamer et al. as modified teaches wherein the SQL tree contains metadata related to the first database statement (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

As to claim 32, Bodamer et al. as modified teaches wherein the first database statement is parsed into logical text pieces which are stored in the SQL tree (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

As to claim 33, Bodamer et al. as modified teaches wherein the second database statement takes into account any second database system-specific query language syntax (see Chow et al., column 3, lines 16-47 and see column 6 line 64 through column 7, line 14).

As to claim 34, Bodamer et al. as modified teaches further comprising the step of:
generating a third database statement for use within a third database system, wherein the third database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the third database statement is compatible with the third database system's query language format (see Bodamer et al., column 4, line 59 through column 5, line 20 and see column 7, line 18 through column 8, line 37).

As to claim 35, Bodamer et al. as modified teaches further comprising the step of:
generating a fourth database statement for use within a fourth database system, wherein the fourth database statement is generated based upon the first database statement and upon the

Art Unit: 2164

accessed database functional language difference data, wherein the fourth database statement is compatible with the fourth database system's query language format (see Bodamer et al., column 4, line 59 through column 5, line 20 and see column 7, line 18 through column 8, line 37).

As to claim 36, Bodamer et al. as modified teaches wherein the first and second database systems are relational database management systems (see Bodamer et al., column 1, lines 44-52).

As to claim 39, Bodamer et al. as modified teaches wherein the first database system comprises a relational database management system (see Bodamer et al., column 1, lines 44-52).

As to claim 38, Bodamer et al. as modified teaches wherein the second database system comprises a relational database management system (see Bodamer et al., column 8, lines 10-46).

As to claim 43, Bodamer et al. as modified teaches wherein the first and second database systems' query language format includes format specifications for insert, select, update, and delete database commands (see Bodamer et al., column 16, line 64 through column 17, line 3).

As to claim 44, Bodamer et al. as modified teaches computer software stored on a computer readable media, the computer software comprising program code (see Bodamer et al., column 3, line 65 through column 4, line 38) for carrying out a method according to claim 1 (for the rejection of the limitations of claim 1, the applicant is directed to the rejection of claim 1 above).

As to claim 45, Bodamer et al. teaches a computer-implemented system, for handling a first database [~~fourth-generation language~~] query that is formatted in a first query format and is executable by a first database system, comprising:

a data structure for storing query specific data that indicates at least one query functional language difference from the first query format, wherein the query functional language difference is a query syntax difference (see column 7, line 18 through column 8, line 37); and

a textualization module having a data access connection to data and the data structure, wherein the textualization module generates a database specific query based on data and the query specific data, wherein the database specific query accounts for the difference from the first query format so that the database specific query may be executed by a different type of database system (see column 8, lines 38-67).

Bodamer et al. does not teach

a. a data store to store tree-structured data that is representative of syntax and metadata of the first database [~~fourth-generation language~~] query;

b. a textualization module having a data access connection to the tree-structured data and the data structure, wherein the textualization module generates a database specific query based upon the tree-structured data and the query specific data.

Chow et al. teaches a. and b., see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of

Art Unit: 2164

Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 46, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first [~~fourth-generation language~~] database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format (see column 7, line 18 through column 8, line 37); and

generating a second [~~fourth-generation language~~] database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format (see column 8, lines 38-67).

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of

Art Unit: 2164

Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 47 and 48, see the rejection of claim 14 and 15 respectively.

As to claim 49, Bodamer et al. teaches a computer-implemented system for handling a database statement from a first database system, comprising:

means for receiving a first database [~~fourth-generation language~~] statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

means for identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format (see column 7, line 18 through column 8, line 37); and

means for generating a second database [~~fourth-generation language~~] statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format (see column 8, lines 38-67);

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

Art Unit: 2164

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 50, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database [~~fourth-generation-language~~] statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system (see column 7, line 18 through column 8, line 37); and

generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call textualization specific data, wherein the generated program call is in a format that is compatible with the second database system (see column 8, lines 38-67);

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call.

Art Unit: 2164

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 51, Bodamer et al. as modified teaches wherein the program call textualization specific data is used to generate application program interface (API) calls from the first database statement (see Bodamer et al., column 7, lines 18-35).

As to claims 52, 53, 54, and 55; see the rejections of claims 3, 4, 5, and 6 respectively.

As to claim 56, Bodamer et al. as modified teaches wherein the first database system's query language format is based upon a standardized structured query language (SQL) version (see Bodamer et al., column 4, lines 40-58).

As to claim 57, Bodamer et al. as modified teaches further comprising the steps of:
accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format (see Bodamer et al., column 5, line 63 through column 6, line 15); and

generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format (see Bodamer et al., column 8, lines 38-67).

As to claim 58, Bodamer et al. as modified teaches further comprising the steps of:
accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format (see Bodamer et al., column 5, line 63 through column 6, line 15); and

generating a second database statement for use within a third database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the third database system's query language format (see Bodamer et al., column 8, lines 38-67 and see figure 2A, reference number 300).

As to claim 59, see the rejection of claim 44 above.

As to claim 60, Bodamer et al. teaches a computer-implemented system for handling a database statement from a first database system, comprising the steps of:

means for receiving a first database [~~fourth-generation language~~] statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

means for accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system (see column 7, line 18 through column 8, line 37); and

means for generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call textualization specific data, wherein the generated program call is in a format that is compatible with the second database system (see column 8, lines 38-67);

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of the metadata associated with the first database system is used in generating the program call for use within the second database system.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 61, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

Art Unit: 2164

receiving a first database [~~fourth-generation language~~] statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

parsing the first database statement to obtain first query metadata of the first database statement; using the obtained first query metadata to access database language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format (see column 7, line 18 through column 8, line 37);

generating a second database [~~fourth-generation language~~] statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format (see column 8, lines 38-67);

receiving another database statement from the first database system, wherein the additional database statement from the first database system is formatted according to the first database system's query language format (see figure 2A, reference number 300 and see column 4, lines 40-58);

parsing the additional database statement to obtain second query metadata of the additional database statement; using the second query metadata to generate a program call to a third database system which utilizes a different query language format than the first database system (see column 7, line 18 through column 8, line 37); and

issuing the program call to the third database system to access data contained in the third database system (see column 8, lines 38-67).

Bodamer et al. does not teach wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

Chow et al. teaches this, see column 3, lines 16-47 and see column 6 line 64 through column 7, line 14. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include the teachings of Chow et al. because these teachings would be a way of translating a query from one language to another language efficiently.

As to claim 62, Bodamer et al. as modified teaches further comprising the steps of:
using object-oriented component means for generating database access instructions to the second database system (see Bodamer et al., column 5, lines 7-20);

determining whether to use an override, wherein the override indicates for the object-oriented component means to use the first query metadata to generate a program call to the second database system (see Bodamer et al., column 5, line 63 through column 6, line 15); and

issuing the program call to the second database system to access data contained in the second database system (see Bodamer et al., column 8, lines 38-67),

wherein if the override did not exist, then the object-oriented component means textualizes a database statement based upon the first query metadata and which is executable

Art Unit: 2164

within the second database system (this limitation is optionally recited and does not properly depend from 61 because it acts as though to exclude limitations of that claim).

13. Claims 37, 40-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bodamer et al. (U.S. patent No. 6,041,344) in view Chow et al. (U.S. patent No. 6,941,298 B2) as applied to claims 1-36, 38-39, and 43-62 above and in further view of the examiner's official notice.

As to claim 37, Bodamer et al. as modified, does not teach wherein the first database system comprises a data mining application.

The examiner takes official notice that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because many databases include data mining applications so that they can acquire new data and so they can more easily search the data they currently contain.

As to claim 40, Bodamer et al. as modified, does not teach wherein the second database system comprises a data mining application.

The examiner takes official notice that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because many databases include data mining applications so that they can acquire new data and so they can more easily search the data they currently contain.

As to claim 41, Bodamer et al. as modified, does not teach wherein the first database system comprises an enterprise resource planning system.

The examiner takes official notice that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because this would allow a company to plan appropriate resources for different projects being performed within a company which is a common use for a database in an corporate environment.

As to claim 42, Bodamer et al. as modified, does not teach wherein the second database system comprises an enterprise resource planning system.

The examiner takes official notice that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because this would allow a company to plan appropriate resources for different projects being performed within a company which is a common use for a database in an corporate environment.

Response to Arguments

14. Applicant's arguments with respect to conversion of a "fourth-generation language" statement have been considered but are moot in view of the new ground(s) of rejection under 35 U.S.C. §112 first paragraph given above.

15. Applicant's arguments with respect to Bodamer not disclosing a tree being used in generating database statements have been considered but are moot in view of the new ground(s) of rejection under 35 U.S.C. § 103(a) given above.

Conclusion

16. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. patent No. 5,421,008 to Banning et al. for teaching graphical construction of a database query and storing the query object links as an object.

U.S. patent No. 5,590,319 to Cohen et al. for teaching query processing for parallel processing in homogenous and *heterogeneous* databases using parse trees.

U.S. patent No. 5,659,725 to Levy et al. for teaching query optimization by predicate move-around.

17. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob F. Betit whose telephone number is (571) 272-4075. The examiner can normally be reached on Monday through Friday 9:30 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Rones can be reached on (571) 272-4085. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

jfb
7 Aug 2006


SAM RIMELL
PRIMARY EXAMINER

Notice of References Cited	Application/Control No. 10/303,106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.	
	Examiner Jacob F. Betit	Art Unit 2164	Page 1 of 1

U.S. PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A US-5,421,008 A	05-1995	Banning et al.	707/4
*	B US-5,590,319 A	12-1996	Cohen et al.	707/4
*	C US-5,659,725	08-1997	Levy et al.	707/3
*	D US-6,941,298	09-2005	Chow et al.	707/3
	E US-			
	F US-			
	G US-			
	H US-			
	I US-			
	J US-			
	K US-			
	L US-			
	M US-			

FOREIGN PATENT DOCUMENTS

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

NON-PATENT DOCUMENTS

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

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

Index of Claims



Application/Control No.

10303106

Applicant(s)/Patent under Reexamination

LEVINE, FREDERICK J.

Examiner

Betit, Jacob F

Art Unit

2164

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claims renumbered in the same order as presented by applicant

CPA

T.D.

R.1.47

CLAIM		DATE									
Final	Original	2/6/06	8/6/06								
	1	✓	✓								
	2	✓	✓								
	3	✓	✓								
	4	✓	✓								
	5	✓	✓								
	6	✓	✓								
	7	✓	✓								
	8	✓	✓								
	9	✓	✓								
	10	✓	✓								
	11	✓	✓								
	12	✓	✓								
	13	✓	✓								
	14	✓	✓								
	15	✓	✓								
	16	✓	✓								
	17	✓	✓								
	18	✓	✓								
	19	✓	✓								
	20	✓	✓								
	21	✓	✓								
	22	✓	✓								
	23	✓	✓								
	24	✓	✓								
	25	✓	✓								
	26	✓	✓								
	27	✓	✓								
	28	✓	✓								
	29	✓	✓								
	30	✓	✓								
	31	✓	✓								
	32	✓	✓								
	33	✓	✓								

Index of Claims



Application/Control No.

10303106

Applicant(s)/Patent under Reexamination

LEVINE, FREDERICK J.

Examiner

Betit, Jacob F

Art Unit

2164

✓	Rejected
=	Allowed

-	Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

Claims renumbered in the same order as presented by applicant

CPA

T.D.

R.1.47

CLAIM		DATE									
Final	Original	2/6/06	8/6/06								
	34	✓	✓								
	35	✓	✓								
	36	✓	✓								
	37	✓	✓								
	38	✓	✓								
	39	✓	✓								
	40	✓	✓								
	41	✓	✓								
	42	✓	✓								
	43	✓	✓								
	44	✓	✓								
	45	✓	✓								
	46	✓	✓								
	47	✓	✓								
	48	✓	✓								
	49	✓	✓								
	50	✓	✓								
	51	✓	✓								
	52	✓	✓								
	53	✓	✓								
	54	✓	✓								
	55	✓	✓								
	56	✓	✓								
	57	✓	✓								
	58	✓	✓								
	59	✓	✓								
	60	✓	✓								
	61	✓	✓								
	62	✓	✓								

Search Notes 	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10303106	LEVINE, FREDERICK J.
	Examiner Betit, Jacob F	Art Unit 2164

Notes	Date	Examiner
searched EAST enclosed	02/06/2006	JFB
updated prior search	8/6/06	JFB
spoke to Pr Sam Rimell about 112 first issue	7/27/06	JFB
U.S. Patent and Trademark Office		Part of Paper No.:

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	13	sql with tree with syntax	US-PGPUB; USPAT; EPO	OR	ON	2006/08/06 18:20
L2	41	sql with tree with translat\$4	US-PGPUB; USPAT; EPO	OR	ON	2006/08/06 18:20
L3	38	sql with tree with translat\$4 with (query statement)	US-PGPUB; USPAT; EPO	OR	ON	2006/08/06 18:57
L4	3	sql with tree with translat\$4 with (query statement) same (syntax metadata format)	US-PGPUB; USPAT; EPO	OR	ON	2006/08/06 18:21
L5	35	sql with tree with translat\$4 with (query statement) not 4	US-PGPUB; USPAT; EPO	OR	ON	2006/08/06 18:59
L6	3	sql with tree with translat\$4 with (query statement) and antlr	US-PGPUB; USPAT; EPO	OR	ON	2006/08/06 18:59

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2001

Application or Docket Number

~~34335600054~~
10 303/06

CLAIMS AS FILED - PART I

(Column 1) (Column 2)

TOTAL CLAIMS	62	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	62 minus 20 =	* 42
INDEPENDENT CLAIMS	7 minus 3 =	* 4
MULTIPLE DEPENDENT CLAIM PRESENT <input type="checkbox"/>		

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE OR

OTHER THAN SMALL ENTITY

RATE	FEE	OR	RATE	FEE
BASIC FEE	370.00	OR	BASIC FEE	740.00
X\$ 9=		OR	X\$18=	756
X42=		OR	X84=	336
+140=		OR	+280=	
TOTAL		OR	TOTAL	1832

CLAIMS AS AMENDED - PART II

(Column 1) (Column 2) (Column 3)

5-12-06

AMENDMENT A	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	* (62)	Minus ** 62	= /
Independent	* 7	Minus *** 7		= /
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <input type="checkbox"/>				

SMALL ENTITY OR

OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

(Column 1) (Column 2) (Column 3)

AMENDMENT B	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus **	=
Independent	*	Minus ***		=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <input type="checkbox"/>				

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

(Column 1) (Column 2) (Column 3)

AMENDMENT C	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA
	Total	*	Minus **	=
Independent	*	Minus ***		=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <input type="checkbox"/>				

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

- * If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
 - ** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
 - *** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
- The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.



JPW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

INVENTOR(S) : F. Levine
TITLE : Computer-Implemented System And Method For
Handling Database Statements
SERIAL NO. : 10/303,106
FILING DATE : 11/22/2002
GROUP ART UNIT : 2164
EXAMINER : J. Betit
ATTORNEY DOCKET NO. : 343355600054

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

CERTIFICATE OF MAILING

*I hereby certify that this correspondence is being deposited today with the United States
Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O.
Box 1450, Alexandria, VA 22313-1450
on May 10, 2006*

By: Kathie J. Koczynski

RESPONSIVE AMENDMENT

Please consider the following amendments and remarks.

CLAIMS

1. (CURRENTLY AMENDED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format; and

generating a second fourth-generation language database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

2. (CURRENTLY AMENDED) The method of claim 1 wherein the database statement functional difference specifies at least a portion of a statement format that is compatible with the second database system's query language format and that is incompatible with the first database system's query language format

wherein the tree contains a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be used in generating the second database statement.

3. (CURRENTLY AMENDED) The method of claim 1 wherein object-oriented techniques are used to access the database functional language difference data.
4. (ORIGINAL) The method of claim 3 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.
5. (ORIGINAL) The method of claim 4 wherein a logical piece is a phrase logical piece.
6. (ORIGINAL) The method of claim 4 wherein a logical piece is an identifier logical piece.
7. (ORIGINAL) The method of claim 4 wherein an SQL component object defaults to a default native SQL textualization method for use in generating the second database statement.
8. (CURRENTLY AMENDED) The method of claim 7 wherein an SQL component object comprises an override to account for functional differences between the first and second database systems' query language formats.
9. (CURRENTLY AMENDED) The method of claim 8 wherein SQL component objects comprise a phrase component object~~means~~.

10. (CURRENTLY AMENDED) The method of claim 8 wherein SQL component objects comprise an identifier component object ~~means~~.
11. (CURRENTLY AMENDED) The method of claim 8 wherein SQL component objects comprise an expression component object ~~means~~.
12. (CURRENTLY AMENDED) The method of claim 8 wherein SQL component objects comprise a parent component object ~~means~~.
13. (ORIGINAL) The method of claim 1 further comprising the steps of:
 - identifying, for the first database statement, query language parts that are common between the first database system's query language format and the second database system's query language format; and
 - generating the second database statement based upon the identified common query language parts.
14. (ORIGINAL) The method of claim 13 wherein the language parts are common based upon a predetermined standardized query language format.
15. (ORIGINAL) The method of claim 14 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.
16. (CURRENTLY AMENDED) The method of claim 1 wherein the database functional language difference data facilitates the generation of the second database statement by specifying common language parts between the first and second database system's language formats.

17. (ORIGINAL) The method of claim 16 wherein the language parts are common based upon a predetermined standardized query language format.
18. (ORIGINAL) The method of claim 17 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.
19. (ORIGINAL) The method of claim 1 wherein the second database system is a different type of database system than the first database system.
20. (ORIGINAL) The method of claim 1 wherein generating the second database statement provides the ability to manipulate data within the second database system from the first database system.
21. (ORIGINAL) The method of claim 1 wherein generating automatically the second database statement provides the ability to transparently manipulate data within the second database system from the first database system.
22. (ORIGINAL) The method of claim 1 wherein the generated second database statement is provided to the second database system for execution by the second database system.
23. (ORIGINAL) The method of claim 1 wherein the generated second database statement is in a format such that the second database statement is directly executable by the second database system.

24. (ORIGINAL) The method of claim 1 wherein the second database statement is a functional equivalent of the first database statement but for differences between the first and second database systems' query language formats.

25. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language formats are based upon a predetermined standardized query language format.

26. (CURRENTLY AMENDED) The method of claim 1 wherein the first and second database systems' query language formats are based upon a standardized fourth-generation structured query language (SQL) version.

27. (ORIGINAL) The method of claim 26 wherein the first database system's query language format utilizes a superset of the SQL standard.

28. (ORIGINAL) The method of claim 26 wherein the second database system's query language format utilizes a superset of the SQL standard.

29. (CURRENTLY AMENDED) The method of claim 1 wherein the first and second database systems' query language formats specify different formats for a preselected query-related function, wherein the first database statement is formatted in the first database system's query language format to perform the query-related function, wherein the second database statement is generated based upon the database functional difference data so as to be formatted in the second database system's query language format, wherein the generated second database statement is executable

within the second database system so as to perform the query-related function within the second database system.

30. (CURRENTLY AMENDED) The method of claim 1 wherein the tree is an SQL tree that is used to generate the second database statement, wherein the SQL tree contains data that represents the syntax of the first database statement.

31. (ORIGINAL) The method of claim 30 wherein the SQL tree contains metadata related to the first database statement.

32. (ORIGINAL) The method of claim 31 wherein the first database statement is parsed into logical text pieces which are stored in the SQL tree.

33. (ORIGINAL) The method of claim 32 wherein the second database statement takes into account any second database system-specific query language syntax.

34. (CURRENTLY AMENDED) The method of claim 1 further comprising the step of:

generating a third database statement for use within a third database system, wherein the third database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the third database statement is compatible with the third database system's query language format.

35. (CURRENTLY AMENDED) The method of claim 34 further comprising the step of:

generating a fourth database statement for use within a fourth database system, wherein the fourth database statement is generated based upon the first database statement and

upon the accessed database functional language difference data, wherein the fourth database statement is compatible with the fourth database system's query language format.

36. (ORIGINAL) The method of claim 1 wherein the first and second database systems are relational database management systems.

37. (ORIGINAL) The method of claim 1 wherein the first database system comprises a data mining application.

38. (ORIGINAL) The method of claim 37 wherein the second database system comprises a relational database management system.

39. (ORIGINAL) The method of claim 1 wherein the first database system comprises a relational database management system.

40. (ORIGINAL) The method of claim 39 wherein the second database system comprises a data mining application.

41. (ORIGINAL) The method of claim 1 wherein the first database system comprises an enterprise resource planning system.

42. (ORIGINAL) The method of claim 41 wherein the second database system comprises an enterprise resource planning system.

43. (ORIGINAL) The method of claim 1 wherein the first and second database systems' query language format includes format specifications for insert, select, update, and delete database commands.

44. (ORIGINAL) Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 1.

45. (CURRENTLY AMENDED) A computer-implemented ~~apparatus-system~~ for handling a first database fourth-generation language query that is formatted in a first query format and is executable by a first database system, comprising:

a data store to store tree-structured data that is representative of syntax and metadata of the first database fourth-generation language query;

a data structure for storing query specific data that indicates at least one query functional language difference from the first query format, wherein the query functional language difference is a query syntax difference; and

a textualization module having a data access connection to the tree-structured data and the data structure, wherein the textualization module generates a database specific query based upon the tree-structured data and the query specific data, wherein the database specific query accounts for the difference from the first query format so that the database, specific query may be executed by a different type of database system.

46. (CURRENTLY AMENDED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first fourth-generation language database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and

generating a second fourth-generation language database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

47. (ORIGINAL) The method of claim 46 wherein the language parts are determined to be common based upon a predetermined standardized query language format.

48. (ORIGINAL) The method of claim 47 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.

49. (CURRENTLY AMENDED) A computer-implemented ~~apparatus~~ system for handling a database statement from a first database system, comprising:

means for receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and

means for generating a second database fourth-generation language statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

50. (CURRENTLY AMENDED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call

textualization specific data, wherein the generated program call is in a format that is compatible with the second database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call.

51. (ORIGINAL) The method of claim 50 wherein the program call textualization specific data is used to generate application program interface (API) calls from the first database statement.

52. (ORIGINAL) The method of claim 50 wherein object-oriented techniques are used to access the program call textualization specific data.

53. (ORIGINAL) The method of claim 52 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.

54. (ORIGINAL) The method of claim 53 wherein a logical piece is a phrase logical piece.

55. (ORIGINAL) The method of claim 53 wherein a logical piece is an identifier logical piece.

56. (ORIGINAL) The method of claim 50 wherein the first database system's query language format is based upon a standardized structured query language (SQL) version.

57. (CURRENTLY AMENDED) The method of claim 50 further comprising the steps of:

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format; and

generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the second database system's query language format.

58. (CURRENTLY AMENDED) The method of claim 50 further comprising the steps of:

accessing database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format; and

generating a second database statement for use within a third database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional language difference data, wherein the second database statement is compatible with the third database system's query language format.

59. (ORIGINAL) Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 50.

60. (CURRENTLY AMENDED) A computer-implemented ~~apparatus~~ system for handling a database statement from a first database system, comprising the steps of:

means for receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

means for generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the

program call textualization specific data, wherein the generated program call is in a format that is compatible with the second database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the program call for use within the second database system.

61. (CURRENTLY AMENDED) A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database fourth-generation language statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

parsing the first database statement to obtain first query metadata of the first database statement;

using the obtained first query metadata to access database functional language difference data, wherein the database functional language difference data indicates a format that contains at least one database functional statement difference from the first database system's query language format;

generating a second database fourth-generation language statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database functional

language difference data, wherein the second database statement is compatible with the second database system's query language format;

receiving another database statement from the first database system, wherein the additional database statement from the first database system is formatted according to the first database system's query language format;

parsing the additional database statement to obtain second query metadata of the additional database statement;

using the second query metadata to generate a program call to a third database system which utilizes a different query language format than the first database system; and

issuing the program call to the third database system to access data contained in the third database system;

wherein a tree representative of the syntax of the database language used within the first database system and of metadata associated with the first database system is used in generating the second database statement.

62. (CURRENTLY AMENDED) The method of claim 61 further comprising the steps of:

using object-oriented component means for generating database access instructions to the second database system;

determining whether to use wherein an override exists, wherein the that override indicates for the object-oriented component means to use the first query metadata to generate a program call to the second database system; and

issuing the program call to the second database system to access data contained in the second database system,

wherein if the override did not exist, then the object-oriented component means textualizes a database statement based upon the first query metadata and which is executable within the second database system.

REMARKS

Claims 1-62 remain pending in the application. Claims 1, 45, 46, 49, 50, 60 and 61 are independent claims. Claim 62 is objected to. Claims 1-62 stand rejected by the examiner. Assignee traverses the instant claim rejections and objections.

Examiner's Interview

Assignee's representative would like to thank Examiner Betit and his Supervisor for the courtesies extended to assignee's representatives, Timothy Wilson, Gary Kuhn, and John Biernacki, during the telephone interview on May 9, 2006. The interview discussed the cited reference Bodamer et al. (USPN 6,041,344) in view of claims 1, 30 and 45. More specifically, the interview discussed Bodamer in reference to the office action's statements with respect to the "generating a second database statement..." step of claim 1. This step in claim 1 is part of a method for translating a first fourth-generation language database statement to a second fourth-generation language database statement. The interview also discussed Bodamer in view of claim 30's and claim 45's recitation of an SQL tree to generate the second database statement. The remarks and the amendments contained herein summarize the interview.

Claim Objections

Claim 62 was objected to under 37 CFR 1.75(c) as being of improper dependent form for failing to further limit the subject matter of a previous claim. Assignee respectfully disagrees with the objection, but has amended claim 62 to expressly recite that there is a determination as to whether to use an override. Accordingly claim 62 (being dependent upon claim 61) requires that a database statement be processed in

accordance with the steps of claim 61 as well as allowing for an override mechanism to be used. Because dependent claim 62 recites further limitations upon claim 61, assignee respectfully requests that the objection for claim 62 be removed and this application proceed to issuance.

Claim Rejections – 35 USC § 112

Claims 9-12 and 45, 49, and 60 stand rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Assignee respectfully traverses this rejection. While assignee disagrees with the rejections, claims 9-12 have been amended to remove the term “means” and claims 45, 49, and 60 have been amended to recite the term “system” instead of “apparatus.” Favorable reconsideration is respectfully requested.

Claim Rejections – 35 USC §§ 102 and 103

Claims 1-36, 38-39, and 43-62 stand rejection under 35 USC § 102(b) as being anticipated by Bodamer et al. (U.S. Patent No. 6,041,344). Claims 37 and 40-42 stand rejection under 35 USC § 103(a) as being unpatentable over Bodamer in view of the examiner’s official notice. Assignee traverses the instant rejections.

Claim 1 is directed to a computer-implemented method for handling a first database statement from a first database system. The database statement is a fourth-generation language database statement formatted according to a language format used by the first database system. Database language difference data is accessed so that a

second fourth-generation language database statement may be generated which is operational within a different type of database system. As part of the process in converting the first fourth-generation language database statement to the second fourth-generation language database statement, a tree (that is representative of the syntax of the database language used within the first database system and of the metadata associated with the first database system) is used in generating the second fourth-generation language database statement.

As a non-limiting example of use of a tree in generating the second statement, assignee's specification (on page 5, lines 3-9) provides an example where an SQL tree is used to process an SQL statement. The SQL tree represents the syntax of a native database's SQL statement and its related metadata. The tree may contain a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be processed. If for example the SQL statement specified that the values from two different columns are to be concatenated, then the SQL tree would contain a node that specifies that a concatenation operation is to be performed.

The Bodamer reference does not disclose the limitations of claim 1. For example, Bodamer lacks details in its disclosure regarding database statement translations, let alone providing any disclosure of claim 1's use of a tree that is representative of the syntax of the database language used within the first database system and of metadata associated with the first database system in generating a second database statement. Instead Bodamer appears to devote most of its disclosure to translations other than the database statement conversion that is the subject matter of claim 1, such as to a data dictionary translation. To the extent that Bodamer might disclose any type of tree-like structure in

this citation, it is with respect to a different type of translation (e.g., the data dictionary translation) than the type of translation that is being performed in claim 1.

Additionally, the office action uses an excerpt from Bodamer to anticipate the “generating...” step of claim 1 that is unrelated to a database statement conversion and instead is related to a different type of translation (i.e., the fourth type of translation discussed in Bodamer, namely the data dictionary translation). This fourth type of translation is directed to handling database schema differences that might occur between two databases. This is different than the database statement translation that is being performed in claim 1. Claim 1 looks at functional database language differences in order to generate a second database functional database statement. In contrast, the data dictionary translation of Bodamer examines two different databases’ schemata to determine schemata differences – that is, a data dictionary translation is performed in Bodamer because a “foreign database system 208, however, may include similar metadata that is organized differently”; see Bodamer at column 8, lines 21-23.). As recognized by Bodamer itself (in establishing four categories of translations; see Bodamer at column 7, lines 18-20), performing database functional statement translation is different than performing data dictionary schemata translations. Accordingly this step of claim 1 cannot be anticipated by Bodamer and thus claim 1 is allowable.

Claim 1 also recites the use of database functional language difference data in order to generate the second SQL database statement. Claim 1’s use of functional language differences to generate the second SQL database statement is advantageous, such as if there are only a few functional language differences between the first and second SQL environments, then only a few functional language differences have to be

specified in order for the translation to occur. It is noted that Bodamer is lacking in any disclosure regarding the use of SQL language functional differences being specified and used to translate from a first database statement into a second database statement.

Furthermore, claim 1 recites that a first 4GL database statement is translated into a second 4GL database statement. As an illustration of why translation from one 4GL statement to a second 4GL statement can be useful, consider the following: while a first database system and a second database system may both support a 4GL SQL function that performs an identical operation, the two systems may use different 4GL SQL function names and/or formats.

Assignee respectfully asserts that Bodamer does not disclose translating a first 4GL statement into a second 4GL statement as required by claim 1. Bodamer does not translate from a first 4GL statement to a second 4GL statement. At best Bodamer translates from an SQL statement into a non-4GL SQL statement, such as a generic function:

The translated SQL statement in foreign database system format can then be passed as an argument of a generic function (e.g., parse). For example, an SQL statement translated by the SQL services module 210b can be an argument for the Oracle-specific call "opiosq," which is then mapped onto the generic API 212 as "parse."
(see Bodamer at column 7, lines 57-62)

Because claim 1 recites that a first 4GL database statement is translated into a second 4GL database statement and Bodamer at best only discloses a translation into a non-4GL SQL statement, Bodamer is significantly different and thus does not anticipate claim 1.

The other independent claims recite the use of a tree in generating database statements. As shown by the above, Bodamer does not disclose such features. Because

of such lack of disclosure, Bodamer cannot anticipate these claims, and they should proceed to issuance.

Assignee disagrees with other positions of the office action. For example, Bodamer does not disclose the details of claim 2 which recites that the tree contains a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be used in generating the second database statement. Because of such lack of disclosure, Bodamer cannot anticipate claim 2, and claim 2 should proceed to issuance.

As another example claim 26 provides that a first 4GL SQL database statement is translated into a second 4GL SQL database statement. Bodamer does not disclose such translations and thus cannot anticipate claim 26. Accordingly assignee respectfully requests that claim 26 should proceed to issuance.

CONCLUSION

For the foregoing reasons, Assignee respectfully submits that claims 1-62 are allowable. Therefore, the examiner is respectfully requested to pass this case to issue.

Respectfully submitted,

Date: May 10, 2006

By: John V. Biernacki

John V. Biernacki
Reg. No. 40,511
JONES DAY
North Point
901 Lakeside Avenue
Cleveland, Ohio 44114
(216) 586-3939



UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/303,106	11/22/2002	Frederick J. Levine	343355600054	2037

7590 05/11/2006

John V. Biernacki
Jones, Day, Reavis & Pogue
North Point
901 Lakeside Avenue
Cleveland, OH 44114

EXAMINER

BETTIT, JACOB F

ART UNIT	PAPER NUMBER
2164	

2164

DATE MAILED: 05/11/2006

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

Interview Summary	Application No. 10/303,106	Applicant(s) LEVINE, FREDERICK J.	
	Examiner Jacob F. Betit	Art Unit 2164	

All participants (applicant, applicant's representative, PTO personnel):

- (1) Jacob F. Betit. (3) John V. Bernacki.
(2) Sam Rimell. (4) _____.

Date of Interview: 09 May 2006.

Type: a) Telephonic b) Video Conference
c) Personal [copy given to: 1) applicant 2) applicant's representative]

Exhibit shown or demonstration conducted: d) Yes e) No.
If Yes, brief description: _____.

Claim(s) discussed: 1 and 45.

Identification of prior art discussed: Bodamer et al..

Agreement with respect to the claims f) was reached. g) was not reached. h) N/A.

Substance of Interview including description of the general nature of what was agreed to if an agreement was reached, or any other comments: The examiners suggested several amendments to overcome the currently recited reference including specifying 4GL SQL statement in the claim and further defining the structure of the difference data i.e. how it is organized and used (tree structure).

(A fuller description, if necessary, and a copy of the amendments which the examiner agreed would render the claims allowable, if available, must be attached. Also, where no copy of the amendments that would render the claims allowable is available, a summary thereof must be attached.)

THE FORMAL WRITTEN REPLY TO THE LAST OFFICE ACTION MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a reply to the last Office action has already been filed, APPLICANT IS GIVEN A NON-EXTENDABLE PERIOD OF THE LONGER OF ONE MONTH OR THIRTY DAYS FROM THIS INTERVIEW DATE, OR THE MAILING DATE OF THIS INTERVIEW SUMMARY FORM, WHICHEVER IS LATER, TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW. See Summary of Record of Interview requirements on reverse side or on attached sheet.

Examiner Note: You must sign this form unless it is an Attachment to a signed Office action.



Examiner's signature, if required



UNITED STATES PATENT AND TRADEMARK OFFICE

19

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/303,106	11/22/2002	Frederick J. Levine	343355600054	2037

7590 02/10/2006
 John V. Biernacki
 Jones, Day, Reavis & Pogue
 North Point
 901 Lakeside Avenue
 Cleveland, OH 44114

EXAMINER

BETIT, JACOB F

ART UNIT	PAPER NUMBER
2164	

2164

DATE MAILED: 02/10/2006

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

Office Action Summary

Application No. 10/303,106	Applicant(s) LEVINE, FREDERICK J.	
Examiner Jacob F. Betit	Art Unit 2164	

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

Period for Reply

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

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

Status

- 1) Responsive to communication(s) filed on _____.
- 2a) This action is **FINAL**.
- 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-62 is/are pending in the application.
4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-62 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

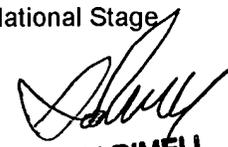
Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

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


SAM RIMELL
PRIMARY EXAMINER

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____
- 5) Notice of Informal Patent Application (PTO-152)
- 6) Other: _____

DETAILED ACTION

Claim Objections

1. Claim 62 is objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim; or amend the claim to place the claim in proper dependent form, or rewrite the claim in independent form. Claims 62 recites limitations that act as though to broaden the subject matter of claim 61, by removing steps that were not optionally recited (i.e. "generating a second database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database language difference data"). A claim in the dependent form must specify a further limitation of the subject matter of the subject matter claimed. See MPEP § 608.01(n), and see 35 U.S.C. § 112 fourth paragraph.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 9-12 and 45, 49, and 60 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

4. Regarding claim 9, the word "means" is preceded by the words "phrase component object" in an attempt to use a "means" clause to recite a claim element as a means for performing a specified function. However, since no function is specified by the words preceding "means," it

Art Unit: 2164

is impossible to determine the equivalents of the element, as required by 35 U.S.C. 112, sixth paragraph. See *Ex parte Klumb*, 159 USPQ 694 (Bd. App. 1967).

Regarding claim 10, the word "means" is preceded by the words "identifier component object" in an attempt to use a "means" clause to recite a claim element as a means for performing a specified function. However, since no function is specified by the words preceding "means," it is impossible to determine the equivalents of the element, as required by 35 U.S.C. 112, sixth paragraph. See *Ex parte Klumb*, 159 USPQ 694 (Bd. App. 1967).

Regarding claim 11, the word "means" is preceded by the words "expression component object" in an attempt to use a "means" clause to recite a claim element as a means for performing a specified function. However, since no function is specified by the words preceding "means," it is impossible to determine the equivalents of the element, as required by 35 U.S.C. 112, sixth paragraph. See *Ex parte Klumb*, 159 USPQ 694 (Bd. App. 1967).

Regarding claim 12, the word "means" is preceded by the words "parent component object" in an attempt to use a "means" clause to recite a claim element as a means for performing a specified function. However, since no function is specified by the words preceding "means," it is impossible to determine the equivalents of the element, as required by 35 U.S.C. 112, sixth paragraph. See *Ex parte Klumb*, 159 USPQ 694 (Bd. App. 1967).

5. Claims 45, 49, and 60 recite the limitation "[a] computer-implemented apparatus" in their preambles. It is not clear how the "apparatus" differs from the computer, and if it differs from the computer it is not clear how one would differentiate between the computer and the apparatus. It is also not clear how the apparatus would be implemented into the computer and not become a functional part of it.

Art Unit: 2164

6. Appropriate corrections are required.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1-36, 38-39, and 43-62 are rejected under 35 U.S.C. 102(b) as being anticipated by Bodamer et al. (U.S. patent No. 6,041,344).

As to claim 1, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

accessing database language difference data, wherein the database language difference data indicates a format that contains at least one database statement difference from the first database system's query language format (see column 7, line 18 through column 8, line 37); and

generating a second database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the second database statement is compatible with the second database system's query language format (see column 8, line 38-67).

Art Unit: 2164

As to claim 2, Bodamer et al. teaches wherein the database statement difference specifies at least a portion of a statement format that is compatible with the second database system's query language format and that is incompatible with the first database system's query language format (see column 7, line 18 through column 8, line 37).

As to claim 3, Bodamer et al. teaches wherein object-oriented techniques are used to access the database language difference data (see column 5, lines 7-54).

As to claim 4, Bodamer et al. teaches wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement (see column 7, lines 43-67).

As to claim 5, Bodamer et al. teaches wherein a logical piece is a phrase logical piece (see column 8, lines 47-67).

As to claim 6, Bodamer et al. teaches wherein a logical piece is an identifier logical piece (see column 8, lines 47-67).

As to claim 7, Bodamer et al. teaches wherein an SQL component object defaults to a default native SQL textualization method for use in generating the second database statement (see column 7, lines 43-67).

Art Unit: 2164

As to claim 8, Bodamer et al. teaches wherein an SQL component object comprises an override to account for differences between the first and second database systems' query language formats (see column 7, lines 43-67)

As to claim 9, Bodamer et al. teaches wherein SQL component objects comprise phrase component object means (see column 8, lines 47-67).

As to claim 10, Bodamer et al. teaches wherein SQL component objects comprise identifier component object means (see column 8, lines 47-67).

As to claim 11, Bodamer et al. teaches wherein SQL component objects comprise expression component object means (see column 8, lines 47-67).

As to claim 12, Bodamer et al. teaches wherein SQL component objects comprise parent component object means (see column 8, lines 47-67).

As to claim 13, Bodamer et al. teaches further comprising the steps of:
identifying, for the first database statement, query language parts that are common between the first database system's query language format and the second database system's query language format; and generating the second database statement based upon the identified common query language parts (see column 7, lines 43-67).

Art Unit: 2164

As to claim 14, Bodamer et al. teaches wherein the language parts are common based upon a predetermined standardized query language format (see column 7, lines 43-67).

As to claim 15, Bodamer et al. teaches wherein the standardized query language format is based upon a standardized structured query language (SQL) version (see column 7, lines 43-67).

As to claim 16, Bodamer et al. teaches wherein the database language difference data facilitates the generation of the second database statement by specifying common language parts between the first and second database system's language formats (see column 7, lines 43-67).

As to claim 17, Bodamer et al. teaches wherein the language parts are common based upon a predetermined standardized query language format (see column 7, lines 18-67).

As to claim 18, Bodamer et al. teaches wherein the standardized query language format is based upon a standardized structured query language (SQL) version (see column 7, lines 18-67).

As to claim 19, Bodamer et al. teaches wherein the second database system is a different type of database system than the first database system (see column 7, lines 18-67).

As to claim 20, Bodamer et al. teaches wherein generating the second database statement provides the ability to manipulate data within the second database system from the first database system (see column 4, line 40 through column 5, line 20).

As to claim 21, Bodamer et al. teaches wherein generating automatically the second database statement provides the ability to transparently manipulate data within the second database system from the first database system (see column 8, lines 38-46).

As to claim 22, Bodamer et al. teaches wherein the generated second database statement is provided to the second database system for execution by the second database system (see column 8, lines 38-46).

As to claim 23, Bodamer et al. teaches wherein the generated second database statement is in a format such that the second database statement is directly executable by the second database system (see column 8, lines 47-67).

As to claim 24, Bodamer et al. teaches wherein the second database statement is a functional equivalent of the first database statement but for differences between the first and second database systems' query language formats (see column 7, lines 9-17).

As to claim 25, Bodamer et al. teaches wherein the first and second database systems' query language formats are based upon a predetermined standardized query language format (see column 7, lines 43-67).

Art Unit: 2164

As to claim 26, Bodamer et al. teaches wherein the first and second database systems' query language formats are based upon a standardized structured query language (SQL) version (see column 7, lines 43-67).

As to claim 27, Bodamer et al. teaches wherein the first database system's query language format utilizes a superset of the SQL standard (see column 7, lines 25-30).

As to claim 28, Bodamer et al. teaches wherein the second database system's query language format utilizes a superset of the SQL standard (see column 7, lines 30-35).

As to claim 29, Bodamer et al. teaches wherein the first and second database systems' query language formats specify different formats for a preselected query-related function, wherein the first database statement is formatted in the first database system's query language format to perform the query-related function, wherein the second database statement is generated based upon the database difference data so as to be formatted in the second database system's query language format, wherein the generated second database statement is executable within the second database system so as to perform the query-related function within the second database system (see column 7, line 18 through column 8, line 37).

As to claim 30, Bodamer et al. teaches wherein an SQL tree is used to generate the second database statement, wherein the SQL tree contains data that represents the syntax of the first database statement (see column 7, line 18 through column 8, line 37).

As to claim 31, Bodamer et al. teaches wherein the SQL tree contains metadata related to the first database statement (see column 7, line 18 through column 8, line 37).

As to claim 32, Bodamer et al. teaches wherein the first database statement is parsed into logical text pieces which are stored in the SQL tree (see column 7, lines 43-67).

As to claim 33, Bodamer et al. teaches wherein the second database statement takes into account any second database system-specific query language syntax (see column 8, lines 38-67).

As to claim 34, Bodamer et al. teaches further comprising the step of:
generating a third database statement for use within a third database system, wherein the third database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the third database statement is compatible with the third database system's query language format (see column 4, line 59 through column 5, line 20 and see column 7, line 18 through column 8, line 37).

As to claim 35, Bodamer et al. teaches further comprising the step of:
generating a fourth database statement for use within a fourth database system, wherein the fourth database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the fourth database statement is compatible

Art Unit: 2164

with the fourth database system's query language format (see column 4, line 59 through column 5, line 20 and see column 7, line 18 through column 8, line 37).

As to claim 36, Bodamer et al. teaches wherein the first and second database systems are relational database management systems (see column 1, lines 44-52).

As to claim 39, Bodamer et al. teaches wherein the first database system comprises a relational database management system (see column 1, lines 44-52).

As to claim 38, Bodamer et al. teaches wherein the second database system comprises a relational database management system (see column 8, lines 10-46).

As to claim 43, Bodamer et al. teaches wherein the first and second database systems' query language format includes format specifications for insert, select, update, and delete database commands (see column 16, line 64 through column 17, line 3).

As to claim 44, Bodamer et al. teaches computer software stored on a computer readable media, the computer software comprising program code (see column 3, line 65 through column 4, line 38) for carrying out a method according to claim 1 (for the rejection of the limitations of claim 1, the applicant is directed to the rejection of claim 1 above).

Art Unit: 2164

As to claim 45, Bodamer et al. teaches a computer-implemented apparatus for handling a first database query that is formatted in a first query format and is executable by a first database system, comprising:

- tree-structured data that is representative of syntax and metadata of the first database query (see column 4, lines 40-58);

- a data structure for storing query specific data that indicates at least one query language difference from the first query format, wherein the query language difference is a query syntax difference (see column 7, line 18 through column 8, line 37); and

- a textualization module having a data access connection to the tree-structured data and the data structure, wherein the textualization module generates a database specific query based upon the tree-structured data and the query specific data, wherein the database specific query accounts for the difference from the first query format so that the database specific query may be executed by a different type of database system (see column 8, lines 38-67).

As to claim 46, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

- receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

- identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format (see column 7, line 18 through column 8, line 37); and

Art Unit: 2164

generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format (see column 8, lines 38-67).

As to claim 47 and 48, see the rejection of claim 14 and 15 respectively.

As to claim 49, Bodamer et al. teaches a computer-implemented apparatus for handling a database statement from a first database system, comprising:

means for receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

means for identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format (see column 7, line 18 through column 8, line 37); and

means for generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format (see column 8, lines 38-67).

As to claim 50, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

Art Unit: 2164

receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system (see column 7, line 18 through column 8, line 37); and

generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call textualization specific data, wherein the generated program call is in a format that is compatible with the second database system (see column 8, lines 38-67).

As to claim 51, Bodamer et al. teaches wherein the program call textualization specific data is used to generate application program interface (API) calls from the first database statement (see column 7, lines 18-35).

As to claims 52, 53, 54, and 55; see the rejections of claims 3, 4, 5, and 6 respectively.

As to claim 56, Bodamer et al. teaches wherein the first database system's query language format is based upon a standardized structured query language (SQL) version (see column 4, lines 40-58).

As to claim 57, Bodamer et al. teaches further comprising the steps of:

Art Unit: 2164

accessing database language difference data, wherein the database language difference data indicates a format that contains at least one database statement difference from the first database system's query language format (see column 5, line 63 through column 6, line 15); and generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the second database statement is compatible with the second database system's query language format (see column 8, lines 38-67).

As to claim 58, Bodamer et al. teaches further comprising the steps of:

accessing database language difference data, wherein the database language difference data indicates a format that contains at least one database statement difference from the first database system's query language format (see column 5, line 63 through column 6, line 15); and generating a second database statement for use within a third database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the second database statement is compatible with the third database system's query language format (see column 8, lines 38-67 and see figure 2A, reference number 300).

As to claim 59, see the rejection of claim 44 above.

As to claim 60, Bodamer et al. teaches a computer-implemented apparatus for handling a database statement from a first database system, comprising the steps of:

Art Unit: 2164

means for receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

means for accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system (see column 7, line 18 through column 8, line 37); and

means for generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call textualization specific data, wherein the generated program call is in a format that is compatible with the second database system (see column 8, lines 38-67).

As to claim 61, Bodamer et al. teaches a computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format (see column 4, lines 40-58);

parsing the first database statement to obtain first query metadata of the first database statement; using the obtained first query metadata to access database language difference data, wherein the database language difference data indicates a format that contains at least one database statement difference from the first database system's query language format (see column 7, line 18 through column 8, line 37);

generating a second database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the second database statement is compatible with the second database system's query language format (see column 8, lines 38-67);

receiving another database statement from the first database system, wherein the additional database statement from the first database system is formatted according to the first database system's query language format (see figure 2A, reference number 300 and see column 4, lines 40-58);

parsing the additional database statement to obtain second query metadata of the additional database statement; using the second query metadata to generate a program call to a third database system which utilizes a different query language format than the first database system (see column 7, line 18 through column 8, line 37); and

issuing the program call to the third database system to access data contained in the third database system (see column 8, lines 38-67).

As to claim 62, Bodamer et al. teaches further comprising the steps of:

using object-oriented component means for generating database access instructions to the second database system (see column 5, lines 7-20);

wherein an override exists that indicates for the object-oriented component means to use the first query metadata to generate a program call to the second database system (see column 5, line 63 through column 6, line 15); and

Art Unit: 2164

issuing the program call to the second database system to access data contained in the second database system (see column 8, lines 38-67),

wherein if the override did not exist, then the object-oriented component means textualizes a database statement based upon the first query metadata and which is executable within the second database system (this limitation is optionally recited and does not properly depend from 61 because it acts as though to exclude limitations of that claim).

Claim Rejections - 35 USC § 103

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

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

10. Claims 37, 40-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bodamer et al. (U.S. patent No. 6,041,344) in view of the examiner's official notice.

As to claim 37, Bodamer et al. does not teach wherein the first database system comprises a data mining application.

The examiner takes official notice that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because many databases include data mining applications so that they can acquire new data and so they can more easily search the data they currently contain.

Art Unit: 2164

As to claim 40, Bodamer et al. does not teach wherein the second database system comprises a data mining application.

The examiner takes official notice that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because many databases include data mining applications so that they can acquire new data and so they can more easily search the data they currently contain.

As to claim 41, Bodamer et al. does not teach wherein the first database system comprises an enterprise resource planning system.

The examiner takes official notice that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because this would allow a company to plan appropriate resources for different projects being performed within a company which is a common use for a database in an corporate environment.

As to claim 42, Bodamer et al. does not teach wherein the second database system comprises an enterprise resource planning system.

The examiner takes official notice that it would have been obvious for one of ordinary skill in the art at the time the invention was made to have modified Bodamer et al. to include this because this would allow a company to plan appropriate resources for different projects being performed within a company which is a common use for a database in an corporate environment.

Art Unit: 2164

Conclusion

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jacob F. Betit whose telephone number is (571) 272-4075. The examiner can normally be reached on Monday through Friday 9:30 am to 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Charles Rones can be reached on (571) 272-4085. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

jfb
3 Feb 2006


JACOB F. BETIT
PRIMARY EXAMINER

Notice of References Cited	Application/Control No. 10/303,106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.	
	Examiner Jacob F. Betit	Art Unit 2164	Page 1 of 1

U.S. PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A US-6,041,344 A	03-2000	Bodamer et al.	709/203
B	US-			
C	US-			
D	US-			
E	US-			
F	US-			
G	US-			
H	US-			
I	US-			
J	US-			
K	US-			
L	US-			
M	US-			

FOREIGN PATENT DOCUMENTS

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

NON-PATENT DOCUMENTS

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

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

Index of Claims



Application/Control No.

10303106

Applicant(s)/Patent under Reexamination

LEVINE, FREDERICK J.

Examiner

Betit, Jacob F

Art Unit

2164

✓	Rejected
=	Allowed

-	(Through numeral) Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

CLAIMS		DATE									
Final	Original	2/8/96									
	1	✓									
	2	✓									
	3	✓									
	4	✓									
	5	✓									
	6	✓									
	7	✓									
	8	✓									
	9	✓									
	10	✓									
	11	✓									
	12	✓									
	13	✓									
	14	✓									
	15	✓									
	16	✓									
	17	✓									
	18	✓									
	19	✓									
	20	✓									
	21	✓									
	22	✓									
	23	✓									
	24	✓									
	25	✓									
	26	✓									
	27	✓									
	28	✓									
	29	✓									
	30	✓									
	31	✓									
	32	✓									
	33	✓									

Index of Claims



Application/Control No.

10303106

Applicant(s)/Patent under Reexamination

LEVINE, FREDERICK J.

Examiner

Betit, Jacob F

Art Unit

2164

✓	Rejected
=	Allowed

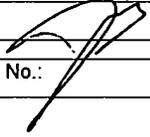
-	(Through numeral) Cancelled
÷	Restricted

N	Non-Elected
I	Interference

A	Appeal
O	Objected

CLAIMS		DATE									
Final	Original	2/8/99									
	34	✓									
	35	✓									
	36	✓									
	37	✓									
	38	✓									
	39	✓									
	40	✓									
	41	✓									
	42	✓									
	43	✓									
	44	✓									
	45	✓									
	46	✓									
	47	✓									
	48	✓									
	49	✓									
	50	✓									
	51	✓									
	52	✓									
	53	✓									
	54	✓									
	55	✓									
	56	✓									
	57	✓									
	58	✓									
	59	✓									
	60	✓									
	61	✓									
	62	✓									

Search Notes 	Application/Control No. 10303106	Applicant(s)/Patent Under Reexamination LEVINE, FREDERICK J.
	Examiner Betit, Jacob F	Art Unit 2164

Notes	Date	Examiner
searched EAST enclosed	2/6/06	JFB 
U.S. Patent and Trademark Office	Part of Paper No.:	

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	608	(sql) and (translat\$4 generat\$4 convert\$4) with query with (format type) not xml	US-PGPUB; USPAT; EPO	OR	ON	2006/01/26 14:35
S2	98	(sql) and (translat\$4 generat\$4 convert\$4) with query with (format type) not xml and Proprietary	US-PGPUB; USPAT; EPO	OR	ON	2006/01/26 14:36
S3	31	(sql) and (translat\$4 generat\$4 convert\$4) with query with (format type) not xml and Proprietary same standard	US-PGPUB; USPAT; EPO	OR	ON	2006/01/26 14:36
S4	15	(sql) and (translat\$4 generat\$4 convert\$4) with query with (format type) not xml and Proprietary same standard and "707"/S.ccls.	US-PGPUB; USPAT; EPO	OR	ON	2006/01/26 16:57
S6	34	(convert\$4 translat\$4) with native with sql	US-PGPUB; USPAT; EPO	OR	ON	2006/01/26 16:59
S7	33	(convert\$4 translat\$4) with native with sql not S4	US-PGPUB; USPAT; EPO	OR	ON	2006/01/27 11:15
S8	0	S7 and S3	US-PGPUB; USPAT; EPO	OR	ON	2006/01/26 16:59
S9	1	("6041344").PN.	US-PGPUB; USPAT; EPO	OR	OFF	2006/01/27 11:15

11-25-02

10303106 112202

A

11/22/02
JC979 U.S. PTO

PTO/SB/05 (03-01)
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.

UTILITY PATENT APPLICATION TRANSMITTAL

Attorney Docket No.	343355600054
First Inventor	Frederick J. Levine
Title	Computer-Implemented System and Method for Handling Database Statements
Express Mail Label No.	EL821237211US

(Only for new nonprovisional applications under 37 CFR 1.53(b))

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

- Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)
- Applicant claims small entity status.
See 37 CFR 1.27.
- Specification [Total Pages 34]
(preferred arrangement set forth below)
 - Descriptive title of the invention
 - Cross Reference to Related Applications
 - Statement Regarding Fed sponsored R & D
 - Reference to sequence listing, a table, or a computer program listing appendix
 - Background of the invention
 - Brief Summary of the Invention
 - Brief Description of the Drawings (if filed)
 - Detailed Description
 - Claim(s)
 - Abstract of the Disclosure
- Drawing(s) (35 U.S.C. 113) [Total Sheets 19]
- Oath or Declaration [Total Pages 2]
 - Newly executed (original or copy)
Copy from a prior application (37 CFR 1.63 (d))
 - (for continuation/divisional with Box 18 completed)
 - DELETION OF INVENTOR(S)**
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
- Application Data Sheet. See 37 CFR 1.76

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

- CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)
- Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
 - Computer Readable Form (CRF)
 - Specification Sequence Listing on:
 - CD-ROM or CD-R (2 copies); or
 - paper
 - Statements verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

- Assignment Papers (cover sheet & document(s))
- 37 CFR 3.73(b) Statement Power of Attorney (when there is an assignee)
- English Translation Document (if applicable)
- Information Disclosure Statement (IDS)/PTO-1449 Copies of IDS Citations
- Preliminary Amendment
- Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
- Certified Copy of Priority Document(s) (if foreign priority is claimed)
- Nonpublication Request under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
- Other:

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:

Continuation Divisional Continuation-in-part (CIP) of prior application No.: _____

Prior application information: Examiner: _____ Group Art Unit: _____

For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. CORRESPONDENCE ADDRESS

Customer Number or Bar Code Label Correspondence address below

Name	John V. Biernacki				
Address	Jones, Day, Reavis & Pogue North Point, 901 Lakeside Avenue				
City	Cleveland	State	OH	Zip Code	44114
Country	US	Telephone	216-586-3939	Fax	216-579-0212

Name (Print/Type)	John V. Biernacki	Registration No. (Attorney/Agent)	40,511
Signature	<i>John V. Biernacki</i>	Date	Nov. 22, 2002

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

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.

<h1 style="margin: 0;">FEE TRANSMITTAL</h1> <h2 style="margin: 0;">for FY 2003</h2> <p style="font-size: small; margin: 0;">Patent fees are subject to annual revision.</p>	<p>Complete if Known</p> <table border="1" style="width:100%; border-collapse: collapse;"> <tr><td>Application Number</td><td></td></tr> <tr><td>Filing Date</td><td></td></tr> <tr><td>First Named Inventor</td><td>Frederick J. Levine</td></tr> <tr><td>Examiner Name</td><td></td></tr> <tr><td>Art Unit</td><td></td></tr> <tr><td>Attorney Docket No.</td><td>343355600054</td></tr> </table>	Application Number		Filing Date		First Named Inventor	Frederick J. Levine	Examiner Name		Art Unit		Attorney Docket No.	343355600054
Application Number													
Filing Date													
First Named Inventor	Frederick J. Levine												
Examiner Name													
Art Unit													
Attorney Docket No.	343355600054												
<input type="checkbox"/> Applicant claims small entity status. See 37 CFR 1.27													
TOTAL AMOUNT OF PAYMENT	(\$) 1872.00												

<p>METHOD OF PAYMENT (check all that apply)</p> <p> <input type="checkbox"/> Check <input type="checkbox"/> Credit card <input type="checkbox"/> Money Order <input type="checkbox"/> Other <input type="checkbox"/> None </p> <p><input checked="" type="checkbox"/> Deposit Account:</p> <table style="width:100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Deposit Account Number</td> <td style="border: 1px solid black; padding: 2px;">501432</td> </tr> <tr> <td>Deposit Account Name</td> <td style="border: 1px solid black; padding: 2px;">Jones, Day, Reavis & Pogue</td> </tr> </table> <p>The Commissioner is authorized to: (check all that apply)</p> <p> <input checked="" type="checkbox"/> Charge fee(s) indicated below <input checked="" type="checkbox"/> Credit any overpayments </p> <p> <input type="checkbox"/> Charge any additional fee(s) during the pendency of this application </p> <p> <input type="checkbox"/> Charge fee(s) indicated below, except for the filing fee to the above-identified deposit account. </p> <p style="text-align: center;">FEE CALCULATION</p> <p>1. BASIC FILING FEE</p> <table style="width:100%; border-collapse: collapse;"> <thead> <tr> <th>Large Entity Fee Code (\$)</th> <th>Small Entity Fee Code (\$)</th> <th>Fee Description</th> <th>Fee Paid</th> </tr> </thead> <tbody> <tr><td>1001 740</td><td>2001 370</td><td>Utility filing fee</td><td style="border: 1px solid black; text-align: center;">740</td></tr> <tr><td>1002 330</td><td>2002 165</td><td>Design filing fee</td><td style="border: 1px solid black;"></td></tr> <tr><td>1003 510</td><td>2003 255</td><td>Plant filing fee</td><td style="border: 1px solid black;"></td></tr> <tr><td>1004 740</td><td>2004 370</td><td>Reissue filing fee</td><td style="border: 1px solid black;"></td></tr> <tr><td>1005 160</td><td>2005 80</td><td>Provisional filing fee</td><td style="border: 1px solid black;"></td></tr> <tr> <td colspan="3" style="text-align: right;">SUBTOTAL (1)</td> <td style="border: 1px solid black; text-align: center;">(\$) 740</td> </tr> </tbody> </table> <p>2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE</p> <table style="width:100%; border-collapse: collapse;"> <tr> <td>Total Claims</td> <td style="border: 1px solid black; text-align: center;">62</td> <td>-20** =</td> <td style="border: 1px solid black; text-align: center;">42</td> <td>x</td> <td style="border: 1px solid black; text-align: center;">18</td> <td>=</td> <td style="border: 1px solid black; text-align: center;">756</td> </tr> <tr> <td>Independent Claims</td> <td style="border: 1px solid black; text-align: center;">7</td> <td>-3** =</td> <td style="border: 1px solid black; text-align: center;">4</td> <td>x</td> <td style="border: 1px solid black; text-align: center;">84</td> <td>=</td> <td style="border: 1px solid black; text-align: center;">336</td> </tr> <tr> <td>Multiple Dependent</td> <td></td> <td></td> <td style="border: 1px solid black; text-align: center;">0</td> <td></td> <td></td> <td>=</td> <td style="border: 1px solid black; text-align: center;">-0-</td> </tr> </table> <table style="width:100%; border-collapse: collapse;"> <thead> <tr> <th>Large Entity Fee Code (\$)</th> <th>Small Entity Fee Code (\$)</th> <th>Fee Description</th> <th>Fee Paid</th> </tr> </thead> <tbody> <tr><td>1202 18</td><td>2202 9</td><td>Claims in excess of 20</td><td style="border: 1px solid black;"></td></tr> <tr><td>1201 84</td><td>2201 42</td><td>Independent claims in excess of 3</td><td style="border: 1px solid black;"></td></tr> <tr><td>1203 280</td><td>2203 140</td><td>Multiple dependent claim, if not paid</td><td style="border: 1px solid black;"></td></tr> <tr><td>1204 84</td><td>2204 42</td><td>** Reissue independent claims over original patent</td><td style="border: 1px solid black;"></td></tr> <tr><td>1205 18</td><td>2205 9</td><td>** Reissue claims in excess of 20 and over original patent</td><td style="border: 1px solid black;"></td></tr> <tr> <td colspan="3" style="text-align: right;">SUBTOTAL (2)</td> <td style="border: 1px solid black; text-align: center;">(\$) 1092</td> </tr> </tbody> </table> <p style="font-size: x-small;">**or number previously paid, if greater; For Reissues, see above</p>	Deposit Account Number	501432	Deposit Account Name	Jones, Day, Reavis & Pogue	Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid	1001 740	2001 370	Utility filing fee	740	1002 330	2002 165	Design filing fee		1003 510	2003 255	Plant filing fee		1004 740	2004 370	Reissue filing fee		1005 160	2005 80	Provisional filing fee		SUBTOTAL (1)			(\$) 740	Total Claims	62	-20** =	42	x	18	=	756	Independent Claims	7	-3** =	4	x	84	=	336	Multiple Dependent			0			=	-0-	Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid	1202 18	2202 9	Claims in excess of 20		1201 84	2201 42	Independent claims in excess of 3		1203 280	2203 140	Multiple dependent claim, if not paid		1204 84	2204 42	** Reissue independent claims over original patent		1205 18	2205 9	** Reissue claims in excess of 20 and over original patent		SUBTOTAL (2)			(\$) 1092	<p>3. ADDITIONAL FEES</p> <table style="width:100%; border-collapse: collapse;"> <thead> <tr> <th>Large Entity Fee Code (\$)</th> <th>Small Entity Fee Code (\$)</th> <th>Fee Description</th> <th>Fee Paid</th> </tr> </thead> <tbody> <tr><td>1051 130</td><td>2051 65</td><td>Surcharge - late filing fee or oath</td><td style="border: 1px solid black;"></td></tr> <tr><td>1052 50</td><td>2052 25</td><td>Surcharge - late provisional filing fee or cover sheet</td><td style="border: 1px solid black;"></td></tr> <tr><td>1053 130</td><td>1053 130</td><td>Non-English specification</td><td style="border: 1px solid black;"></td></tr> <tr><td>1812 2,520</td><td>1812 2,520</td><td>For filing a request for <i>ex parte</i> reexamination</td><td style="border: 1px solid black;"></td></tr> <tr><td>1804 920*</td><td>1804 920*</td><td>Requesting publication of SIR prior to Examiner action</td><td style="border: 1px solid black;"></td></tr> <tr><td>1805 1,840*</td><td>1805 1,840*</td><td>Requesting publication of SIR after Examiner action</td><td style="border: 1px solid black;"></td></tr> <tr><td>1251 110</td><td>2251 55</td><td>Extension for reply within first month</td><td style="border: 1px solid black;"></td></tr> <tr><td>1252 400</td><td>2252 200</td><td>Extension for reply within second month</td><td style="border: 1px solid black;"></td></tr> <tr><td>1253 920</td><td>2253 460</td><td>Extension for reply within third month</td><td style="border: 1px solid black;"></td></tr> <tr><td>1254 1,440</td><td>2254 720</td><td>Extension for reply within fourth month</td><td style="border: 1px solid black;"></td></tr> <tr><td>1255 1,960</td><td>2255 980</td><td>Extension for reply within fifth month</td><td style="border: 1px solid black;"></td></tr> <tr><td>1401 320</td><td>2401 160</td><td>Notice of Appeal</td><td style="border: 1px solid black;"></td></tr> <tr><td>1402 320</td><td>2402 160</td><td>Filing a brief in support of an appeal</td><td style="border: 1px solid black;"></td></tr> <tr><td>1403 280</td><td>2403 140</td><td>Request for oral hearing</td><td style="border: 1px solid black;"></td></tr> <tr><td>1451 1,510</td><td>1451 1,510</td><td>Petition to institute a public use proceeding</td><td style="border: 1px solid black;"></td></tr> <tr><td>1452 110</td><td>2452 55</td><td>Petition to revive - unavoidable</td><td style="border: 1px solid black;"></td></tr> <tr><td>1453 1,280</td><td>2453 640</td><td>Petition to revive - unintentional</td><td style="border: 1px solid black;"></td></tr> <tr><td>1501 1,280</td><td>2501 640</td><td>Utility issue fee (or reissue)</td><td style="border: 1px solid black;"></td></tr> <tr><td>1502 460</td><td>2502 230</td><td>Design issue fee</td><td style="border: 1px solid black;"></td></tr> <tr><td>1503 620</td><td>2503 310</td><td>Plant issue fee</td><td style="border: 1px solid black;"></td></tr> <tr><td>1460 130</td><td>1460 130</td><td>Petitions to the Commissioner</td><td style="border: 1px solid black;"></td></tr> <tr><td>1807 50</td><td>1807 50</td><td>Processing fee under 37 CFR 1.17(q)</td><td style="border: 1px solid black;"></td></tr> <tr><td>1806 180</td><td>1806 180</td><td>Submission of Information Disclosure Stmt</td><td style="border: 1px solid black;"></td></tr> <tr><td>8021 40</td><td>8021 40</td><td>Recording each patent assignment per property (times number of properties)</td><td style="border: 1px solid black; text-align: center;">40</td></tr> <tr><td>1809 740</td><td>2809 370</td><td>Filing a submission after final rejection (37 CFR 1.129(a))</td><td style="border: 1px solid black;"></td></tr> <tr><td>1810 740</td><td>2810 370</td><td>For each additional invention to be examined (37 CFR 1.129(b))</td><td style="border: 1px solid black;"></td></tr> <tr><td>1801 740</td><td>2801 370</td><td>Request for Continued Examination (RCE)</td><td style="border: 1px solid black;"></td></tr> <tr><td>1802 900</td><td>1802 900</td><td>Request for expedited examination of a design application</td><td style="border: 1px solid black;"></td></tr> </tbody> </table> <p>Other fee (specify) _____</p> <p>*Reduced by Basic Filing Fee Paid</p> <p style="text-align: right;">SUBTOTAL (3) (\$)</p>	Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid	1051 130	2051 65	Surcharge - late filing fee or oath		1052 50	2052 25	Surcharge - late provisional filing fee or cover sheet		1053 130	1053 130	Non-English specification		1812 2,520	1812 2,520	For filing a request for <i>ex parte</i> reexamination		1804 920*	1804 920*	Requesting publication of SIR prior to Examiner action		1805 1,840*	1805 1,840*	Requesting publication of SIR after Examiner action		1251 110	2251 55	Extension for reply within first month		1252 400	2252 200	Extension for reply within second month		1253 920	2253 460	Extension for reply within third month		1254 1,440	2254 720	Extension for reply within fourth month		1255 1,960	2255 980	Extension for reply within fifth month		1401 320	2401 160	Notice of Appeal		1402 320	2402 160	Filing a brief in support of an appeal		1403 280	2403 140	Request for oral hearing		1451 1,510	1451 1,510	Petition to institute a public use proceeding		1452 110	2452 55	Petition to revive - unavoidable		1453 1,280	2453 640	Petition to revive - unintentional		1501 1,280	2501 640	Utility issue fee (or reissue)		1502 460	2502 230	Design issue fee		1503 620	2503 310	Plant issue fee		1460 130	1460 130	Petitions to the Commissioner		1807 50	1807 50	Processing fee under 37 CFR 1.17(q)		1806 180	1806 180	Submission of Information Disclosure Stmt		8021 40	8021 40	Recording each patent assignment per property (times number of properties)	40	1809 740	2809 370	Filing a submission after final rejection (37 CFR 1.129(a))		1810 740	2810 370	For each additional invention to be examined (37 CFR 1.129(b))		1801 740	2801 370	Request for Continued Examination (RCE)		1802 900	1802 900	Request for expedited examination of a design application	
Deposit Account Number	501432																																																																																																																																																																																																								
Deposit Account Name	Jones, Day, Reavis & Pogue																																																																																																																																																																																																								
Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid																																																																																																																																																																																																						
1001 740	2001 370	Utility filing fee	740																																																																																																																																																																																																						
1002 330	2002 165	Design filing fee																																																																																																																																																																																																							
1003 510	2003 255	Plant filing fee																																																																																																																																																																																																							
1004 740	2004 370	Reissue filing fee																																																																																																																																																																																																							
1005 160	2005 80	Provisional filing fee																																																																																																																																																																																																							
SUBTOTAL (1)			(\$) 740																																																																																																																																																																																																						
Total Claims	62	-20** =	42	x	18	=	756																																																																																																																																																																																																		
Independent Claims	7	-3** =	4	x	84	=	336																																																																																																																																																																																																		
Multiple Dependent			0			=	-0-																																																																																																																																																																																																		
Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid																																																																																																																																																																																																						
1202 18	2202 9	Claims in excess of 20																																																																																																																																																																																																							
1201 84	2201 42	Independent claims in excess of 3																																																																																																																																																																																																							
1203 280	2203 140	Multiple dependent claim, if not paid																																																																																																																																																																																																							
1204 84	2204 42	** Reissue independent claims over original patent																																																																																																																																																																																																							
1205 18	2205 9	** Reissue claims in excess of 20 and over original patent																																																																																																																																																																																																							
SUBTOTAL (2)			(\$) 1092																																																																																																																																																																																																						
Large Entity Fee Code (\$)	Small Entity Fee Code (\$)	Fee Description	Fee Paid																																																																																																																																																																																																						
1051 130	2051 65	Surcharge - late filing fee or oath																																																																																																																																																																																																							
1052 50	2052 25	Surcharge - late provisional filing fee or cover sheet																																																																																																																																																																																																							
1053 130	1053 130	Non-English specification																																																																																																																																																																																																							
1812 2,520	1812 2,520	For filing a request for <i>ex parte</i> reexamination																																																																																																																																																																																																							
1804 920*	1804 920*	Requesting publication of SIR prior to Examiner action																																																																																																																																																																																																							
1805 1,840*	1805 1,840*	Requesting publication of SIR after Examiner action																																																																																																																																																																																																							
1251 110	2251 55	Extension for reply within first month																																																																																																																																																																																																							
1252 400	2252 200	Extension for reply within second month																																																																																																																																																																																																							
1253 920	2253 460	Extension for reply within third month																																																																																																																																																																																																							
1254 1,440	2254 720	Extension for reply within fourth month																																																																																																																																																																																																							
1255 1,960	2255 980	Extension for reply within fifth month																																																																																																																																																																																																							
1401 320	2401 160	Notice of Appeal																																																																																																																																																																																																							
1402 320	2402 160	Filing a brief in support of an appeal																																																																																																																																																																																																							
1403 280	2403 140	Request for oral hearing																																																																																																																																																																																																							
1451 1,510	1451 1,510	Petition to institute a public use proceeding																																																																																																																																																																																																							
1452 110	2452 55	Petition to revive - unavoidable																																																																																																																																																																																																							
1453 1,280	2453 640	Petition to revive - unintentional																																																																																																																																																																																																							
1501 1,280	2501 640	Utility issue fee (or reissue)																																																																																																																																																																																																							
1502 460	2502 230	Design issue fee																																																																																																																																																																																																							
1503 620	2503 310	Plant issue fee																																																																																																																																																																																																							
1460 130	1460 130	Petitions to the Commissioner																																																																																																																																																																																																							
1807 50	1807 50	Processing fee under 37 CFR 1.17(q)																																																																																																																																																																																																							
1806 180	1806 180	Submission of Information Disclosure Stmt																																																																																																																																																																																																							
8021 40	8021 40	Recording each patent assignment per property (times number of properties)	40																																																																																																																																																																																																						
1809 740	2809 370	Filing a submission after final rejection (37 CFR 1.129(a))																																																																																																																																																																																																							
1810 740	2810 370	For each additional invention to be examined (37 CFR 1.129(b))																																																																																																																																																																																																							
1801 740	2801 370	Request for Continued Examination (RCE)																																																																																																																																																																																																							
1802 900	1802 900	Request for expedited examination of a design application																																																																																																																																																																																																							

SUBMITTED BY		(Complete if applicable)	
Name (Print/Type)	John Y. Biernacki	Registration No. (Attorney/Agent)	40,511
Signature	<i>John Y. Biernacki</i>	Telephone	216-586-3939
		Date	Nov. 22, 2002

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

This collection of information is required by 37 CFR 1.17 and 1.27. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, Washington, DC 20231.

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

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Title

Computer-Implemented System And Method For Handling Database Statements

Inventor

Frederick J. Levine

TITLE

EL821237211US

Computer-Implemented System And Method For Handling Database Statements

TECHNICAL FIELD

The present invention relates generally to computer-implemented database systems and more particularly to database statement operations.

BACKGROUND

Data access across different database platforms proves difficult due to the
 5 platforms using varying database commands. For example, although the structured query
 language (SQL) is based on a well-documented ANSI standard, in reality most database systems,
 such as those from Oracle, Sybase, Business Objects, SAS, or Brio, implement a superset of the
 ANSI standard. Variations in the superset provide an obstacle in cross-platform database
 operations.

10

SUMMARY

In accordance with the teachings provided herein, a system and method are
 provided for handling a database statement from a first database system. The database statement
 is formatted according to a language format used by the first database system. Database
 15 language difference data is accessed so that a database specific statement may be generated
 which is operational within a different type of database system.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram depicting software and computer components that allow database statements to be automatically converted so that they may be used in a different type of database system;

5 FIG. 2 is a block diagram showing an example where the textualization process uses a tree to represent a database statement;

FIGS. 3-5 are block diagrams illustrating object-oriented approaches to creating disparate SQL text for third party data access;

FIG. 6 is a tabular representation depicting phrase component examples;

10 FIG. 7 is a tabular representation depicting identifier component examples;

FIG. 8 is a tabular representation depicting an example of component processing using SELECT and UPDATE commands;

FIG. 9 is a tabular representation listing exemplary expression components.

15 FIG. 10 is a process flow diagram showing exemplary processing of SQL statements;

FIG. 11 is a process flow diagram showing processing of an example query statement;

FIG. 12 is a tabular representation illustrating several exemplary components that may be involved in processing an SQL SELECT statement;

20 FIG. 13 is a block diagram illustrating use of an additional component that may be used in conjunction with overrides to standard components;

FIGS. 14 and 15 are listings of computer instructions to illustrate textualization examples involving different types of database systems;

processed SQL command 34 for the third party database system 42 that employs the third party's outer join syntax. The processed SQL command 34 is then able to be executed within the third party database system 42. As another example, a native database system 40 and a third party database system 42 may both support a function that performs an identical operation but differs in name and/or format. Using the specific textualizations 52, the textualization process 50 translates the SQL statement 32 having the function in the native format into an SQL statement 34 having the function in the third party's format. It should be understood that the terms "native" and "third party" are relative terms in that what is a native database system for one company may be a third party database system for another company. Accordingly, the terms "native" and "third party" database systems may be generalized to a first type of database system that generates a database statement that is processed by the textualization process 50 and provided to a different type of database system. It should be further understood that different types of databases refer to database systems that contain differences in their respective database statement format and/or syntax, such as utilizing a different superset of an ANSI database statement standard.

The textualization system 30 is highly flexible in that a third party database system 42 may utilize the textualization process 50 to convert and send database commands to the native database system 40. It is also noted that a native database system 40 may send database statements 32 to other third party database systems (44 and 46). The textualization system 30 contains textualization information and/or operations 52 that are specific to each of the third party database systems (42, 44, 46). The textualization system 30 has the flexibility of providing an SQL statement 32 from the native database system 40 to a single third party

database system 42, or may provide the same native SQL statement 32 to multiple third party database systems (42, 44, 46) substantially concurrently or serially.

FIG. 2 shows an example where an SQL tree 60 is used by the textualization process 50 to process an SQL statement 32. The SQL tree 60 represents the syntax of a native database's SQL statement 32 and its related metadata (e.g., table names, column names, etc.). The tree 60 may contain a hierarchical arrangement of nodes representative of the SQL syntax and metadata to be processed. If for example the SQL statement 32 specified that the values from two different columns are to be concatenated, then the SQL tree 60 would contain a node that specifies that a concatenation operation is to be performed.

10 The textualization process 50 compartmentalizes an SQL statement 32 into logical text pieces or components which are initially provided based on a default SQL dialect. The logical text pieces are represented in the SQL tree 60. Any of these text pieces can be overridden by a third party SQL provider that utilizes a different SQL dialect than the default, hence allowing for granular customization and code reuse. As an illustration, a database system from
15 SAS Institute Inc. has an SQL language which has differences from other vendor's SQL. The textualization process 50 allows a SAS SQL statement to be converted into a third party vendor-specific SQL in order to successfully submit a table request to the third party's relational database system (RDBMS). This is accomplished by representing the SAS SQL statement as an SQL tree 60. The SQL tree 60 is passed to the textualization process 50 to convert the tree 60
20 into the text of the third party vendor-specific SQL query, taking into account any DBMS-specific SQL. The textualization operation happens in this example just prior to the call to a prepare() or executeDirect() routine. These standardized routines then pass the SQL query to an RDBMS in the form of text. It is noted that in an SQL-centric table services model, an SQL

query typically gets passed to either the prepare() or executeDirect() routines (depending on context). A call to either of these routines, therefore, constitutes a request to an RDBMS.

An application or program that operates within a native database system may wish to access data in a remote third party database system under such situations as when the application is a data mining application that needs data from the third party system for operations to be performed within the data mining application. Other application examples include requests for metadata that are stored in a different type of database system. It should be understood that any system that uses SQL or a similar type of database technique may utilize the textualization system.

10 The textualization system may be implemented in many ways and through different types of programming languages. FIG. 3 illustrates an object-oriented approach 100 to creating disparate SQL text for third party data access. The object-oriented approach 100 contains SQL component objects 102 where each component corresponds to a logical "piece" 104 of an SQL statement (as may be found in an SQL tree). An SQL component defaults to a provided base or default native SQL text method 106. However, when there are third party-specific differences for a particular component, then the component utilizes the third party specific textualization method(s) 108 to handle the differences.

As shown in FIG. 4, when there are third party-specific differences for a particular component, a driver object 110 is responsible for creating an "override" 112 to the default method 106. The driver object 110 specifies to a component object 102 when a component object 102 is to point to specific textualization method 112 instead of its base textualization methods. Optionally, only the driver 110 knows about its datasource-specific SQL syntax 112.

In order to textualize for multiple different types of database systems, different drivers (110, 112) are associated with different third party platforms. For example, a first driver object 110 might point a component object 102 to use an "override" 108 to the default method 106 so that the component object 102 may textualize an SQL statement that can be used within a Sybase datasource system. A second driver object 114 might point the component object 102 to use an "override" 116 to the default method 106 so that the component object 102 may textualize an SQL statement that can be used within an Oracle datasource system.

The net effect of this object-oriented mechanism 100 is a driver-customized set of components where the driver need only supply an override method when a specific SQL construct differs from the default SAS SQL syntax. This design maximizes code reusability while pushing any DBMS-specific SQL text processing down to the driver, thereby distributing functionality more equitably.

With reference to FIG. 5, components may be assigned to handle different pieces of the native SQL statement. The different components may be: phrase components 120; identifier components 122; and expression components 124. Phrase components 120 handle textualization of SQL statement clauses or phrases, such as WHERE clauses or FROM clauses found in SQL select statements. To understand phrase components 120, the following exemplary SQL query statement is dissected:

```
select a.empid, b.sal from emp a, hr b where b.hdat > '01jan1998'd order by b.sal;
```

At the highest level we can think of this query as a group of ordered phrases (140, 142, 144, 146) as shown in FIG. 6. Phrase 140 of the query is a SELECT phrase; phrase 142 is a FROM phrase; phrase 144 is a WHERE phrase; and phrase 146 is an ORDER BY phrase. These phrases (140, 142, 144, 146) provide the high-level context for the query. Each phrase is

represented and processed by a component object (150, 152, 154, 156) whose default text method can be overridden by a driver at this "high level". The phrase components (150, 152, 154, 156) textualize large pieces of a query that correspond to high-level SQL operations, e.g., FROM clauses, WHERE clauses, ORDER BY clauses, etc.

5 Typically, a driver would not have to override phrase component methods because the general syntactical layout of phrases tends to be relatively standard across RDBMSs. However, exceptions may exist, such as those involving RDBMSs that support non-standard outer joins. In addition, some ERP (enterprise resource planning) systems may have the need for phrase overrides since their "SQL" tends to be proprietary and very non-standard.

10 Identifier components are next discussed in reference to FIG. 7 using the query example above. We can further break down the phrases as collections of low-level "physical" entities. These low-level "entities" can be thought of as identifiers because they represent a real physical entity in an SQL query. Fully qualified table names, column names, literals, and aliases fall into this category of components. As shown in FIG. 7, the identifier component

15 GenColName 160 textualizes the column names a.empid 162 and b.sal 164 within the SELECT phrase; the identifier component GenTableName 170 textualizes the table names emp 172 and hr 174 within the FROM phrase; the identifier component GenAlias textualizes the aliases a 182 and b 184 within the FROM phrase. It is noted that aliases are a unique class of identifiers in that they are only valid in GenSelectList and GenFrom (and typically not allowed in filter

20 clauses). The identifier component GenColName 190 textualizes the column name b.hdat 192 within the WHERE phrase; the identifier component GenDateValue 200 textualizes the date value 202 within the WHERE phrase; the identifier component GenColName 210 textualizes the column name b.sal 212 within the ORDER BY phrase.

- `select min(sal+20000) from emp where (select dept from emp) = 'SALES';`

In order to efficiently process such SQL syntaxes and entities, expression components are used as a more abstract type of component. Optionally, expression components provide a common entry point into all forms of an SQL entity and to this end, SQL entities in a query are initially

5 processed as expressions. (Note that aliases may be an exception since they are only valid within the GenSelectList and GenFrom phrase components as described later in reference to FIG. 11).

A generic expression component, GenExpression, may be used which is the "catch-all" expression method through which SQL entities are initially processed. At the point where the GenExpression component is called, the SQL entity can be an identifier (e.g., column, table,

10 literal) or it can be another, more granular expression that has more context than the generic GenExpression. Such "granular" expressions processed by GenExpression are SQL functions, compound expressions, third party (or native) functions (e.g., SAS functions), and SQL commands. Function expressions treat each function argument as a generic expression (since arguments can typically be any form of SQL entity). A compound expression includes an SQL

15 keyword or operator combined with one or more other expressions, i.e., "empid is null", "a + b", IN clauses. It is noted that if an SQL is used, then the tree node type representing the keyword or operator may identify an expression as compound. Phrase components correspond to the clauses (or phrases) of a command and may be processed in an order of precedence.

An example of component processing is shown in FIG. 8 using SELECT and

20 UPDATE commands. Two SQL commands are shown in column 250 -- a SELECT command and an UPDATE command. Column 252 lists that the command expression component GenQuery is used to determine what command phrase components are needed to textualize the SELECT command. Column 254 lists the order in which the phrases are processed, and column

components, such as a SELECT or UPDATE command expression. Because commands contain phrases, command expression components 358 invoke phrase components 370 to textualize phrases as shown in block 372.

Similar to GenExpression 352, granular expression components (354, 356, 358) also call identifier components 360 and other expression components, but within a more specific context. For example, the GenSASFunction component and the GenSQLFunction component (shown in block 364) have the context to know the type of function and how many arguments to process. They then call the GenExpression component 352 for each function argument.

To further illustrate the exemplary component operational flow, the example query described above is used and is as follows:

```
select a.empid, b.sal from emp a, hr b where b.hdat > '01jan1998'd order by b.sal;
```

We can see that at the highest level the entire query is a command expression that will be processed by the GenQuery expression component 358 shown in FIG. 10. Since GenQuery is itself an expression (as are all SQL commands), GenExpression 352 is the entry point into the text component system. It should be understood that the system may have different entry points, such as an entry point where the converted tree calls directly other component objects or the entry point is to a program that checks the syntax of the input SQL command with respect to the native database system's query language format.

After the driver loads the software appendage, exports its overrides, and does setup operations, it calls GenExpression 352 once to produce the entire SQL text from a provided SQL tree. In this example, this is true for all SQL commands, and all driver-provided overrides automatically get applied as needed because components utilize a common call interface.

Drivers may also be free to perform setup operations. It should be noted that a query may also appear as an SQL entity within a query (referred to as a subquery or inline view), so GenQuery 358 may be called multiple times.

FIG. 11 shows in greater detail the order in which different components are invoked in processing the example query. The driver calls GenExpression 380 to textualize the query. GenExpression 380 calls GenQuery 390 to process the SELECT statement. GenQuery 390 then calls the following phrase components: GenSelectList 400A; GenFrom 404A; GenWhere 410A; GenOrderBy 420A. The phrase components (400A, 404A, 410A, 420A) call GenExpression (401, 405, 411, 421) which, in turn, calls the appropriate identifier/expression components: GenExpression 401 calls GenColName 402A; GenExpression 405 calls GenTableName 406A; GenExpression 411 calls GenCompoundExpr 412A which calls GenColName 414A, and GenColName 414A calls GenDateValue 416A; GenExpression 421 calls GenColName 422A. Note that GenFrom 404A calls GenAlias 408A directly (because aliases are only valid for select list items and result sets). As illustrated in FIG. 11, recursion is used as a mechanism of textualization.

The following table lists what statement portions are textualized by which components (shown in FIG. 11):

COMPONENT	STATEMENT PIECE
GenSelectList 400A	select 400B
GenColName 402A	a.empid 402B b.sal 402C
GenFrom 404A	from 404B
GenTableName 406A	emp 406B hr 406C
GenAlias 408A	a 408B b 408C
GenWhere 410A	where 410B
GenCompoundExpr 412A	(...) 412B and 412C
GenColName 414A	b.hdat 414B
GenDateValue 416A	'01jan98'd 416B
GenOrderBy 420A	order by 420B
GenColName 422A	b.sal 422B

A driver may have no need to override either GenExpression 380 or GenQuery 390 because they are both high-level expression components that utilize components common to all standard SQL implementations.

It is noted that GenSASFunction and GenSQLFunction components are much more driver-specific. Most third party drivers will need overrides to GenSASFunction since any SAS function would require a driver replacement function for a successful prepare.

GenSQLFunction processes SQL-defined functions. This includes the standard aggregate functions along with any other SQL-defined functions (e.g., COALESCE, any new SQL-99 functions, etc.). Some drivers may need an override to GenSQLFunction.

5 Some drivers will also need overrides to GenCompoundExpr as well. Consider the case where a driver does not support "a || b" but instead supports "concat(a,b)". Or the driver may not support a default operator -- for example, Sybase uses '+' instead of '||' for a concatenation operator). Driver overrides will be required for such cases.

10 FIG. 12 provides an exemplary component listing of different phrase components 450, expression components 452, and identifier components 454 that a textualization system might wish to use for a select SQL statement. It should be understood that this list may be extended for non-SELECT components.

15 While examples have been used to disclose the invention, including the best mode, and also to enable any person skilled in the art to make and use the invention, the patentable scope of the invention is defined by the claims, and may include other examples that occur to those skilled in the art. For example, different component objects may be used instead of or in addition to the above-listed component objects. As an illustration and with reference to FIG. 13, a "parent" component type 470 may be used in conjunction with overrides to other components. A "parent" component 470 is a static component that always points to a specific default method and is called from a corresponding override method when necessary. That is, 20 when a driver exports an override to a given method, the parent component 470 gives the driver a mechanism to call back to the overridden default method. This may be used when an override method wishes to call back into the corresponding default (or parent) method to do the work

dealing with a newer version of Oracle that does support standard outer join syntax. The default GenWhere may be used at 570 for those cases.

From the above examples we can see how the parent components are named with respect to the standard components. Optionally, a parent component has the same name as its
5 corresponding standard component but prefixed with the term "Parent", and standard components have a corresponding parent as shown in FIG. 16.

We can now see how the parent components allow the drivers to be very granular in how they implement their override methods. Using this override mechanism combined with parent components, a driver may write just those cases of a method that differ from the default.
10 Such a design maximizes code reuse. However, it should be noted that less optimal implementations may be used and still achieve advantages of the textualization system. As another example of the applicability and extensions of the textualization system, the textualization system may be applied to database system dialects other than SQL. As yet another example of the many applications and extensions of the system, the textualization system has
15 general applicability to third party data store systems which are SQL-based. Such drivers can access RDBMSs as well as ERP systems and other such systems.

As yet another example of the wide variation of the textualization system, a textualization process 602 may textualize an input SQL statement 32 (which is formatted in a particular native database query format) as one or a set of application programming interface
20 (API) calls 604. The API calls 604 are textualized so as to be compatible and directly executable within another database system 606 that utilizes API calls to perform database operations. It should be understood that the textualized API calls 604 may need to be compiled into machine code for execution within the second database system. The textualization process 602 accesses

providing of textualized database statements or API calls to third party database systems may be across one or more network connections in order to access the third party database systems.

4. The method of claim 3 wherein the object-oriented techniques contain SQL component objects, wherein a component object corresponds to a logical piece of an SQL statement.

5. The method of claim 4 wherein a logical piece is a phrase logical piece.

5

6. The method of claim 4 wherein a logical piece is an identifier logical piece.

7. The method of claim 4 wherein an SQL component object defaults to a default native SQL textualization method for use in generating the second database statement.

10

8. The method of claim 7 wherein an SQL component object comprises an override to account for differences between the first and second database systems' query language formats.

9. The method of claim 8 wherein SQL component objects comprise phrase component object means.

15

10. The method of claim 8 wherein SQL component objects comprise identifier component object means.

11. The method of claim 8 wherein SQL component objects comprise expression component object means.

20

18. The method of claim 17 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.

19. The method of claim 1 wherein the second database system is a different type of database system than the first database system.

20. The method of claim 1 wherein generating the second database statement provides the ability to manipulate data within the second database system from the first database system.

21. The method of claim 1 wherein generating automatically the second database statement provides the ability to transparently manipulate data within the second database system from the first database system.

22. The method of claim 1 wherein the generated second database statement is provided to the second database system for execution by the second database system.

23. The method of claim 1 wherein the generated second database statement is in a format such that the second database statement is directly executable by the second database system.

24. The method of claim 1 wherein the second database statement is a functional equivalent of the first database statement but for differences between the first and second database systems' query language formats.

30. The method of claim 1 wherein an SQL tree is used to generate the second database statement, wherein the SQL tree contains data that represents the syntax of the first database statement.

5 31. The method of claim 30 wherein the SQL tree contains metadata related to the first database statement.

32. The method of claim 31 wherein the first database statement is parsed into logical text pieces which are stored in the SQL tree.

10

33. The method of claim 32 wherein the second database statement takes into account any second database system-specific query language syntax.

34. The method of claim 1 further comprising the step of:

15 generating a third database statement for use within a third database system, wherein the third database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the third database statement is compatible with the third database system's query language format.

20 35. The method of claim 34 further comprising the step of:

 generating a fourth database statement for use within a fourth database system, wherein the fourth database statement is generated based upon the first database statement and

upon the accessed database language difference data, wherein the fourth database statement is compatible with the fourth database system's query language format.

5 36. The method of claim 1 wherein the first and second database systems are relational database management systems.

37. The method of claim 1 wherein the first database system comprises a data mining application.

10 38. The method of claim 37 wherein the second database system comprises a relational database management system.

39. The method of claim 1 wherein the first database system comprises a relational database management system.

15

40. The method of claim 39 wherein the second database system comprises a data mining application.

20 41. The method of claim 1 wherein the first database system comprises an enterprise resource planning system.

42. The method of claim 41 wherein the second database system comprises an enterprise resource planning system.

43. The method of claim 1 wherein the first and second database systems' query language format includes format specifications for insert, select, update, and delete database commands.

5 44. Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 1.

45. A computer-implemented apparatus for handling a first database query that is formatted in a first query format and is executable by a first database system, comprising:

10 tree-structured data that is representative of syntax and metadata of the first database query;

a data structure for storing query specific data that indicates at least one query language difference from the first query format, wherein the query language difference is a query syntax difference; and

15 a textualization module having a data access connection to the tree-structured data and the data structure, wherein the textualization module generates a database specific query based upon the tree-structured data and the query specific data, wherein the database specific query accounts for the difference from the first query format so that the database specific query may be executed by a different type of database system.

46. A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

5

identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and

generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format.

10

15

47. The method of claim 46 wherein the language parts are determined to be common based upon a predetermined standardized query language format.

48. The method of claim 47 wherein the standardized query language format is based upon a standardized structured query language (SQL) version.

49. A computer-implemented apparatus for handling a database statement from a first database system, comprising:

means for receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for identifying, for the first database statement, query language parts that are common between the first database system's query language format and a second database system's query language format; and

means for generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the identified common query language parts, wherein the generated second database statement is compatible with the second database system's query language format.

50. A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call

generating a second database statement for use within the second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the second database statement is compatible with the second database system's query language format.

5

58. The method of claim 50 further comprising the steps of:

accessing database language difference data, wherein the database language difference data indicates a format that contains at least one database statement difference from the first database system's query language format; and

10

generating a second database statement for use within a third database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the second database statement is compatible with the third database system's query language format.

15

59. Computer software stored on a computer readable media, the computer software comprising program code for carrying out a method according to claim 50.

60. A computer-implemented apparatus for handling a database statement from a first database system, comprising the steps of:

20

means for receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

means for accessing program call textualization specific data, wherein the program call textualization specific data indicates formatting of a program call to access data contained in a second database system; and

5 means for generating a program call for use within the second database system, wherein the program call is generated based upon the first database statement and upon the program call textualization specific data, wherein the generated program call is in a format that is compatible with the second database system.

61. A computer-implemented method for handling a database statement from a first database system, comprising the steps of:

10 receiving a first database statement from the first database system, wherein the first database statement is formatted according to the first database system's query language format;

15 parsing the first database statement to obtain first query metadata of the first database statement;

using the obtained first query metadata to access database language difference data, wherein the database language difference data indicates a format that contains at least one database statement difference from the first database system's query language format;

20 generating a second database statement for use within a second database system, wherein the second database statement is generated based upon the first database statement and upon the accessed database language difference data, wherein the second database statement is compatible with the second database system's query language format;

receiving another database statement from the first database system, wherein the additional database statement from the first database system is formatted according to the first database system's query language format;

5 parsing the additional database statement to obtain second query metadata of the additional database statement;

using the second query metadata to generate a program call to a third database system which utilizes a different query language format than the first database system; and

issuing the program call to the third database system to access data contained in the third database system.

10

62. The method of claim 61 further comprising the steps of:

using object-oriented component means for generating database access instructions to the second database system;

15 wherein an override exists that indicates for the object-oriented component means to use the first query metadata to generate a program call to the second database system; and

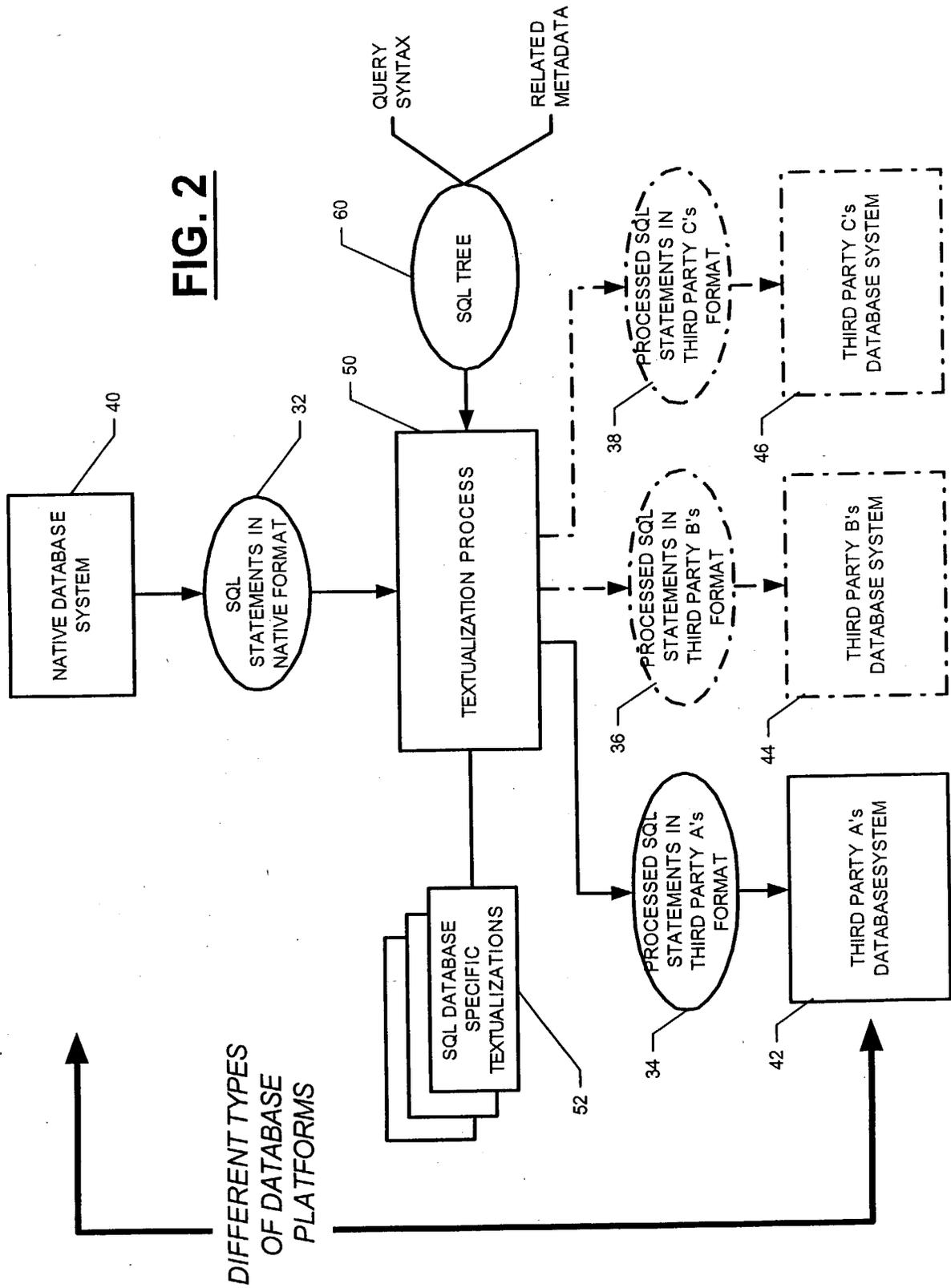
issuing the program call to the second database system to access data contained in the second database system,

20 wherein if the override did not exist, then the object-oriented component means textualizes a database statement based upon the first query metadata and which is executable within the second database system.

Computer-Implemented System And Method For Handling Database Statements**ABSTRACT**

A computer-implemented system and method for handling a database statement from a first database system. The database statement is formatted according to a language format used by the first database system. Database language difference data is accessed so that a database specific statement may be generated which is operational within a different type of database system.

FIG. 2



DIFFERENT TYPES OF DATABASE PLATFORMS

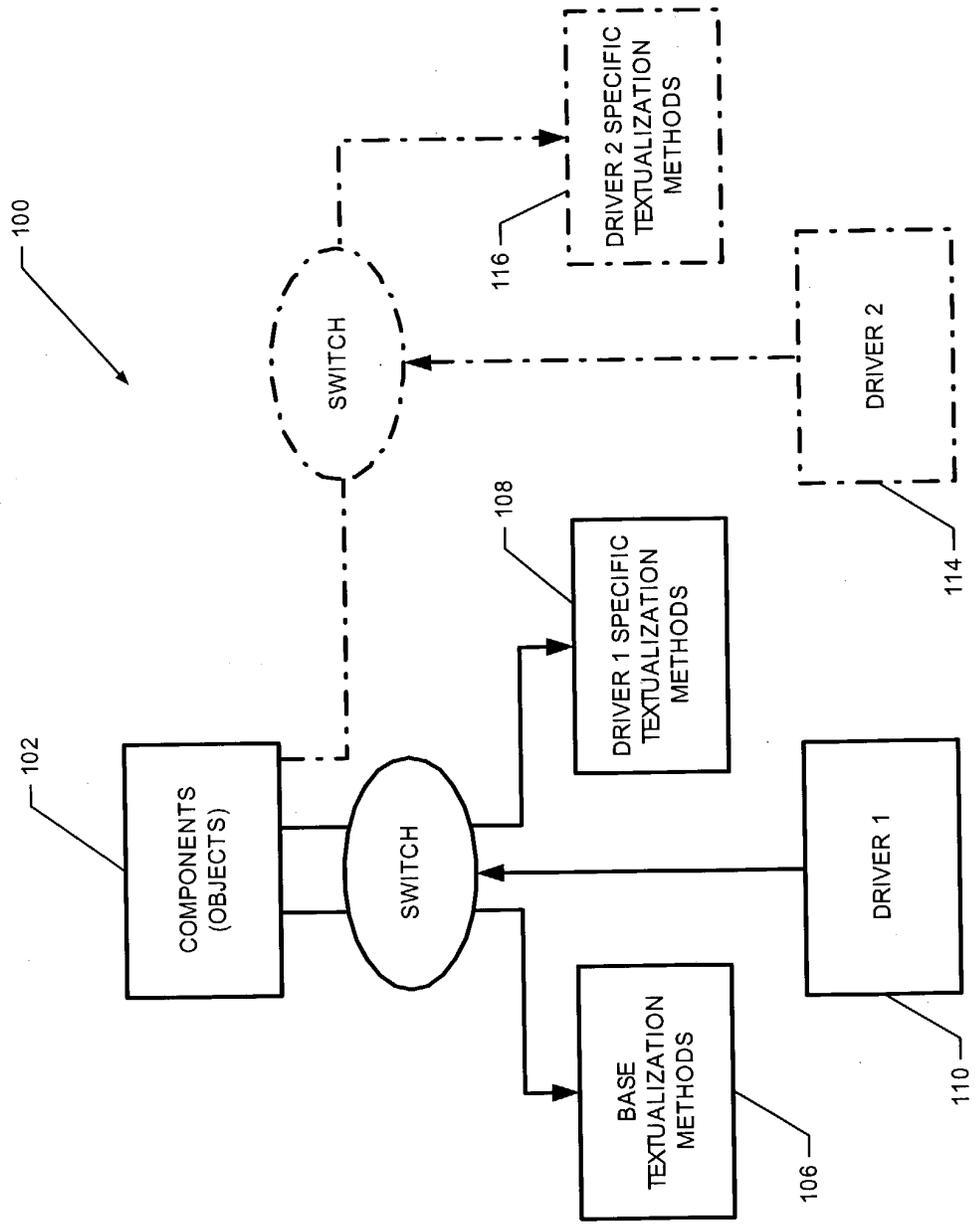


FIG. 4

select a.empid, b.sal	from emp a, hr b	where b.hdat > '01jan98'd	order by b.sal;
GenColName	GenTableName	GenColName	GenColName
	GenAlias	GenDateValue	

FIG. 7

SQL Command	Command Expression Component	Command Phrase	Corresponding Phrase Component
SELECT	GenQuery	select list	GenSelectList
		from	GenFrom
		where	GenWhere
		group by	GenGroupBy
		having	GenHaving
		order by	GenOrderBy
UPDATE	GenUpdate	update table	GenUpdTable
		set list	GenSetList

FIG. 8

Expression Component	Expression Type	Examples
GenExpression	Generic	ALL SQL entities
GenSASFunction	Function	"SUBSTR(lastname, 1, 3)"
GenSQLFunction	Function	"COUNT(*)", "AVG(x)"
GenCompoundExpr	Compound	"(a + 1)", "(a b)"
GenQuery	Command	"SELECT * from a where a > 0"

FIG. 9

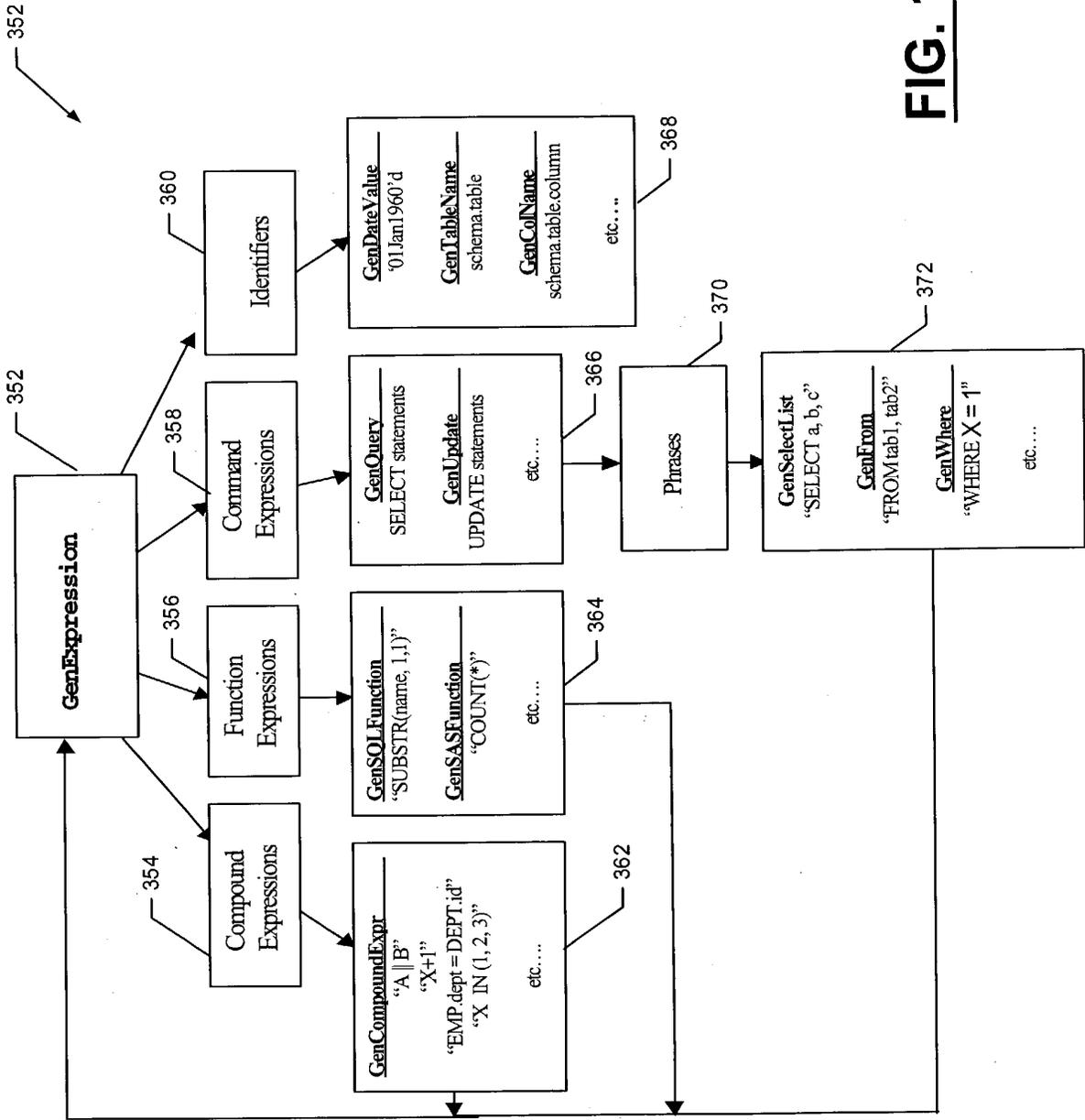


FIG. 10

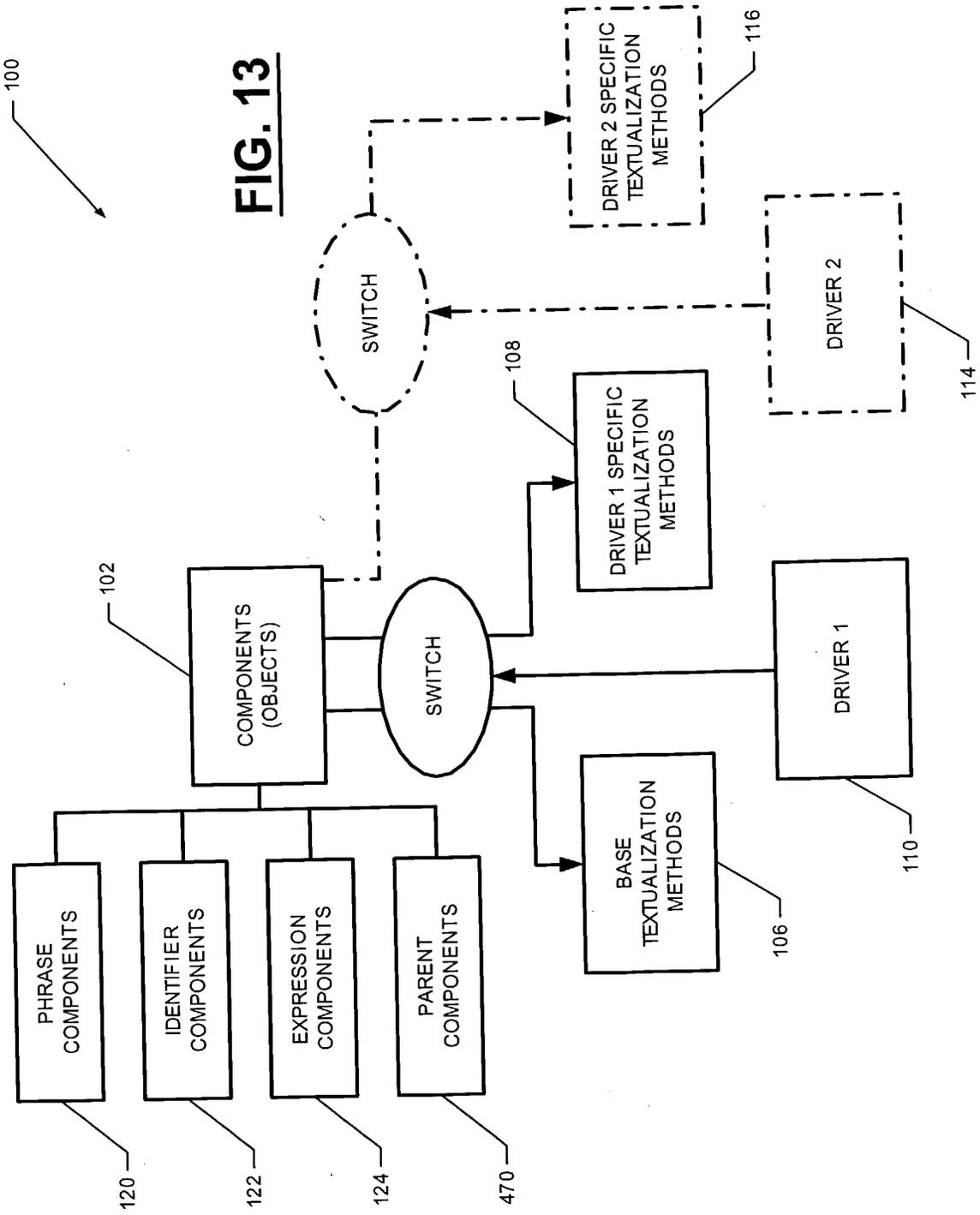


FIG. 13

```
sybGenCompoundExpr(tqueryh)
{
  if (tqueryh->tree->code == SQL_CONC) // CONCATENATION node
  {
    // Textualize Sybase-specific concatenation 510
  }
  else
  {
    tkrc = ParentGenCompoundExpr(tqueryh); // Let the default method
    // handle the rest!
  }
}
```

FIG. 14

```

oraGenWhere(tqueryh)
{
550   if ( (tqueryh->tree->code == SQL_OTRJ) &&      // NON-standard Outer joins?
        (non-standard_OJ_server) )
    {
560       // Textualize Oracle-specific outer join WHERE conditions
    }
    else
570       tkrc = ParentGenWhere(tqueryh); // Let the default method handle the rest
}

```

FIG. 15

NO. 500005, 4/12/05

Standard Component	Parent Component
GenColName	ParentGenColName
GenCompoundExpr	ParentGenCompoundExpr
GenFrom	ParentGenFrom
etc.	etc.

FIG. 16

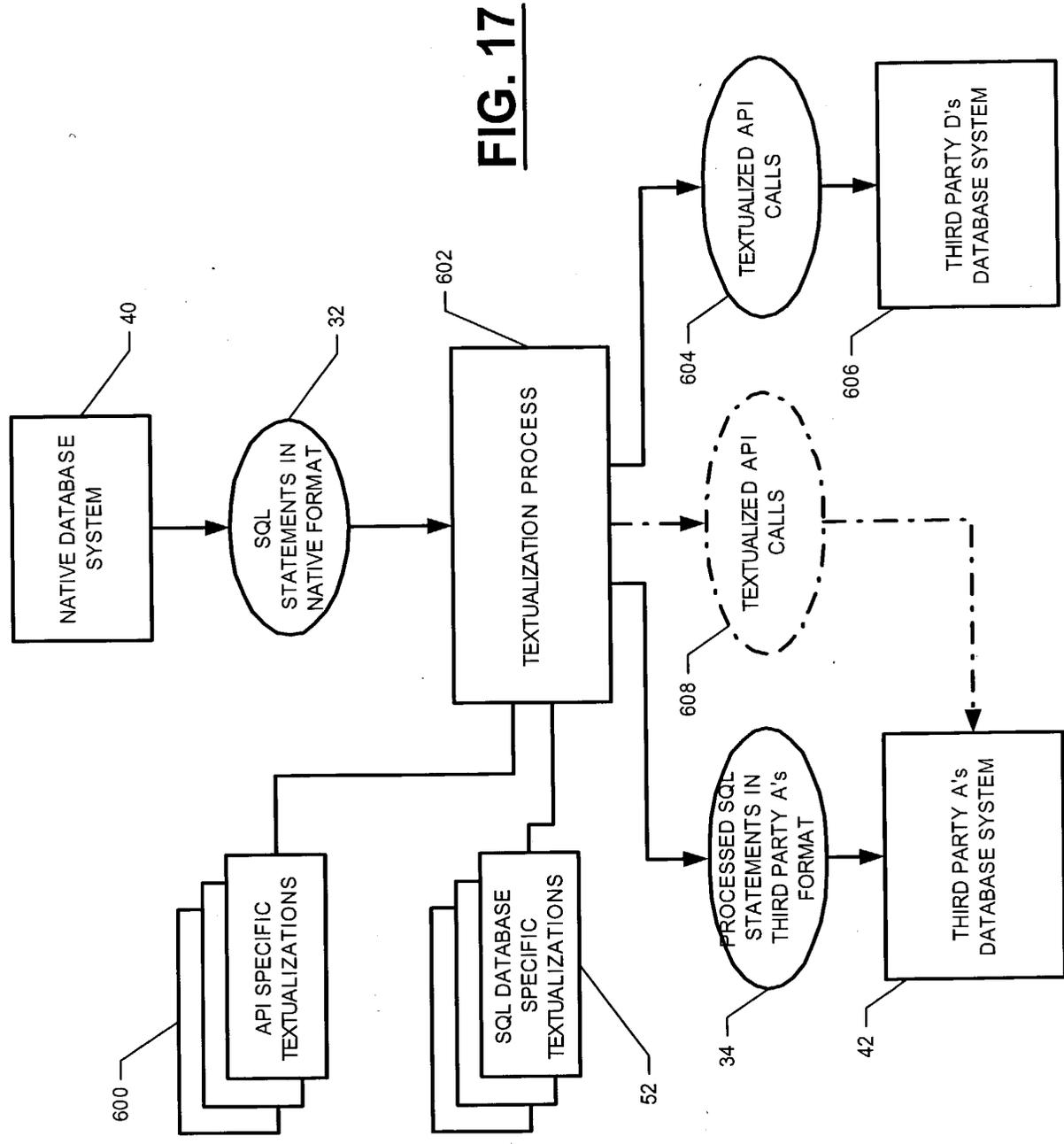
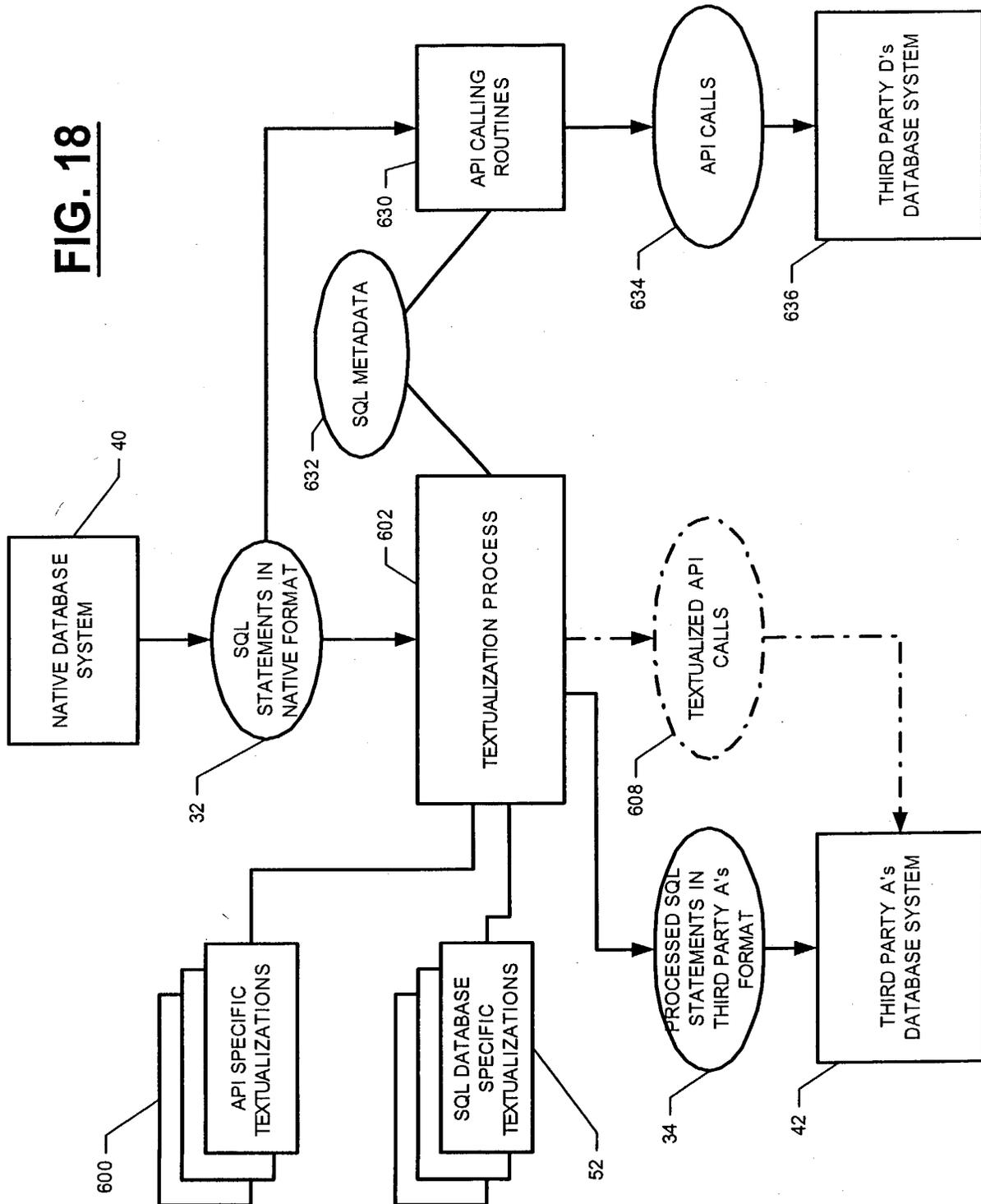


FIG. 17

FIG. 18



punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Frederick J. Levine Date 11-21-02
FREDERICK J. LEVINE

Post Office Address: 1302 Vickers Avenue
Durham, NC 27707

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

11/27/2002 EHAILE1 00000015 501432 10303106

01 FC:1001	740.00 CH
02 FC:1201	336.00 CH
03 FC:1202	756.00 CH

PTO-1556
(5/87)

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 1, 2001

Application or Docket Number

343355600054

CLAIMS AS FILED - PART I

(Column 1) (Column 2)

TOTAL CLAIMS	62	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	62 minus 20 =	* 42
INDEPENDENT CLAIMS	7 minus 3 =	* 4
MULTIPLE DEPENDENT CLAIM PRESENT <input type="checkbox"/>		

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE

OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
BASIC FEE	370.00	OR	BASIC FEE	740.00
X\$ 9=		OR	X\$18=	756
X42=		OR	X84=	336
+140=		OR	+280=	
TOTAL		OR	TOTAL	1832

CLAIMS AS AMENDED - PART II

(Column 1) (Column 2) (Column 3)

	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR		PRESENT EXTRA
AMENDMENT A	Total	*	Minus	**	=
	Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <input type="checkbox"/>					

SMALL ENTITY

OR OTHER THAN SMALL ENTITY

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR		PRESENT EXTRA
AMENDMENT B	Total	*	Minus	**	=
	Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <input type="checkbox"/>					

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR		PRESENT EXTRA
AMENDMENT C	Total	*	Minus	**	=
	Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM <input type="checkbox"/>					

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X\$ 9=		OR	X\$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

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

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

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

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

BEST AVAILABLE COPY

CLAIMS ONLY

SERIAL NO. 10/303106 FILING DATE 11/22/02
 APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT			*		*		*	
	IND.	DEP.	IND.	DEP.	IND.	DEP.		IND.	DEP.	IND.	DEP.	IND.	DEP.
1	1												
2		1					51		1				
3		1					52		1				
4		1					53		1				
5		1					54		1				
6		1					55		1				
7		1					56		1				
8		1					57		1				
9		1					58		1				
10		1					59		1				
11		1					60	1					
12		1					61	1					
13	1						62	1	1				
14		1					63						
15		1					64						
16		1					65						
17		1					66						
18		1					67						
19		1					68						
20		1					69						
21		1					70						
22		1					71						
23		1					72						
24		1					73						
25		1					74						
26		1					75						
27		1					76						
28		1					77						
29		1					78						
30		1					79						
31		1					80						
32		1					81						
33		1					82						
34		1					83						
35		1					84						
36		1					85						
37		1					86						
38		1					87						
39		1					88						
40		1					89						
41		1					90						
42		1					91						
43		1					92						
44		1					93						
45		1					94						
46		1					95						
47		1					96						
48		1					97						
49		1					98						
50		1					99						
							100						
TOTAL IND.							TOTAL IND.	2					
TOTAL DEP.							TOTAL DEP.	10					
TOTAL CLAIMS	50						TOTAL CLAIMS	12					

* MAY BE USED FOR ADDITIONAL CLAIMS OR ADMMENDMENTS