AO 120 (Rev. 08/10)

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

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 for the District of Delaware on the following

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

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
	7/5/2022	for the District of Delaware

PLAINTIFF	DEFENDANT
INQUISIENT INC., a Delaware corporation	SERVICENOW, INC., a Delaware corporation

	PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1	US 7,979,468 B2	7/12/2011	InQuisient Inc.
2	US 8,219,585 B2	7/10/2012	InQuisient Inc.
3	US 8,224,855 B2	7/17/2012	InQuisient Inc.
4			
5			

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

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ 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

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.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/917,410	07/12/2011	7979468	09801.0002	6234

22852 7590 06/22/2011

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER
LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

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 618 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 Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

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

Bruce Randall DeWoolfson, Potomac Falls, VA;

PART B - FEE(S) TRANSMITTAL

OPAP
JUN 03 2011
PATENT & TRADEMARK OFFICE

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)

22852 7590 04/12/2011

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

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.
11/917,410	12/13/2007	Bruce Randall DeWoolfson	09801.0002	6234

TITLE OF INVENTION: DATABASE DATA DICTIONARY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	07/12/2011

EXAMINER	ART UNIT	CLASS-SUBCLASS
MORRISON, JAY A	2168	707-790000

06/06/2011 AWONDAF2 00000078 11917410
01 FC:1501 1510.00 OP
02 FC:1504 300.00 OP

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,
(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 Finnegan, Henderson,
2 Farabow, Garrett &
3 Dunner, LLP

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: ENTERPRISE ELEMENTS, INC.

(B) RESIDENCE: (CITY and STATE OR COUNTRY): RESTON, VIRGINIA

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 06-0916 (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 Robert E. Converse, Jr. Date June 3, 2011

Typed or printed name Robert E. Converse, Jr. Registration No. 27,432

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

NOTICE OF ALLOWANCE AND FEE(S) DUE

22852 7590 04/12/2011

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER
MORRISON, JAY A

ART UNIT	PAPER NUMBER
2168	

DATE MAILED: 04/12/2011

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/917,410	12/13/2007	Bruce Randall DeWoolfson	09801.0002	6234

TITLE OF INVENTION: DATABASE DATA DICTIONARY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	07/12/2011

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)

22852 7590 04/12/2011

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

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.
11/917,410	12/13/2007	Bruce Randall DeWoolfson	09801.0002	6234

TITLE OF INVENTION: DATABASE DATA DICTIONARY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1510	$300	$0	$1810	07/12/2011

EXAMINER	ART UNIT	CLASS-SUBCLASS
MORRISON, JAY A	2168	707-790000

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,

(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 _____

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/917,410	12/13/2007	Bruce Randall DeWoolfson	09801.0002	6234

22852 7590 04/12/2011

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER	
MORRISON, JAY A	

ART UNIT	PAPER NUMBER
2168	

DATE MAILED: 04/12/2011

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 408 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 408 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.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Notice of Allowability	Application No.	Applicant(s)	
	11/917,410	DEWOOLFSON, BRUCE RANDALL	
	Examiner	Art Unit	
	JAY A. MORRISON	2168	

-- 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 *1/21/2011*.

2. ☒ The allowed claim(s) is/are *1-39*.

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 Draftperson'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 _____ .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/Jay A Morrison/
Primary Examiner, Art Unit 2168

Examiner-Initiated Interview Summary	Application No.	Applicant(s)	
	11/917,410	DEWOOLFSON, BRUCE RANDALL	
	Examiner	Art Unit	
	JAY A. MORRISON	2168	

All Participants:

(1) *JAY A. MORRISON*.

(2) *Lionel Lavenue (Reg No. 46,859)*.

(3) _____.

(4) _____.

Status of Application: *non-final response received*

Date of Interview: *7 April 2011*

Time: *11am*

Type of Interview:
☒ Telephonic
☐ Video Conference
☐ Personal (Copy given to: ☐ Applicant ☐ Applicant's representative)

Exhibit Shown or Demonstrated: ☐ Yes ☒ No
If Yes, provide a brief description: .

Part I.

Rejection(s) discussed:
35 USC 103

Claims discussed:
1

Prior art documents discussed:
None

Part II.

SUBSTANCE OF INTERVIEW DESCRIBING THE GENERAL NATURE OF WHAT WAS DISCUSSED:
See Continuation Sheet

Part III.

☒ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview directly resulted in the allowance of the application. The examiner will provide a written summary of the substance of the interview in the Notice of Allowability.
☐ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview did not result in resolution of all issues. A brief summary by the examiner appears in Part II above.

/Jay A Morrison/
Primary Examiner, Art Unit 2168

(Applicant/Applicant's Representative Signature – if appropriate)

Continuation of Substance of Interview including description of the general nature of what was discussed: Applicant was contacted to propose an examiner amendment moving dependant claim 2 into independent claim 1 in order to overcome the rejections of record. After negotiation, the applicant agreed to move appropriate portions of claim 2 into claim 1 in order to place the claims in condition for allowance. Applicant agreed to make these same changes to the all the independent claims and provide the amended claim set, which are attached to this action as the examiner amendment..

DETAILED ACTION

EXAMINER'S AMENDMENT

An examiner's amendment to the record appears below. 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.

Authorization for this examiner's amendment was given in a telephone interview with Lionel Lavenue (Reg. No. 46,859) on 4/7/2011.

The application has been amended as follows:

Please amend the claims as per the attached Examiner's Amendment document.

Reasons for Allowance

The following is an examiner's statement of reasons for allowance:

Claims 1-39 are allowable over the prior art of record. The closest prior art of record Habela, teaches an object-oriented DBMS metadata management system. The reasons for allowance of these claims are that the prior art of record neither anticipates,

nor renders obvious the recited combination as a whole, including the limitations being indicated in independent claim 1 as “a data repository, the data repository being adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, comprising: an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification; an element relation module configured to store one or more relationships between the elements in the element module; a class module configured to define at least one class of the elements and store the class; an attribute module configured to define one or more attributes and store the attributes; a class attribute module configured to define and store one or more class attribute associations between at least one of the attributes and the class; a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements; a state machine module configured to store one or more state machine types associated with at least one of the elements; and a status module configured to store one or more statuses of each state machine”. Independent claims 14 and 27 are allowable for substantially the same reasons. Respectfully, it is noted that for the purposes of examination the applicant acted as their own lexicographer on pages 5-8 of the “Remarks” section of their response received 1/21/2011. In these remarks, the applicants stated that certain portions of the claims drawn to interpretations of a class, type, and attribute, among other language in the claims, should "not be defined as standard object-oriented structures" (Applicants “Remarks” 1/21/2011, page 5). Accordingly, for the purposes of

examination the appropriate language in the claims was construed according to the definitions found in the instant specification, as noted by the applicant. Respectfully, these interpretations were made according to MPEP 2111.01(IV), where the applicant may be their own lexicographer, since the applicant in the instant application rebutted the presumption that the claim terms were to be given their ordinary and customary meaning and clearly stated definitions different from their ordinary and customary meaning, as found on pages 5-8 in the “Remarks” section of the applicant's response received 1/21/2011.

Dependent claims 2-13, 15-26 and 28-39 are allowable at least for the reasons recited above including all the limitations of the allowable independent base claim upon which they depend.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled “Comments on Statement of Reasons for Allowance.”

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jay A. Morrison whose telephone number is (571) 272-7112. The examiner can normally be reached on M-F 8-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached on (571) 272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

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

/Jay A Morrison/
Primary Examiner, Art Unit 2168

Jay Morrison
TC2100

Issue Classification

Application/Control No.	Applicant(s)/Patent Under Reexamination
11917410	DEWOOLFSON, BRUCE RANDALL
Examiner	**Art Unit**
JAY A MORRISON	2168

ORIGINAL		INTERNATIONAL CLASSIFICATION															
CLASS	SUBCLASS	CLAIMED									NON-CLAIMED						
707	790	G	0	6	F	7 / 00 (2006.01.01)											
CROSS REFERENCE(S)		G	0	6	F	17 / 30 (2006.01.01)											
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)																
707	793	802															

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

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	17	17	33	33										
2	2	18	18	34	34										
3	3	19	19	35	35										
4	4	20	20	36	36										
5	5	21	21	37	37										
6	6	22	22	38	38										
7	7	23	23	39	39										
8	8	24	24												
9	9	25	25												
10	10	26	26												
11	11	27	27												
12	12	28	28												
13	13	29	29												
14	14	30	30												
15	15	31	31												
16	16	32	32												

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	39	
/JAY A MORRISON/ Primary Examiner.Art Unit 2168	4/7/2011	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	1

EXAMINER'S AMENDMENTS

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

1. (Currently Amended) A computerized system for manipulating data sets, comprising:

a processor; and

a data repository, the data repository being adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, comprising:

an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification;

an element relation module configured to store one or more relationships between the elements in the element module;

a class module configured to define at least one class of the elements and store the class;

an attribute module configured to define one or more attributes and store the attributes;

a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class; [[and]]

ServiceNow Ex. 1002 Page 16 of 410

a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements;

a state machine module configured to store one or more state machine types associated with at least one of the elements; and

a status module configured to store one or more statuses of each state machine.

2. (Currently Amended) The computerized system of claim 1, wherein the data repository further comprises:

~~a state machine module, the state machine module including~~ a state machine transition module ~~and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and~~ configured to store a transition from a first status to a second status ~~of the one element~~.

3. (Currently Amended) The computerized system of claim 2, wherein the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status ~~of the one element~~.

ServiceNow Ex. 1002 Page 17 of 410

4. (Original) The computerized system of claim 1, wherein the type definition module defines a type including at least one of a boolean type, a picklist type, and a state machine type.

5. (Original) The computerized system of claim 4, wherein the data repository further comprises:

a list option module configured to enumerate one or more picklist attributes of the picklist type.

6. (Original) The computerized system of claim 2, wherein the data repository further comprises:

a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements.

7. (Original) The computerized system of claim 6, wherein the data repository further comprises:

a tuple module configured to generate and store one or more tuples, wherein each of the tuples associates a subset of the elements in accordance to one of the types associated with the subset of the elements.

8. (Original) The computerized system of claim 1, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

ServiceNow Ex. 1002 Page 18 of 410

9. (Original) The computerized system of claim 8, wherein the data repository further comprises:

an element attribute module for each of the at least one class, the element attribute module configured to define and store the attributes of the elements in the class, wherein the element attribute module is identified using the name of the class.

10. (Original) The computerized system of claim 9, wherein the data repository further comprises:

an element history module configured to store one or more indicators of modification to the elements.

11. (Original) The computerized system of claim 10, wherein the data repository further comprises:

an element document module configured to store one or more formatting data for grouping and displaying a subset of the elements.

12. (Original) The computerized system of claim 1, wherein the processor is configured to generate a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

13. (Original) The computerized system of claim 1, wherein the processor is configured to generate a dynamic user-interface for accessing and manipulating the

ServiceNow Ex. 1002 Page 19 of 410

elements based on the data in the data sets and the layers of metadata stored in the data repository.

14. (Currently Amended) A method for manipulating data sets, comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; [[and]]

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository;

defining one or more state machines and defining and storing a type for each state machine associated with at least one of the elements; and

defining one or more statuses for each state machine and storing the statuses in the data repository.

ServiceNow Ex. 1002 Page 20 of 410

15. (Currently Amended) The method of claim 14, further comprising:

~~defining a state machine type, the state machine type storing a first status of one element and~~ defining and storing a transition from a first status to a second status in the data repository ~~of the one element, and storing the state machine type in the data repository~~.

16. (Currently Amended) The method of claim 15, wherein the type for each state machine [[type]] comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status ~~of the one element~~.

17. (Currently Amended) The method of claim 14, wherein the ~~type comprises~~ one or more types of the class comprise a type including at least one of a boolean type, a picklist type, and a state machine type.

18. (Original) The method of claim 17, further comprising:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

19. (Original) The method of claim 15, further comprising:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

ServiceNow Ex. 1002 Page 21 of 410

20. (Original) The method of claim 19, further comprising:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

21. (Original) The method of claim 14, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

22. (Original) The method of claim 21, further comprising:

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

23. (Original) The method of claim 22, further comprising:

storing, in the data repository, one or more indicators of modification to the elements.

24. (Original) The method of claim 23, further comprising:

storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

25. (Original) The method of claim 14, further comprising:

ServiceNow Ex. 1002 Page 22 of 410

generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

26. (Original) The method of claim 14, further comprising:

generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

27. (Currently Amended) A non-transitory computer-readable medium storing a computer-executable program which, when executed by a processor performs a method comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; [[and]]

ServiceNow Ex. 1002 Page 23 of 410

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository;

defining one or more state machines and defining and storing a type for each state machine associated with at least one of the elements; and

defining one or more statuses for each state machine and storing the statuses in the data repository.

28. (Currently Amended) The computer-readable medium of claim 27, wherein the method further comprises:

~~defining a state machine type, the state machine type storing a first status of one element and~~ defining and storing a transition from a first status to a second status in the data repository ~~of the one element, and storing the state machine type in the data repository~~.

29. (Currently Amended) The computer-readable medium of claim 28, wherein the type for each state machine [[type]] comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status ~~of the one element~~.

30. (Currently Amended) The computer-readable medium of claim 27, wherein the ~~type comprises~~ one or more types of the class comprise a type including at least one of a boolean type, a picklist type, and a state machine type.

ServiceNow Ex. 1002 Page 24 of 410

31. (Original) The computer-readable medium of claim 30, wherein the method further comprises:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

32. (Original) The computer-readable medium of claim 28, wherein the method further comprises:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

33. (Original) The computer-readable medium of claim 32, wherein the method further comprises:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

34. (Original) The computer-readable medium of claim 27, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

35. (Original) The computer-readable medium of claim 34, wherein the method further comprises:

ServiceNow Ex. 1002 Page 25 of 410

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

36. (Original) The computer-readable medium of claim 35, wherein the method further comprises:

storing, in the data repository, one or more indicators of modification to the elements.

37. (Original) The computer-readable medium of claim 36, wherein the method further comprises:

storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

38. (Original) The computer-readable medium of claim 27, wherein the method further comprises:

generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

39. (Original) The computer-readable medium of claim 27, wherein the method further comprises:

ServiceNow Ex. 1002 Page 26 of 410

generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

ServiceNow Ex. 1002 Page 27 of 410

Google scholar database rdbms inherit attribute metadata Search Advanced Scholar Search

Scholar Articles and patents anytime include citations | Create email alert Results 1 - 10 of about 2,460. (0.12 sec)

Method and system for representing and accessing object-oriented data in a relational **database** system

U Srinivasan, H Madan... - US Patent 6,587,856, 2003 - Google Patents

... CAN HAVE „ Attnbute_TyP« NAME-'STATE- ONLY ONE STATE m, **Attribute**.^ TIMESTAMP-PARAMETER=DATE SYNTAX; ONLY USER NAME ... PUBLICATIONS Ozsoyoglu, Gultekin et al., "Query Processing Techniques in the Summary-Table-by-Example **Database** Query Lan ...

Cited by 50 - Related articles - All 2 versions

The Hybrid Object-Relational Architecture (HORA): an integration of object-oriented and relational technology [PDF] from psu.edu

J Sutherland, M Pope... - ... on Applied computing: states of the ..., 1993 - portal.acm.org

... implementation of multiple **inheritance**. The EOIIA Metaclasa When the Object Manager receives a request to open a new **database**, it open a **database** in the **RDBMS** and constmcts the eight tables shown in Figure 2. Initially, there are no data in the rows of these tablea. ...

Cited by 42 - Related articles - All 5 versions

[PDF] **Metadata** for object-relational data warehouse [PDF] from psu.edu

TN Huynh, O Mangisengi... - Proc. of DMDW, 2000 - Citeseer

... interfaces obtain requirement data and deliver the corresponding object data from the O-**RDBMS** to the ... [CoBA99] C Connolly, C. Begg, and A. Strachan, **Database** Systems, A ... Chen, NM Mattos, BT Tran, et al: Implementation of SQL3 Structured Types with **Inheritance** and Value ...

Cited by 28 - Related articles - View as HTML - All 6 versions

Automatically generated objects within extensible object frameworks and links to enterprise resources

RG Goodwin, MA Farrar, M Messina... - US Patent 6,199,195, 2001 - Google Patents

... DATADIRECT includes support for object **data- base** connectivity (ODBC), Java **Database** Connectivity (JDBC ... H++ also supports dynamic Structured Query Lan- guage (SQL), asynchronous **database** access, and ... Thus the patient class can **inherit** certain **attributes**, such as name ...

Cited by 219 - Related articles - All 2 versions

Integrated three-tier application framework with automated class and table generation

B Skinner, AK Turk, K McDonnell, C Yang... - US Patent ..., 2000 - Google Patents

... Connection 214 between application server 210 and **data- base** server 212 represents the transmission ... Further, methods and **attributes** are **inherited** from framework superclasses that confer the ... a flow diagram of a process for generating an **attribute database** declaration from ...

Cited by 156 - Related articles - All 2 versions

Frame-based knowledge representation system and methods

RB Atman, NF Abernethy - US Patent 6,442,566, 2002 - Google Patents

... such as classes and instances, are organized with their associated **attributes** into frames. ... The **database** also includes tables for security definitions, logging, and other features ... Record Thing Atom Physical-Thing Atom Thing Base-Conservation **Data Base**-Conservation Measured ...

Cited by 28 - Related articles - All 3 versions

Measuring the contributions of (O) **RDBMS** to object-oriented software development

WP Zhang... - **Database** Engineering and Applications ..., 2000 - ieeexplore.ieee.org

... deals with the support of software dcvelopment processes by object-rela- tional **database** technology. ... Section 2), the mapping layer has to explicitly control the semantics of **inheritance** rela- tionships ... navigation- al OOPL and the set-oriented query language of (0)**RDBMS** and to ...

Cited by 4 - Related articles - All 7 versions

Context representation using hierarchical method and heuristic based context selection in context aware computing

B Vanathi... - Proceedings of the 1st Amrita ACM-W ..., 2010 - portal.acm.org

... It supports only exact matching and no **inheritance**. ... Table 2. Comparision of **rdbms** and ordbms Necessary Features Relational Approach Ontology Approach Object Relational (ordbms ... Using the rules, relevant context from **database** are forwarded to the reasoning component. ...

Cited by 5 - Related articles

Bridging XML-schema and relational databases: a system for generating and manipulating relational databases using valid XML documents [PDF] from psu.edu

I Varlamis... - ... of the 2001 ACM Symposium on ..., 2001 - portal.acm.org

... 2.2.1 The **database** The system was developed over the Oracle 8i **RDBMS** and the Oracle ... the **database** contents based on the logical structure of information rather than the **database** schema. ... DVD" are declared as extensions of the abstract entity "Medium" so they **inherit** all the ...

Cited by 94 - Related articles - All 15 versions

[PDF] A concrete UML-based graphical transformation syntax: The UML to **RDBMS** example in UMLX [PDF] from psu.edu

E Willink - Metamodelling for MDA, 2003 - Citeseer

... are to be suppressed, a work-list of names visible lower in the **inheritance** hierarchy must ... After the **DataBase** has been created, it may be op- timised by NormaliseDataBase. ... Preservation The UML to **RDBMS** example is a total model translation, and so it makes extensive use of ...

Cited by 13 - Related articles - View as HTML - All 6 versions

Create email alert

ServiceNow Ex. 1002 Page 29 of 410

database rdbms inherit attribute met Search

Go to Google Home - About Google - About Google Scholar

©2011 Google

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. 6234

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/917,410	12/13/2007	707	2168	09801.0002

APPLICANTS

Bruce Randall DeWoolfson, Potomac Falls, VA;

**** CONTINUING DATA ***************************

This application is a 371 of PCT/US2006/023242 06/14/2006
which claims benefit of 60/690,147 06/14/2005

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

09/02/2009

Foreign Priority claimed	35 USC 119(a-d) conditions met	Verified and Acknowledged	Met after Allowance	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
☑ Yes ☐ No	☑ Yes ☐ No	/JAY A MORRISON/ Examiner's Signature	☑ /JM/ Initials	VA	19	39	3

ADDRESS

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER
LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413
UNITED STATES

TITLE

Database Data Dictionary

FILING FEE RECEIVED		
1880	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	193	metadata with class same (element or attribute) with (relation or relationship or association)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:24
L2	103	metadata with class same (element or attribute) with (relation or relationship or association) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:24
L3	5	metadata with class same (element or attribute) with (relation or relationship or association) and 707/790,793,802.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:30
L4	3	metadata with class same (element or attribute) with (relation or relationship or association) with (inherit or inheritance)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:38
L5	3	metadata with class same (element or attribute) with (relation or relationship or association) with (inherit or inheritance) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:39
L6	0	metadata with class same (element or attribute) with (relation or relationship or association) with (inherit or inheritance) and 707/790,793,802.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:39
L7	26	metadata with class same (attribute) near3 (inherit or inheritance)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:43

ServiceNow Ex. 1002 Page 32 of 410

L8	0	metadata with class same (attribute) near3 (inherit or inheritance) with overrid$3	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:48
L9	0	metadata with class same (attribute) near3 (inherit or inheritance) with overrid$5	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:49
L10	0	metadata with class same (attribute) with (inherit or inheritance) with overrid$5	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:49
L11	0	metadata with class same (attribute) with (inherit or inheritance) with overrid$5 and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:49
L12	0	metadata with class same (attribute) with (inherit or inheritance) with overrid$5 and 707/790,793,802.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:49
L13	0	metadata same class same (attribute) with (inherit or inheritance) with overrid$5	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:49
L14	9	metadata with class same (attribute) near3 (inherit or inheritance) and state adj machine	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:49
L15	6	metadata with class same (attribute) near3 (inherit or inheritance) and state adj machine and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:50

ServiceNow Ex. 1002 Page 33 of 410

L16	0	metadata with class same (attribute) near3 (inherit or inheritance) and state adj machine and 707/790,793,802.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:50
L17	0	metadata with class same (attribute) near3 (inherit or inheritance) same state adj machine	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:50
L18	0	metadata with class same (attribute) near3 (inherit or inheritance) same state adj machine and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:50
L19	0	metadata with class same (attribute) near3 (inherit or inheritance) same state adj machine and 707/790,793,802.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:50
L20	1	(database or dbms or rdbms) with class same (attribute) with (inherit or inheritance) with overrid$5	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:53
L21	1	(database or dbms or rdbms) same class same (attribute) with (inherit or inheritance) with overrid$5	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:53
L22	1	(database or dbms or rdbms) same class same (attribute) with (inherit or inheritance) with overrid$5 and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:53
L23	0	(database or dbms or rdbms) same class same (attribute) with (inherit or inheritance) with overrid$5 and 707/790,793,802.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:53

ServiceNow Ex. 1002 Page 34 of 410

L24	4	(database or dbms or rdbms) same (object or class) same (attribute) with (inherit or inheritance) with overrid $5	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:53
L25	33	(database or dbms or rdbms) same (object or class) same (attribute) with (association or relation or relationship or reference) with (inherit or inheritance)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:55
L26	23	(database or dbms or rdbms) same (object or class) same (attribute) with (association or relation or relationship or reference) with (inherit or inheritance) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:55
L27	0	(database or dbms or rdbms) same (object or class) same (attribute) with (association or relation or relationship or reference) with (inherit or inheritance) and 707/790,793,802.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:57
L28	1	(database or dbms or rdbms) same (object or class) same (attribute) with (association or relation or relationship or reference) with (inherit or inheritance) and state adj machine	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:58
L29	0	(database or dbms or rdbms) same (object or class) same (attribute) with (association or relation or relationship or reference) with (inherit or inheritance) and state adj machine and 707/790,793,802.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2011/04/07 16:58

EAST Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp

ServiceNow Ex. 1002 Page 35 of 410

L30	0	metadata with class same (element or attribute) with (relation or relationship or association) with (inherit or inheritance) and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 16:59
L31	0	metadata with class same (attribute) with (inherit or inheritance) with overrid $5 and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 16:59
L32	0	metadata with class same (attribute) with (inherit or inheritance) same overrid $5 and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 16:59
L33	0	metadata with class same (attribute) near3 (inherit or inheritance) and state adj machine and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 16:59
L34	0	metadata with class same (attribute) same (inherit or inheritance) and state adj machine and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 16:59
L35	0	(database or dbms or rdbms) same class same (attribute) with (inherit or inheritance) with overrid $5 and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 17:00
L36	0	(database or dbms or rdbms) same (object or class) same (attribute) with (association or relation or relationship or reference) with (inherit or inheritance) and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 17:00
L37	0	(database or dbms or rdbms) same (object or class) same (attribute) with (association or relation or relationship or reference) with (inherit or inheritance) and state adj machine and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 17:00

ServiceNow Ex. 1002 Page 36 of 410

L38	0	(database or dbms or rdbms) same (object or class) same (attribute) same (association or relation or relationship or reference) with (inherit or inheritance) and state adj machine and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 17:00
L39	0	(database or dbms or rdbms) same (object or class) same (attribute) same (association or relation or relationship or reference) same (inherit or inheritance) and state adj machine and 707/790,793,802.ccls.	USPAT; UPAD	OR	ON	2011/04/07 17:00

4/ 7/ 2011 5:01:23 PM
H:\ Workspaces\ 11917410-allowance.wsp

ServiceNow Ex. 1002 Page 37 of 410

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11917410	DEWOOLFSON, BRUCE RANDALL
	Examiner	Art Unit
	JAY A MORRISON	2168

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

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

CLAIM		DATE								
Final	Original	04/22/2010	11/07/2010	04/07/2011						
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	✓	✓	=						

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11917410	DEWOOLFSON, BRUCE RANDALL
	Examiner	Art Unit
	JAY A MORRISON	2168

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

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

CLAIM		DATE								
Final	Original	04/22/2010	11/07/2010	04/07/2011						
36	36	✓	✓	=						
37	37	✓	✓	=						
38	38	✓	✓	=						
39	39	✓	✓	=						

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11917410	DEWOOLFSON, BRUCE RANDALL
	Examiner	Art Unit
	JAY A MORRISON	2168

SEARCHED

Class	Subclass	Date	Examiner
707	790,793,802 (keyword search of subclass listed)	4/7/2011	JM

SEARCH NOTES

Search Notes	Date	Examiner
East	4/7/2011	JM
Keyword search of 707/790,793,802 plus limited text search of 707	4/7/2011	JM
NPL Search - Google Scholar	4/7/2011	JM
Inventor Search	4/7/2011	JM

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner
707	790,793,802	4/7/2011	JM

PATENT
Customer No. 22,852
Attorney Docket No. 09801.0002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	Group Art Unit: 2168
Bruce R. DeWoolfson	)	
	)	
Application No.: 11/917,410	)	Examiner: Morrison, Jay A.
	)	
Filed: December 13, 2007	)	
	)	Confirmation No.: 6234
For: DATABASE DATA DICTIONARY	)	

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

Sir:

REPLY TO OFFICE ACTION

In response to the Office Action mailed November 12, 2010, the period for response to which extends through February 12, 2011, Applicant respectfully requests reconsideration of this application in light of the following remarks.

Remarks begin on page 2.

REMARKS

As a preliminary matter, Applicant thanks the Examiner for the courtesies extended to Applicant and Applicant's representatives during the interview on December 29, 2010. During the interview, the rejections contained in the Office Action mailed on November 12, 2010 were discussed. The substance of the interview is incorporated in this response.

In the Office Action,[1] the Examiner rejected claims 1, 8, 9, 12-14, 21, 22, 25-27, 34, 35, 38, and 39 under 35 U.S.C. § 102(b) as being anticipated by "Overcoming the Complexity of Object-Oriented DBMS Metadata Management" to Habela et al. ("*Habela*"); and rejected claims 2-7, 10, 11, 15-20, 28-33, 36, and 37 under 35 U.S.C. § 103(a) as obvious over *Habela* in view of "Ontologies and Diagrams for Software and Systems Engineering: processes and objects" to Wieringa ("*Wieringa*").

I. Rejection of Claims 1, 8, 9, 12-14, 21, 22, 25-27, 34, 35, 38, and 39 under 35 U.S.C. §102(b)

Applicant respectfully traverses the rejection of claims 1, 8, 9, 12-14, 21, 22, 25-27, 34, 35, 38, and 39 under 35 U.S.C. § 102(b) as anticipated by *Habela*. In order to properly establish that *Habela* anticipates under 35 U.S.C. § 102, each and every element of each of the claims in issue must be found, either expressly described or under principles of inherency, in that single reference. Furthermore, "[t]he identical invention must be shown in as complete detail as is contained in the ... claim." *See*

[1] The Office Action may contain statements reflecting characterizations of the related art and the claims. Regardless of whether any such statement is identified herein, Applicant declines to automatically subscribe to any statement or characterization in the Office Action.

ServiceNow Ex. 1002 Page 42 of 410

M.P.E.P. § 2131, quoting *Richardson v. Suzuki Motor Co.*, 868 F.2d 1126, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989).

Independent claim 1 recites a system, including the following six modules:

> . . .
> a data repository . . . comprising:
> an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification;
> an element relation module configured to store one or more relationships between the elements in the element module;
> a class module configured to define at least one class of the elements and store the class;
> an attribute module configured to define one or more attributes and store the attributes;
> a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class; and
> a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements.

Habela discloses a structure that "improves database metadata flexibility" (Abstract). In general, *Habela* is directed to a metamodel to simplify metadata operations. In particular, *Habela* discloses a language for querying an object oriented metamodel, and *Habela* uses this "query language" on a non-relational object oriented model (Abstract).

Habela does not create any relational database schema. Moreover, the metamodel in *Habela* does not include the claimed six modules. As described below, while *Habela* discloses objects, relationships, and classes, *Habela* lacks the claimed six data modules or any relationship between a class attribute and a type definition.

The Examiner cites pages 221, lines 16-30 of *Habela* to allegedly disclose the claimed six data modules, namely (1) the "element module," (2) the "element relation module," (3) the "class module," (4) the "attribute module," (5) the "class attribute module," and (6) the "type definition module" (Office Action at page 3). This is not correct.

According to *Habela*, a "flattened metamodel" includes 1) "[p]redefined values of the meta-attribute 'kind' in the metaclass 'MetaObject' (e.g. 'class', 'attribute', etc.)," 2) "[p]redefined values of meta-attribute names (e.g. 'isAbstract') and of meta-attributes 'kind' of 'MetaRelationship' metaclasses (e.g. 'specialization')," and 3) "[c]onstraints defining the allowed combinations and context of those predefined elements" (page 221, lines 16-22). Further, based on the disclosure of *Habela*, the "metamodel" includes values of the "meta-attribute 'kind'" in the "MetaObject", values of the "meta-attribute names," and "meta-attributes 'kind'"of "'MetaRelationship' metaclasses." However, neither this passage nor any other passage of *Habela* teaches or suggests the claimed combination of six modules and the relationships between the modules.

For example, claim 1 requires (3) "a class module configured to define at least one class of the elements and store the class" and (4) "an attribute module configured to define one or more attributes and store the attributes." *Habela* mentions the words "class" and "attribute" as examples of a metaclass "MetaObject." Even assuming that these examples could correspond to the claimed (3) "class module" and (4) "attribute module," which Applicant does not concede, *Habela* fails to teach or suggest the combination of (5) "a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class" and (6)

ServiceNow Ex. 1002 Page 44 of 410

"a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements," as further recited in claim 1.

The Examiner cites a single "attribute" in *Habela* to allegedly disclose both the claimed (4) "attribute module" and (5) "class attribute module." However, a single attribute cannot disclose two distinct modules.

The Examiner also cites page 221, lines 20-30 of *Habela* to allegedly disclose the claimed (6) "type definition module" (Office Action at page 3), but Applicant finds no teaching or suggestion in the cited passage or any other passage of *Habela* of the claimed "type definition module."

Therefore, *Habela* does not teach or suggest either (5) "a class attribute module" or (6) "a type definition module," or a combination thereof.

Moreover, the combination of (5) "a class attribute module" and (6) "a type definition module" achieves a feature of using an attribute definition with any number of class definitions, resulting in a unique mix of reusable attributes within the resulting schema.

During the interview, Applicant presented arguments to the Examiner that the (5) "class attribute module" and the (6) "type definition module" are not defined as standard object-oriented structures. In response, the Examiner requested that Applicant "make clear where these definitions are found in the specification so that the claims can be construed based on these non-standard definitions" (Interview Summary at page 2). In response to the Examiner's request, Applicant has provided the followings passages of the specification to provide exemplary definitions and support for Applicant's arguments.

ServiceNow Ex. 1002 Page 45 of 410

The claimed "class attribute module" is a module that provides class attributes, which are definitions that can be used with zero, one, or more than one class definitions, thereby providing a unique mix of reusable attributes within a resulting schema. As stated in paragraph 035 of the specification, class_attribute table 112, which corresponds to the claimed "class attribute module," "may associate a given attribute to one or more classes (in class table 150 [which corresponds to the claimed "class module"]). *Because class_attribute table 112 allows for one attribute to be referenced by many classes, there is attribute inheritance among the classes. Attribute inheritance among the classes is a powerful notion not found in traditional relational database management systems (RDBMS).* Class_attribute table 112 stores the logical name that may be used by the attribute when presented to the user for data manipulation such as insert or update or delete of values. *The ability to reference the same attribute via multiple names depending on which class it appears in is akin to the concept of overridden inheritance, also a concept not found in RDBMS*" (emphasis added).

In addition, the claimed "type definition module" is a module that stores types of data and maintains the data language. As stated in paragraph 032 of the specification, "[u]nlike a conventional relational database, the *EMR extends the notion of available attributes types beyond the standard string, number, and date types.* To do so, the EMR allows dynamically extendable entries in a type definition table 102. Type definition table 102 [which corresponds to the claimed "type definition module"] *may store articulated types that may be, for example, an M4 layer while still maintaining the meta meta meta meta data language. By using the type of the person's name for an*

attribute, the user can declare more than what a normal relational database may store. . . . Moreover, *the EMR is capable of storing a process or state machine type which is not found in a relational database. All these data types provide flexibility in the type of data management that can be performed*" (emphasis added).

The above passages of the specification provide exemplary definitions and support for Applicant's argument that the claimed modules are not drawn to standard object-oriented structures. In addition, Applicant directs the Examiner to paragraph 054 of the specification for an exemplary disclosure of conventional RDBMS. According to conventional RDBMS, they implement their "own data model according to the user interface and the business rules specification for that system. For the conventional system, the system's developers write dedicated code and user interface code that displays or manages certain data inputs such as check boxes, lists, combo boxes or other data inputs. The conventional system's dedicated code and user interface manipulate the data that the system's application stores and retrieves during operation. Moreover, the conventional system stores and retrieves data locally on a local disk, like a C drive on a Windows system."

In contrast to the conventional systems described in paragraph 054, the claims are directed to an element-based approach where "each of the elements conforms to the rules of generalized things in the universe, and therefore may be reused or referenced to other things in the universe, e.g., through entries in element_relation table 162. Moreover, each of the elements in its own context is also meaningful and reusable" (specification at paragraph 056).

ServiceNow Ex. 1002 Page 47 of 410

The features described above (such as attribute inheritance and using an attribute definition with any number of class definitions) are not taught in the conventional model disclosed in *Habela*. According, *Habela* fails to teach or suggest the claimed "data repository" that is "adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets" and includes (1) "an element module," (2) "an element relation module," (3) "a class module," and (4) "an attribute module" in combination with (5) "a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class" and (6) "a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements," as recited in claim 1.

Accordingly, *Habela* cannot anticipate claim 1. Thus, claim 1 is allowable for at least these reasons. Claims 8, 9, 12, and 13 depend from claim 1 and are thus also allowable for at least the same reasons as claim 1.

Independent claims 14 and 27, while of different scope, recite features similar to those of claim 1 and are thus allowable over *Habela* for at least similar reasons to those discussed above in regard to claim 1. Dependent claims 21, 22, 25, 26, 34, 35, 38, and 39 are also allowable at least due to their dependence from the independent claims 14 and 27.

II. Rejection of claims 2-7, 10, 11, 15-20, 28-33, 36, and 37 under 35 U.S.C. § 103(a)

Regarding the rejection of claims 2-7, 10, 11, 15-20, 28-33, 36, and 37, which depend from independent claims 1, 14, and 27, the Examiner relies on *Wieringa* for

ServiceNow Ex. 1002 Page 48 of 410

allegedly disclosing the elements recited in these claims (Office Action at pages 5-10). Even assuming this is true, which Applicant does not concede, *Wieringa* fails to cure the deficiencies of *Habela* with respect to the independent claims.

Therefore, a *prima facie* case of obviousness has not been established for claims 2-7, 10, 11, 15-20, 28-33, 36, and 37, and the Examiner should withdraw the rejection of the claims under 35 U.S.C. § 103(a).

III. Conclusion

In view of the foregoing, Applicant respectfully requests reconsideration of this application and the timely allowance of the pending claims.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: January 21, 2011

By: Paul M. Gurzo
Paul M. Gurzo
Reg. No. 67,478

Electronic Acknowledgement Receipt

EFS ID:	9278563
Application Number:	11917410
International Application Number:	
Confirmation Number:	6234
Title of Invention:	Database Data Dictionary
First Named Inventor/Applicant Name:	Bruce Randall DeWoolfson
Customer Number:	22852
Filer:	Paul Michael Gurzo/Kimberly Slazer
Filer Authorized By:	Paul Michael Gurzo
Attorney Docket Number:	09801.0002
Receipt Date:	21-JAN-2011
Filing Date:	13-DEC-2007
Time Stamp:	12:53:30
Application Type:	U.S. National Stage under 35 USC 371

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		Reply_to_Office_Action_1-21-11.pdf	361879 0a267189bcb6a62449c3e060b5283865a4e8b7ac	yes	9

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Amendment/Req. Reconsideration-After Non-Final Reject	1	1
	Applicant Arguments/Remarks Made in an Amendment	2	9
Warnings:			
Information:			
	Total Files Size (in bytes):	361879	

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

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

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

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

UNITED STATES PATENT AND TRADEMARK OFFICE

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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/917,410	12/13/2007	Bruce Randall DeWoolfson	09801.0002	6234

22852 7590 01/03/2011

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER	
MORRISON, JAY A	

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
01/03/2011	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.

Interview Summary	Application No.	Applicant(s)	
	11/917,410	DEWOOLFSON, BRUCE RANDALL	
	Examiner	Art Unit	
	JAY A. MORRISON	2168	

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

(1) *JAY A. MORRISON*.

(2) *Bruce R. Woolfson, inventor*.

(3) *Paul Gurzo (Reg No.67478)*.

(4) *Lionel Lavenue (Reg No 46859)*.

Date of Interview: *29 December 2010*.

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*.

Identification of prior art discussed: *Habela*.

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: *Discussed claim 1 in view of the cited references. Inventor explained that the objected-oriented teachings of the Habela reference do not read on the claimed invention since the invention is not drawn to standard object-oriented structures. The applicant intended to use non-standard definitions for the normal terms such as the class and element that are used in the claims. It was noted that the applicant should make clear where these definitions are found in the application so that the claims can be construed based on these non-standard definitions*.

(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.

/Jay A Morrison/ Primary Examiner, Art Unit 2168	

Summary of Record of Interview Requirements

Manual of Patent Examining Procedure (MPEP), Section 713.04, Substance of Interview Must be Made of Record
A complete written statement as to the substance of any face-to-face, video conference, or telephone interview with regard to an application must be made of record in the application whether or not an agreement with the examiner was reached at the interview.

Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews
Paragraph (b)

In every instance where reconsideration is requested in view of an interview with an examiner, a complete written statement of the reasons presented at the interview as warranting favorable action must be filed by the applicant. An interview does not remove the necessity for reply to Office action as specified in §§ 1.111, 1.135. (35 U.S.C. 132)

37 CFR §1.2 Business to be transacted in writing.
All business with the Patent or Trademark Office should be transacted in writing. The personal attendance of applicants or their attorneys or agents at the Patent and Trademark Office is unnecessary. The action of the Patent and Trademark Office will be based exclusively on the written record in the Office. No attention will be paid to any alleged oral promise, stipulation, or understanding in relation to which there is disagreement or doubt.

The action of the Patent and Trademark Office cannot be based exclusively on the written record in the Office if that record is itself incomplete through the failure to record the substance of interviews.

It is the responsibility of the applicant or the attorney or agent to make the substance of an interview of record in the application file, unless the examiner indicates he or she will do so. It is the examiner's responsibility to see that such a record is made and to correct material inaccuracies which bear directly on the question of patentability.

Examiners must complete an Interview Summary Form for each interview held where a matter of substance has been discussed during the interview by checking the appropriate boxes and filling in the blanks. Discussions regarding only procedural matters, directed solely to restriction requirements for which interview recordation is otherwise provided for in Section 812.01 of the Manual of Patent Examining Procedure, or pointing out typographical errors or unreadable script in Office actions or the like, are excluded from the interview recordation procedures below. Where the substance of an interview is completely recorded in an Examiners Amendment, no separate Interview Summary Record is required.

The Interview Summary Form shall be given an appropriate Paper No., placed in the right hand portion of the file, and listed on the "Contents" section of the file wrapper. In a personal interview, a duplicate of the Form is given to the applicant (or attorney or agent) at the conclusion of the interview. In the case of a telephone or video-conference interview, the copy is mailed to the applicant's correspondence address either with or prior to the next official communication. If additional correspondence from the examiner is not likely before an allowance or if other circumstances dictate, the Form should be mailed promptly after the interview rather than with the next official communication.

The Form provides for recordation of the following information:
- Application Number (Series Code and Serial Number)
- Name of applicant
- Name of examiner
- Date of interview
- Type of interview (telephonic, video-conference, or personal)
- Name of participant(s) (applicant, attorney or agent, examiner, other PTO personnel, etc.)
- An indication whether or not an exhibit was shown or a demonstration conducted
- An identification of the specific prior art discussed
- An indication whether an agreement was reached and if so, a description of the general nature of the agreement (may be by attachment of a copy of amendments or claims agreed as being allowable). Note: Agreement as to allowability is tentative and does not restrict further action by the examiner to the contrary.
- The signature of the examiner who conducted the interview (if Form is not an attachment to a signed Office action)

It is desirable that the examiner orally remind the applicant of his or her obligation to record the substance of the interview of each case. It should be noted, however, that the Interview Summary Form will not normally be considered a complete and proper recordation of the interview unless it includes, or is supplemented by the applicant or the examiner to include, all of the applicable items required below concerning the substance of the interview.

A complete and proper recordation of the substance of any interview should include at least the following applicable items:

1) A brief description of the nature of any exhibit shown or any demonstration conducted,
2) an identification of the claims discussed,
3) an identification of the specific prior art discussed,
4) an identification of the principal proposed amendments of a substantive nature discussed, unless these are already described on the Interview Summary Form completed by the Examiner,
5) a brief identification of the general thrust of the principal arguments presented to the examiner,
   (The identification of arguments need not be lengthy or elaborate. A verbatim or highly detailed description of the arguments is not required. The identification of the arguments is sufficient if the general nature or thrust of the principal arguments made to the examiner can be understood in the context of the application file. Of course, the applicant may desire to emphasize and fully describe those arguments which he or she feels were or might be persuasive to the examiner.)
6) a general indication of any other pertinent matters discussed, and
7) if appropriate, the general results or outcome of the interview unless already described in the Interview Summary Form completed by the examiner.

Examiners are expected to carefully review the applicant's record of the substance of an interview. If the record is not complete and accurate, the examiner will give the applicant an extendable one month time period to correct the record.

Examiner to Check for Accuracy

If the claims are allowable for other reasons of record, the examiner should send a letter setting forth the examiner's version of the statement attributed to him or her. If the record is complete and accurate, the examiner should place the indication, "Interview Record OK" on the paper recording the substance of the interview along with the date and the examiner's initials.

RECEIVED
CENTRAL FAX CENTER
DEC 2 8 2010

Applicant Initiated Interview Request Form

Application No.: 11/917,410 Examiner: Jay Morrison
First Named Applicant: Bruce R. DeWoolfson Group Art Unit: 2168
Status of Application: Non-Final

Tentative Participants:

(1) Bruce R. (Randy) DeWoolfson (2) Lionel Lavenue (3) Paul Gurzo

Proposed Date of Interview: December 29, 2010 **Proposed Time:** 3:00 p.m.

Type of Interview Requested:
- ☐ Telephonic
- ☒ Personal
- ☐ Video Conference

Exhibit to be Shown or Demonstrated?
- ☐ Yes
- ☒ No

Issues to be Discussed

Issues (Rej., Obj. etc.)	Claims/Fig. #s	Prior Art	Discussed	Agreed	Not Agreed
1. Rej (102)	Cl. 1, 8, 9, 12-14, 21, 22, 25-27, 34, 35, 38, and 39	*Habela*	☐	☐	☐
3. Rej (103)	Cl. 2-7, 10, 11, 15-20, 23, 24, 28-33, 36, and 37	*Habela* and *Wieringa*	☐	☐	☐

Brief Description of Arguments to be Presented:

1. Regarding the 102 rejection, Applicant will present at least the following arguments: (1) that *Habela* does not teach or suggest the claimed six modules disclosed in independent claim 1; (2) that a single attribute cannot disclose two distinct modules, for example, the citation of a single "attribute" in *Habela* (Office Action at page 3) to allegedly disclose both the claimed "attribute module" and "class attribute module" in independent claim 1; and (3) that there is no teaching in *Habela* of the claimed "type definition module" of independent claim 1. Therefore, *Habela* fails to anticipate claim 1.

2. Regarding the 103 rejection, Applicant will present arguments that *Wieringa* does not cure the deficiencies of *Habela* with respect to independent claims 1, 14, or 27.

Note: This form should be completed by applicant and submitted to the Examiner in advance of the interview (see MPEP § 713.01). This application will not be delayed from issue because of applicants' failure to file a statement of the substance of this interview (37 CFR § 1.133(b)) as soon as possible.

Paul M. Gurzo

Paul M. Gurzo, Reg. No. 67,478 Examiner/SPE Signature

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.
11/917,410	12/13/2007	Bruce Randall DeWoolfson	09801.0002	6234

22852 7590 11/12/2010

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER	
MORRISON, JAY A	

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
11/12/2010	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.

Office Action Summary	Application No.	Applicant(s)	
	11/917,410	DEWOOLFSON, BRUCE RANDALL	
	Examiner	Art Unit	
	JAY A. MORRISON	2168	

-- 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 *25 August 2010*.

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) *1-39* 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) *1-39* is/are rejected.

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

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

Application Papers

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

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

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

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

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

Priority under 35 U.S.C. § 119

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

a)☐ All b)☐ Some * c)☐ None of:

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

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

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

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

Attachment(s)

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

DETAILED ACTION

Remarks

1. Claims 1-39 are pending.

Claim Rejections - 35 USC § 102

2. 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.

3. Claims 1, 8-9, 12-14, 21-22, 25-27, 34-35 and 38-39 are rejected under 35 U.S.C. 102(b) as being anticipated by Habela et al. ('Habela' hereinafter) ("Overcoming the Complexity of Object-Oriented DBMS Metadata Management", Proc of the 9th International Conference Object-Oriented Information System (OOIS 2003), Geneva, Switzerland, September 2nd, 2003).

As per claim 1, Habela teaches

A computerized system for manipulating data sets, comprising: (see abstract)

a processor; and a data repository, the data repository being adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, comprising: (query language and database metamodel, abstract)

an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification; an element relation module configured to store one or more relationships between the elements in the element module; (MetaAttribute, page 221, lines 16-20)

a class module configured to define at least one class of the elements and store the class; (class, page 221, lines 16-20; figure 4)

an attribute module configured to define one or more attributes and store the attributes; a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class; (attribute, page 221, lines 16-20; figure 4)

and a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements. (page 221, lines 20-30)

As per claim 8, Habela teaches

each of the class-attribute associations is identified by a logical name and a name of the class. (figure 4, relationships)

As per claim 9, Habela teaches

the data repository further comprises: an element attribute module for each of the at least one class, the element attribute module configured to define and store the attributes of the elements in the class, wherein the element attribute module is identified using the name of the class. (figure 4, named class)

As per claim 12,

the processor is configured to generate a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository. (page 216, lines 20-25)

As per claim 13,

the processor is configured to generate a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository. (page 216, lines 20-25)

As per claims 14, 21-22 and 25-26,

These claims are rejected on grounds corresponding to the arguments given above for rejected claims 1, 8-9 and 12-13, respectively, and are similarly rejected.

As per claims 27, 34-35 and 38-39,

These claims are rejected on grounds corresponding to the arguments given above for rejected claims 1, 8-9 and 12-13 and are similarly rejected.

Claim Rejections - 35 USC § 103

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

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

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

5. Claims 2-7, 10-11, 15-20, 23-24, 28-33 and 36-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Habela et al. ('Habela' hereinafter) ("Overcoming the Complexity of Object-Oriented DBMS Metadata Management", Proc of the 9th International Conference Object-Oriented Information System (OOIS 2003), Geneva, Switzerland, September 2nd, 2003) in view of Wieringa ("Ontologies and Diagrams for

Software and Systems Engineering: processes and objects", by Roel Wieringa, January 23, 2000).

As per claim 2,

Habela does not explicitly indicate “the data repository further comprises: a state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element”.

However, Wieringa discloses “the data repository further comprises: a state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element” (section 4.3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Habela and Wieringa because using the steps of “the data repository further comprises: a state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element” would have given those skilled in the art the tools to improve the invention by merging ideas from structured and object-oriented analysis to get a blend that fits well

into a systems engineering approach. This gives the user the advantage of being able to better understand software design issues.

As per claim 3,

Habela does not explicitly indicate "the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element".

However, Wieringa discloses "the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element" (section 4.3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Habela and Wieringa because using the steps of "the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element" would have given those skilled in the art the tools to improve the invention by merging ideas from structured and object-oriented analysis to get a blend that fits well into a systems engineering approach. This gives the user the advantage of being able to better understand software design issues.

As per claim 4,

Neither Habela nor Wieringa expressly show "the type definition module defines a type including at least one of a boolean type, a picklist type, and a state machine type"

However these differences are only found in the nonfunctional descriptive material and are not functionally involved in the steps recited. The definition of a type would be performed the same regardless of the data. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, see In re Gulack, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); In re Lowry, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to select a type having any type of content because such data does not functionally relate to the steps in the system claimed and because the subjective interpretation of the data does not patentably distinguish the claimed invention.

As per claim 5, Habela teaches

the data repository further comprises: a list option module configured to enumerate one or more picklist attributes of the picklist type. (page 223, lines 10-15)

As per claim 6, Habela teaches

the data repository further comprises: a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements. (page 223, lines 14-20)

As per claim 7, Habela teaches

the data repository further comprises: a tuple module configured to generate and store one or more tuples, wherein each of the tuples associates a subset of the elements in accordance to one of the types associated with the subset of the elements. (figure 4, MetaRelationship)

As per claim 10,

Habela does not explicitly indicate "the data repository further comprises: an element history module configured to store one or more indicators of modification to the elements".

However, Wieringa discloses "the data repository further comprises: an element history module configured to store one or more indicators of modification to the elements" (section 4.3).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine Habela and Wieringa because using the steps of "the data repository further comprises: an element history module configured to store one or more indicators of modification to the elements" would have given those skilled in the art the tools to improve the invention by merging ideas from structured and object-oriented analysis to get a blend that fits well into a systems engineering approach. This gives the user the advantage of being able to better understand software design issues.

As per claim 11, Habela teaches

the data repository further comprises: an element document module configured to store one or more formatting data for grouping and displaying a subset of the elements. (page 224, lines 18-22)

As per claims 15-20 and 23-24,

These claims are rejected on grounds corresponding to the arguments given above for rejected claims 2-7 and 10-11, respectively, and are similarly rejected.

As per claims 28-33 and 36-37,

These claims are rejected on grounds corresponding to the arguments given above for rejected claims 2-7 and 10-11, respectively, and are similarly rejected.

Response to Arguments

6. Applicant's arguments, see pages 13-14, filed 8/25/2010, with respect to the 35 USC 101 rejections of claims 27-39 have been fully considered and are persuasive. The 35 USC 101 rejections of claims 27-39 have been withdrawn.

7. Applicant's arguments filed 8/25/2010 regarding claims 1 and 5-6 have been fully considered but they are not persuasive.

Regarding claim 1, Applicant argues that Habela does not disclose “a class attribute mobule configured to define and store one or more class-attribute associations between at least one of the attributes and the class" or “a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationship between the elements”. Respectfully, it is noted that Habela discloses using metadata to describe object structures including associations between the various parts of the object like classes and attributes (section 4.2). It is submitted that the claimed elements such as classes, elements, attributes, and the various associations which exist are simply common elements of object-oriented programming, and that using metadata to represent these relationships is taught by Habela. Therefore Habela discloses the limitation.

Regarding claim 5, Applicant argues that Habela does not disclose a data repository that includes a list option module configured to enumerate one or more picklist attributes of the picklist type. Respectfully, it is noted that Habela discloses predefined values of a kind (section 4.2), which enumerates an attribute. Enumeration is simply identifying values in a set, and it is submitted that this common feature of object-oriented programming is taught by Habela.

Regarding claim 6, Applicant argues that Habela does not disclose a data repository that includes a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements. Respectfully, it is

noted that Habela discloses textual representations of elements within its definition (figure 4), and since the query, control, and display mechanisms have no claimed function, the naming of Habela reads on this limitation.

8. Applicant's arguments with respect to claims 2-3 and 10 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

9. The prior art made of record, listed on form PTO-892, and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jay A. Morrison whose telephone number is (571) 272-7112. The examiner can normally be reached on M-F 8-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached on (571) 272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

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

/Jay A Morrison/
Primary Examiner, Art Unit 2168

Jay Morrison
TC2100

Notice of References Cited	Application/Control No. 11/917,410	Applicant(s)/Patent Under Reexamination DEWOOLFSON, BRUCE RANDAL	
	Examiner JAY A. MORRISON	Art Unit 2168	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	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	"Ontologies and Diagrams for Software and Systems Engineering: processes and objects", by Roel Wieringa, January 23, 2000.
	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.

Google scholar transition restriction "state machine" Search Advanced Scholar Search

O Search only in Engineering, Computer Science, and Mathematics.

O Search in all subject areas.

Scholar Articles excluding pate - 2005 include citations Create email alert Results **1** - **10** of about **8,580**. (**0.33** sec)

The algebra of security

J McLean - Security and Privacy, 1988. Proceedings., 1988 ..., 2002 - ieeexplore.ieee.org

... The system C(V,R,T,vinit) is the **state machine** consisting of states V, requests R, and **transition** function T that starts in initial state winit ... Second, we could state our **transition restriction** as a state **restriction**, eg, as the **restriction** that s can have write access to the security level of a ...

Cited by 63 - Related articles - All 2 versions

Synthesis of asynchronous state machines using a local clock

SM Nowick... - ... VLSI in Computers and Processors, 1991 ..., 2002 - ieeexplore.ieee.org

... a distinguished start state (initial wire values are either specified or as- sumed 0). Each **transition** is labelled ... set of input values; that is, each state has a unique entry point (this **restriction** simplifies minimization). ... Figure 2: Detailed block diagram of asynchronous **state machine**. ...

Cited by 93 - Related articles - All 2 versions

A DNA and **restriction** enzyme implementation of Turing machines [PDF] from psu.edu

PWK Rothemund - DNA based computers, 1996 - books.google.com

... the **transition** table. Even the implementation of the smallest UTM known—a 4 symbol, 7 **state machine** [21]—would require 24 different **restriction** enzymes and many multiple digests to effect each **transition**. In the Section 4.1 ...

Cited by 124 - Related articles - All 10 versions

A formal semantics of UML Statecharts by model **transition** systems [PDF] from psu.edu

D Varró - Graph Transformation, 2002 - Springer

... The UML semantics assumes a dispatcher, which selects one event at a time from the environment and offers it to the **state machine**. ... The original source state(s) of **transition** is denoted by the source **restriction** relation (sourceRest). ...

Cited by 77 - Related articles - BL Direct - All 17 versions

Automatic synthesis of locally-clocked asynchronous state machines

SM Nowick... - ... -Aided Design, 1991. ICCAD-91. Digest ..., 2002 - ieeexplore.ieee.org
... **restriction** simplifies " zation) . Given a **transition** r in some state s, we define srari(r) as the input value at the entry point of s, and cube(r) as the set of all input values that can occur during r. An example of a simple finite-state specification is shown in figure 1. 2.2 **State Machine** ...
Cited by 108 - Related articles

The specification and modeling of computer security [PDF] from osu.edu
J McLean - COMPUTER,, 1990 - computer.org
... A request r is issued in state v and moves the system from v to its successor. A system Z(V, T, vlnl,) is the finite **state machine** consisting of states V, **transition** function T, and initial state v ml,. ... It seems to offer a set of **transition restrictions** ade- quate for explicating access security. ...
Cited by 124 - Related articles - Library Search - All 36 versions

Analysis of modularly composed nets by siphons
MD Jeng... - Systems, Man and Cybernetics, Part A: ..., 2002 - ieeexplore.ieee.org
... **machine** component remains constant whatever **transition** firings. Hence, Property 4: An RCN-merged net is conservative and structurally bounded. In the following, additional **restrictions** concerning the merging of RCN's are considered. **Restriction** 1: At each common **transition**, ...
Cited by 50 - Related articles - BL Direct - All 3 versions

Transition network grammars for natural language analysis [PDF] from usu.edu
WA Woods - Communications of the ACM, 1970 - portal.acm.org
... arcs connecting them, where the nodes correspond to states in a finite **state machine** and the ... has no "natural" mechanism, there is no problem extending the augmented **transition** network model to ... This requires only the relaxation of the **restrictions** on the types of conditions and ...
Cited by 1215 - Related articles - Library Search - All 17 versions

Unifying synchronous/asynchronous **state machine** synthesis [PDF] from osu.edu
KY Yun... - Computer-Aided Design, 1993. ICCAD-93. ..., 2002 - ieeexplore.ieee.org
Unifying Synchronous/Asynchronous **State Machine** Synthesis * ... In this section, we describe the extended-burst-mode specification as defined in [21] and propose lifting some **restrictions** we imposed in ... from 1 to 0 (ifs initially 1) or remains 0 (ifs initially 0). A signal **transition** s+ or ...
Cited by 38 - Related articles - BL Direct - All 6 versions

Synthesis of 3D asynchronous state machines [PDF] from osu.edu
KY Yun, DL Dill... - Computer Design: VLSI in ..., 2002 - ieeexplore.ieee.org
... The operation of the 3D **state machine** is similar to a Mealy- mode synchronous **state machine**. ... The burst-mode **restriction** allows us to generate the table so that there are no function ... during the input burst - outputs are specified to remain unchanged until the last **transition** of the ...
Cited by 155 - Related articles - BL Direct - All 13 versions

Create email alert

Gooooooooogle

Result Page: 1 2 3 4 5 6 7 8 9 10 Next

transition restriction "state machine" Search

Go to Google Home - About Google - About Google Scholar

©2010 Google

Google scholar transition "state machine" object-oriented Search Advanced Scholar Search

O Search only in Engineering, Computer Science, and Mathematics.

O Search in all subject areas.

Real-time **object-oriented** modeling (room)

B Selic - rtas, 1996 - computer.org

... An important characteristic of ROOM is that it is an **object-oriented** language. ... This represents an "event" which can trigger a state **transition** in the receiving behavior's **state machine**. The **transition** can have an action function attached. ...

Cited by 1462 - Related articles - Library Search - All 8 versions

Introducing objectcharts or how to use statecharts in **object-oriented** design

D Coleman, F Hayes... - Software Engineering, IEEE ..., 2002 - ieeexplore.ieee.org

... figuration diagrams are similar to the Object Communication Model used in Shlaer and Mellor's **object-oriented** systems analysis ... introduces the services provided and required by the class, and defines-by the underlying **state machine** and **transition** specifications-which ...

Cited by 274 - Related articles - All 7 versions

Weaving executability into **object-oriented** meta-languages [PDF] from psu.edu

PA Muller, F Fleurey... - Model Driven Engineering ..., 2005 - Springer

... Page 10. Weaving Executability into **Object-Oriented** Meta-languages 273 3.6 The Result: KerMeta ... Fig. 10. A simple **state machine** to translate "hello" into "world" ... A **transition** contains an input character and an output character and it refers both to a source and to a target state. ...

Cited by 257 - Related articles - BL Direct - All 14 versions

[BOOK] Testing **object-oriented** systems: models, patterns, and tools

R Binder - 2000 - books.google.com

... 175 7.1 Motivation 176 7.2 The Basic Model 177 7.2.1 What Is a **State Machine**? ... 7.2.4 Guarded Transitions 183 7.2.5 Mealy and Moore 185 7.2.6 State **Transition** Tables 189 ... the Basic Model 193 7.2.8 Statecharts 194 7.2.9 State Machines and **Object-oriented** Development 201 ...

Cited by 1252 - Related articles - Library Search - All 8 versions

Automatic extraction of **object-oriented** component interfaces [PDF] from gmu.edu

J Whaley, MC Martin... - ACM SIGSOFT Software ..., 2002 - portal.acm.org
... We can represen tthese constraints as a finite **state machine** (FSM), as shown in Figure 1 ... in this area, we are interested in techniques that ap- ply to large **object-oriented** component-level ... Instances of the same class may all follow the same state **transition** model, but they can be ...
Cited by 268 - Related articles - All 11 versions

Testing of **object-oriented** programs based on finite state machines [PDF] from osu.edu
HS Hong, YR Kwon... - apsec, 1995 - computer.org
... states of it11 objwt and member functions represent a change of states, ie, a **transition** of FSMs. Therefore, FSMs are often used as a specification of classes in most **object-oriented** anal- ysis and design ... We propose class **state machine** (CSM) for automatic test cases generation ...
Cited by 55 - Related articles - All 22 versions

On object state testing
D Kung, N Suchak, J Gao, P Hsia... - Computer Software ..., 2002 - ieeexplore.ieee.org
... and phraseus: state testing, **object- oriented** program testing, software testing, symbolic execution. ... The notions of state and **transition** in OSD are aesociated with s ecific programming con ... Chow's method [2], where test casea are generated from a single **state machine**; and hence ...
Cited by 106 - Related articles - All 8 versions

Reengineering procedural into **object-oriented** systems
P Newcomb... - Reverse Engineering, 1995., ..., 2002 - ieeexplore.ieee.org
... Figure 5 shows a **state machine** and state **transition** table that has been derived from an **object oriented** model.In the: upper left quadrant is a **State Machine** Model. In the lower left quadrant is a Source Code display for a subroutine. The State **Transition** Table is in ...
Cited by 81 - Related articles - All 5 versions

Generating tests from UML specifications [PDF] from osu.edu
J Offutt... - «UML»'99—The Unified Modeling Language, 1999 - Springer
... A constraint is a restriction on one or more values within an **object-oriented** model or system. ... State **Transition**: Logical ----- **State Machine** object State /* StartState, Normal, EndState */ State **Transition State Machine** Object State /* Normal */ ...
Cited by 362 - Related articles - BL Direct - All 24 versions

Transformation of data flow analysis models to **object oriented** design
B Alabiso - Conference proceedings on **Object-oriented** ..., 1988 - portal.acm.org
... full/close inlet FIgure 4 - State **TransItIon** Diagram for Control Tank 2 TW ORJFCT OWNTED DBIGE(MODE& The **Object Oriented** Design Model used in this paper is based on the Smalltalk-80'D" [GR80] model, with two fundamental differences: ...
Cited by 96 - Related articles - All 2 versions

ServiceNow Ex. 1002 Page 75 of 410

Create email alert

Gooooooooogle

Result Page: 1 2 3 4 5 6 7 8 9 10 Next

transition "state machine" object-orie Search

Go to Google Home - About Google - About Google Scholar

©2010 Google

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11917410	DEWOOLFSON, BRUCE RANDALL
	Examiner	Art Unit
	JAY A MORRISON	2168

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

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

CLAIM		DATE								
Final	Original	04/22/2010	11/07/2010							
	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	✓	✓							

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11917410	DEWOOLFSON, BRUCE RANDALL
	Examiner	Art Unit
	JAY A MORRISON	2168

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

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

CLAIM		DATE								
Final	Original	04/22/2010	11/07/2010							
	36	✓	✓							
	37	✓	✓							
	38	✓	✓							
	39	✓	✓							

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11917410	DEWOOLFSON, BRUCE RANDALL
	Examiner	Art Unit
	JAY A MORRISON	2168

SEARCHED			
Class	Subclass	Date	Examiner
707	999.102 (keyword search of subclass listed)	11/8/2010	JM

SEARCH NOTES		
Search Notes	Date	Examiner
East	11/8/2010	JM
Keyword search of 707 plus limited text search of 707/999.102	11/8/2010	JM
NPL Search - Google Scholar	11/8/2010	JM

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	9349	state adj machine with transition	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:25
L2	366	state adj machine with transition with status	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:27
L3	11	state adj machine with transition with status and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:27
L4	2	state adj machine with transition with status and 707/999.102.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:27
L5	207	state adj machine with transition with (restrict$3 or limit$3 or limitation)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:28
L6	8	state adj machine with transition with (restrict$3 or limit$3 or limitation) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:28

ServiceNow Ex. 1002 Page 80 of 410

L7	2	state adj machine with transition with (restrict$3 or limit$3 or limitation) and 707/999.102.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:28
L8	1	state adj machine with transition with forward with (restrict$3 or limit $3 or limitation)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:29
L9	1	state adj machine with transition near3 forward with (restrict$3 or limit $3 or limitation)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:29
L10	1	state adj machine with transition near3 forward with (restrict$3 or limit $3 or limitation) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:29
L11	1	state adj machine with transition near3 forward with (restrict$3 or limit $3 or limitation) and 707/999.102. ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:29
L12	1	state adj machine same transition near3 forward with (restrict$3 or limit $3 or limitation)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:29
L13	353	state adj machine same transition with (restrict$3 or limit$3 or limitation)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:30
L14	12	state adj machine same transition with (restrict$3 or limit$3 or limitation) with status	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:30

ServiceNow Ex. 1002 Page 81 of 410

L15	1	state adj machine same transition with (restrict$3 or limit$3 or limitation) with status and "707". clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:30
L16	1	state adj machine same transition with (restrict$3 or limit$3 or limitation) with status and 707/999.102.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:30
L17	518	state adj machine same transition with (class or element)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:32
L18	68	state adj machine same transition with (class or element) same (object or object-oriented)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:32
L19	10	state adj machine same transition with (class or element) same (object or object-oriented) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:33
L20	0	state adj machine same transition with (class or element) same (object or object-oriented) and 707/999.102.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:33
L21	7	state adj machine with transition with (class or element) same (object or object-oriented) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:34

ServiceNow Ex. 1002 Page 82 of 410

L22	47	state adj machine with transition with (class or element) same (object or object-oriented)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:34
L23	3	state adj machine with transition with (class or element) same (object or object-oriented) with (stor $3 or sav$3)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:34
L24	3	state adj machine with transition with (class or element) same (object or object-oriented) with (stor $3 or sav$3 or history)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:35
L25	7	state adj machine same transition with (class or element) same (object or object-oriented) with (stor $3 or sav$3 or history)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:35
L26	13	state adj machine same transition same (class or element) same (object or object-oriented) with (stor $3 or sav$3 or history)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:36
L27	1	state adj machine same transition same (class or element) same (object or object-oriented) with (stor $3 or sav$3 or history) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/11/08 09:36

11/8/2010 9:38:11 AM
H:\Workspaces\11917410-nf2.wsp

ServiceNow Ex. 1002 Page 83 of 410

OPAP IAP19
AUG 2 5 2010
PATENT & TRADEMARK OFFICE

PATENT
Customer No. 22,852
Attorney Docket No. 09801.0002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	
Bruce R. DeWoolfson	Group Art Unit: 2168
Application No.: 11/917,410	Examiner: Morrison, Jay A.
Filed: December 13, 2007	Confirmation No.: 6234
For: DATABASE DATA DICTIONARY	

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

Sir:

REPLY TO OFFICE ACTION

In response to the Office Action mailed April 27, 2010, the period for response to which has been extended through August 27, 2010, by a request for extension of one month and fee payment filed concurrently herewith, please amend the above-identified application as follows:

Amendments to the Claims are reflected in the listing of claims in this paper and begin on page 2.

Remarks follow the amendment section of this paper.

AMENDMENTS TO THE CLAIMS:

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

1. (Original) A computerized system for manipulating data sets, comprising:

a processor; and

a data repository, the data repository being adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, comprising:

an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification;

an element relation module configured to store one or more relationships between the elements in the element module;

a class module configured to define at least one class of the elements and store the class;

an attribute module configured to define one or more attributes and store the attributes;

a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class; and

ServiceNow Ex. 1002 Page 85 of 410

a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements.

2. (Original) The computerized system of claim 1, wherein the data repository further comprises:

a state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element.

3. (Original) The computerized system of claim 2, wherein the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

4. (Original) The computerized system of claim 1, wherein the type definition module defines a type including at least one of a boolean type, a picklist type, and a state machine type.

5. (Original) The computerized system of claim 4, wherein the data repository further comprises:

ServiceNow Ex. 1002 Page 86 of 410

a list option module configured to enumerate one or more picklist attributes of the picklist type.

6. (Original) The computerized system of claim 2, wherein the data repository further comprises:

a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements.

7. (Original) The computerized system of claim 6, wherein the data repository further comprises:

a tuple module configured to generate and store one or more tuples, wherein each of the tuples associates a subset of the elements in accordance to one of the types associated with the subset of the elements.

8. (Original) The computerized system of claim 1, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

9. (Original) The computerized system of claim 8, wherein the data repository further comprises:

an element attribute module for each of the at least one class, the element attribute module configured to define and store the attributes of the elements in the class, wherein the element attribute module is identified using the name of the class.

ServiceNow Ex. 1002 Page 87 of 410

10. (Original) The computerized system of claim 9, wherein the data repository further comprises:

an element history module configured to store one or more indicators of modification to the elements.

11. (Original) The computerized system of claim 10, wherein the data repository further comprises:

an element document module configured to store one or more formatting data for grouping and displaying a subset of the elements.

12. (Original) The computerized system of claim 1, wherein the processor is configured to generate a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

13. (Original) The computerized system of claim 1, wherein the processor is configured to generate a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

14. (Original) A method for manipulating data sets, comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository

ServiceNow Ex. 1002 Page 88 of 410

is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; and

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

15. (Original) The method of claim 14, further comprising:

defining a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element, and storing the state machine type in the data repository.

16. (Original) The method of claim 15, wherein the state machine type comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

17. (Original) The method of claim 14, wherein the type comprises a type including at least one of a boolean type, a picklist type, and a state machine type.

ServiceNow Ex. 1002 Page 89 of 410

18. (Original) The method of claim 17, further comprising:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

19. (Original) The method of claim 15, further comprising:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

20. (Original) The method of claim 19, further comprising:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

21. (Original) The method of claim 14, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

22. (Original) The method of claim 21, further comprising:

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

23. (Original) The method of claim 22, further comprising:

ServiceNow Ex. 1002 Page 90 of 410

storing, in the data repository, one or more indicators of modification to the elements.

24. (Original) The method of claim 23, further comprising:

storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

25. (Original) The method of claim 14, further comprising:

generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

26. (Original) The method of claim 14, further comprising:

generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

27. (Currently Amended) A non-transitory computer-readable medium storing a computer-executable program which, when executed by ~~containing instructions to configure~~ a processor ~~to perform~~ performs a method comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

ServiceNow Ex. 1002 Page 91 of 410

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; and

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

28. (Original) The computer-readable medium of claim 27, wherein the method further comprises:

defining a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element, and storing the state machine type in the data repository.

29. (Original) The computer-readable medium of claim 28, wherein the state machine type comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

ServiceNow Ex. 1002 Page 92 of 410

30. (Original) The computer-readable medium of claim 27, wherein the type comprises a type including at least one of a boolean type, a picklist type, and a state machine type.

31. (Original) The computer-readable medium of claim 30, wherein the method further comprises:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

32. (Original) The computer-readable medium of claim 28, wherein the method further comprises:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

33. (Original) The computer-readable medium of claim 32, wherein the method further comprises:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

34. (Original) The computer-readable medium of claim 27, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

ServiceNow Ex. 1002 Page 93 of 410

35. (Original) The computer-readable medium of claim 34, wherein the method further comprises:

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

36. (Original) The computer-readable medium of claim 35, wherein the method further comprises:

storing, in the data repository, one or more indicators of modification to the elements.

37. (Original) The computer-readable medium of claim 36, wherein the method further comprises:

storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

38. (Original) The computer-readable medium of claim 27, wherein the method further comprises:

generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

39. (Original) The computer-readable medium of claim 27, wherein the method further comprises:

ServiceNow Ex. 1002 Page 94 of 410

generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

ServiceNow Ex. 1002 Page 95 of 410

REMARKS

In the Office Action[1], the Examiner rejected claims 27-39 under 35 U.S.C. § 101; and rejected claims 1-39 under 35 U.S.C. § 102(b) as being anticipated by "Overcoming the Complexity of Object-Oriented DBMS Metadata Management" to Habela et al. ("*Habela*").

By this Amendment, Applicant has amended claim 27. Claims 1-39 remain pending.

I. Rejection of Claims 27-39 under 35 U.S.C. §101

Regarding the rejection of claims 27-39 under 35 U.S.C. §101, the Examiner states that claims 27-39 are "directed to non-statutory subject matter" (Office Action at page 2). In response, Applicant has amended independent claim 27 to recite a "non-transitory computer-readable medium storing a computer-executable program which, when executed by a processor performs a method comprising . . . ."

As set forth in In re Bilski, "a claimed process is surely patent-eligible under § 101 if: (1) it is tied to a particular machine or apparatus, or (2) it transforms a particular article into a different state or thing." In re Bilski, 545 F.3d 943, at 954 (Fed. Cir. 2008) (en banc). Because the claims are tied to machines (i.e., computer), Applicant submits that independent claim 27 recites statutory subject matter. Claims 28-39 depend from independent claim 27 and, accordingly, meet the requirements of 35 U.S.C. § 101 at least due to their dependence.

[1] The Office Action contains a number of statements reflecting characterizations of the related art and the claims. Regardless of whether any such statement is identified herein, Applicant declines to automatically subscribe to any statement or characterization in the Office Action.

ServiceNow Ex. 1002 Page 96 of 410

Accordingly, Applicant respectfully requests reconsideration and withdrawal of the rejection of claims 27-39 under 35 U.S.C. § 101.

II. Rejection of Claims 1-39 under 35 U.S.C. §102(b)

Applicant respectfully traverses the rejection of claims 1-39 under 35 U.S.C. § 102(b) as anticipated by *Habela*. In order to properly establish that *Habela* anticipates Applicant's claimed invention under 35 U.S.C. § 102, each and every element of each of the claims in issue must be found, either expressly described or under principles of inherency, in that single reference. Furthermore, "[t]he identical invention must be shown in as complete detail as is contained in the ... claim." *See* M.P.E.P. § 2131, quoting *Richardson v. Suzuki Motor Co.*, 868 F.2d 1126, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989).

Independent claim 1 recites a system including, for example:

> . . .
> a data repository . . . comprising:
> an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification;
> an element relation module configured to store one or more relationships between the elements in the element module;
> a class module configured to define at least one class of the elements and store the class;
> an attribute module configured to define one or more attributes and store the attributes;
> a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class; and
> a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements.

ServiceNow Ex. 1002 Page 97 of 410

Habela discloses a structure that "improves database metadata flexibility" (Abstract). The Examiner cites pages 221, lines 16-30 of *Habela* to allegedly disclose the claimed "element module," "element relation module," "class module," "attribute module," "class attribute module," and "type definition module" (Office Action at page 4). This is not correct.

According to this passage of *Habela*, a "flattened metamodel" includes 1) "[p]redefined values of the meta-attribute 'kind' in the metaclass 'MetaObject' (e.g. 'class', 'attribute', etc.)," 2) "[p]redefined values of meta-attribute names (e.g. 'isAbstract') and of meta-attributes 'kind' of 'MetaRelationship' metaclasses (e.g. 'specialization')," and 3) "[c]onstraints defining the allowed combinations and context of those predefined elements" (page 221, lines 16-22).

Based on the disclosure of *Habela*, the "metamodel" includes values of the "meta-attribute 'kind'" in the "MetaObject", values of the "meta-attribute names," and "meta-attributes 'kind'"of "'MetaRelationship' metaclasses." However, neither this passage nor any other passage of *Habela* teaches or suggests the claimed combination of modules and the relationships between the modules.

For example, claim 1 requires "class module configured to define at least one class of the elements and store the class" and "an attribute module configured to define one or more attributes and store the attributes." *Habela* mentions the words "class" and "attribute" as examples of a metaclass "MetaObject." Even assuming that these examples could correspond to the claimed "class module" and "attribute module," which Applicant does not concede, *Habela* fails to teach or suggest the combination of "a class attribute module configured to define and store one or more class-attribute

ServiceNow Ex. 1002 Page 98 of 410

associations between at least one of the attributes and the class" and "a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements," as further recited in claim 1.

The Examiner cites a single "attribute" in *Habela* to allegedly disclose both the claimed "attribute module" and "class attribute module" and page 221, lines 20-30 of *Habela* to allegedly disclose the claimed "type definition module" (Office Action at page 4). However, a single attribute cannot disclose two distinct modules. Moreover, Applicant finds no teaching or suggestion in the cited passage or any other passage of *Habela* of the claimed "type definition module." Therefore, *Habela* fails to teach or suggest "a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class" in combination with "a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements," as recited in claim 1.

Accordingly, *Habela* cannot anticipate claim 1. Thus, claim 1 is allowable for at least these reasons. Claims 2-13 depend from claim 1 and are thus also allowable for at least the same reasons as claim 1.

Independent claims 14 and 27, while of different scope, recite features similar to those of claim 1 and are thus allowable over *Habela* for at least similar reasons to those discussed above in regard to claim 1. Dependent claims 15-26 and 28-39 are also allowable at least due to their dependence from the independent claims 14 and 27.

ServiceNow Ex. 1002 Page 99 of 410

Dependent claim 2 recites "a state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element." Dependent claims 15 and 28 recite similar elements. The Examiner cites page 216, lines 30-45 of *Habela* to allegedly disclose these elements (Office Action at page 4). This is not correct.

According to this passage of *Habela*, a Unified Modeling Language (UML) is an example of an object metamodel. "To specify additional constraints not expressible by standard graphical notation elements, a precise declarative . . . language named OCL (Object Constraint Language) has been introduced." Applicant submits that the OCL in *Habela* is a precise textual language that provides constraint and object query expressions. However, these languages do not teach or suggest the claimed "state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element." Accordingly, claims 2, 15, and 28 are also allowable for at least these additional reasons.

Dependent claim 3 recites "wherein the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element." Dependent claims 16 and 29 recite similar elements. The Examiner cites page 224, lines 10-15 of

ServiceNow Ex. 1002 Page 100 of 410

Habela to allegedly disclose these elements (Office Action at page 5). This is not correct.

According to this passage of *Habela*, a collection of references are selected and explored. However, "there is no need of checking the kind of those metaobjects, since we assume that metaobjects of kind 'method' are the only legal targets of the meta-relationships of kind 'toMethod.'" This passage of *Habela* appears to reference a proposed query language that includes a meta object specification labeled "toMethod." However, a proposed query string that requires a OO Class Kind as part of the specification does not teach or suggest the claimed "wherein the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element." Accordingly, claims 3, 16, and 29 are also allowable for at least these additional reasons.

Dependent claim 5 recites a "data repository" that includes "a list option module configured to enumerate one or more picklist attributes of the picklist type." Dependent claims 18 and 31 recite similar elements. The Examiner cites page 223, lines 10-15 of *Habela* to allegedly disclose these elements (Office Action at page 5). This is not correct.

According to this passage of *Habela*, "programming through reflection" and "Dynamic SQL" may be examples of performing queries on a metamodel. However, this passage of *Habela* does not teach or suggest the claimed "data repository" that includes "a list option module configured to enumerate one or more picklist attributes of

ServiceNow Ex. 1002 Page 101 of 410

the picklist type." Accordingly, claims 5, 18, and 31 are also allowable for at least these additional reasons.

Dependent claim 6 recites a "data repository" that includes "a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements." Dependent claims 19 and 32 recite similar elements. The Examiner cites page 223, lines 14-20 of *Habela* to allegedly disclose these elements (Office Action at page 6). This is not correct.

This passage of *Habela* also discloses "programming through reflection" and "Dynamic SQL" may be examples of performing queries on a metamodel. However, this passage of *Habela* does not teach or suggest the claimed "data repository" that includes "a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements." Accordingly, claims 6, 19, and 32 are also allowable for at least these additional reasons.

Dependent claim 10 recites a "data repository" that includes "an element history module configured to store one or more indicators of modification to the elements." Dependent claims 23 and 36 recite similar elements. The Examiner cites page 224, lines 8-14 of *Habela* to allegedly disclose these elements (Office Action at page 7). This is not correct.

This passage of *Habela* also discloses selecting "a collection of references to its meta-relationships pointing at methods' definition . . . . The references are explored and metaobjects describing methods are accessed." However, this passage of *Habela* does not teach or suggest the claimed "data repository" that includes "an element history module configured to store one or more indicators of modification to the elements."

ServiceNow Ex. 1002 Page 102 of 410

Accordingly, claims 10, 23, and 36 are also allowable for at least these additional reasons.

III. Conclusion

In view of the foregoing, Applicant respectfully requests reconsideration of this application and the timely allowance of the pending claims.

Please grant any extensions of time required to enter this response and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: August 25, 2010

By: /David W. Hill/
David W. Hill
Reg. No. 28,220

ServiceNow Ex. 1002 Page 103 of 410



PATENT
Customer No. 22,852
Attorney Docket No. 09801.0002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	Group Art Unit: 2168
Bruce R. DeWoolfson	)	
	)	
Application No.: 11/917,410	)	Examiner: Morrison, Jay A.
	)	
Filed: December 13, 2007	)	
	)	Confirmation No.: 6234
For: DATABASE DATA DICTIONARY	)	

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

Sir:

PETITION FOR EXTENSION OF TIME

Applicant petitions for a one month extension of time to file a reply to the Office Action mailed April 27, 2010. A fee of $130.00 is enclosed.

Please grant any additional extensions of time required to enter the attached reply and charge any additional required fees to our deposit account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: August 25, 2010

By: /David W. Hill/
David W. Hill
Reg. No. 28,220

08/26/2010 AWONDAF1 00000084 11917410
01 FC:1251 130.00 OP

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

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	☐ To be Mailed
11/917,410	12/13/2007	

APPLICATION AS FILED – PART I

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

APPLICATION AS AMENDED – PART II

AMENDMENT **08/25/2010**

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 39	Minus	** 39	= 0	X $ =		OR	X $52=	0
Independent (37 CFR 1.16(h))	* 3	Minus	***3	= 0	X $ =		OR	X $220=	0
☐ Application Size Fee (37 CFR 1.16(s))									
☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
					TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT

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

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

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

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

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

Legal Instrument Examiner:
/MONIQUE BRUNSON/

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.

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.
11/917,410	12/13/2007	Bruce Randall DeWoolfson	09801.0002	6234

22852 7590 04/27/2010

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

EXAMINER	
MORRISON, JAY A	

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
04/27/2010	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.

Office Action Summary	Application No.	Applicant(s)	
	11/917,410	DEWOOLFSON, BRUCE RANDALL	
	Examiner	Art Unit	
	JAY A. MORRISON	2168	

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

Period for Reply

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

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

Status

1)☒ Responsive to communication(s) filed on *10 August 2009*.

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) *1-39* 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) *1-39* is/are rejected.

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

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

Application Papers

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

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

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

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

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

Priority under 35 U.S.C. § 119

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

a)☒ All b)☐ Some * c)☐ None of:

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

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

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

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

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *12/13/07*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Remarks

1. Claims 1-39 are pending.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 27-39 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The specification has provided evidence (paragraph [098]) that Applicant intends the medium include signals and as such is drawn to a form of energy. Energy is not one of the four categories of invention and therefore these claims are not statutory. Energy is not a series of steps or acts and thus is not a process. Energy is not a physical article or object and as such is not a machine or manufacture. Energy is not combination of substances and therefore not a composition of matter. Accordingly, this claim is rejected as non-statutory.

Claim Rejections - 35 USC § 102

4. 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.

5. Claims 1-39 are rejected under 35 U.S.C. 102(b) as being anticipated by Habela et al. ('Habela' hereinafter) ("Overcoming the Complexity of Object-Oriented DBMS Metadata Management", Proc of the 9th International Conference Object-Oriented Information System (OOIS 2003), Geneva, Switzerland, September 2nd, 2003).

As per claim 1, Habela teaches

A computerized system for manipulating data sets, comprising: (see abstract)

a processor; and a data repository, the data repository being adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, comprising: (query language and database metamodel, abstract)

an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification;

an element relation module configured to store one or more relationships between the elements in the element module; (MetaAttribute, page 221, lines 16-20)

a class module configured to define at least one class of the elements and store the class; (class, page 221, lines 16-20; figure 4)

an attribute module configured to define one or more attributes and store the attributes; a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class; (attribute, page 221, lines 16-20; figure 4)

and a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements. (page 221, lines 20-30)

As per claim 2, Habela teaches

the data repository further comprises: a state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element. (page 216, lines 30-45)

As per claim 3, Habela teaches

the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element. (page 224, lines 10-15)

As per claim 4, Habela teaches

Habela does not expressly show "the type definition module defines a type including at least one of a boolean type, a picklist type, and a state machine type"

However these differences are only found in the nonfunctional descriptive material and are not functionally involved in the steps recited. The definition of a type would be performed the same regardless of the data. Thus, this descriptive material will not distinguish the claimed invention from the prior art in terms of patentability, see In re Gulack, 703 F.2d 1381, 1385, 217 USPQ 401, 404 (Fed. Cir. 1983); In re Lowry, 32 F.3d 1579, 32 USPQ2d 1031 (Fed. Cir. 1994).

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to select a type having any type of content because such data does not functionally relate to the steps in the system claimed and because the subjective interpretation of the data does not patentably distinguish the claimed invention.

As per claim 5, Habela teaches

the data repository further comprises: a list option module configured to enumerate one or more picklist attributes of the picklist type. (page 223, lines 10-15)

As per claim 6, Habela teaches

the data repository further comprises: a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements. (page 223, lines 14-20)

As per claim 7, Habela teaches

the data repository further comprises: a tuple module configured to generate and store one or more tuples, wherein each of the tuples associates a subset of the elements in accordance to one of the types associated with the subset of the elements. (figure 4, MetaRelationship)

As per claim 8, Habela teaches

each of the class-attribute associations is identified by a logical name and a name of the class. (figure 4, relationships)

As per claim 9, Habela teaches

the data repository further comprises: an element attribute module for each of the at least one class, the element attribute module configured to define and store the attributes of the elements in the class, wherein the element attribute module is identified using the name of the class. (figure 4, named class)

As per claim 10,

the data repository further comprises: an element history module configured to store one or more indicators of modification to the elements. (page 224, lines 8-14)

As per claim 11,

the data repository further comprises: an element document module configured to store one or more formatting data for grouping and displaying a subset of the elements. (page 224, lines 18-22)

As per claim 12,

the processor is configured to generate a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository. (page 216, lines 20-25)

As per claim 13,

the processor is configured to generate a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository. (page 216, lines 20-25)

As per claims 14-26,

These claims are rejected on grounds corresponding to the arguments given above for rejected claims 1-13 and are similarly rejected.

As per claims 27-39,

These claims are rejected on grounds corresponding to the arguments given above for rejected claims 1-13 and are similarly rejected.

Conclusion

6. The prior art made of record, listed on form PTO-892, and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jay A. Morrison whose telephone number is (571) 272-7112. The examiner can normally be reached on M-F 8-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached on (571) 272-3642. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

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

/Jay A Morrison/
Examiner, Art Unit 2168

Jay Morrison
TC2100

Notice of References Cited	Application/Control No. 11/917,410	Applicant(s)/Patent Under Reexamination DEWOOLFSON, BRUCE RANDAL	
	Examiner JAY A. MORRISON	Art Unit 2168	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,587,856 B1	07-2003	Srinivasan et al.	1/1
*	B	US-6,708,186 B1	03-2004	Claborn et al.	1/1
*	C	US-2006/0190478 A1	08-2006	Owens et al.	707/102
*	D	US-2006/0218177 A1	09-2006	Chang et al.	707/102
*	E	US-7,266,565 B2	09-2007	Diab, Salleh	1/1
*	F	US-2008/0201358 A1	08-2008	Calusinski, Jeffrey David	707/102
*	G	US-7,418,459 B2	08-2008	Gupta, Arun K.	1/1
	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.

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11917410	DEWOOLFSON, BRUCE RANDALL
	Examiner	Art Unit
	JAY A MORRISON	2168

SEARCHED			
Class	Subclass	Date	Examiner
707	102 (keyword search of subclass listed)	4/23/2010	JM

SEARCH NOTES		
Search Notes	Date	Examiner
East	4/23/2010	JM
Keyword search of 707 plus limited text search of 707/999.102	4/23/2010	JM
Inventor Search	4/23/2010	JM

INTERFERENCE SEARCH			
Class	Subclass	Date	Examiner

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11917410	DEWOOLFSON, BRUCE RANDALL
	Examiner	Art Unit
	JAY A MORRISON	2168

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

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

CLAIM		DATE								
Final	Original	04/22/2010								
	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	✓								

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	11917410	DEWOOLFSON, BRUCE RANDALL
	Examiner	Art Unit
	JAY A MORRISON	2168

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

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

CLAIM		DATE								
Final	Original	04/22/2010								
	36	✓								
	37	✓								
	38	✓								
	39	✓								

Substitute for form 1449A/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				Application Number	
				Filing Date	December 13, 2007
				First Named Inventor	Bruce Randall DEWOOLFSON
				Art Unit	
(Use as many sheets as necessary)				Examiner Name	
Sheet	1	of	1	Attorney Docket Number	09801.0002

U.S. PATENTS AND PUBLISHED U.S. PATENT APPLICATIONS

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

Note: Copies of the U.S. Patent Documents are not Required in IDS filed after October 21, 2004

FOREIGN PATENT DOCUMENTS

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

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	Translation[6]
		JODLOWSKI, ET AL., "Objects and Roles in the Stack-Based Approach", XP-002401006, Proc. of the 13th International Conference on Database and Expert Systems Applications (DEXA 2002), Vol. 2453, pages 514-523, (September 2, 2002)	
		HABELA, ET AL., "Overcoming the Complexity of Object-Oriented DBMS Metadata Management", XP-002401007, Proc. of the 9th International Conference Object-Oriented Information Systems (00IS 2003), pages 214-225, (September 2, 2003)	
		BLAHA, ET AL., "Converting OO Models into RDBMS Schema", XP-002401008, IEEE Software USA, Vol. 11, No. 3, pages 28-39, (May 1994)	
		ARSOV, ET AL., "An Approach for Design of the Object-Oriented Databases with Restricted Natural Language Access", XP-002401009, Proc. of the 4th International Conference Conference on Computer Systems and Technologies: E-Learning, pages 147-152, (June 19, 2003)	
		JERKE, ET AL., "Building Database-Driven Flash Applications", XP-002401199, APRESS, SPRINGER-VERLAG BERLIN AND HEIDELBERG GMBH & CO. KG, pages 119-170, (October 2003)	

Examiner Signature	/Jay Morrison/	Date Considered	04/13/2010

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /J.M./

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. 6234

SERIAL NUMBER	FILING or 371(c) DATE RULE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
11/917,410	12/13/2007	707	2168	09801.0002

APPLICANTS

Bruce Randall DeWoolfson, Potomac Falls, VA;

**** CONTINUING DATA ***************************

This application is a 371 of PCT/US2006/023242 06/14/2006
which claims benefit of 60/690,147 06/14/2005

**** FOREIGN APPLICATIONS ***************************

**** IF REQUIRED, FOREIGN FILING LICENSE GRANTED ****

09/02/2009

Foreign Priority claimed ☐ Yes ☑ No 35 USC 119(a-d) conditions met ☐ Yes ☑ No Verified and Acknowledged /JAY A MORRISON/ Examiner's Signature	☐ Met after Allowance Initials	STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
		VA	19	39	3

ADDRESS

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER
LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413
UNITED STATES

TITLE

Database Data Dictionary

FILING FEE RECEIVED		
1880	FEES: Authority has been given in Paper No._____ to charge/credit DEPOSIT ACCOUNT No._____ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other _____ ☐ Credit

EAST Search History

EAST Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	6	(US-20060218177-$ or US-20060190478-$ or US-20030014424-$). did. or (US-7440969-$ or US-7418459-$ or US-7266565-$). did.	US-PGPUB; USPAT	OR	OFF	2010/04/23 09:31
L2	1	l1 and (metadata or metamodel or meta)	USPAT	OR	OFF	2010/04/23 09:31
L16	1833	(database or rdbms or dbms or db) same (class or object-oriented) same (metadata or metamodel or meta)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:37
L17	1015	(database or rdbms or dbms or db) same (class or object-oriented) same (metadata or metamodel or meta) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:37
L18	105	(database or rdbms or dbms or db) same (class or object-oriented) same (metadata or metamodel or meta) and 707/999.102. ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:37
L19	557	(database or rdbms or dbms or db) same (class or object-oriented) with (metadata or metamodel or meta) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:38

ServiceNow Ex. 1002 Page 122 of 410

L20	331	(database or rdbms or dbms or db) with (class or object-oriented) with (metadata or metamodel or meta) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:38
L21	1052	(database or rdbms or dbms or db) same (class or object-oriented) same (metadata or metamodel or meta) same (element or attribute or type)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:38
L22	614	(database or rdbms or dbms or db) same (class or object-oriented) same (metadata or metamodel or meta) same (element or attribute or type) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:38
L23	68	(database or rdbms or dbms or db) same (class or object-oriented) same (metadata or metamodel or meta) same (element or attribute or type) and 707/999.102.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:38
L24	306	(database or rdbms or dbms or db) with (class or object-oriented) same (metadata or metamodel or meta) same (element or attribute or type) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:39

ServiceNow Ex. 1002 Page 123 of 410

L25	187	(database or rdbms or dbms or db) with (class or object-oriented) with (metadata or metamodel or meta) same (element or attribute or type) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:39
L26	114	(database or rdbms or dbms or db) with (class or object-oriented) with (metadata or metamodel or meta) with (element or attribute or type) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:39
L27	80	(database or rdbms or dbms or db) with (class or object-oriented) with (metadata or metamodel) with (element or attribute or type) and "707".clas.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:39
L29	17	(database or rdbms or dbms or db) with (class or object-oriented) with (metadata or metamodel) with (element or attribute or type) and 707/999.102.ccls.	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:40
L30	8	(database or rdbms or dbms or db) with (class or object-oriented) with (metadata or metamodel) with attribute with type	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:48

ServiceNow Ex. 1002 Page 124 of 410

L31	0	(database or rdbms or dbms or db) with (class or object-oriented) with (metadata or metamodel) with attribute with type with element	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:48
L32	0	(database or rdbms or dbms or db) same (class or object-oriented) with (metadata or metamodel) with attribute with type with element	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:48
L33	4	(database or rdbms or dbms or db) same (class or object-oriented) same (metadata or metamodel) with attribute with type with element	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:48
L34	1	(database or rdbms or dbms or db) with (class or object-oriented) with (metadata or metamodel) same attribute same type same element	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2010/04/23 09:56

4/23/2010 10:06:01 AM
C:\Documents and Settings\jmorrison\My Documents\EAST\Workspaces\11917410.wsp

ServiceNow Ex. 1002 Page 125 of 410

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 NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
11/917,410	12/13/2007	Bruce Randall DeWoolfson	09801.0002

CONFIRMATION NO. 6234

22852
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

PUBLICATION NOTICE

OC00000039287500

Title: Database Data Dictionary

Publication No.US-2009-0313278-A1
Publication Date:12/17/2009

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently http://pair.uspto.gov/. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

OIPE
DEC 04 2009
PATENT & TRADEMARK OFFICE

IFW

PATENT
Customer No. 22,852
Attorney Docket No. 09801.0002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	
Bruce Randall DEWOOLFSON	Group Art Unit: 2161
Application No.: 11/917,410	Examiner: Unknown
Filed: December 13, 2007	Confirmation No.: 6234
For: DATABASE DATA DICTIONARY	

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

Sir:

STATUS INQUIRY

According to our records, the above-identified application was filed in the U.S. Patent and Trademark Office ("Office") on December 13, 2007. An Office Action on the merits has not yet been issued in this application.

In view of these circumstances, the undersigned attorney respectfully requests that the Office advise him of the status of this application as soon as possible in order to determine whether further action by the Applicant is required at this time.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: December 4, 2009

By: /David W. Hill/
David W. Hill
Reg. No. 28,220

UNITED STATES NATIONAL STAGE WORKSHEET (DO/EO)

B. C., Patent Application Specialist, PCT /DO/EO/National Stage Division

Publication No. WO 06 / 138420 Publication Date 12-28-06

Not Published: ___U.S. only ___Requested

Basic fee paid: $ 310 PER fee paid: $ 210 / EP ISA fee paid: $ 410 / EP

U. S. Application No. 11/ 917,410

International Application Number: PCT/ US2006 / 023242 International Filing Date: 6-14-06

Application Filed by 30 Month: ✓ Yes ___No Language: Eng

Copy of IA: yes ✓ no___ Translation: yes___ no___ Defective: ___Yes___Date satisfied:___

371 Filing Fees ✓ paid ___Insufficient Meet Article 33 Requirement: ___Yes ___No & why not___

Original claims: 1-39 Added Claims: ___Total Claims: 39 Chargeable 39 Independent 3 multiple NL

After entry of Article 19 or 34 Amdt: ___

Cancelled Claims via Article 19 or 34___ &/or Pre-Amdt. ___Clms. added via Article 19 or 34 ___

Total Number of Drawing Sheets: 19 Foreign Text: ___

Oath/Declaration: yes ✓ no___: signed ✓ unsigned ___ defective___ Date Satisfied: 12.13.07

PCT/RO/101/Request Form Declaration: ___yes properly executed No ___ not properly executed Date: ___

Small Entity: ___Yes Small Entity Statement ___Assertion by filing fee paid ___ Large Entity: ✓

1st Submission: Biochemical Seq. Diskette: yes___ no___ entered & date ___ not entered & date ___

2nd Submission: Biochemical Seq. Diskette: yes___ no___ entered & date ___ not entered & date___

Biochemical Seq. Listing: yes __ no __ statement ___ yes ___ no other submission date(s): ___

Biochemical Diskette/Listing not needed: ___

Copy of ISR (210): ___with references without references ✓ Non-Establishment of ISR PCT/ISA/203 ___

Article 19 Amendment: ___entered ___not entered ___ Replaced by Article 34 Amendment ___

Copy of IPER: ___without Annexes: ___with Annexes: ___ Annexes entered ___Annexes not entered ___237

Reason Annexes have not been entered: ___

Preliminary Amendment(s): yes ___not entered___ & why ___Other Amendment dates: ___

IDS: ✓ yes ___with references ___without references Other IDS Dates: ___

Request for Immediate Examination: yes___ no___, Other Early Processing Date: ___

Substitute Specification: yes___ Date filed: ___Other documents filed: ___

Assignment: yes ✓ Date filed: 12.13.07 Assignment for PG Pub: ___Yes Date filed: ___

Power of Attorney ✓ Date filed: 12.13.07 Application Data Sheet ___Priority Document(s): yes ✓

**Other: ___

**Other Problems: ___

Application Size: Spec. 33 Claims 11 Abstract 1 Drawings 19 Seq. List. ___Seq. RF___Total: ___

Application Fees: owed (paid)

___Declaration ✓ Claims ___Multiple ___Translation ___Extension ___Petition ___Application size___

Date of 35 USC Receipt of Request: 12-13-07

Date Completion USC 371 Requirements: 11

Notice of Missing Requirements: ___Response to Missing Requirements___

371 Formalities Letter: (Sequence) 922___922 Response ___ ___ or (Fees Owed) 923___923-Repsonse ___

Notice of Defective Response: ___Defective Response Reply___

Notice of Acceptance: 09/02/09

Petition to Revive: ___ Petition 1.47: ___Petition Granted:___

Notification of Abandonment:___

PATENT APPLICATION FEE DETERMINATION RECORD

Effective October 2, 2008*

Application or Docket Number: 11/917,410

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)	SMALL ENTITY TYPE ☐ RATE	FEE	OR	OTHER THAN SMALL ENTITY RATE	FEE
U.S. NATIONAL STAGE FEES							
BASIC FEE	SMALL ENT. = $ 165	LARGE ENT. = $ 330	BASIC FEE		OR	BASIC FEE	310
EXAMINATION FEE	Satisfies PCT Article 33(1)-(4) = $ 0 / $0	All other situations = $ 110 / $ 220	EXAM. FEE			EXAM. FEE	210
SEARCH FEE	U.S. is ISA = $ 50 / $ 100 ALL other countries = $ 215 /$ 430	ALL other situations = $ 270/ $ 540	SEARCH FEE			SEARCH FEE	410
FEE FOR EXTRA SPEC. PGS.	minus 100 =	/ 50 =	X $ 135 =			X $ 270 =	
TOTAL CHARGEABLE CLAIMS	39 minus 20 =	* 19	X $ 26 =		OR	X $ ~~52~~ 50 =	950
INDEPENDENT CLAIMS	3 minus 3 =	*	X $ 110 =		OR	X $ 220 =	
MULTIPLE DEPENDENT CLAIM PRESENT		☐	+ $ 195 =		OR	+ $ 390 =	
			TOTAL		OR	TOTAL	1880

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
AMENDMENT A	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐				

SMALL ENTITY RATE	ADDI-TIONAL FEE	OR	OTHER THAN SMALL ENTITY RATE	ADDI-TIONAL FEE
X $ 26 =		OR	X $ 52 =	
X $ 110 =		OR	X $ 220 =	
+ $ 195 =		OR	+ $ 390 =	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
AMENDMENT B	Total	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM ☐				

SMALL ENTITY RATE	ADDI-TIONAL FEE	OR	OTHER THAN SMALL ENTITY RATE	ADDI-TIONAL FEE
X $ 26 =		OR	X $ 52 =	
X $ 115=		OR	X $ 220 =	
+ $ 195 =		OR	+ $ 390 =	
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.

Barbara Campbell, PCT National Stage Division

MULTIPLE DEPENDENT CLAIM FEE CALCULATION SHEET

(FOR USE WITH FORM PTO-875)

SERIAL NO. 11/917,410

FILING DATE 12-13-07

APPLICANT(S)

CLAIMS

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
1	1					
2		1				
3		1				
4		1				
5		1				
6		1				
7		1				
8		1				
9		1				
10		1				
11		1				
12		1				
13		1				
14	1					
15		1				
16		1				
17		1				
18		1				
19		1				
20		1				
21		1				
22		1				
23		1				
24		1				
25		1				
26		1				
27	1					
28		1				
29		1				
30		1				
31		1				
32		1				
33		1				
34		1				
35		1				
36		1				
37		1				
38		1				
39		1				
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
TOTAL IND.	3	↓		↓		↓
TOTAL DEP.	36	←		←		←
TOTAL CLAIMS	39					

	AS FILED		AFTER 1st AMENDMENT		AFTER 2nd AMENDMENT	
	IND.	DEP.	IND.	DEP.	IND.	DEP.
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						
TOTAL IND.		↓		↓		↓
TOTAL DEP.		←		←		←
TOTAL CLAIMS						

Barbara Campbell, PCT National Stage

U.S. DEPARTMENT of COMMERCE
Patent and Trademark Office

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

U.S. APPLICATION NUMBER NO.	FIRST NAMED APPLICANT	ATTY. DOCKET NO.
11/917,410	Bruce Randall DeWoolfson	09801.0002

22852
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

INTERNATIONAL APPLICATION NO.	
PCT/US2006/023242	
I.A. FILING DATE	PRIORITY DATE
06/14/2006	06/14/2005

CONFIRMATION NO. 6234
371 ACCEPTANCE LETTER

OC000000037641640

Date Mailed: 09/04/2009

NOTICE OF ACCEPTANCE OF APPLICATION UNDER 35 U.S.C 371 AND 37 CFR 1.495

The applicant is hereby advised that the United States Patent and Trademark Office in its capacity as a Designated / Elected Office (37 CFR 1.495), has determined that the above identified international application has met the requirements of 35 U.S.C. 371, and is ACCEPTED for national patentability examination in the United States Patent and Trademark Office.

The United States Application Number assigned to the application is shown above and the relevant dates are:

12/13/2007	12/14/2007
DATE OF RECEIPT OF 35 U.S.C. 371(c)(1), (c)(2) and (c)(4) REQUIREMENTS	DATE OF COMPLETION OF ALL 35 U.S.C. 371 REQUIREMENTS

A Filing Receipt (PTO-103X) will be issued for the present application in due course. **THE DATE APPEARING ON THE FILING RECEIPT AS THE " FILING DATE" IS THE DATE ON WHICH THE LAST OF THE 35 U.S.C. 371 (c)(1), (c)(2) and (c)(4) REQUIREMENTS HAS BEEN RECEIVED IN THE OFFICE. THIS DATE IS SHOWN ABOVE.** *The filing date of the above identified application is the international filing date of the international application (Article 11(3) and 35 U.S.C. 363).* Once the Filing Receipt has been received, send all correspondence to the Group Art Unit designated thereon.

The following items have been received:

- Copy of the International Application filed on 12/13/2007
- Copy of the International Search Report filed on 12/13/2007
- Information Disclosure Statements filed on 12/13/2007
- Oath or Declaration filed on 12/13/2007
- U.S. Basic National Fees filed on 12/13/2007
- Assignment filed on 12/13/2007
- Priority Documents filed on 12/13/2007
- Power of Attorney filed on 12/13/2007
- Specification filed on 12/13/2007
- Claims filed on 12/13/2007
- Abstracts filed on 12/13/2007
- Drawings filed on 12/13/2007

ServiceNow Ex. 1002 Page 131 of 410

Applicant is reminded that any communications to the United States Patent and Trademark Office must be mailed to the address given in the heading and include the U.S. application no. shown above (37 CFR 1.5)

BARBARA A CAMPBELL

Telephone: (703) 756-1461

FORM PCT/DO/EO/903 (371 Acceptance Notice)

ServiceNow Ex. 1002 Page 132 of 410

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 NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
11/917,410	12/13/2007	2161	1880	09801.0002	39	3

CONFIRMATION NO. 6234

22852
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

FILING RECEIPT



Date Mailed: 09/04/2009

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections**

Applicant(s)

Bruce Randall DeWoolfson, Potomac Falls, VA;

Assignment For Published Patent Application

ENTERPRISE ELEMENTS, INC., Reston, VA

Power of Attorney: The patent practitioners associated with Customer Number 22852

Domestic Priority data as claimed by applicant

This application is a 371 of PCT/US2006/023242 06/14/2006
which claims benefit of 60/690,147 06/14/2005

Foreign Applications

If Required, Foreign Filing License Granted: 09/02/2009

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 11/917,410**

Projected Publication Date: 12/17/2009

Non-Publication Request: No

Early Publication Request: No

ServiceNow Ex. 1002 Page 133 of 410

Title

Database Data Dictionary

Preliminary Class

707

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as

ServiceNow Ex. 1002 Page 134 of 410

set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

ServiceNow Ex. 1002 Page 135 of 410

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US2006/023242

International filing date: 14 June 2006 (14.06.2006)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/690,147
Filing date: 14 June 2005 (14.06.2005)

Date of receipt at the International Bureau: 15 August 2006 (15.08.2006)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



1500192

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

August 03, 2006

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: *60/690,147*
FILING DATE: *June 14, 2005*
RELATED PCT APPLICATION NUMBER: *PCT/US06/23242*

THE COUNTRY CODE AND NUMBER OF YOUR PRIORITY APPLICATION, TO BE USED FOR FILING ABROAD UNDER THE PARIS CONVENTION, IS *US60/690,147*

Certified by

Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

15

15/01/2007

13281 U.S. PTO 061405

PROVISIONAL APPLICATION COVER SHEET

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

U.S. PTO 60/690147 061405

Docket Number 09801.6000-00		Type a plus sign (+) inside this box ▫	+

INVENTOR(s)/APPLICANT(s)

LAST NAME	FIRST NAME	MIDDLE INITIAL	RESIDENCE (CITY AND EITHER STATE OR FOREIGN COUNTRY)
DeWoolfson	Bruce	R	Potomac Falls, Virginia

TITLE OF INVENTION (500 characters max)

Database Data Dictionary

CORRESPONDENCE ADDRESS

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P., **Customer Number 22,852**

ENCLOSED APPLICATION PARTS (check all that apply)

☒ Specification: 35 Pages ☒ Drawing(s): 17 Sheets/17 Figures

☐ Other: N/A Pages; N/A

METHOD OF PAYMENT

☒ A check or money order is enclosed to cover the Provisional filing fees	PROVISIONAL FILING FEE $200.00
☒ The Commissioner is hereby authorized to charge any deficiency or credit any overpayment i n the filing fees to Deposit Account Number 06-0916.	Total Number of Pages of specification, drawings, sequence or computer listing, or other papers: 52. If more than 100 pages, add $250 for each additional 50 pages or fraction thereof. $0.00 (Size Fee) Reduction by ½ For Small Entity $N/A TOTAL FILING FEE $200.00

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

☒ No.

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

Respectfully submitted,

SIGNATURE _____ Date: June 14, 2005

Joshua C. Liu (Reg. No. 55,391)

TYPED OR PRINTED NAME: Lionel M. Lavenue for REGISTRATION NO.: 46,859

☐ Additional inventors are being named on separately numbered sheets attached hereto.

PROVISIONAL APPLICATION FILING ONLY

Provisional Patent Application
Customer No.: 22,852
Attorney Docket No.: 09801.6000-00

UNITED STATES PROVISIONAL PATENT APPLICATION

FOR

DATABASE DATA DICTIONARY

BY

BRUCE RANDALL DEWOOLFSON

ServiceNow Ex. 1002 Page 139 of 410

13281 U.S.PTO
061405

Provisional Patent Application
Customer No.: 22,852
Attorney Docket No.: 09801.6000-00

TECHNICAL FIELD

[001] This invention relates to the field of computer data storage. More particularly, the present invention relates to methods and systems for storing, managing, indexing, interrelating, and/or retrieving data sets.

BACKGROUND

[002] Currently, a conventional data storage system implements its own data model according to the user interface and the business rules specification for that system. For the conventional system, the system's developers write dedicated code and user interface code that displays or manages certain data inputs such as check boxes, lists, combo boxes or other data inputs. Such systems depend upon a specific data model and lack flexibility and portability. The conventional system's dedicated code and user interface in some respects manipulate the data that the conventional system's application stores and retrieves during operation. Moreover, the conventional system stores and retrieves data locally on a local disk, such as a C drive on a Windows system. It is well known that conventional systems may also store data on a remote storage media, e.g. a file server or database.

[003] There are efforts to overcome the lack of flexibility and portability of conventional data storage systems, such as the adoption and utilization of extensible markup language for storing data (e.g., Microsoft Office® XML Open Format, OASIS Open Office XML Format, etc). Data files in open data formats (e.g., XML) facilitate data transfer and are typically stored on a desktop machine. XML data files rely on the

ServiceNow Ex. 1002 Page 140 of 410

structures defined by XML standards, which allows standardized XML parsers to read data files expressed in XML format.

[004] However, open data formats based on XML and other open standards do not interrelate and manage dynamic data sets through process and collaboration and state machines. Open data formats (e.g., XML) also do not provide for inheritance and override behaviors, collaboration or state machine process enforcement, and/or element style normalization.

[005] Accordingly, there is a need for systems and methods that store, manage, index, interrelate, and/or retrieve data sets in a manner independent of the data model.

BRIEF DESCRIPTION OF THE DRAWINGS

[006] Both the foregoing general description and the following detailed description are exemplary and explanatory only. They do not restrict the invention, as claimed. Furthermore, the accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate embodiments of the invention and together with the description, serve to explain the principles of the invention. In the drawings:

[007] Figure 1 is an exemplary entity-relationship diagram of an elements metadata repository (EMR) for storing, managing, indexing, interrelating, and/or retrieving data sets, consistent with the principles of the present invention;

[008] Figure 2 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing articulated types, workspaces and workspace objects, class and class attributes, parameters, list options, etc;

[009] Figure 3 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing elements, element relations, element attributes, element history, element documents, etc;

[010] Figure 4 illustrates an exemplary diagrammatic visualization for visually selecting and relating elements using a graphical user interface (GUI);

[011] Figure 5 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing state machines, state machine transitions, state machine statuses, etc;

[012] Figure 6 illustrates an exemplary state machine diagram;

[013] Figure 7 illustrates a GUI of an exemplary diagrammatic depiction displaying congregations of requirements that are related to subcomponents of a system;

[014] Figure 8 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing tuples, named texts and metadata for named texts, etc;

[015] Figure 9 illustrates a GUI of an exemplary elements explorer;

[016] Figure 10 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing collaboration status history, polls, discussions, system components, etc;

[017] Figure 11 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing terminology, glossary, etc;

ServiceNow Ex. 1002 Page 142 of 410

[018] Figure 12 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing emails, email details, etc;

[019] Figure 13 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing formatted dataset, SQL extensible stylesheet language transformations, etc;

[020] Figures 14 and 15 depict GUIs of exemplary editing environments that may be invoked from a displayed element;

[021] Figure 16 is a diagram depicting an exemplary Human Resource Management workspace; and

[022] Figure 17 is a diagram depicting an exemplary computer system in which systems and methods consistent with the principles of the elements metadata repository of Figure 1 may be implemented.

DETAILED DESCRIPTION

[023] Reference will now be made in detail to present embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Entity-Relationship Configuration

[024] FIG. 1 shows an entity relationship diagram 100 of an elements metadata repository (EMR) for storing, managing, indexing, interrelating, and/or retrieving data sets, consistent with principles of the present invention. In accordance with embodiments of the invention, a repository, such as the EMR, is a central place where

data and metadata is stored and maintained. Metadata is data that is used to describe other data. Examples of metadata include schema, tables, indices, views, column definitions, etc. Metadata is recursive such that data about data about the data may be stored. This recursive nature may be expressed as levels, such as "M" number of levels. For example, the actual data being stored is M0. Data about that actual data (i.e., M0) may be referred to as M1. Data about the M1 layer data may be referred to as M2 and so on. In this example, the M1 and M2 layers are metadata, i.e., data about data. The EMR of FIG. 1 allows for storage of any number of data layers.

[025] Embodiments of the present invention, as shown in FIG. 2, include a class table 150, which may be analogous to a relational database table. Just as a table contains columns, a class, such as an entry in class table 150, contains attributes. In an exemplary implementation of the EMR, using database systems such as Oracle®, Microsoft SQL Server®, or MySQL, a user may create a class and its attributes by creating a table and constructing columns within the table. In other words, the user creates a class and populates the class with attributes where attributes are conceptually other pieces of information of interest to the things that the class is trying to describe. For instance, if the class was a person, an attribute may be the person's name, birth date, or social security number. Any number of attributes can be assigned to any number of classes.

[026] Unlike a conventional relational database, the EMR extends the notion of available attributes types beyond the standard string, number, and date types. To do so, the EMR allows dynamically extendable entries in a type definition table 102. Type

definition table 102 may store articulated types that may be, for example, an M4 layer while still maintaining the meta meta meta meta data language. By using the type of the person's name for an attribute, the user can declare more than what a normal relational database may store. For example, the EMR provides for a boolean type which is not normally found in a relational database. The EMR also provides for a pick list type, which stores a plurality of optional selections, and restricts actual data settings to none, or one and only one of the optional selections. The pick list type is also not found in a relational database. Moreover, the EMR is capable of storing a process or state machine type which is not found in a relational database. All these data types provide flexibility in the type of data management that can be performed.

[027] In accordance with still further embodiments of the invention, the EMR includes a parameter table 104. Parameter table 104 allows for any unforeseen attribute values to be placed upon system level components without redefining the base schema. For instance, if one type of named text is a SQL query, and the system requires security settings on that one type of named text (further described below), then a parameter entry can be used to specify that setting instead of redefining named_text table (e.g. named_text table 140 in FIG. 8) to include an additional column, which may or may not be needed for all other named text types.

[028] Embodiments of the present invention, as shown in FIG. 2, include a workspace table 108 and a workspace_object table 106. Workspace table 108 and workspace_object table 106 provide for logical grouping of elements into subsets that are usable for particular users or purposes. For example, if an EMR user is a tester,

then a "Testing" workspace may be declared. Objects, including elements, classes, named texts, and "things" that is defined by a numeric identifier (which may be every "thing") may be placed into 'Testing' workspace via entries in workspace_object table 106. The tester would be assigned to this workspace (also through workspace_object table 106), and therefore only be permitted to see the other objects in the workspace. The effect is a basic level of security and data volume reduction to small localized easily navigable subsets.

[029] Consistent with embodiments of the invention, an attribute table 110 holds the definition of each attribute used in each class in the EMR. Attribute table 110 may store the physical name and type of the data which corresponds to the database column name and column type where the actual values are stored. A class_attribute table 112 may associate a given attribute to one or more classes (in class table 150). Because class_attribute table 112 allows for one attribute to be referenced by many classes, there is attribute inheritance among the classes. Attribute inheritance among the classes is a powerful notion not found in traditional relational database management systems (RDBMS). Class_attribute table 112 stores the logical name that may be used by the attribute when presented to the user for data manipulation such as insert or update or delete of values. The ability to reference the same attribute via multiple names depending on which class it appears in is akin to the concept of overridden inheritance, also a concept not found in RDBMS.

[030] In an embodiment of the invention, the EMR may immediately create a physical instantiation of the logical class and attribute model as soon as a class attribute

ServiceNow Ex. 1002 Page 146 of 410

definition is entered and committed into class_attribute table 112. The EMR may immediately represent the table and columns that the class and attribute definitions dictate. Moreover, the EMR's editors (see, for example, FIG. 14 and 15) may immediately conform to this definition and support full data manipulation of any values in any defined attribute. Furthermore, when certain types of attributes are attached to a class, appropriate named text entries are automatically created and tupled to the class and may be immediately available for use in the EMR. For example, if a pick list attribute type is added to a class, then queries that returns, for instance, all rows in the class and how many rows fall into each of the available pick list options, are automatically created, which allows the EMR to present the results (via charts or other presentation formats) to the users. This immediate construction of physical storage, complete editing capability, and automated reporting allows the EMR to become operational immediately upon logical schema definition.

[031] Embodiments of the present invention, as shown in FIG. 2, further include a list_option table 114. List_option table 114 may hold all possible enumerations in all possible pick list attributes, which allows for structured user interface and predictable query structures when displaying and managing pick list data. A color table 116 may hold a record for each enumerated color referenced by any object in the EMR. A color can be declared and associated to an object such as a class, list option selection, or element, etc, and when detected during presentation, the id reference value for the color can be replaced with a swatch of the actual color or any other run time decoration as appropriate.

[032] Consistent with embodiments of the invention, the EMR includes a symbol table 118. Symbol table 118 is similar to color table 116 because it may hold a list of references to graphical objects that can be used in the system. The number and type of graphical objects are not restricted, and can be extended by addition of type definition entries but typically include standard graphic files, such as .BMP, .GIF, .JPG, .SVG, and .SVGz. Once declared in this table, these graphical objects can be referenced by any object in the system, and when the reference is detected at display time, the reference may be replaced with a rendering of the actual image. For example, reports may be constructed that show check marks, colored balls, military insignia, or any other relevant image depending on the needs of the visualization.

[033] The EMR of FIG. 1 conceptually breaks the world into an articulation of "things." The EMR places these "things" into one pile in an element table 160. An element stored in element table 160 implies that the user can find similarity between any "thing" that the user would like to discuss, report, store, or manage. Element table 160 provides the user the capability to add those "things" into a congruous pile so that the user can deal with them without regard to their exact nature. This provides an over-normalization of any data captured by the EMR into one record type called an element.

[034] FIG. 3 illustrates an exemplary element, which has a unique identifier (e.g., element ID), a name (e.g., description), and a classification (e.g., class identifier). Consistent typing of the element ID permits the user to refer to elements across the schema without having to convert the type of the element ID. In one embodiment, the element ID takes on a number format, e.g., a long value or an integer value. In this

embodiment, any "thing" in the world may be uniquely identified by a number, e.g., an integer. Moreover, once any "thing" has been differentiated from nothingness, the "thing" has a name. This name may be called a description. In one embodiment, the key components of the element are the element ID and the name, as shown in element table 160. The description does not have to be unique. Some "things" may be described generally with the same name but have differences that render those things unique from each other. Another component of the element is a class identifier. The class identifier is a pointer or an indicator of the type or classification of this element in the universe. The class identifier allows the EMR to articulate the differences between types of elements that otherwise appear to be similar.

[035] Embodiments of the present invention include an element_relation table 162 for storing relationships between elements. Further to storing and managing elements in the universe, the EMR of FIG. 1 records and manages the fact that one element relates to another element in element_relation table 162. Each entry in element_relation table 162 may contain a pointer to at least two elements in elements table 160 and a type definition describing the type of relationship. Using the over-normalized model, the EMR ends up with a collection of elements and a collection of indicators of how those elements relate to each other.

[036] Embodiments of the present invention, as shown in FIG. 3, include an element_attr_c# table 164. Element_attr_c# table 164 represents a plurality of physical tables that are constructed into the base RDBMS. A physical table may be constructed in element_attr_c# table 164 for each class that is declared in class table 150. A

column is inserted into this table for each class_attribute entry that specifies an attribute that should be in this class. In the physical implementation the # symbol is replaced with the identifier of the class to ensure uniqueness in table names, which is a requirement in the RDBMS but not necessarily in the class names of the element system.

[037] For each element_attr_c# table constructed, a corresponding entry in an element_history table 166 may also be constructed. Element_history table 166 records each version of each element through its lifespan in the EMR. In addition to all values of all attributes, element_history table 166 may also store indicators of the user making the change, and the time the change was made. This provides the EMR the capability to provide a complete history for all tracked elements throughout the life of the elements. This is useful for recovering a point in time configuration, or mixing historic version of various element subsets into named groups sometimes referred to as Baselines, Blocks, Revisions, Versions, Variants, or the like.

[038] Consistent with embodiments of the invention, the EMR includes an element_document table 168, which may represent a series of records that logically group a set of elements into what would traditionally be called a document. Element_document table 168 provides for formatting information that is separated from the elements themselves, yet are important for the reconstruction for an ordered formatted display of the elements.

[039] From the collection of elements and the collection of indicators, the EMR can create visualizations for users by placing the elements in context with other

elements that they relate to. FIG. 4 illustrates an exemplary visualization 400. Visualization 400 may also be called a diagrammatic visualization, which may be implemented by an elements diagram manager. In a diagrammatic visualization, the elements diagram manager selects or allows the user to select subsets of elements in the universe and place them on a drawing canvas. The elements diagram manager then arranges the elements in relation to other elements based on the conductivity rules found in element_relation table 162.

[040] In visualization 400, several elements are arranged on the canvas in proximity to each other and connected by directional lines that indicate their relation to each other, as determined by element_relation table 162. Each element in the canvas may be expanded to find and display elements in proximity to itself that is not yet on the canvas. Visualization 400 provides the user the capability to dynamically explore related elements.

[041] In addition to expansion of an element, each element in the EMR may invoke an editing screen that manages the values for the attributes of that element, based on the classification of that element. FIG. 14 displays a screen shot illustrating an exemplary editing environment 1400 that may be invoked from any displayed element, whether it appears in a diagram canvas, element explorer list, or other context.

[042] As illustrated in FIG. 14, editing environment 1400 shows several tabs, a General Tab, an Attributes tab, and a History tab. The selected tab in the example shown in FIG. 14 is the Attributes tab, which displays a listing of each attribute associated with the elements class, and an editor for the value of that attribute. In this

example, the type of attribute is indicated by small icons to the left of the attribute name and the editor appears to the right of the attribute name. The order in which the attributes appear and certain display characteristics such as the number of rows of text to display in the edit field are determined by the entry in class_attribute table 112. Specialized editors can be defined and used such as date selectors, pick list combo boxes, and boolean checkboxes, all of which are invoked dynamically based on the definition of the attributes that happen to be in this class.

[043] FIG. 15 shows a screen shot of another exemplary representation 1500 of the dynamic editing capability of an element. In this embodiment, the EMR provides dynamic edit capability through web-based programming language and a standard browser without a structured programming language such as Java, as seen in editing environment 1400 (of FIG. 14). By providing web-based edit capability, the EMR allows users to edit attributes of an element by invoking one or more hyperlinks to the element when the element is referenced in other web-based reporting or outputs.

[044] State machines are a very powerful extension of normal relational database attributes or column types. As shown in FIG. 5, a state_machine table 120, a state_machine_transition table 122, and a status table 124 describe the structure and behavior of state machines. The term state machine may be used interchangeably with the terms process or workflow. A state machine (e.g., an entry in state_machine table 120) has one or more statuses (e.g., entries in status table 124) which are connected through transitions (e.g., entries in state_machine_transition table 122). The user may

ServiceNow Ex. 1002 Page 152 of 410

classify the elements by declaring one or more possible state machine statuses for each of the elements.

[045] Using the person example, if the person was an employee, then the EMR of FIG. 1 would be able to declare at least one possible status of the person. The EMR could declare the person's status in the state machine model as shown in FIG. 6, such as whether the person is a prospective employee 600, an actively hired employee 610, an employee on probation 620, an employee terminated for cause 630, etc. The EMR may restrict the transition from one status to another by limiting each status's forward transition connections. For instance, the user may first connect prospective employee status 600 to hired status 610. Next, the user may connect hired status 610 to on probation status 620. Furthermore, on probation status 620 may be connected to terminated for cause status 630 or back to hired status 610. Loops and branches are allowable in a stateful process.

[046] The EMR may enforce data behavior and constrain the user's ability to manipulate data based on the state machine model. For one or more stateful models, state machines may be applied to a data classification in a manner that forces data to follow at least one prescribed process declared by the user or a state administrator in a given enterprise. For instance, a prospective employee may not be immediately transitioned to terminated for cause because that violates the rules declared by the process. A prospective employee may only be propagated forward to available or declared next statuses, e.g., hired. Furthermore, the EMR may grant certain users the ability to restrict the condition under which a transition is allowable, for example, only to

users in the organization who are authorized to terminate an employee. Although the transition from actively hired to terminated with cause is generally allowed, there may be one or more exclusionary rules that declare that only a certain user or subset of users may transition an actively hired employee to the status of terminated with cause. Other exclusionary rules might state that the user may not transition an actively hired employee to terminated for cause unless the user has entered a cause and linked it by relation to the employee.

[047] There are many types of data models beyond employees, e.g., the Department of Defense core architectural data model (CADM), the Defense Department architectural framework (DODAF), or the Meta Objects Facility (MOF). These are examples of standardized data models for users to store data, and the models are independent of the data definition that the user wishes to implement. The user can implement multiple concurrent disparate data definitions within a single enterprise repository, such as the EMR of FIG. 1.

[048] Currently, when a system is declared, it implements its own data model according to the user interface and the business rules specification for that system. For the conventional system, the system's developers write dedicated code and user interface code that displays or manages certain data inputs such as check boxes, lists, combo boxes or other data inputs. The conventional system's dedicated code and user interface manipulate the data that the system's application stores and retrieves during operation. Moreover, the conventional system stores and retrieves data locally on a local disk, like a C drive on a Windows system.

[049] There are several drawbacks to conventional systems, one of which is that the data is typically stored in a proprietary file, i.e., a file that is a Microsoft Word® document in .doc format that contains proprietary binary code. The contents of that file are not clear unless one uses Microsoft Word® or another tool that can decode the contents. While Microsoft Word® stores textual information and formats information in a proprietary format contained within an operating system level object called a file, the EMR foregoes the file-based implementation in favor of a elements-based implementation. In the elements-based implementation, the contents of what otherwise would be a file are flexibility represented as elements. Therefore, the contents are not restricted to use by a proprietary application, but are concurrently available to all database-aware applications.

[050] In the elements-based implementation, a document would not be represented by a collection of proprietarily stored formatting and textual information. Instead, the document would be stored as an element with, e.g., one element for each sentence in the document. The elements represent a usable chunk of contextualized information, such as a sentence. In the field of ontologies, words in a sentence by themselves have no meaning. An ontology implies context. Words by themselves may have a definition, but for the words to truly convey meaning, the words need to included in at least a phrase and probably a sentence. In the elements-based approach, each of the elements conforms to the rules of generalized things in the universe, and therefore may be reused or referenced to other things in the universe, e.g., through entries in

element_relation table 162. Moreover, each of the elements in its own context is also meaningful and reusable.

[051] In implementing a solution for a requirements management problem, people would typically use a conventional document to articulate thousands or hundreds or tens of requirements that they would like a system or systems to achieve. In an embodiment of the elements-based implementation in the EMR, each requirement statement would be an element that the user may associate a process to, independent of any kind of hard-coded association. The user has a choice to declare, on each element, such a process which is automatically enforced for that user.

[052] Using the EMR, an element or, in this case, a requirement, would be listed and available as generally readable in an open query-able fashion. This allows the user to organize and aggregate that requirement in more than just the context of one requirements document as the user would in a conventional document-based system. For example, the user may organize and aggregate requirements in a diagrammatic depiction 700 as shown in FIG. 7 that shows congregations of requirements that are related to specific subcomponents of the main system that the user is trying to build.

[053] Diagrammatic depiction 700 shows a subset of elements that may represent requirements as in the previous example. In the example illustrated in diagrammatic depiction 700, there are at least two (2) requirements that are common to the major system components represented in diagrammatic depiction 700. If each generally circular grouping is considered a system component and its related requirements, then the left hand component is related to the right hand component

through the two elements generally in the middle of the diagram and linked simultaneously to both.

[054] For example, consider a situation where the user declares that an engine is needed in the process of building an airplane. In a file-based system, the user either declares that through the requirements themselves or through the practice of engineering the airplane, for which there are a thousand requirements that describe how the engine should mount to the wing. In the file-based system, the requirements may be in several spreadsheets or documents. The requirements may be repeated in the spreadsheet and a document, and may be repeated again in a testing suite application. Moreover, the requirements may be further repeated on the architectural drawing of the component. In the EMR of FIG. 1 there would be one instance of the requirement and it would be shared among all applications that needed to reference that instance of that requirement. Items such as a diagrammatic depiction (e.g. diagrammatic depiction 700), spreadsheet-like depiction, or documentation depiction, are now file independent. The items are just depictions of the core data set.

[055] Embodiments of the present invention include a named_text table 140, as shown in FIG. 8. Named_text table 140 provides a powerful normalization of all text that the user may refer to in a system. Text is a ubiquitous descriptor of many things, including graphics. With current techniques, there are textual representations of graphics, notably scalable vector graphic (SVG) from companies like Adobe®, which are textual descriptions that can be rendered into graphics. For example, a textual description may state "draw a red circle with a 50 pixel radius." The textual description

can be translated into an actual depiction on the screen. This illustrates that text represents most things or most things can be represented by text, not only graphics through the SVG, but also any data structure through extended markup language. The EMR takes advantage of this by declaring these sets of texts and identifying them with a name and a unique identifier. These sets of texts then become usable in the elements-based implementation in many ways.

[056] Embodiments of the present invention include a tuple table 144, as shown in FIG. 9. Tuple table 144 is similar to element_relation table 162, except that tuple table 144 has been separated to store information about the system schema as opposed to the generated user schema. Tuple table 144 relates any identified item to any other identified item and that relationship is created according to a type (e.g., type_id). Given that an entry in named_text table 140 is a thing and that the user may relate the thing to other types of named text, the EMR declares, for the user's convenience, a named_text item that is akin to a folder in an operating system hierarchy. The folder may be called, for example, "My Favorites," or "My Documents," or in this instance, "My Elements." That folder can then be referenced by the user. Next, the EMR may declare 2, 3, 5 or 100 (or any number of) sub-folders. The sub-folders may be referenced by entries in tuple table 144 to the folder and other sub-folders.

[057] The combination of named_text table 140 and tuple table 144 provides to the EMR the capability to create a recursion between named_text types that represent folders to other named_text types that represent folders in a hierarchy. In an

embodiment shown in FIG. 9, the EMR may contain a folder called root (e.g. "EE Root"), another folder called "My elements" (e.g. "My Elements (MARK)" and "My Elements (RONO)"), and tuples the root entry in named_text table 140 to the "My Elements" entry in named_text table 140. In doing so, the EMR indicates through elements explorer 900 a hierarchy or a tree or an indenture of the folders. This particular implementation approximates a standard operating system file explorer, in which the user can click and drag and drop folders and other items into a folder in a tree or navigable folder set. Through the same mechanism, an element may be tupled into a folder. In the elements-based implementation, the EMR may use a folder to organize and aggregate random sets of things in the universe.

[058] The EMR may utilize entries in named_text table 140 to name structured query language queries. SQL is a textual language that describes an interaction or request to a relational database. SQL queries are the mechanism to retrieve data from the database. SQL query is a powerful notion for relational databases and may be implemented as a named_text entry. Unlike a conventional database, e.g., Oracle® or Microsoft SQL Server®, that stores user-generated SQL as a "view," the EMR stores user-generated SQL as a named_text entry. Because the user-generated SQL is an entry in named_text table 140, and since folders exist as entries in named_text table 140 or tuple table 144, the EMR may filter or query entries in named_text table 140 or tuple table 144 into one or more folders.

[059] Thus, if the SQL statement serves as a folder, the results of the SQL statement or the items returned by the query would be the contents of that folder. The

SQL statement to the user is therefore a dynamic folder or a SQL folder and no tuple exists between that data set. In other words, elements in the dynamic folder can be placed into other folders. The dynamic folder is executed at run-time if the user selects the dynamic folder and the resulting elements that come back from the query are presented as if they were tupled into that folder. This is a dynamic process and the result set may differ depending on the dynamic data changes. This dynamic process simplifies the organization and provides to the user the ability to locate data through a dynamic folder hierarchy. The dynamic folder hierarchy further includes dynamic folders that change the data set appropriately every time the user visits them.

[060] An example of the dynamic folder capability provided by the EMR is a "My Tasks" folder. Assume that task is an element classification and that user is an attribute of the task. If there are tasks in the system that are assigned to the user, the user may perform a query that directs the EMR to select one or more tasks for which the user is the assigned user. The result of that query may change over time. The administrator may assign a task to someone else, so the task may be deleted or moved to a completed state and the user's folder would only contain a current task set that matches the query as defined by a named_text of that folder. This allows for an extensible method of organizing data of interest and allowing the user to easily find and retrieve data of interest in a large, complex system.

[061] Embodiments of the present invention, as shown in FIG. 10, include a poll table 182. Poll table 182 may represent a set of identified questions that are in one of several stages of life cycle of an element or set of elements. A poll is a collaborative

mechanism that restricts the responses of the participants to one or more predetermined choices which are enumerated in a poll_option table 188. The replies of the participants are recorded in a poll_reply table 184. These poll replies can be analyzed for preference among the voting community and used to record decision making processes regarding any given element or set of elements. This relation is achieved through entries in tuple table 144 (in FIG. 8).

[062] In accordance with still further embodiments of the invention, the EMR includes a discussion table 186, which is similar to poll table 182 as discussion table 186 records collaborative information regarding one or more elements in the system. However, a discussion, which may be stored as entry in discussion table 186, is not forced into a set of structured responses. Instead discussion table 186 allows free-form narrative to be recorded, and replied to by any number of users in the system.

[063] Embodiments of the present invention may further include a collaboration_status_history table 180, which may record all the states that either a discussion or poll moves through in a defined life cycle, similar to element history. Moreoever, the user and date and time is recorded with each status change.

[064] In accordance with still further embodiments of the invention, the EMR includes a system_component table 190. The EMR may register components that are programmatic in nature (e.g. Java classes), wherein the components may be executed dynamically as needed (through a process called reflection) as part of element decoration. For example, one or more Java classes may be defined and saved as a record in system_component table 190, then the Java classes may be tupled to

elements. At runtime, the Java classes that are referenced may be executed when an element of the Java class is edited or viewed.

[065] Embodiments of the present invention, as shown in FIG. 11, include a terminology table 130 and a glossary 132. Terminology table 130 may store a list of terminologies that can be used in the system. Glossary table 132 may uniquely identify and define each of the terms stored in terminology table 130.

[066] In accordance with embodiments of the invention as shown in FIG. 12, the EMR includes an email table 172 and an email_detail table 170. These tables allow the EMR to populate records in the database that can programmatically be converted from rows in the RDBMS to actual emails using a known protocols such as SMTP and POP. Since the emails stored in email table 172 and email_detail table 170 are also data, the EMR may apply stateful process logic to each email - using status, state_machine, and state_machine_transition tables (FIG. 5, elements 120, 122, 124, respectively) to define the lifecycle of an email. Full email history can be maintained in the EMR and new emails can be generated either programmatically when an event occurs, or manually when a user enters an email. Individual messages may be aggregated into one larger email at a time interval specified by an administrator. The aggregated messages may also be sent on demand.

[067] In accordance with embodiments of the invention as shown in FIG. 13, the EMR includes a formatted_dataset table 176 and a SQL_XSLT table 178. Together, formatted_dataset table 176 and SQL_XSLT table 178 allow the EMR to create a named "formatted dataset," which applies formatting to a SQL result set.

ServiceNow Ex. 1002 Page 162 of 410

Formatted_dataset table 176 and SQL_XSLT table 178 may work in conjunction with named_text table 140 to relate one or more named text entries (representing SQL statements), which provide the result set(s), with another named text entry (representing XSLT or a formatting definition) which generates a final layout.

[068] To further illustrate the EMR as described above and in FIGS. 1-15, an example is provided to demonstrate the utilization and capability of the EMR, consistent with the present invention, for storing, managing, indexing, interrelating, and/or retrieving data sets, such as large scale and dynamic data sets. One example, as shown in FIG. 16, may be Human Resource Management (HRM). Further examples may include, for instance, a vertical implementation, or schema and associated processes, that may serve the needs of a well defined market, e.g., Enterprise Architecture, Requirements Management, Six Sigma, Resource Planning, Federal Regulatory Compliance, Legal Document Management, Portfolio Management, etc.

[069] In the HRM example, it may be determined that tracking "resources" is desired. Those resources may be people, but may not necessarily be restricted to people, which is a distinction that may be made when defining the implementation. It may further be determined that the resources of interest will in fact be people, and those people will be employees of a specified company. The company may have written policies that determine employment, such as the amount of vacation time each employee may receive, a level of compensation, etc.

[070] Furthermore, it may be determined that the resources may exist in a hierarchy, e.g., have a place in an organization chart, and that tasks should be identified

and assigned to these resources as needed. Steps may be taken within the EMR to support such system requirements.

[071] A workspace (e.g., HRM Workspace 1600 of FIG. 16) may optionally be entered into the ERM (e.g. workspace_object table 106 and workspace table 108 of FIGS. 1 and 2) to segregate the HRM information from other possible data sets in the EMR if desired.

[072] A schema would be logically defined that encompassed the articulated needs of the system. In this case perhaps a resource class 1602 (of FIG. 16) would be declared with attributes of employee_name, employee_ID, and employee_status. The employee_status attribute may be of type state_machine, allowing the stateful process of employment to be declared according to the rules of the company.

[073] It may further be determined that a resource may change title and salary over time, and other classes, such as a resource_title class 1606 and a resource_compensation class 1608, would be declared each with a pointer to resource class 1602 such as resource_ID and other appropriate attributes such as resource_title, effective_start_date, and effective_end_date. Furthermore, resource_compensation class 1608 may similarly contain compensation_amount, compensation_interval, compensation_start_date, and compensation_end_date.

[074] Tasks would also be similarly defined as a task class 1604 with, for example, a name, a description, a milestone indicator, a status, an assignee, an estimated_duration, and a project indicator.

ServiceNow Ex. 1002 Page 164 of 410

[075] To implement these classes, data would be entered into the EMR representing these structures. The order of implementation, as illustrated below, is exemplary and does not limit embodiments of the present invention.

[076] Entries may be made into class table 150 (of FIGS. 1 and 2), one each for resource, task, resource_title, and resource_compensation. Entries may be made into attribute table 110 (of FIGS. 1 and 2) for each employee_name, employee_ID, employee_status, employee_title, compensation_interval, compensation_amount, compensation_start_date, compensation_end_date, status, assignee, estimated_duration, and project.

[077] As further definition, entries may be made into state_machine table 120, state_machine_transition table 122, and status table 124 (of FIGS. 1 and 5) to declare the available selections of the status and employee_status attributes. Further entries may be placed into list_option table 114 (of FIGS. 1 and 2) to specify the available resource_titles that may be assigned to resource or resources.

[078] Entries may then be made into class_attribute table 112 (of FIGS. 1 and 2) to associate the appropriate attributes with their respective classes. This may all be done via any data insertion method available, e.g. a user interface, etc.

[079] Once these class_attribute records have been entered into class_attribute table 112, the EMR is ready for use. Users have the capability to add resources, create tasks, assign them to the resources, report on how many tasks are completed, etc. Users may also create folders to organize projects (see, e.g., FIG. 9) and view diagrams (see, e.g., FIGS. 4 and 7) of which resources are committed to which tasks without any

further programmatic effort. Users may also be able to collaborate on tasks by discussing ramifications or polling a subset of users to determine specific answers to to-be-determined elements.

[080] In this example, the resource hierarchy may be constructed by linking one resource to another recursively (see, e.g., FIG. 16). This may be done either via the record editing capabilities or by using the diagramming tool (see, e.g., FIGS. 4 and 7) to draw a line from one resource to another. The diagrammatic representation of the organization chart is capable of immediately expand an individual to supervisors, peers, or subordinates. The diagrammatic representation may also show all related tasks to one or more of those individuals if desired. By providing those capabilities, the diagrammatic representation allows for visualization of tasking overloads or underutilized resources, etc.

[081] If the EMR does not provide a specific aggregation or subsetting of tasks or resources, one or more named_text entries in named_text table 140 (of FIGS. 1 and 8) may be defined that return the appropriate results. If the default display of those results is insufficient for the business needs, then one or more formatted_dataset entries in formatted_dataset table 176 (of FIGS. 1 and 13) and SQL_XSLT entries in SQL_XSLT table 178 (of FIGS. 1 and 13) may be defined to convert the available results into any desired final output.

[082] In this example, when the user edits or inserts data in the EMR, the data may be stored in an appropriate element_attr_c# table 164 (of FIGS. 1 and 3) that was automatically constructed as part of the class definition step above. Moreover, any

modifications may automatically be recorded as history in element_history table 166 (of FIGS. 1 and 3).

[083] The above-described exemplary scenario may be followed for any given schema. A full schema does not need to be defined before the EMR may be used. If for instance only resources were defined, then resource information could be entered, and organization hierarchies may be defined at a later date; the tasking information may be added without disruption to the EMR. As a further example, additional attributes may be added to the resources themselves at a later date if the attributes are deemed necessary. As another example, the EMR's user interfaces may automatically adjust to the current definition to allow seamless upgrades of the data models.

[084] Systems and methods consistent with the present invention may be implemented by computers organized in a conventional distributed processing system architecture. FIG. 17 is a block diagram illustrating a computer system 200 in which an embodiment consistent with the invention may be implemented. Computer system 200 includes a computer 201 having a bus 202 or other communication mechanism for communicating information, and a processor 204 coupled with bus 202 for processing information. Computer 201 also includes a main memory, such as random access memory (RAM) 206 or other dynamic storage device, coupled to bus 202 for storing information and instructions by processor 204. RAM 206 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 204. During operation, some or all portions of an operating system (not shown) are loaded into RAM 206. Computer 201 further includes a read

only memory (ROM) 208 or other static storage device coupled to bus 202 for storing static information and instructions for processor 204. A storage device 210, such as a magnetic disk or optical disk, is provided and coupled to bus 202 for storing information and instructions.

[085] Computer 201 may be coupled via bus 202 to a display 212, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 214, including alphanumeric and other keys, is coupled to bus 202 for communicating information and command selections to processor 204. Another type of user input device is cursor control 216, such as a mouse, a trackball or cursor direction keys for communicating direction information and command selections to processor 204 and for controlling cursor movement on display 212. This input device may have two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), which allow the device to specify positions in a plane.

[086] System 200 may further include an input drive device 217, such as a CD reader, for reading high-capacity computer-readable media, such as CD-ROMs and CDRs.

[087] The invention is related to the use of computer system 200 for manipulating data sets for, to take an example, corporate or engineering exercises. According to one implementation, systems and methods consistent with the present invention collect system information and store it in a central database in response to processor 204 executing one or more sequences of one or more instructions contained in RAM 206. Such instructions may be read into RAM 206 from a computer-readable

medium via an input device such as storage device 210. Execution of the sequences of instructions contained in main memory 206 causes processor 204 to perform the process steps described herein. In an alternative implementation, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, implementations consistent with the principles of the present invention are not limited to any specific combination of hardware circuitry and software.

[088] Computer 201 also includes a communication interface 218 coupled to bus 202. Communication interface 218 provides a two-way data communication coupling to a network link 220 that is connected to local network 222. For example, communication interface 218 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 218 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 218 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[089] Network link 220 typically provides data communication through one or more networks to other data devices. For example, network link 220 may provide a connection through local network 222 to host computer 224 and/or to data equipment operated by Internet Service Provider (ISP) 226. ISP 226, in turn, provides data communication services through the Internet 228. Local network 222 and Internet 228 both use electric, electromagnetic or optical signals that carry digital data streams. The

ServiceNow Ex. 1002 Page 169 of 410

signals through the various networks and the signals on network link 220 and through communication interface 218, which carry the digital data to and from computer system 200, are exemplary forms of carrier waves transporting the information.

[090] Computer system 200 can send messages and receive data, including program code, through the network(s), network link 220 and communication interface 218. In the Internet example, a server 230 might transmit a requested code for an application program through Internet 228, ISP 226, local network 222 and communication interface 218. The received code may be loaded in RAM 206 and executed by processor 204 as it is received. Alternatively, or in addition, it may be stored in storage device 210, or other non-volatile storage for later execution. In this manner, computer system 200 may obtain application code in the form of a carrier wave.

[091] Although computer system 200 is shown in FIG. 17 as being connectable to one server 230, those skilled in the art will recognize that computer system 200 may establish connections to multiple servers on Internet 228. Such servers may include an HTML-based Internet application, which may provide information to computer system 200 upon request in a manner consistent with the present invention.

[092] The foregoing descriptions of the invention have been presented for purposes of illustration and description. They are not exhaustive and do not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but the present invention may

be implemented as a combination of hardware and software or in hardware alone. Additionally, although aspects of the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM.

ServiceNow Ex. 1002 Page 171 of 410

WHAT IS CLAIMED IS:

1. A computerized system for manipulating data sets, comprising:

an element module configured to store and uniquely identify a plurality of elements;

an element relation module configured to store relationships between the plurality of elements in the element module;

a class module configured to store attributes of the plurality of elements in a class;

a type definition module configured to define the class and the attributes related to the class; and

a state machine module, the state machine module including a state machine transition module and a status module.

2. The computerized system of claim 1, wherein the state machine module is configured to store a first status of one element and a transition to a second status of the one element.

3. The computerized system of claim 2, further comprising:

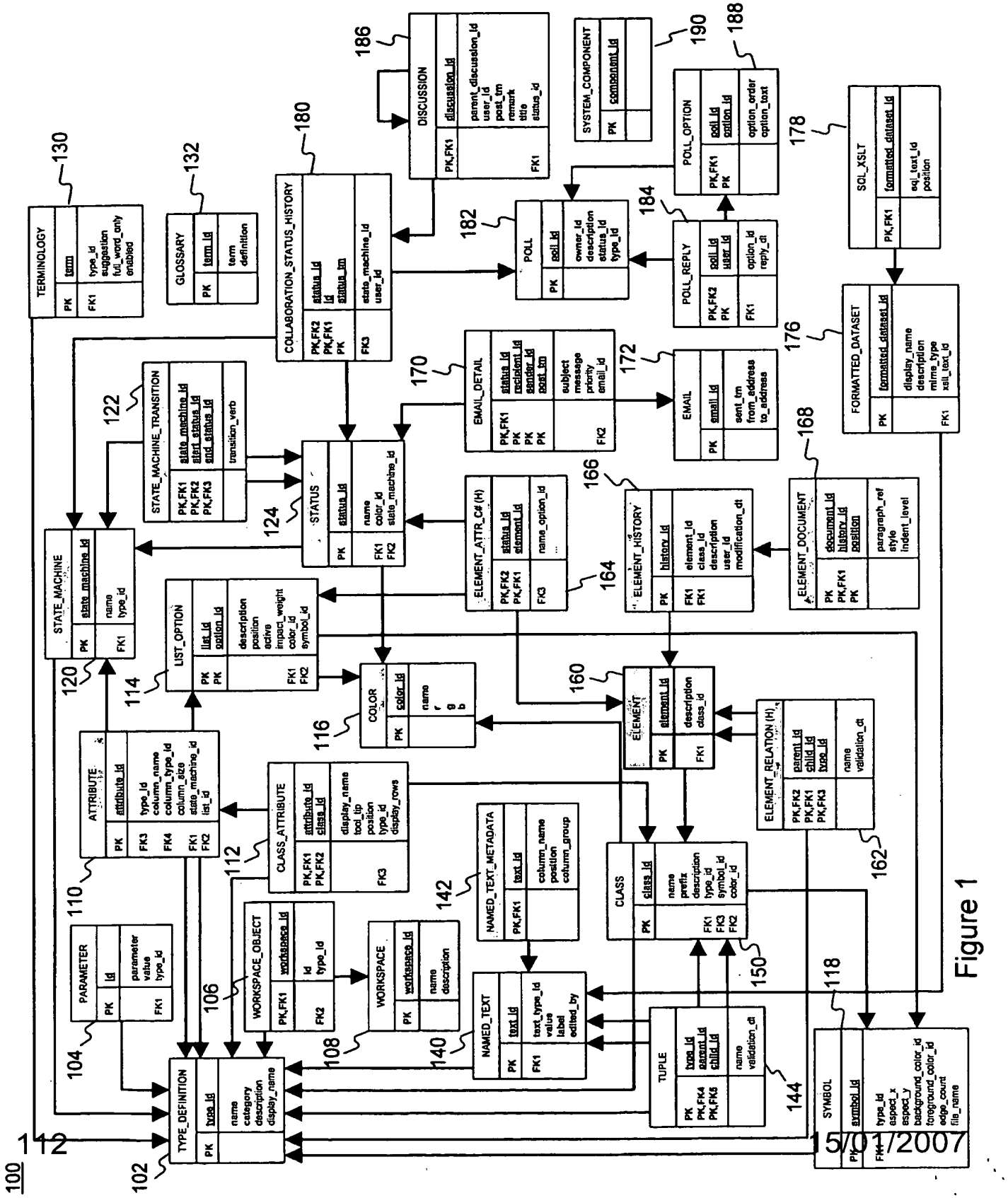
a named text module configured to store textual representations of query, control, and display mechanisms that relate to the plurality of elements.

Provisional Patent Application
Customer No.: 22,852
Attorney Docket No.: 09801.6000-00

ABSTRACT OF THE DISCLOSURE

Systems and methods are provided for manipulating data sets. In accordance with one implementation, a computerized system is provided for storing, managing, indexing, interrelating, and/or retrieving data sets in a manner independent of the data model. The system includes an element module configured to store and uniquely identify elements and an element relation module configured to store relationships between the elements in the element module. The computerized system may also comprise a class module configured to store attributes of elements in a class and a type definition module configured to define the class and the attributes related to the class. The computerized system may further comprise a state machine module, the state machine module including a state machine transition module and a status module.

ServiceNow Ex. 1002 Page 173 of 410


100
102
104
106
108
110
112
114
116
118
120
122
124
130
132
140
142
144
150
160
162
164
166
168
170
172
176
178
180
182
184
186
188
190
TERMINOLOGY
GLOSSARY
DISCUSSION
SYSTEM_COMPONENT
COLLABORATION_STATUS_HISTORY
POLL
POLL_OPTION
SQL_XSLT
POLL_REPLY
FORMATTED_DATASET
EMAIL_DETAIL
EMAIL
STATE_MACHINE_TRANSITION
STATE_MACHINE
STATUS
ELEMENT_ATTR_C# (H)
ELEMENT_HISTORY
ELEMENT_DOCUMENT
LIST_OPTION
COLOR
ELEMENT
ATTRIBUTE
CLASS_ATTRIBUTE
ELEMENT_RELATION (H)
PARAMETER
WORKSPACE_OBJECT
NAMED_TEXT_METADATA
CLASS
WORKSPACE
NAMED_TEXT
TUPLE
TYPE_DEFINITION
SYMBOL
Figure 1
15/01/2007

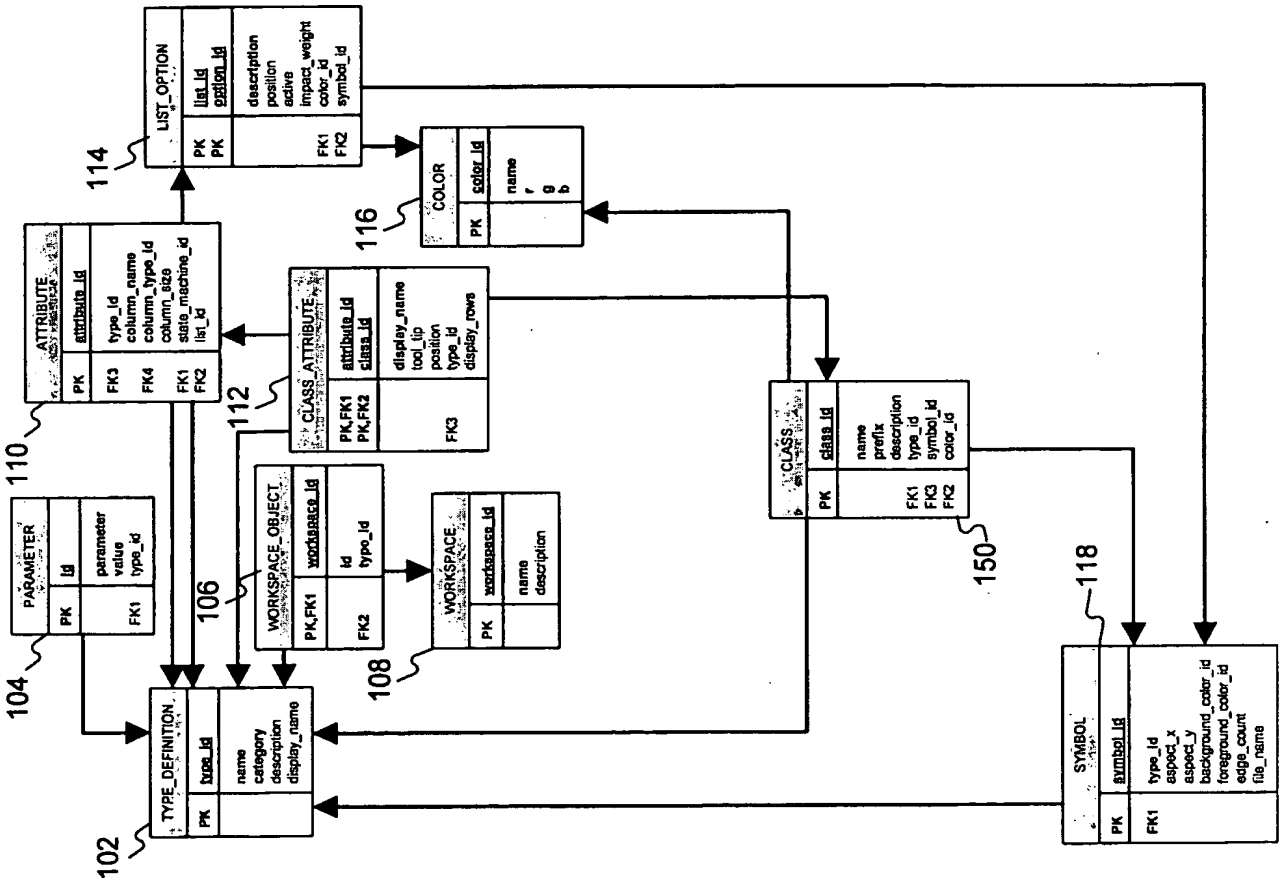


Figure 2

113

15/01/2007

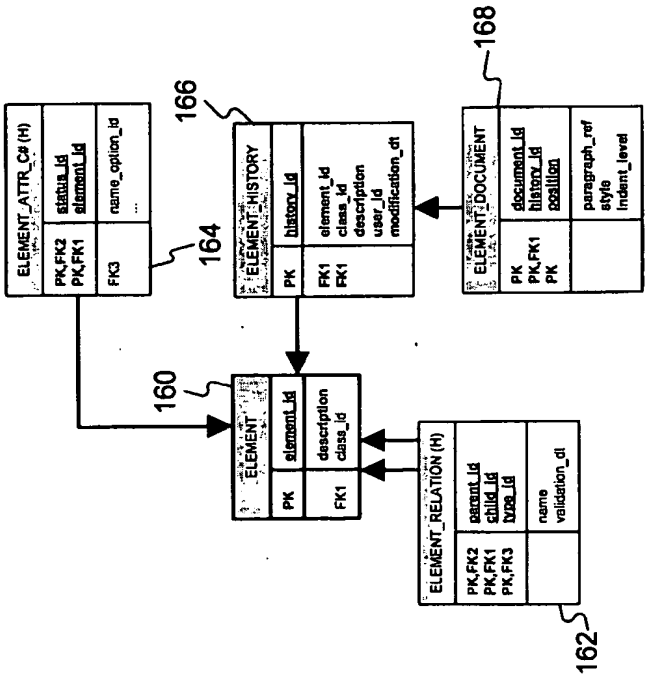
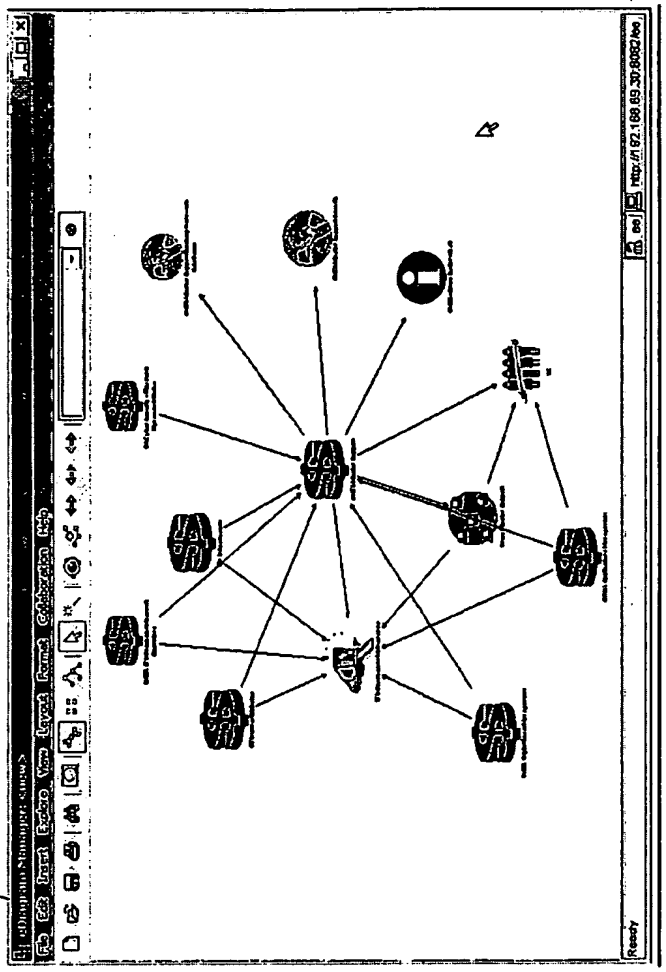

ELEMENT_ATTR_C# (H)
PK,FK2 status_id
PK,FK1 element_id
FK3 name_option_id
...
164
166
ELEMENT_HISTORY
PK history_id
FK1 element_id
FK1 class_id
description
user_id
modification_dt
168
ELEMENT_DOCUMENT
PK document_id
PK,FK1 history_id
PK position
paragraph_ref
style
indent_level
160
ELEMENT
PK element_id
description
FK1 class_id
ELEMENT_RELATION (H)
PK,FK2 parent_id
PK,FK1 child_id
PK,FK3 type_id
name
validation_dt
162

Figure 3



115

Figure 4

15/01/2007

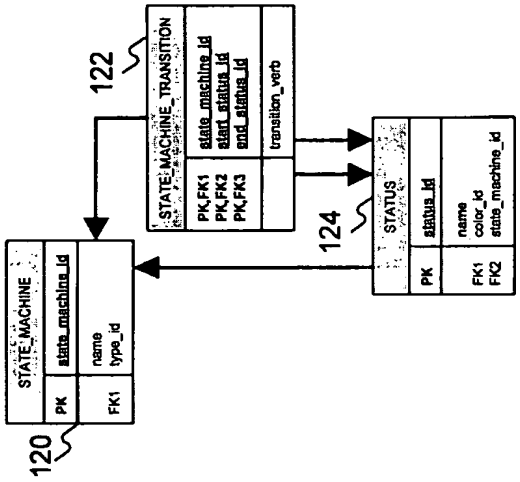


Figure 5

116

15/01/2007

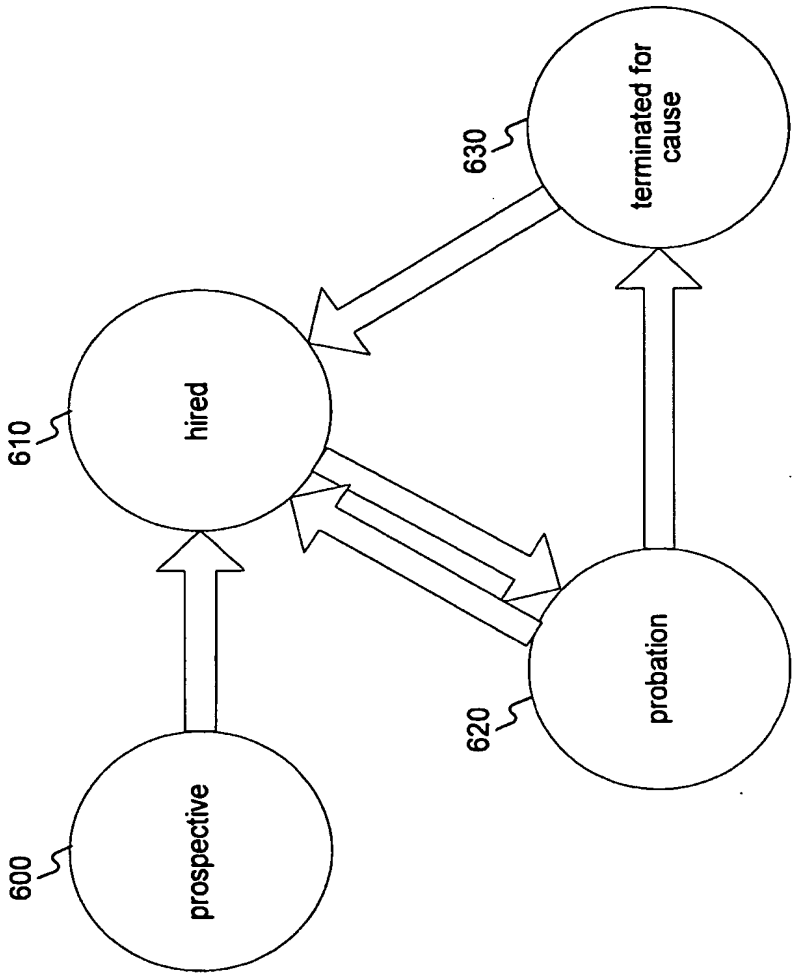
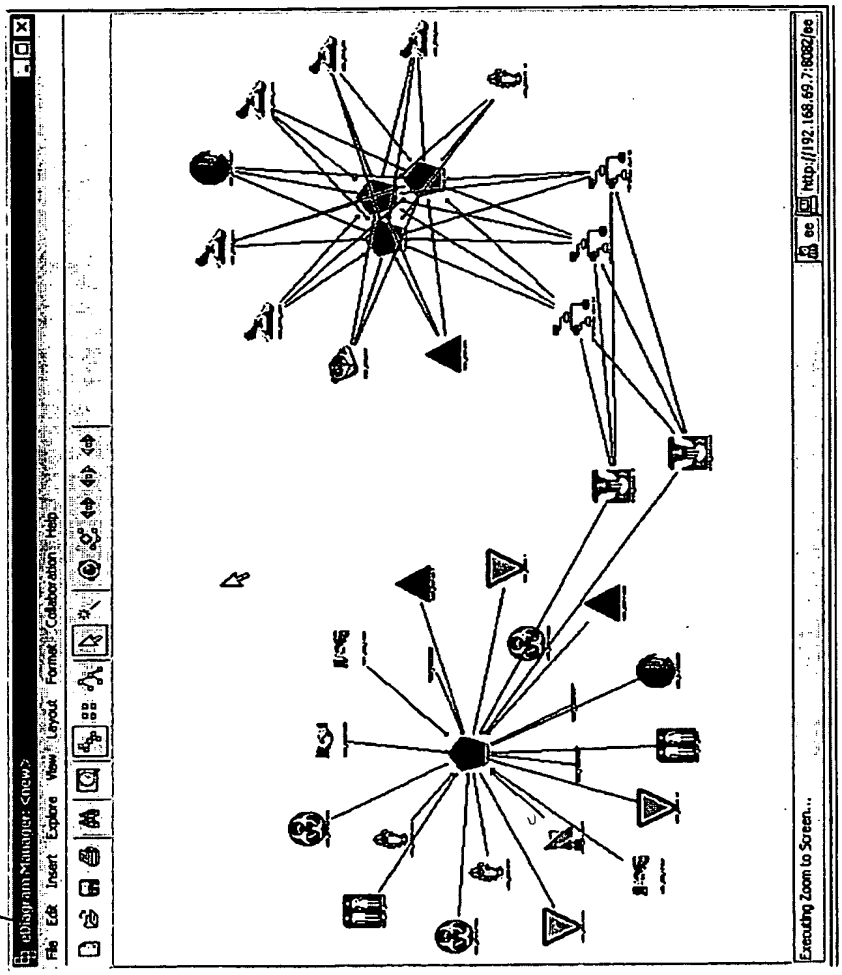


Figure 6

ServiceNow Ex. 1002 Page 179 of 410



118

15/01/2007

Figure 7

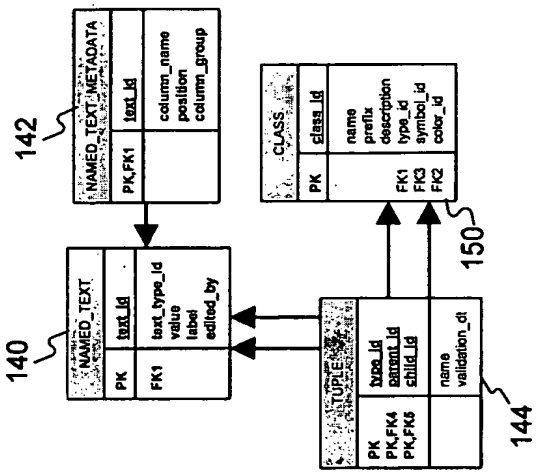
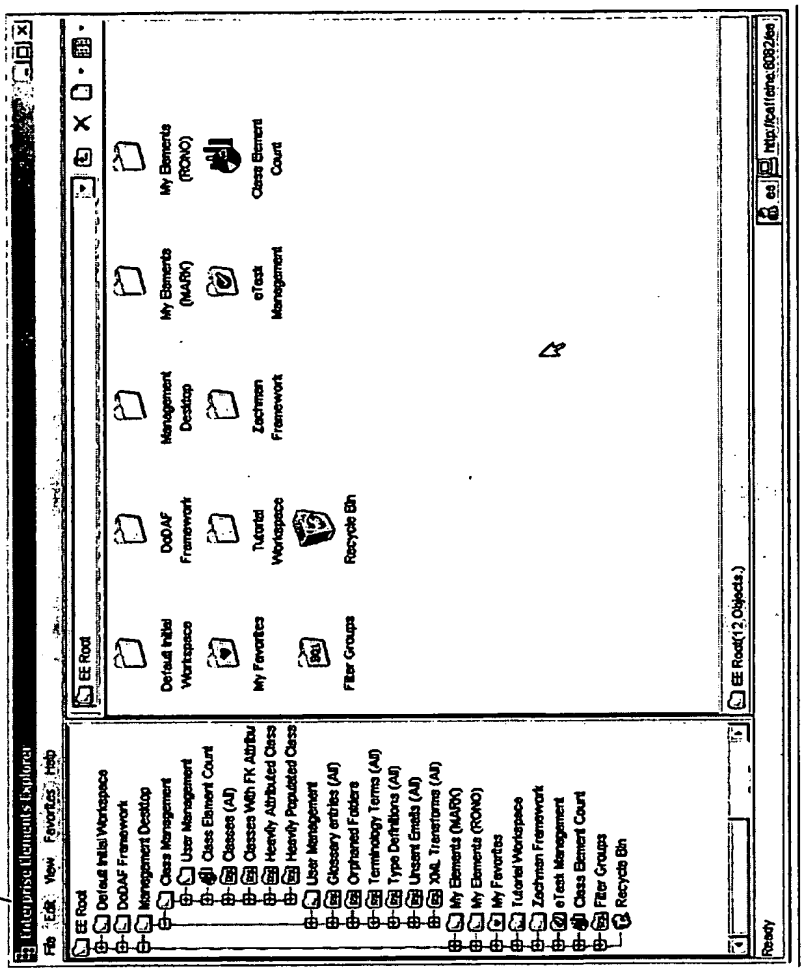

142
NAMED_TEXT_METADATA
PK,FK1
text_id
column_name
position
column_group
140
NAMED_TEXT
PK
FK1
text_id
text_type_id
value
label
edited_by
CLASS
PK
class_id
name
prefix
description
type_id
symbol_id
color_id
FK1
FK3
FK2
150
TUPLE
PK
PK,FK4
PK,FK5
type_id
parent_id
child_id
name
validation_dt
144

Figure 8



120

15/01/2007

Figure 9

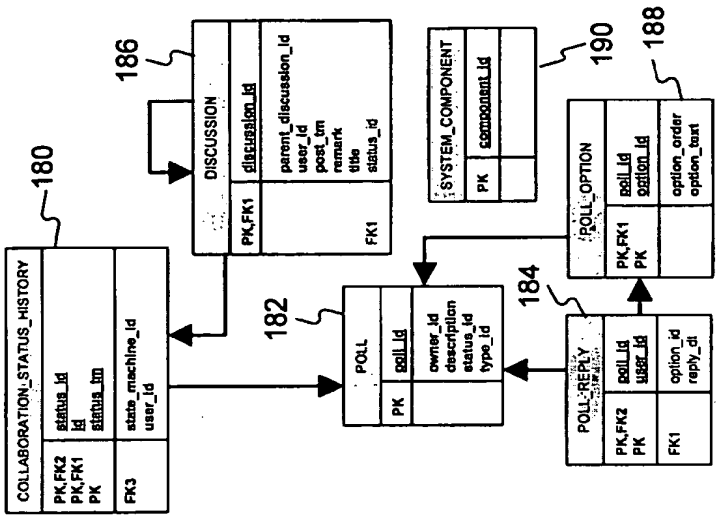

COLLABORATION_STATUS_HISTORY
PK,FK2 status_id
PK,FK1 id
PK status_tm
FK3 state_machine_id
user_id
180
DISCUSSION
PK,FK1 discussion_id
parent_discussion_id
user_id
post_tm
remark
title
FK1 status_id
186
SYSTEM_COMPONENT
PK component_id
190
188
POLL_OPTION
PK,FK1 poll_id
PK option_id
option_order
option_text
182
POLL
PK poll_id
owner_id
description
status_id
type_id
184
POLL_REPLY
PK,FK2 poll_id
PK user_id
FK1 option_id
reply_dt

Figure 10

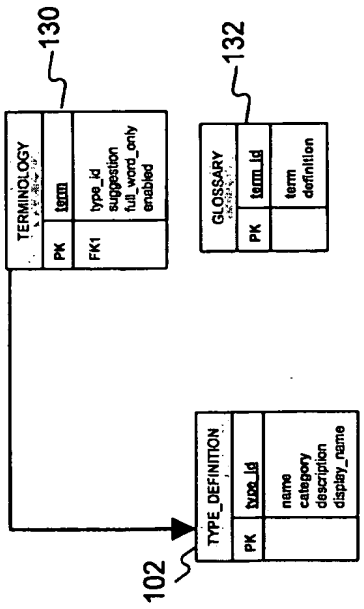


Figure 11

122

15/01/2007

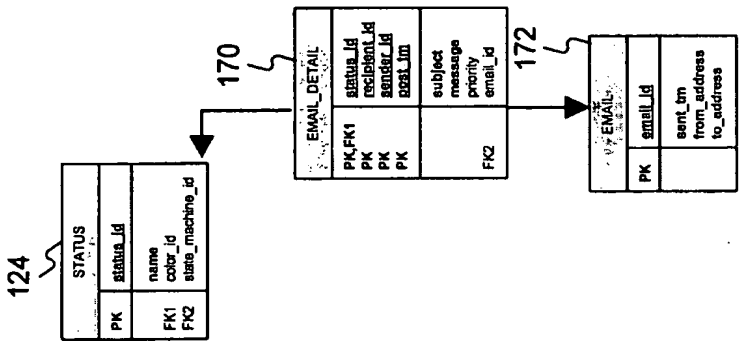


Figure 12

ServiceNow Ex. 1002 Page 185 of 410

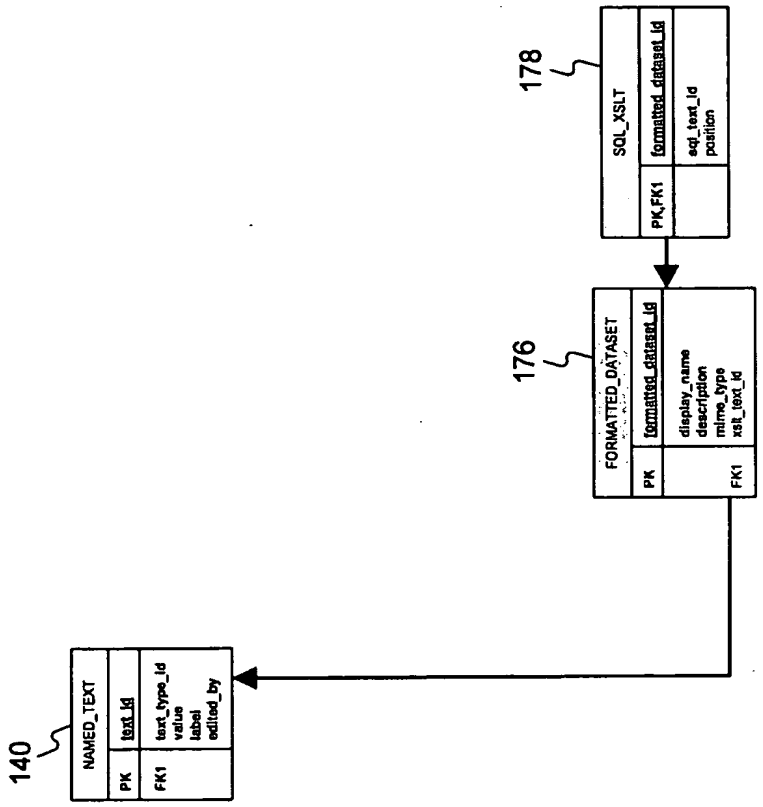

178
SQL_XSLT
formatted_dataset_id
PK,FK1
seq_text_id
position
176
FORMATTED_DATASET
formatted_dataset_id
PK
display_name
description
mime_type
xslt_text_id
FK1
140
NAMED_TEXT
text_id
PK
text_type_id
value
label
edited_by
FK1

Figure 13

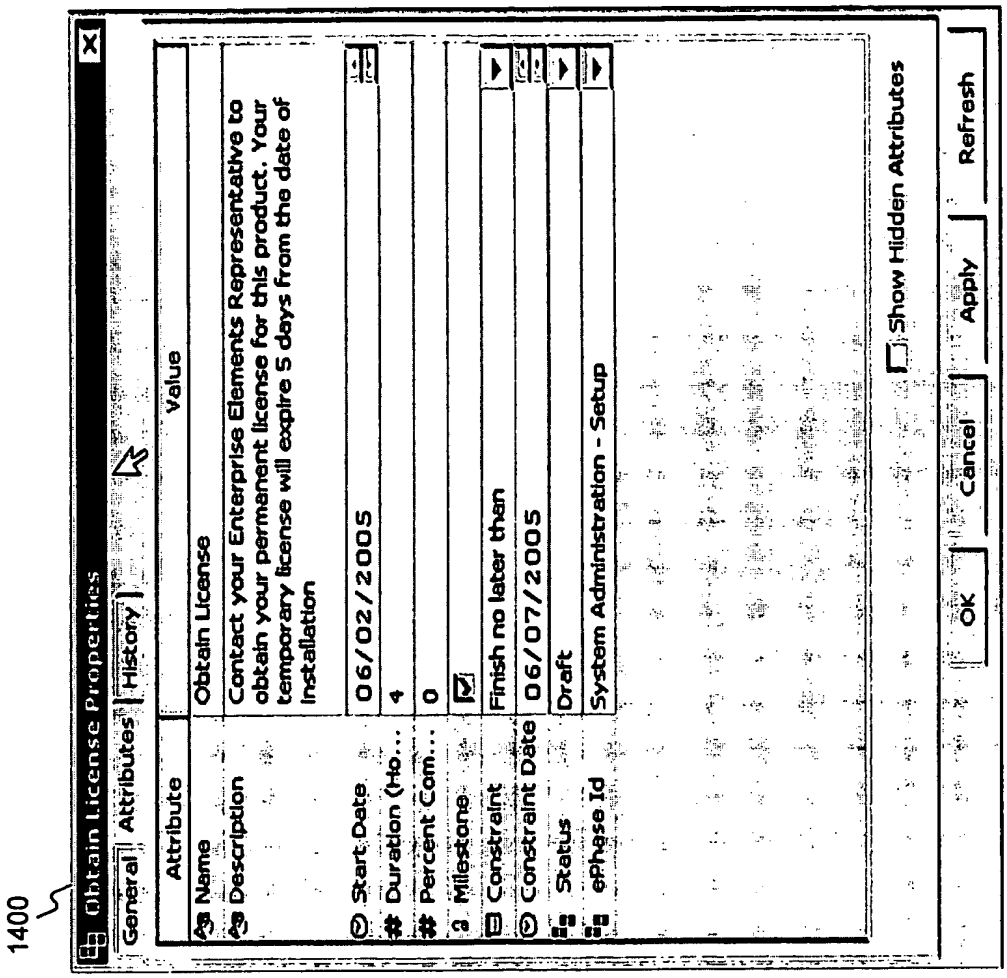

1400
Obtain License Properties
General
Attributes
History
Attribute
Value
Name
Obtain License
Description
Contact your Enterprise Elements Representative to obtain your permanent license for this product. Your temporary license will expire 5 days from the date of installation
Start Date
06/02/2005
Duration (Ho...
4
Percent Com...
0
Milestone
Constraint
Finish no later than
Constraint Date
06/07/2005
Status
Draft
ePhase Id
System Administration - Setup
Show Hidden Attributes
OK
Cancel
Apply
Refresh

Figure 14

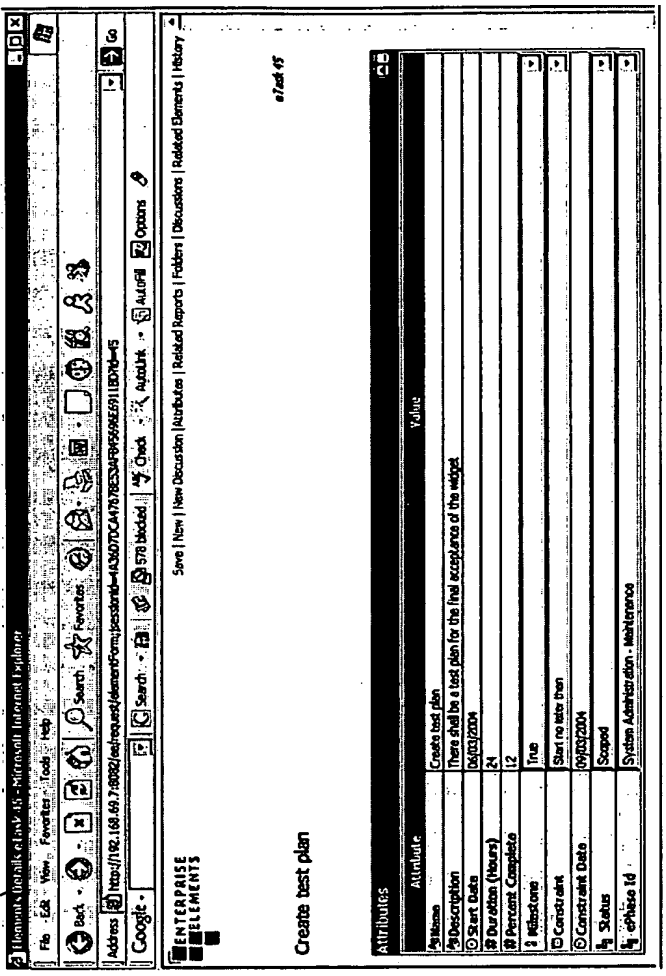

1500
ENTERPRISE ELEMENTS
Save | New | New Discussion | Attributes | Related Reports | Folders | Discussions | Related Elements | History
Create test plan
Attributes
Attribute
Value
Name
Create test plan
Description
There shall be a test plan for the final acceptance of the widget
Start Date
06/03/2004
Duration (Hours)
24
Percent Complete
12
Milestone
True
Constraint
Start no later than
Constraint Date
09/03/2004
Status
Scoped
ePulse Id
System Administration - Maintenance

Figure 15

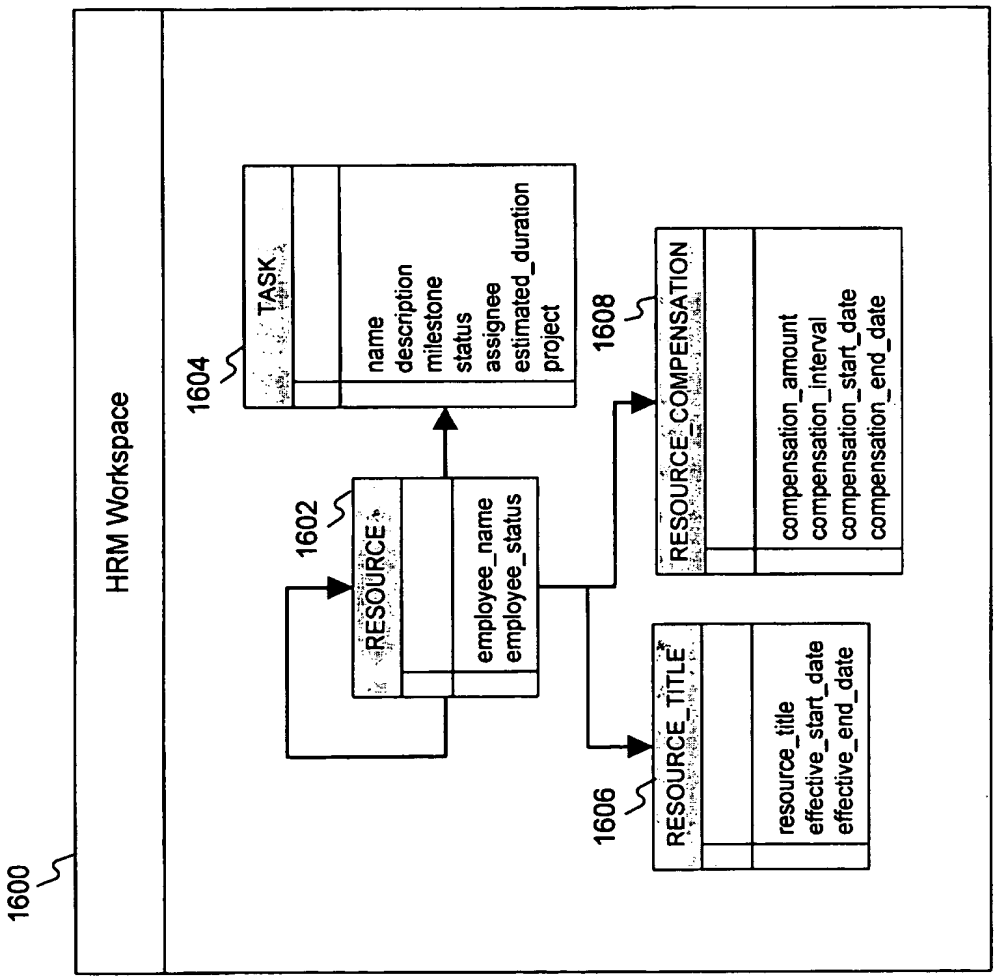

HRM Workspace
1600
TASK
1604
name
description
milestone
status
assignee
estimated_duration
project
RESOURCE
1602
employee_name
employee_status
RESOURCE_COMPENSATION
1608
compensation_amount
compensation_interval
compensation_start_date
compensation_end_date
RESOURCE_TITLE
1606
resource_title
effective_start_date
effective_end_date

Figure 16

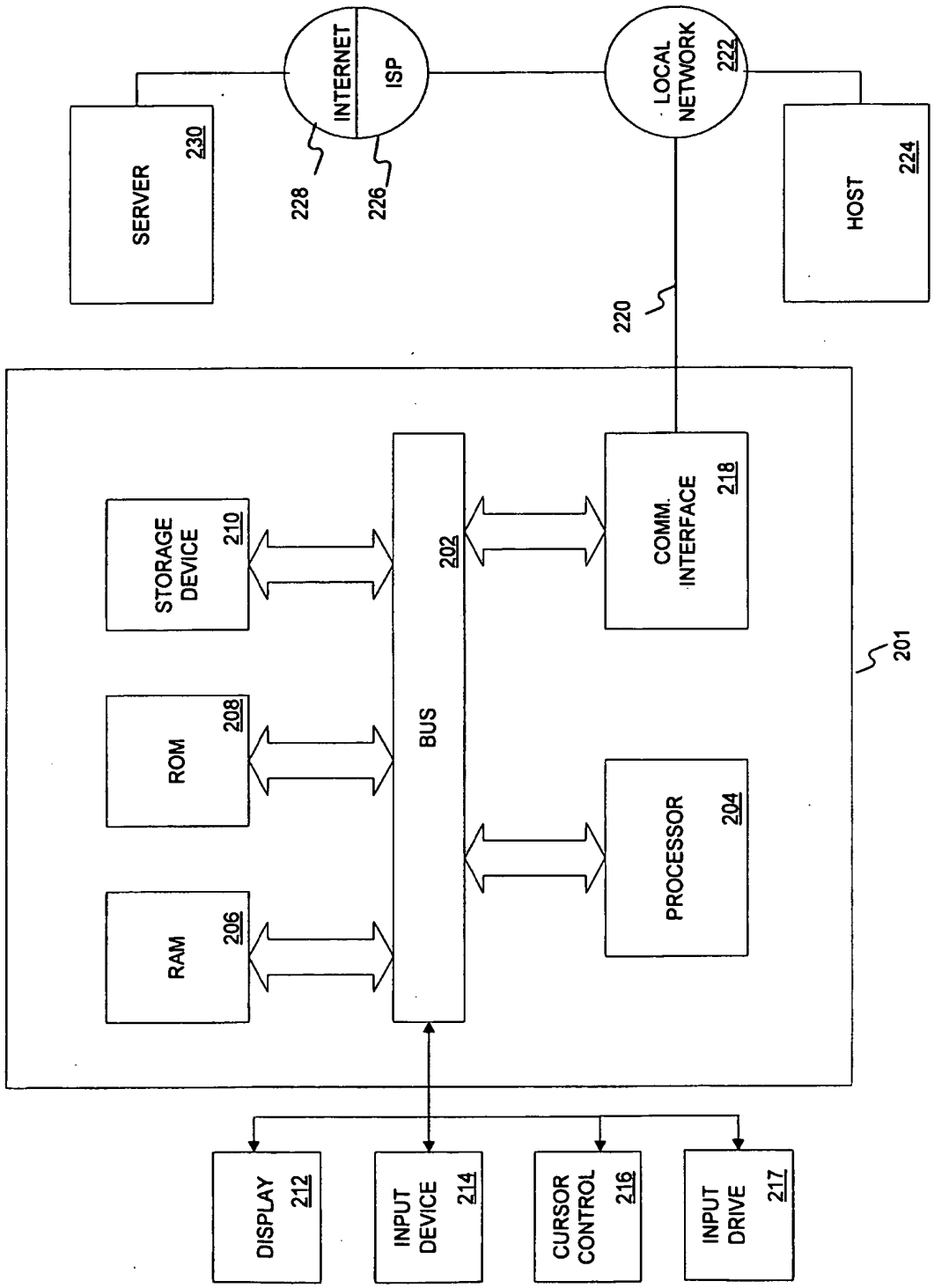

INTERNET
ISP
LOCAL NETWORK
222
SERVER
230
228
226
220
HOST
224
201
STORAGE DEVICE
210
ROM
208
RAM
206
BUS
202
COMM. INTERFACE
218
PROCESSOR
204
DISPLAY
212
INPUT DEVICE
214
CURSOR CONTROL
216
INPUT DRIVE
217
200

Figure 17

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☑ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☑ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

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

(19) World Intellectual Property Organization
International Bureau

PCT

(43) International Publication Date
28 December 2006 (28.12.2006)

(10) International Publication Number
WO 2006/138420 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/US2006/023242

(22) International Filing Date: 14 June 2006 (14.06.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/690,147 14 June 2005 (14.06.2005) US

(71) Applicant *(for all designated States except US)*: **ENTERPRISE ELEMENTS, INC.** [US/US]; 11654 Plaza America Drive, Suite 639, Reston, VA 20190-4700 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: **DEWOOLFSON, Bruce, Randall** [US/US]; 46290 Summerhill Place, Potomac Falls, VA 20165 (US).

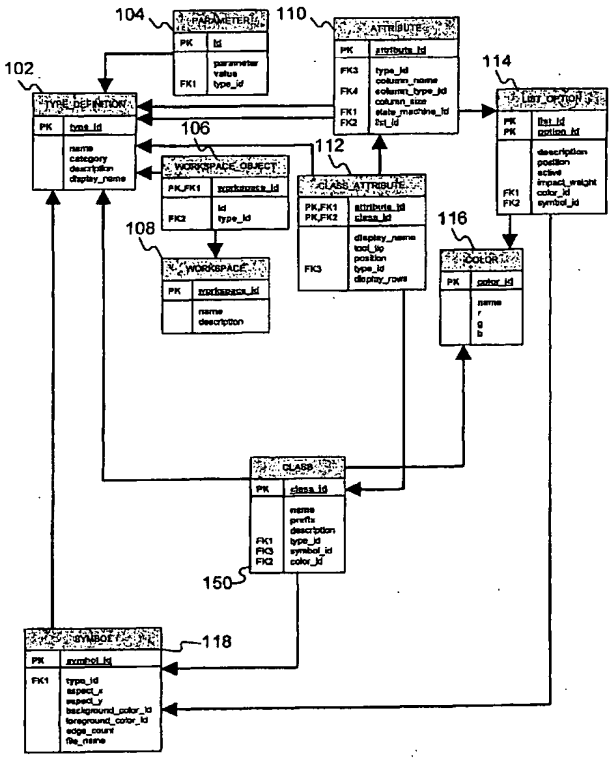
(74) Agents: **GARRETT, Arthur, S.** et al.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.l.p., 901 New York Avenue, Nw, Washington, DC 20001-4413 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,

[Continued on next page]

(54) Title: DATABASE DATA DICTIONARY



(57) Abstract: Systems and methods are provided for manipulating data sets. In accordance with one implementation, a computerized system is provided for storing, managing, indexing, interrelating, and/or retrieving data sets in a manner independent of the data model. The system includes an element module configured to store and uniquely identify elements and an element relation module configured to store relationships between the elements in the element module. The computerized system may also comprise a class module configured to store attributes of elements in a class and a type definition module configured to define the class and the attributes related to the class. The computerized system may further comprise a state machine module, the state machine module including a state machine transition module and a status module.

WO 2006/138420 A1

130

28/12/2006

FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

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

DATABASE DATA DICTIONARY

DESCRIPTION

Cross-Reference to Related Application(s)

[001] This application claims the benefit of priority from U.S. Provisional Application No. 60/690,147, entitled "Database Data Dictionary," filed June 14, 2005, the disclosure of which is expressly incorporated herein by reference to its entirety.

Technical Field

[002] This invention relates to the field of computer data storage. More particularly, the present invention relates to methods and systems for storing, managing, indexing, interrelating, and/or retrieving data sets.

Background Information

[003] Currently, a conventional data storage system implements its own data model according to the user interface and the business rules specification for that system. For the conventional system, the system's developers write dedicated code and user interface code that displays or manages certain data inputs such as check boxes, lists, combo boxes or other data inputs. Such systems depend upon a specific data model and lack flexibility and portability. The conventional system's dedicated code and user interface in some respects manipulate the data that the conventional system's application stores and retrieves during operation. Moreover, the conventional system stores and retrieves data locally on a local disk, such as a C drive on a Windows system. It is well known that conventional systems may also store data on a remote storage media, e.g. a file server or database.

ServiceNow Ex. 1002 Page 194 of 410

[004] There are efforts to overcome the lack of flexibility and portability of conventional data storage systems, such as the adoption and utilization of extensible markup language for storing data (e.g., Microsoft Office® XML Open Format, OASIS Open Office XML Format, etc). Data files in open data formats (e.g., XML) facilitate data transfer and are typically stored on a desktop machine. XML data files rely on the structures defined by XML standards, which allows standardized XML parsers to read data files expressed in XML format.

[005] However, open data formats based on XML and other open standards do not interrelate and manage dynamic data sets through process and collaboration and state machines. Open data formats (e.g., XML) also do not provide for inheritance and override behaviors, collaboration or state machine process enforcement, and/or element style normalization.

[006] Accordingly, there is a need for systems and methods that store, manage, index, interrelate, and/or retrieve data sets in a manner independent of the data model.

SUMMARY

[007] In accordance with the principles of the present invention, as embodied and broadly described herein, methods, systems and computer program products are provided for manipulating data sets. The data set manipulation may be achieved through utilization of a processor and a data repository adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets.

[008] According to one embodiment, a system is provided for manipulating data sets. The system comprises a processor and a repository, the data repository including

an element module configured to store and uniquely identify a plurality of elements each identified by a unique element identification, an element relation module configured to store one or more relationships between the elements in the element module, and a class module configured to define at least one class of the elements and store the class. The data repository further includes an attribute module configured to define and store the attributes, a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class, and a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements.

[009] According to another embodiment, a method is provided for manipulating data sets. The method comprises storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, and storing one or more relationships between the elements in the data repository. The method further comprises defining at least one class of the elements and storing the class in the data repository, defining one or more attributes and storing the attributes in the data repository, defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository, and defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

[010] Another exemplary embodiment provides for a computer-readable medium containing instructions to configure a processor to perform a method for manipulating data sets. The method comprises storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, and storing one or more relationships between the elements in the data repository. The method further comprises defining at least one class of the elements and storing the class in the data repository, defining one or more attributes and storing the attributes in the data repository, defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository, and defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

[011] Additional aspects of the invention are set forth in the detailed description which follows or may be learned by practice of methods, systems, and articles of manufacture consistent with the present invention. It is understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[012] Both the foregoing general description and the following detailed description are exemplary and explanatory only. They do not restrict the invention, as claimed. Furthermore, the accompanying drawings, which are incorporated in and

constitute a part of this specification, illustrate embodiments of the invention and together with the description, serve to explain the principles of the invention. In the drawings:

[013] Figure 1 is an exemplary entity-relationship diagram of an elements metadata repository (EMR) for storing, managing, indexing, interrelating, and/or retrieving data sets, consistent with the principles of the present invention;

[014] Figure 2 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing articulated types, workspaces and workspace objects, class and class attributes, parameters, list options, etc;

[015] Figure 3 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing elements, element relations, element attributes, element history, element documents, etc;

[016] Figure 4 illustrates an exemplary diagrammatic visualization for visually selecting and relating elements using a graphical user interface (GUI);

[017] Figure 5 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing state machines, state machine transitions, state machine statuses, etc;

[018] Figure 6 illustrates an exemplary state machine diagram;

[019] Figure 7 illustrates a GUI of an exemplary diagrammatic depiction displaying congregations of requirements that are related to subcomponents of a system;

[020] Figure 8 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing tuples, named texts and metadata for named texts, etc;

[021] Figure 9 illustrates a GUI of an exemplary elements explorer;

[022] Figure 10 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing collaboration status history, polls, discussions, system components, etc;

[023] Figure 11 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing terminology, glossary, etc;

[024] Figure 12 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing emails, email details, etc;

[025] Figure 13 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing formatted dataset, SQL extensible stylesheet language transformations, etc;

[026] Figures 14 and 15 depict GUIs of exemplary editing environments that may be invoked from a displayed element;

[027] Figure 16 is a diagram depicting an exemplary Human Resource Management workspace; and

[028] Figure 17 is a diagram depicting an exemplary computer system in which systems and methods consistent with the principles of the elements metadata repository of Figure 1 may be implemented.

DETAILED DESCRIPTION

[029] Reference will now be made in detail to present embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Entity-Relationship Configuration

[030] FIG. 1 shows an entity relationship diagram 100 of an elements metadata repository (EMR) for storing, managing, indexing, interrelating, and/or retrieving data sets, consistent with principles of the present invention. In accordance with embodiments of the invention, a repository, such as the EMR, is a central place where data and metadata is stored and maintained. Metadata is data that is used to describe other data. Examples of metadata include schema, tables, indices, views, column definitions, etc. Metadata is recursive such that data about data about the data may be stored. This recursive nature may be expressed as levels, such as "M" number of levels. For example, the actual data being stored is M0. Data about that actual data (i.e., M0) may be referred to as M1. Data about the M1 layer data may be referred to as M2 and so on. In this example, the M1 and M2 layers are metadata, i.e., data about data. The EMR of FIG. 1 allows for storage of any number of data layers.

[031] Embodiments of the present invention, as shown in FIG. 2, include a class table 150, which may be analogous to a relational database table. Just as a table contains columns, a class, such as an entry in class table 150, contains attributes. In an exemplary implementation of the EMR, using database systems such as Oracle®, Microsoft SQL Server®, or MySQL, a user may create a class and its attributes by

creating a table and constructing columns within the table. In other words, the user creates a class and populates the class with attributes where attributes are conceptually other pieces of information of interest to the things that the class is trying to describe. For instance, if the class was a person, an attribute may be the person's name, birth date, or social security number. Any number of attributes can be assigned to any number of classes.

[032] Unlike a conventional relational database, the EMR extends the notion of available attributes types beyond the standard string, number, and date types. To do so, the EMR allows dynamically extendable entries in a type definition table 102. Type definition table 102 may store articulated types that may be, for example, an M4 layer while still maintaining the meta meta meta meta data language. By using the type of the person's name for an attribute, the user can declare more than what a normal relational database may store. For example, the EMR provides for a boolean type which is not normally found in a relational database. The EMR also provides for a pick list type, which stores a plurality of optional selections, and restricts actual data settings to none, or one and only one of the optional selections. The pick list type is also not found in a relational database. Moreover, the EMR is capable of storing a process or state machine type which is not found in a relational database. All these data types provide flexibility in the type of data management that can be performed.

[033] In accordance with still further embodiments of the invention, the EMR includes a parameter table 104. Parameter table 104 allows for any unforeseen attribute values to be placed upon system level components without redefining the base schema. For instance, if one type of named text is a SQL query, and the system

requires security settings on that one type of named text (further described below), then a parameter entry can be used to specify that setting instead of redefining named_text table (e.g. named_text table 140 in FIG. 8) to include an additional column, which may or may not be needed for all other named text types.

[034] Embodiments of the present invention, as shown in FIG. 2, include a workspace table 108 and a workspace_object table 106. Workspace table 108 and workspace_object table 106 provide for logical grouping of elements into subsets that are usable for particular users or purposes. For example, if an EMR user is a tester, then a "Testing" workspace may be declared. Objects, including elements, classes, named texts, and "things" that is defined by a numeric identifier (which may be every "thing") may be placed into 'Testing' workspace via entries in workspace_object table 106. The tester would be assigned to this workspace (also through workspace_object table 106), and therefore only be permitted to see the other objects in the workspace. The effect is a basic level of security and data volume reduction to small localized easily navigable subsets.

[035] Consistent with embodiments of the invention, an attribute table 110 holds the definition of each attribute used in each class in the EMR. Attribute table 110 may store the physical name and type of the data which corresponds to the database column name and column type where the actual values are stored. A class_attribute table 112 may associate a given attribute to one or more classes (in class table 150). Because class_attribute table 112 allows for one attribute to be referenced by many classes, there is attribute inheritance among the classes. Attribute inheritance among the classes is a powerful notion not found in traditional relational database management

systems (RDBMS). Class_attribute table 112 stores the logical name that may be used by the attribute when presented to the user for data manipulation such as insert or update or delete of values. The ability to reference the same attribute via multiple names depending on which class it appears in is akin to the concept of overridden inheritance, also a concept not found in RDBMS.

[036] In an embodiment of the invention, the EMR may immediately create a physical instantiation of the logical class and attribute model as soon as a class attribute definition is entered and committed into class_attribute table 112. The EMR may immediately represent the table and columns that the class and attribute definitions dictate. Moreover, the EMR's editors (see, for example, FIG. 14 and 15) may immediately conform to this definition and support full data manipulation of any values in any defined attribute. Furthermore, when certain types of attributes are attached to a class, appropriate named text entries are automatically created and tupled to the class and may be immediately available for use in the EMR. For example, if a pick list attribute type is added to a class, then queries that returns, for instance, all rows in the class and how many rows fall into each of the available pick list options, are automatically created, which allows the EMR to present the results (via charts or other presentation formats) to the users. This immediate construction of physical storage, complete editing capability, and automated reporting allows the EMR to become operational immediately upon logical schema definition.

[037] Embodiments of the present invention, as shown in FIG. 2, further include a list_option table 114. List_option table 114 may hold all possible enumerations in all possible pick list attributes, which allows for structured user interface and predictable

query structures when displaying and managing pick list data. A color table 116 may hold a record for each enumerated color referenced by any object in the EMR. A color can be declared and associated to an object such as a class, list option selection, or element, etc, and when detected during presentation, the id reference value for the color can be replaced with a swatch of the actual color or any other run time decoration as appropriate.

[038] Consistent with embodiments of the invention, the EMR includes a symbol table 118. Symbol table 118 is similar to color table 116 because it may hold a list of references to graphical objects that can be used in the system. The number and type of graphical objects are not restricted, and can be extended by addition of type definition entries but typically include standard graphic files, such as .BMP, .GIF, .JPG, .SVG, and .SVGz. Once declared in this table, these graphical objects can be referenced by any object in the system, and when the reference is detected at display time, the reference may be replaced with a rendering of the actual image. For example, reports may be constructed that show check marks, colored balls, military insignia, or any other relevant image depending on the needs of the visualization.

[039] The EMR of FIG. 1 conceptually breaks the world into an articulation of "things." The EMR places these "things" into one pile in an element table 160. An element stored in element table 160 implies that the user can find similarity between any "thing" that the user would like to discuss, report, store, or manage. Element table 160 provides the user the capability to add those "things" into a congruous pile so that the user can deal with them without regard to their exact nature. This provides an over-normalization of any data captured by the EMR into one record type called an element.

[040] FIG. 3 illustrates an exemplary element, which has a unique identifier (e.g., element ID), a name (e.g., description), and a classification (e.g., class identifier). Consistent typing of the element ID permits the user to refer to elements across the schema without having to convert the type of the element ID. In one embodiment, the element ID takes on a number format, e.g., a long value or an integer value. In this embodiment, any "thing" in the world may be uniquely identified by a number, e.g., an integer. Moreover, once any "thing" has been differentiated from nothingness, the "thing" has a name. This name may be called a description. In one embodiment, the key components of the element are the element ID and the name, as shown in element table 160. The description does not have to be unique. Some "things" may be described generally with the same name but have differences that render those things unique from each other. Another component of the element is a class identifier. The class identifier is a pointer or an indicator of the type or classification of this element in the universe. The class identifier allows the EMR to articulate the differences between types of elements that otherwise appear to be similar.

[041] Embodiments of the present invention include an element_relation table 162 for storing relationships between elements. Further to storing and managing elements in the universe, the EMR of FIG. 1 records and manages the fact that one element relates to another element in element_relation table 162. Each entry in element_relation table 162 may contain a pointer to at least two elements in elements table 160 and a type definition describing the type of relationship. Using the over-normalized model, the EMR ends up with a collection of elements and a collection of indicators of how those elements relate to each other.

[042] Embodiments of the present invention, as shown in FIG. 3, include an element_attr_c# table 164. Element_attr_c# table 164 represents a plurality of physical tables that are constructed into the base RDBMS. A physical table may be constructed in element_attr_c# table 164 for each class that is declared in class table 150. A column is inserted into this table for each class_attribute entry that specifies an attribute that should be in this class. In the physical implementation the # symbol is replaced with the identifier of the class to ensure uniqueness in table names, which is a requirement in the RDBMS but not necessarily in the class names of the element system.

[043] For each element_attr_c# table constructed, a corresponding entry in an element_history table 166 may also be constructed. Element_history table 166 records each version of each element through its lifespan in the EMR. In addition to all values of all attributes, element_history table 166 may also store indicators of the user making the change, and the time the change was made. This provides the EMR the capability to provide a complete history for all tracked elements throughout the life of the elements. This is useful for recovering a point in time configuration, or mixing historic version of various element subsets into named groups sometimes referred to as Baselines, Blocks, Revisions, Versions, Variants, or the like.

[044] Consistent with embodiments of the invention, the EMR includes an element_document table 168, which may represent a series of records that logically group a set of elements into what would traditionally be called a document. Element_document table 168 provides for formatting information that is separated from

the elements themselves, yet are important for the reconstruction for an ordered formatted display of the elements.

[045] From the collection of elements and the collection of indicators, the EMR can create visualizations for users by placing the elements in context with other elements that they relate to. FIG. 4 illustrates an exemplary visualization 400. Visualization 400 may also be called a diagrammatic visualization, which may be implemented by an elements diagram manager. In a diagrammatic visualization, the elements diagram manager selects or allows the user to select subsets of elements in the universe and place them on a drawing canvas. The elements diagram manager then arranges the elements in relation to other elements based on the conductivity rules found in element_relation table 162.

[046] In visualization 400, several elements are arranged on the canvas in proximity to each other and connected by directional lines that indicate their relation to each other, as determined by element_relation table 162. Each element in the canvas may be expanded to find and display elements in proximity to itself that is not yet on the canvas. Visualization 400 provides the user the capability to dynamically explore related elements.

[047] In addition to expansion of an element, each element in the EMR may invoke an editing screen that manages the values for the attributes of that element, based on the classification of that element. FIG. 14 displays a screen shot illustrating an exemplary editing environment 1400 that may be invoked from any displayed element, whether it appears in a diagram canvas, element explorer list, or other context.

[048] As illustrated in FIG. 14, editing environment 1400 shows several tabs, a General Tab, an Attributes tab, and a History tab. The selected tab in the example shown in FIG. 14 is the Attributes tab, which displays a listing of each attribute associated with the elements class, and an editor for the value of that attribute. In this example, the type of attribute is indicated by small icons to the left of the attribute name and the editor appears to the right of the attribute name. The order in which the attributes appear and certain display characteristics such as the number of rows of text to display in the edit field are determined by the entry in class_attribute table 112. Specialized editors can be defined and used such as date selectors, pick list combo boxes, and boolean checkboxes, all of which are invoked dynamically based on the definition of the attributes that happen to be in this class.

[049] FIG. 15 shows a screen shot of another exemplary representation 1500 of the dynamic editing capability of an element. In this embodiment, the EMR provides dynamic edit capability through web-based programming language and a standard browser without a structured programming language such as Java, as seen in editing environment 1400 (of FIG. 14). By providing web-based edit capability, the EMR allows users to edit attributes of an element by invoking one or more hyperlinks to the element when the element is referenced in other web-based reporting or outputs.

[050] State machines are a very powerful extension of normal relational database attributes or column types. As shown in FIG. 5, a state_machine table 120, a state_machine_transition table 122, and a status table 124 describe the structure and behavior of state machines. The term state machine may be used interchangeably with the terms process or workflow. A state machine (e.g., an entry in state_machine table

120) has one or more statuses (e.g., entries in status table 124) which are connected through transitions (e.g., entries in state_machine_transition table 122). The user may classify the elements by declaring one or more possible state machine statuses for each of the elements.

[051] Using the person example, if the person was an employee, then the EMR of FIG. 1 would be able to declare at least one possible status of the person. The EMR could declare the person's status in the state machine model as shown in FIG. 6, such as whether the person is a prospective employee 600, an actively hired employee 610, an employee on probation 620, an employee terminated for cause 630, etc. The EMR may restrict the transition from one status to another by limiting each status's forward transition connections. For instance, the user may first connect prospective employee status 600 to hired status 610. Next, the user may connect hired status 610 to on probation status 620. Furthermore, on probation status 620 may be connected to terminated for cause status 630 or back to hired status 610. Loops and branches are allowable in a stateful process.

[052] The EMR may enforce data behavior and constrain the user's ability to manipulate data based on the state machine model. For one or more stateful models, state machines may be applied to a data classification in a manner that forces data to follow at least one prescribed process declared by the user or a state administrator in a given enterprise. For instance, a prospective employee may not be immediately transitioned to terminated for cause because that violates the rules declared by the process. A prospective employee may only be propagated forward to available or declared next statuses, e.g., hired. Furthermore, the EMR may grant certain users the

ability to restrict the condition under which a transition is allowable, for example, only to users in the organization who are authorized to terminate an employee. Although the transition from actively hired to terminated with cause is generally allowed, there may be one or more exclusionary rules that declare that only a certain user or subset of users may transition an actively hired employee to the status of terminated with cause. Other exclusionary rules might state that the user may not transition an actively hired employee to terminated for cause unless the user has entered a cause and linked it by relation to the employee.

[053] There are many types of data models beyond employees, e.g., the Department of Defense core architectural data model (CADM), the Defense Department architectural framework (DODAF), or the Meta Objects Facility (MOF). These are examples of standardized data models for users to store data, and the models are independent of the data definition that the user wishes to implement. The user can implement multiple concurrent disparate data definitions within a single enterprise repository, such as the EMR of FIG. 1.

[054] Currently, when a system is declared, it implements its own data model according to the user interface and the business rules specification for that system. For the conventional system, the system's developers write dedicated code and user interface code that displays or manages certain data inputs such as check boxes, lists, combo boxes or other data inputs. The conventional system's dedicated code and user interface manipulate the data that the system's application stores and retrieves during operation. Moreover, the conventional system stores and retrieves data locally on a local disk, like a C drive on a Windows system.

[055] There are several drawbacks to conventional systems, one of which is that the data is typically stored in a proprietary file, i.e., a file that is a Microsoft Word® document in .doc format that contains proprietary binary code. The contents of that file are not clear unless one uses Microsoft Word® or another tool that can decode the contents. While Microsoft Word® stores textual information and formats information in a proprietary format contained within an operating system level object called a file, the EMR foregoes the file-based implementation in favor of a elements-based implementation. In the elements-based implementation, the contents of what otherwise would be a file are flexibility represented as elements. Therefore, the contents are not restricted to use by a proprietary application, but are concurrently available to all database-aware applications.

[056] In the elements-based implementation, a document would not be represented by a collection of proprietarily stored formatting and textual information. Instead, the document would be stored as an element with, e.g., one element for each sentence in the document. The elements represent a usable chunk of contextualized information, such as a sentence. In the field of ontologies, words in a sentence by themselves have no meaning. An ontology implies context. Words by themselves may have a definition, but for the words to truly convey meaning, the words need to included in at least a phrase and probably a sentence. In the elements-based approach, each of the elements conforms to the rules of generalized things in the universe, and therefore may be reused or referenced to other things in the universe, e.g., through entries in element_relation table 162. Moreover, each of the elements in its own context is also meaningful and reusable.

[057] In implementing a solution for a requirements management problem, people would typically use a conventional document to articulate thousands or hundreds or tens of requirements that they would like a system or systems to achieve. In an embodiment of the elements-based implementation in the EMR, each requirement statement would be an element that the user may associate a process to, independent of any kind of hard-coded association. The user has a choice to declare, on each element, such a process which is automatically enforced for that user.

[058] Using the EMR, an element or, in this case, a requirement, would be listed and available as generally readable in an open query-able fashion. This allows the user to organize and aggregate that requirement in more than just the context of one requirements document as the user would in a conventional document-based system. For example, the user may organize and aggregate requirements in a diagrammatic depiction 700 as shown in FIG. 7 that shows congregations of requirements that are related to specific subcomponents of the main system that the user is trying to build.

[059] Diagrammatic depiction 700 shows a subset of elements that may represent requirements as in the previous example. In the example illustrated in diagrammatic depiction 700, there are at least two (2) requirements that are common to the major system components represented in diagrammatic depiction 700. If each generally circular grouping is considered a system component and its related requirements, then the left hand component is related to the right hand component through the two elements generally in the middle of the diagram and linked simultaneously to both.

[060] For example, consider a situation where the user declares that an engine is needed in the process of building an airplane. In a file-based system, the user either declares that through the requirements themselves or through the practice of engineering the airplane, for which there are a thousand requirements that describe how the engine should mount to the wing. In the file-based system, the requirements may be in several spreadsheets or documents. The requirements may be repeated in the spreadsheet and a document, and may be repeated again in a testing suite application. Moreover, the requirements may be further repeated on the architectural drawing of the component. In the EMR of FIG. 1 there would be one instance of the requirement and it would be shared among all applications that needed to reference that instance of that requirement. Items such as a diagrammatic depiction (e.g. diagrammatic depiction 700), spreadsheet-like depiction, or documentation depiction, are now file independent. The items are just depictions of the core data set.

[061] Embodiments of the present invention include a named_text table 140, as shown in FIG. 8. Named_text table 140 provides a powerful normalization of all text that the user may refer to in a system. Text is a ubiquitous descriptor of many things, including graphics. With current techniques, there are textual representations of graphics, notably scalable vector graphic (SVG) from companies like Adobe®, which are textual descriptions that can be rendered into graphics. For example, a textual description may state "draw a red circle with a 50 pixel radius." The textual description can be translated into an actual depiction on the screen. This illustrates that text represents most things or most things can be represented by text, not only graphics through the SVG, but also any data structure through extended markup language. The

EMR takes advantage of this by declaring these sets of texts and identifying them with a name and a unique identifier. These sets of texts then become usable in the elements-based implementation in many ways.

[062] Embodiments of the present invention include a tuple table 144, as shown in FIG. 9. Tuple table 144 is similar to element_relation table 162, except that tuple table 144 has been separated to store information about the system schema as opposed to the generated user schema. Tuple table 144 relates any identified item to any other identified item and that relationship is created according to a type (e.g., type_id). Given that an entry in named_text table 140 is a thing and that the user may relate the thing to other types of named text, the EMR declares, for the user's convenience, a named_text item that is akin to a folder in an operating system hierarchy. The folder may be called, for example, "My Favorites," or "My Documents," or in this instance, "My Elements." That folder can then be referenced by the user. Next, the EMR may declare 2, 3, 5 or 100 (or any number of) sub-folders. The sub-folders may be referenced by entries in tuple table 144 to the folder and other sub-folders.

[063] The combination of named_text table 140 and tuple table 144 provides to the EMR the capability to create a recursion between named_text types that represent folders to other named_text types that represent folders in a hierarchy. In an embodiment shown in FIG. 9, the EMR may contain a folder called root (e.g. "EE Root"), another folder called "My elements" (e.g. "My Elements (MARK)" and "My Elements (RONO)"), and tuples the root entry in named_text table 140 to the "My Elements" entry in named_text table 140. In doing so, the EMR indicates through elements explorer 900

a hierarchy or a tree or an indenture of the folders. This particular implementation approximates a standard operating system file explorer, in which the user can click and drag and drop folders and other items into a folder in a tree or navigable folder set. Through the same mechanism, an element may be tupled into a folder. In the elements-based implementation, the EMR may use a folder to organize and aggregate random sets of things in the universe.

[064] The EMR may utilize entries in named_text table 140 to name structured query language queries. SQL is a textual language that describes an interaction or request to a relational database. SQL queries are the mechanism to retrieve data from the database. SQL query is a powerful notion for relational databases and may be implemented as a named_text entry. Unlike a conventional database, e.g., Oracle® or Microsoft SQL Server®, that stores user-generated SQL as a "view," the EMR stores user-generated SQL as a named_text entry. Because the user-generated SQL is an entry in named_text table 140, and since folders exist as entries in named_text table 140 or tuple table 144, the EMR may filter or query entries in named_text table 140 or tuple table 144 into one or more folders.

[065] Thus, if the SQL statement serves as a folder, the results of the SQL statement or the items returned by the query would be the contents of that folder. The SQL statement to the user is therefore a dynamic folder or a SQL folder and no tuple exists between that data set. In other words, elements in the dynamic folder can be placed into other folders. The dynamic folder is executed at run-time if the user selects the dynamic folder and the resulting elements that come back from the query are presented as if they were tupled into that folder. This is a dynamic process and the

result set may differ depending on the dynamic data changes. This dynamic process simplifies the organization and provides to the user the ability to locate data through a dynamic folder hierarchy. The dynamic folder hierarchy further includes dynamic folders that change the data set appropriately every time the user visits them.

[066] An example of the dynamic folder capability provided by the EMR is a "My Tasks" folder. Assume that task is an element classification and that user is an attribute of the task. If there are tasks in the system that are assigned to the user, the user may perform a query that directs the EMR to select one or more tasks for which the user is the assigned user. The result of that query may change over time. The administrator may assign a task to someone else, so the task may be deleted or moved to a completed state and the user's folder would only contain a current task set that matches the query as defined by a named_text of that folder. This allows for an extensible method of organizing data of interest and allowing the user to easily find and retrieve data of interest in a large, complex system.

[067] Embodiments of the present invention, as shown in FIG. 10, include a poll table 182. Poll table 182 may represent a set of identified questions that are in one of several stages of life cycle of an element or set of elements. A poll is a collaborative mechanism that restricts the responses of the participants to one or more predetermined choices which are enumerated in a poll_option table 188. The replies of the participants are recorded in a poll_reply table 184. These poll replies can be analyzed for preference among the voting community and used to record decision making processes regarding any given element or set of elements. This relation is achieved through entries in tuple table 144 (in FIG. 8).

[068] In accordance with still further embodiments of the invention, the EMR includes a discussion table 186, which is similar to poll table 182 as discussion table 186 records collaborative information regarding one or more elements in the system. However, a discussion, which may be stored as entry in discussion table 186, is not forced into a set of structured responses. Instead discussion table 186 allows free-form narrative to be recorded, and replied to by any number of users in the system.

[069] Embodiments of the present invention may further include a collaboration_status_history table 180, which may record all the states that either a discussion or poll moves through in a defined life cycle, similar to element history. Moreoever, the user and date and time is recorded with each status change.

[070] In accordance with still further embodiments of the invention, the EMR includes a system_component table 190. The EMR may register components that are programmatic in nature (e.g. Java classes), wherein the components may be executed dynamically as needed (through a process called reflection) as part of element decoration. For example, one or more Java classes may be defined and saved as a record in system_component table 190, then the Java classes may be tupled to elements. At runtime, the Java classes that are referenced may be executed when an element of the Java class is edited or viewed.

[071] Embodiments of the present invention, as shown in FIG. 11, include a terminology table 130 and a glossary 132. Terminology table 130 may store a list of terminologies that can be used in the system. Glossary table 132 may uniquely identify and define each of the terms stored in terminology table 130.

[072] In accordance with embodiments of the invention as shown in FIG. 12, the EMR includes an email table 172 and an email_detail table 170. These tables allow the EMR to populate records in the database that can programmatically be converted from rows in the RDBMS to actual emails using a known protocols such as SMTP and POP. Since the emails stored in email table 172 and email_detail table 170 are also data, the EMR may apply stateful process logic to each email - using status, state_machine, and state_machine_transition tables (FIG. 5, elements 120, 122, 124, respectively) to define the lifecycle of an email. Full email history can be maintained in the EMR and new emails can be generated either programmatically when an event occurs, or manually when a user enters an email. Individual messages may be aggregated into one larger email at a time interval specified by an administrator. The aggregated messages may also be sent on demand.

[073] In accordance with embodiments of the invention as shown in FIG. 13, the EMR includes a formatted_dataset table 176 and a SQL_XSLT table 178. Together, formatted_dataset table 176 and SQL_XSLT table 178 allow the EMR to create a named "formatted dataset," which applies formatting to a SQL result set. Formatted_dataset table 176 and SQL_XSLT table 178 may work in conjunction with named_text table 140 to relate one or more named text entries (representing SQL statements), which provide the result set(s), with another named text entry (representing XSLT or a formatting definition) which generates a final layout.

[074] To further illustrate the EMR as described above and in FIGS. 1-15, an example is provided to demonstrate the utilization and capability of the EMR, consistent with the present invention, for storing, managing, indexing, interrelating,

and/or retrieving data sets, such as large scale and dynamic data sets. One example, as shown in FIG. 16, may be Human Resource Management (HRM). Further examples may include, for instance, a vertical implementation, or schema and associated processes, that may serve the needs of a well defined market, e.g., Enterprise Architecture, Requirements Management, Six Sigma, Resource Planning, Federal Regulatory Compliance, Legal Document Management, Portfolio Management, etc.

[075] In the HRM example, it may be determined that tracking "resources" is desired. Those resources may be people, but may not necessarily be restricted to people, which is a distinction that may be made when defining the implementation. It may further be determined that the resources of interest will in fact be people, and those people will be employees of a specified company. The company may have written policies that determine employment, such as the amount of vacation time each employee may receive, a level of compensation, etc.

[076] Furthermore, it may be determined that the resources may exist in a hierarchy, e.g., have a place in an organization chart, and that tasks should be identified and assigned to these resources as needed. Steps may be taken within the EMR to support such system requirements.

[077] A workspace (e.g., HRM Workspace 1600 of FIG. 16) may optionally be entered into the ERM (e.g. workspace_object table 106 and workspace table 108 of FIGS. 1 and 2) to segregate the HRM information from other possible data sets in the EMR if desired.

[078] A schema would be logically defined that encompassed the articulated needs of the system. In this case perhaps a resource class 1602 (of FIG. 16) would be

declared with attributes of employee_name, employee_ID, and employee_status. The employee_status attribute may be of type state_machine, allowing the stateful process of employment to be declared according to the rules of the company.

[079] It may further be determined that a resource may change title and salary over time, and other classes, such as a resource_title class 1606 and a resource_compensation class 1608, would be declared each with a pointer to resource class 1602 such as resource_ID and other appropriate attributes such as resource_title, effective_start_date, and effective_end_date. Furthermore, resource_compensation class 1608 may similarly contain compensation_amount, compensation_interval, compensation_start_date, and compensation_end_date.

[080] Tasks would also be similarly defined as a task class 1604 with, for example, a name, a description, a milestone indicator, a status, an assignee, an estimated_duration, and a project indicator.

[081] To implement these classes, data would be entered into the EMR representing these structures. The order of implementation, as illustrated below, is exemplary and does not limit embodiments of the present invention.

[082] Entries may be made into class table 150 (of FIGS. 1 and 2), one each for resource, task, resource_title, and resource_compensation. Entries may be made into attribute table 110 (of FIGS. 1 and 2) for each employee_name, employee_ID, employee_status, employee_title, compensation_interval, compensation_amount, compensation_start_date, compensation_end_date, status, assignee, estimated_duration, and project.

[083] As further definition, entries may be made into state_machine table 120, state_machine_transition table 122, and status table 124 (of FIGS. 1 and 5) to declare the available selections of the status and employee_status attributes. Further entries may be placed into list_option table 114 (of FIGS. 1 and 2) to specify the available resource_titles that may be assigned to resource or resources.

[084] Entries may then be made into class_attribute table 112 (of FIGS. 1 and 2) to associate the appropriate attributes with their respective classes. This may all be done via any data insertion method available, e.g. a user interface, etc.

[085] Once these class_attribute records have been entered into class_attribute table 112, the EMR is ready for use. Users have the capability to add resources, create tasks, assign them to the resources, report on how many tasks are completed, etc. Users may also create folders to organize projects (see, e.g., FIG. 9) and view diagrams (see, e.g., FIGS. 4 and 7) of which resources are committed to which tasks without any further programmatic effort. Users may also be able to collaborate on tasks by discussing ramifications or polling a subset of users to determine specific answers to to-be-determined elements.

[086] In this example, the resource hierarchy may be constructed by linking one resource to another recursively (see, e.g., FIG. 16). This may be done either via the record editing capabilities or by using the diagramming tool (see, e.g., FIGS. 4 and 7) to draw a line from one resource to another. The diagrammatic representation of the organization chart is capable of immediately expand an individual to supervisors, peers, or subordinates. The diagrammatic representation may also show all related tasks to one or more of those individuals if desired. By providing those capabilities, the

diagrammatic representation allows for visualization of tasking overloads or underutilized resources, etc.

[087] If the EMR does not provide a specific aggregation or subsetting of tasks or resources, one or more named_text entries in named_text table 140 (of FIGS. 1 and 8) may be defined that return the appropriate results. If the default display of those results is insufficient for the business needs, then one or more formatted_dataset entries in formatted_dataset table 176 (of FIGS. 1 and 13) and SQL_XSLT entries in SQL_XSLT table 178 (of FIGS. 1 and 13) may be defined to convert the available results into any desired final output.

[088] In this example, when the user edits or inserts data in the EMR, the data may be stored in an appropriate element_attr_c# table 164 (of FIGS. 1 and 3) that was automatically constructed as part of the class definition step above. Moreover, any modifications may automatically be recorded as history in element_history table 166 (of FIGS. 1 and 3).

[089] The above-described exemplary scenario may be followed for any given schema. A full schema does not need to be defined before the EMR may be used. If for instance only resources were defined, then resource information could be entered, and organization hierarchies may be defined at a later date; the tasking information may be added without disruption to the EMR. As a further example, additional attributes may be added to the resources themselves at a later date if the attributes are deemed necessary. As another example, the EMR's user interfaces may automatically adjust to the current definition to allow seamless upgrades of the data models.

[090] Systems and methods consistent with the present invention may be implemented by computers organized in a conventional distributed processing system architecture. FIG. 17 is a block diagram illustrating a computer system 200 in which an embodiment consistent with the invention may be implemented. Computer system 200 includes a computer 201 having a bus 202 or other communication mechanism for communicating information, and a processor 204 coupled with bus 202 for processing information. Computer 201 also includes a main memory, such as random access memory (RAM) 206 or other dynamic storage device, coupled to bus 202 for storing information and instructions by processor 204. RAM 206 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 204. During operation, some or all portions of an operating system (not shown) are loaded into RAM 206. Computer 201 further includes a read only memory (ROM) 208 or other static storage device coupled to bus 202 for storing static information and instructions for processor 204. A storage device 210, such as a magnetic disk or optical disk, is provided and coupled to bus 202 for storing information and instructions.

[091] Computer 201 may be coupled via bus 202 to a display 212, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 214, including alphanumeric and other keys, is coupled to bus 202 for communicating information and command selections to processor 204. Another type of user input device is cursor control 216, such as a mouse, a trackball or cursor direction keys for communicating direction information and command selections to processor 204 and for controlling cursor movement on display 212. This input device may have two degrees

of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), which allow the device to specify positions in a plane.

[092] System 200 may further include an input drive device 217, such as a CD reader, for reading high-capacity computer-readable media, such as CD-ROMs and CDRs.

[093] The invention is related to the use of computer system 200 for manipulating data sets for, to take an example, corporate or engineering exercises. According to one implementation, systems and methods consistent with the present invention collect system information and store it in a central database in response to processor 204 executing one or more sequences of one or more instructions contained in RAM 206. Such instructions may be read into RAM 206 from a computer-readable medium via an input device such as storage device 210. Execution of the sequences of instructions contained in main memory 206 causes processor 204 to perform the process steps described herein. In an alternative implementation, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, implementations consistent with the principles of the present invention are not limited to any specific combination of hardware circuitry and software.

[094] Computer 201 also includes a communication interface 218 coupled to bus 202. Communication interface 218 provides a two-way data communication coupling to a network link 220 that is connected to local network 222. For example, communication interface 218 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 218 may be a local area

network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 218 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[095] Network link 220 typically provides data communication through one or more networks to other data devices. For example, network link 220 may provide a connection through local network 222 to host computer 224 and/or to data equipment operated by Internet Service Provider (ISP) 226. ISP 226, in turn, provides data communication services through the Internet 228. Local network 222 and Internet 228 both use electric, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 220 and through communication interface 218, which carry the digital data to and from computer system 200, are exemplary forms of carrier waves transporting the information.

[096] Computer system 200 can send messages and receive data, including program code, through the network(s), network link 220 and communication interface 218. In the Internet example, a server 230 might transmit a requested code for an application program through Internet 228, ISP 226, local network 222 and communication interface 218. The received code may be loaded in RAM 206 and executed by processor 204 as it is received. Alternatively, or in addition, it may be stored in storage device 210, or other non-volatile storage for later execution. In this manner, computer system 200 may obtain application code in the form of a carrier wave.

[097] Although computer system 200 is shown in FIG. 17 as being connectable to one server 230, those skilled in the art will recognize that computer system 200 may establish connections to multiple servers on Internet 228. Such servers may include an HTML-based Internet application, which may provide information to computer system 200 upon request in a manner consistent with the present invention.

[098] The foregoing descriptions of the invention have been presented for purposes of illustration and description. They are not exhaustive and do not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but the present invention may be implemented as a combination of hardware and software or in hardware alone. Additionally, although aspects of the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM.

WHAT IS CLAIMED IS:

1. A computerized system for manipulating data sets, comprising:

a processor; and

a data repository, the data repository being adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, comprising:

an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification;

an element relation module configured to store one or more relationships between the elements in the element module;

a class module configured to define at least one class of the elements and store the class;

an attribute module configured to define one or more attributes and store the attributes;

a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class; and

a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements.

2. The computerized system of claim 1, wherein the data repository further comprises:

a state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element.

3. The computerized system of claim 2, wherein the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

4. The computerized system of claim 1, wherein the type definition module defines a type including at least one of a boolean type, a picklist type, and a state machine type.

5. The computerized system of claim 4, wherein the data repository further comprises:

a list option module configured to enumerate one or more picklist attributes of the picklist type.

6. The computerized system of claim 2, wherein the data repository further comprises:

a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements.

7. The computerized system of claim 6, wherein the data repository further comprises:

a tuple module configured to generate and store one or more tuples, wherein each of the tuples associates a subset of the elements in accordance to one of the types associated with the subset of the elements.

8. The computerized system of claim 1, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

9. The computerized system of claim 8, wherein the data repository further comprises:

an element attribute module for each of the at least one class, the element attribute module configured to define and store the attributes of the elements in the class, wherein the element attribute module is identified using the name of the class.

10. The computerized system of claim 9, wherein the data repository further comprises:

an element history module configured to store one or more indicators of modification to the elements.

11. The computerized system of claim 10, wherein the data repository further comprises:

an element document module configured to store one or more formatting data for grouping and displaying a subset of the elements.

12. The computerized system of claim 1, wherein the processor is configured to generate a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

13. The computerized system of claim 1, wherein the processor is configured to generate a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

14. A method for manipulating data sets, comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; and

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

15. The method of claim 14, further comprising:

defining a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element, and storing the state machine type in the data repository.

16. The method of claim 15, wherein the state machine type comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

17. The method of claim 14, wherein the type comprises a type including at least one of a boolean type, a picklist type, and a state machine type.

18. The method of claim 17, further comprising:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

19. The method of claim 15, further comprising:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

20. The method of claim 19, further comprising:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

21. The method of claim 14, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

22. The method of claim 21, further comprising:

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

23. The method of claim 22, further comprising:

storing, in the data repository, one or more indicators of modification to the elements.

24. The method of claim 23, further comprising:

storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

25. The method of claim 14, further comprising:

generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

26. The method of claim 14, further comprising:

generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

27. A computer-readable medium containing instructions to configure a processor to perform a method comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; and

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

28. The computer-readable medium of claim 27, wherein the method further comprises:

defining a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element, and storing the state machine type in the data repository.

29. The computer-readable medium of claim 28, wherein the state machine type comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

30. The computer-readable medium of claim 27, wherein the type comprises a type including at least one of a boolean type, a picklist type, and a state machine type.

31. The computer-readable medium of claim 30, wherein the method further comprises:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

32. The computer-readable medium of claim 28, wherein the method further comprises:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

33. The computer-readable medium of claim 32, wherein the method further comprises:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

34. The computer-readable medium of claim 27, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

35. The computer-readable medium of claim 34, wherein the method further comprises:

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

36. The computer-readable medium of claim 35, wherein the method further comprises:

storing, in the data repository, one or more indicators of modification to the elements.

37. The computer-readable medium of claim 36, wherein the method further comprises:

storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

38. The computer-readable medium of claim 27, wherein the method further comprises:

generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

39. The computer-readable medium of claim 27, wherein the method further comprises:

generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

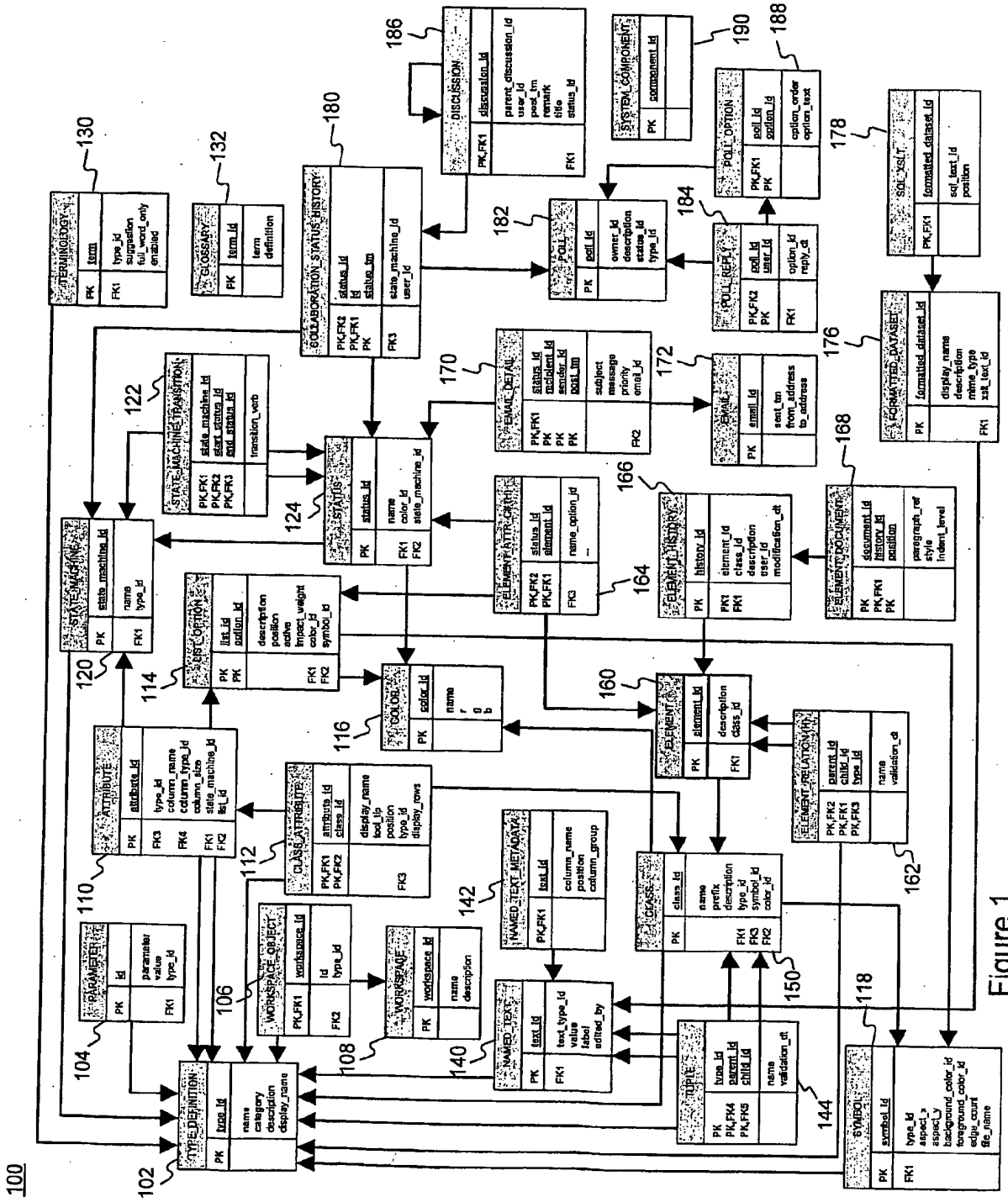

100
102
104
106
108
110
112
114
116
118
120
122
124
130
132
140
142
144
150
160
162
164
166
168
170
172
176
178
180
182
184
186
188
190
TERMINOLOGY
GLOSSARY
DISCUSSION
SYSTEM_COMPONENT
POLL_OPTION
SQL_XSLT
COLLABORATION_STATUS_HISTORY
POLL
POLL_REPLY
FORMATTED_DATASET
STATE_MACHINE_TRANSITION
EMAIL_DETAIL
EMAIL
STATUS
ELEMENT_ATTR_CATALOG
ELEMENT_HISTORY
ELEMENT_DOCUMENT
STATE_MACHINE
LIST_OPTION
COLOR
ELEMENT
ATTRIBUTE
CLASS_ATTRIBUTE
ELEMENT_RELATIONSHIP
PARAMETER
WORKSPACE_OBJECT
WORKSPACE
NAMED_TEXT_METADATA
CLASS
NAMED_TEXT
RULE
TYPE_DEFINITION
SYMBOL

Figure 1

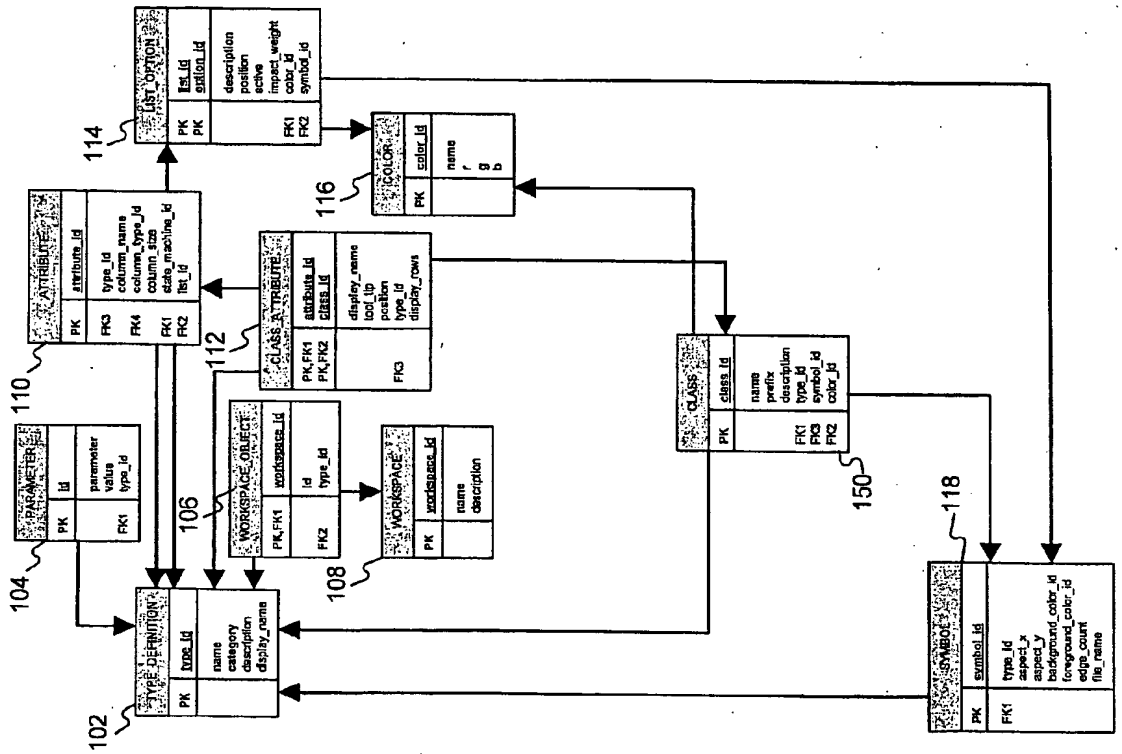

LIST_OPTION
list_id
option_id
description
position
active
impact_weight
color_id
symbol_id
PK
PK
FK1
FK2
114
COLOR
color_id
name
r
g
b
PK
116
ATTRIBUTE
attribute_id
type_id
column_name
column_type_id
column_size
date_machine_id
list_id
PK
FK3
FK4
FK1
FK2
110
CLASS_ATTRIBUTE
attribute_id
class_id
display_name
tool_tip
position
type_id
display_rows
PK,FK1
PK,FK2
FK3
112
CLASS
class_id
name
prefix
description
type_id
symbol_id
color_id
PK
FK1
FK3
FK2
150
PARAMETER
id
parameter
value
type_id
PK
FK1
104
WORKSPACE_OBJECT
workspace_id
id
type_id
PK,FK1
FK2
106
WORKSPACE
workspace_id
name
description
PK
108
TYPE_DEFINITION
type_id
name
category
description
display_name
PK
102
SYMBOL
symbol_id
type_id
aspect_x
aspect_y
background_color_id
foreground_color_id
edge_count
file_name
PK
FK1
118

Figure 2

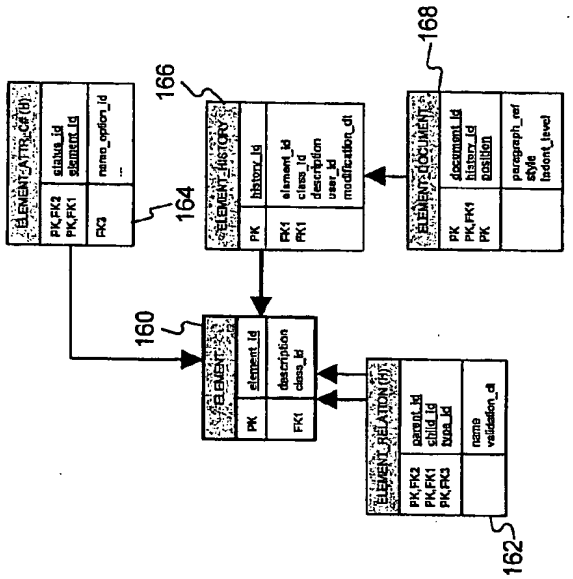

168
166
164
160
162
ELEMENT_ATTR_CF (0)
PK,FK2
PK,FK1
FK3
status_id
element_Id
name_option_id
...
ELEMENT_HISTORY
PK
history_Id
FK1
FK1
element_id
class_id
description
user_id
modification_dt
ELEMENT_DOCUMENT
PK
PK,FK1
PK
document_Id
history_Id
position
paragraph_ref
style
indent_level
ELEMENT
PK
element_Id
description
class_id
FK1
ELEMENT_RELATION (0)
PK,FK2
PK,FK1
PK,FK3
parent_id
child_Id
type_id
name
validation_id

Figure 3

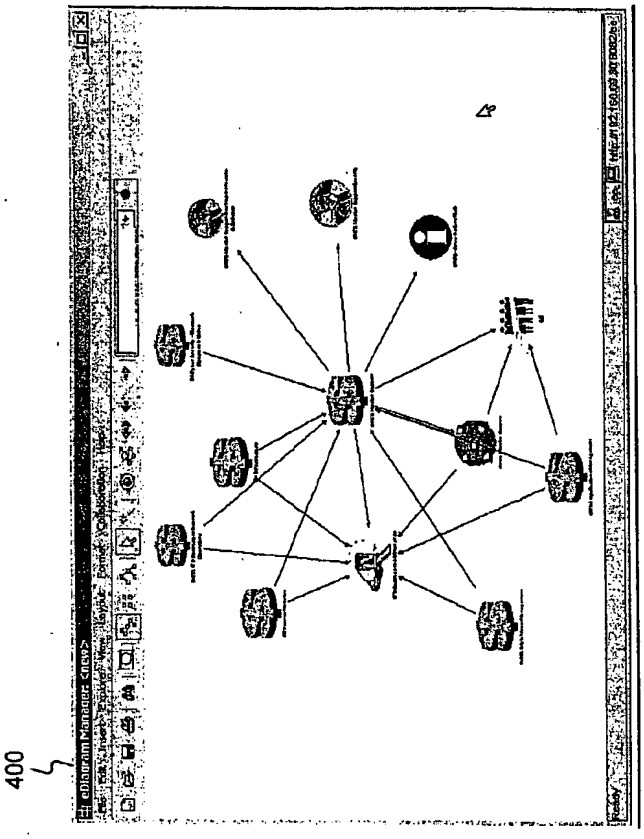


Figure 4

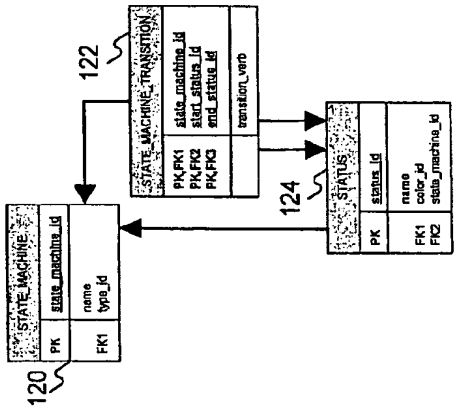

122
STATE_MACHINE_TRANSITION
PK,FK1
PK,FK2
PK,FK3
state_machine_id
start_status_id
end_status_id
transition_verb
124
STATUS
PK
status_id
name
color_id
state_machine_id
FK1
FK2
STATE_MACHINE
PK
state_machine_id
name
type_id
FK1
120

Figure 5

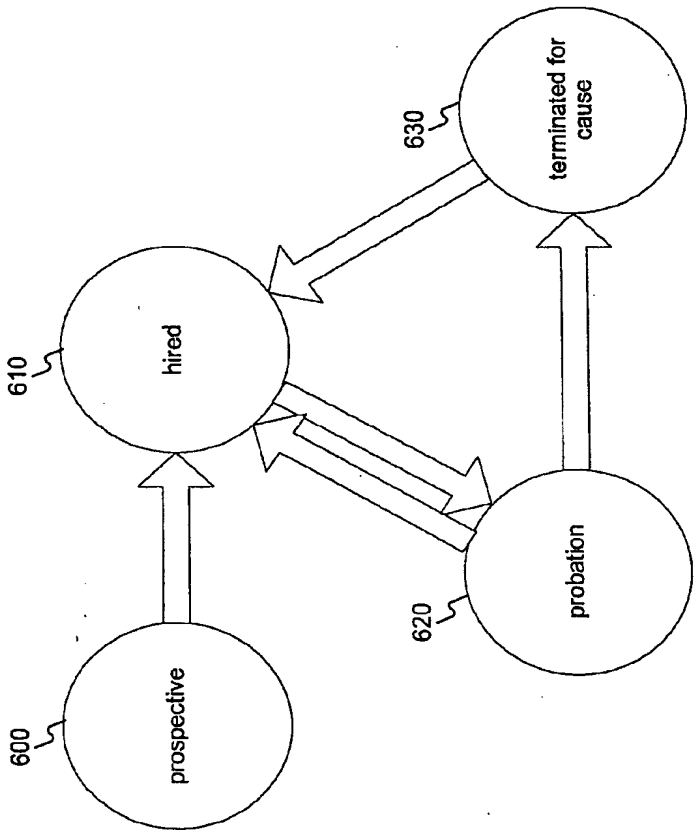

terminated for cause
630
hired
610
probation
620
prospective
600

Figure 6

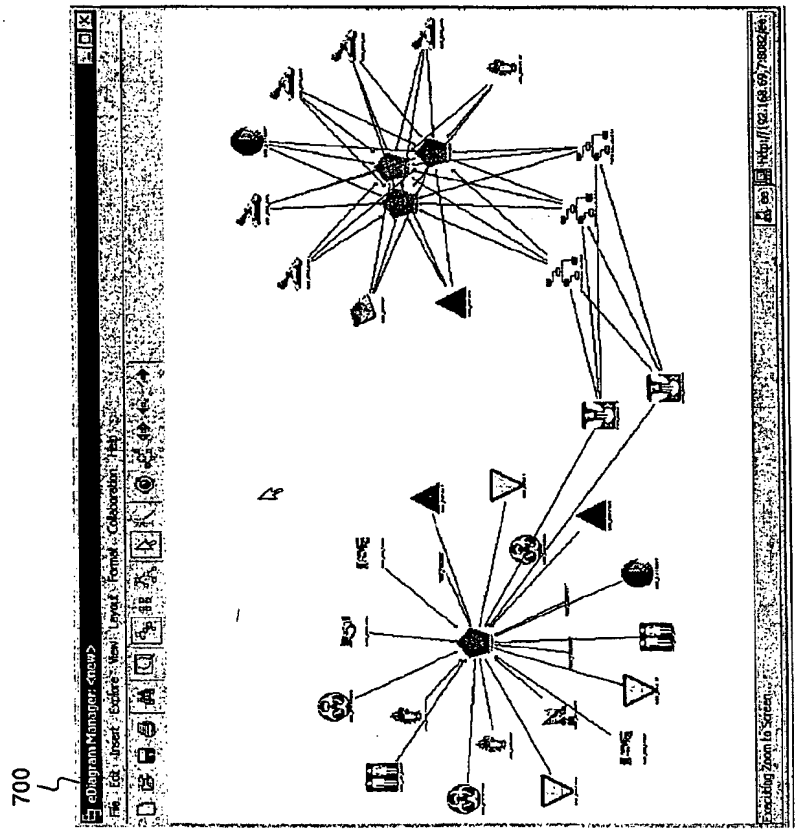


Figure 7

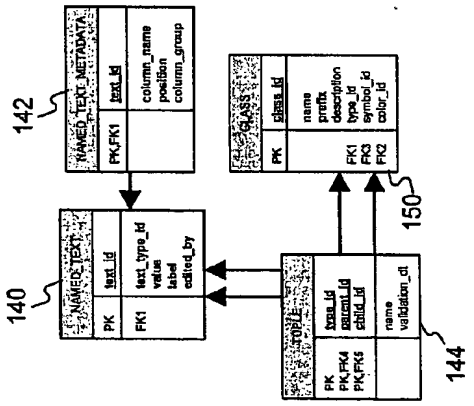

142
NAMED_TEXT_METADATA
text_id
PK,FK1
column_name
position
column_group
140
NAMED_TEXT
PK
text_id
FK1
text_type_id
value
label
edited_by
CLASS
class_id
PK
name
prefix
description
type_id
symbol_id
color_id
FK1
FK3
FK2
150
TYPE
PK
PK,FK4
PK,FK5
name
validation_dt
144

Figure 8

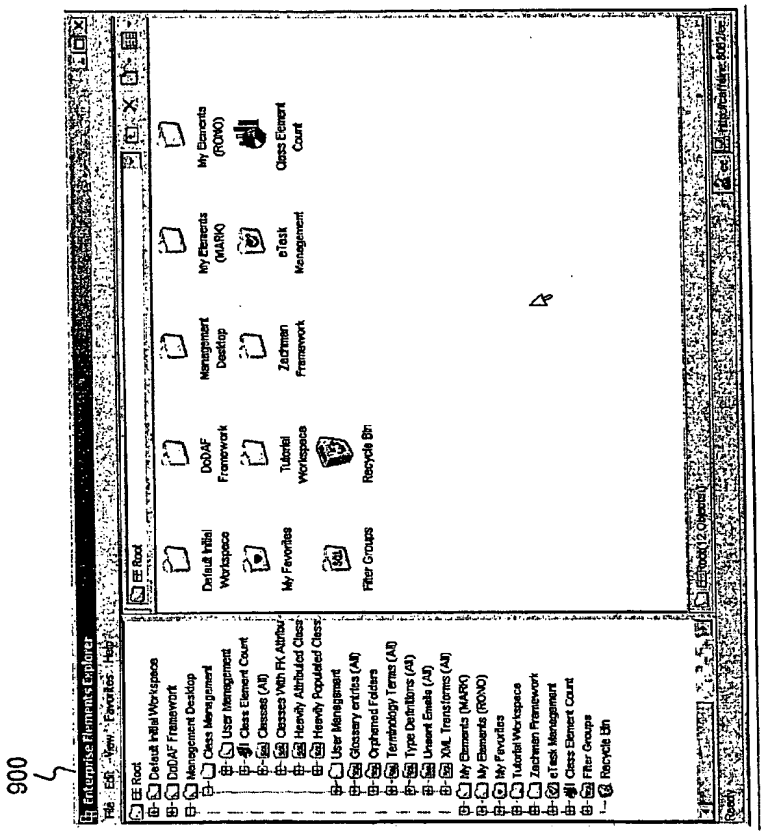

900
Enterprise Elements Explorer
File Edit View Favorites Help
EE Root
Default Initial Workspace
DoDAF Framework
Management Desktop
Class Management
User Management
Class Element Count
Classes (All)
Classes With FK Attribu
Heavily Attributed Class
Heavily Populated Class
User Management
Glossary entries (All)
Orphaned Folders
Terminology Terms (All)
Type Definitions (All)
Unsent Emails (All)
XML Transforms (All)
My Elements (MARK)
My Elements (RONO)
My Favorites
Tutorial Workspace
Zachman Framework
eTask Management
Class Element Count
Filter Groups
Recycle Bin
Default Initial Workspace
DoDAF Framework
Management Desktop
My Elements (MARK)
My Elements (RONO)
My Favorites
Tutorial Workspace
Recycle Bin
Zachman Framework
eTask Management
Class Element Count
Filter Groups
Ready

Figure 9

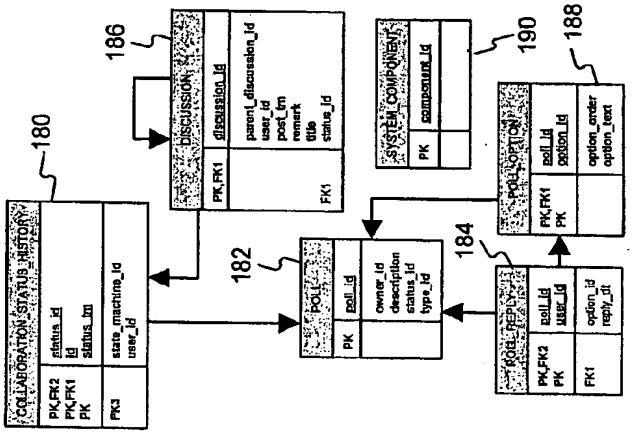

180
186
182
184
188
190
COLLABORATION_STATUS_HISTORY
DISCUSSION
POLL
POLL_REPLY
POLL_OPTION
SYSTEM_COMPONENT

Figure 10

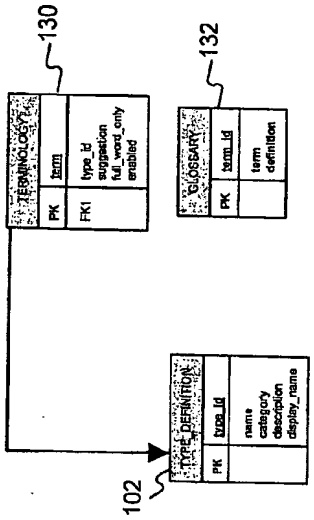

130
132
TERMINOLOGY
PK
FK1
term
type_id
suggestion
full_word_only
enabled
GLOSSARY
PK
term_id
term
definition
TYPE_DEFINITION
PK
type_id
name
category
description
display_name
102

Figure 11

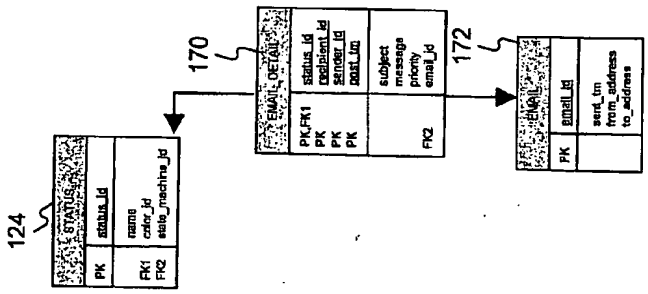

124
170
172
STATUS
status_id
name
color_id
state_machine_id
PK
FK1
FK2
EMAIL_DETAIL
status_id
recipient_id
sender_id
post_tm
subject
message
priority
email_id
PK,FK1
PK
PK
PK
FK2
EMAIL
email_id
send_tm
from_address
to_address
PK

Figure 12

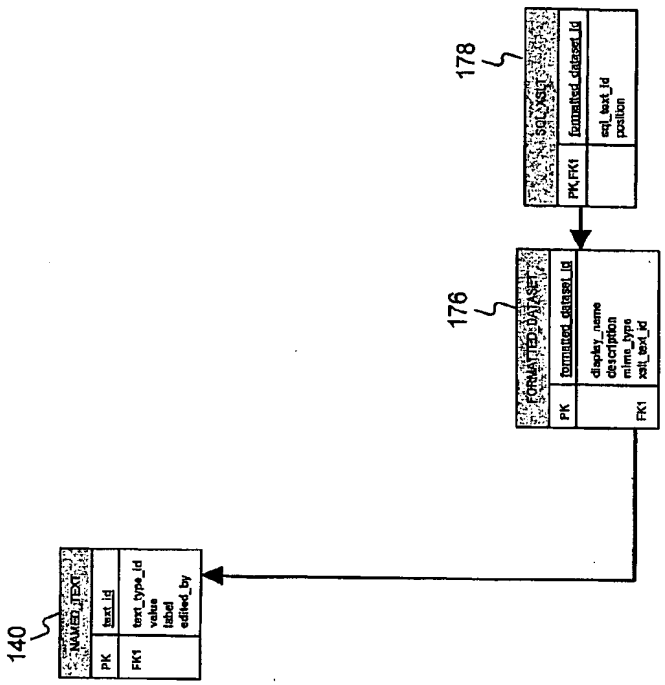

178
SQL_ASSET
PK,FK1
formatted_dataset_id
sql_text_id
position
176
FORMATTED_DATASET
PK
formatted_dataset_id
display_name
description
mime_type
xslt_text_id
FK1
140
NAMED_TEXT
PK
text_id
FK1
text_type_id
value
label
edited_by

Figure 13

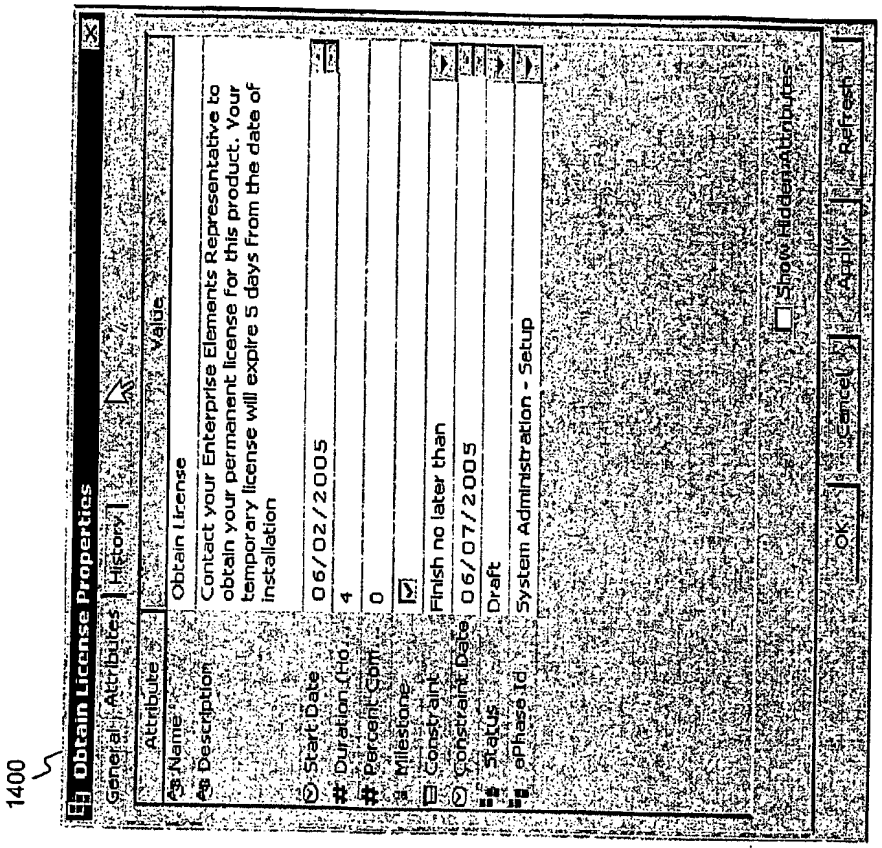

1400
Obtain License Properties
General | Attributes | History
Attribute
Value
Name
Obtain License
Description
Contact your Enterprise Elements Representative to obtain your permanent license for this product. Your temporary license will expire 5 days from the date of installation
Start Date
06/02/2005
Duration (Ho...
4
Percent Com...
0
Milestone
Constraint
Finish no later than
Constraint Date
06/07/2005
Status
Draft
ePhase Id
System Administration - Setup
Show Hidden Attributes
OK
Cancel
Apply
Refresh

Figure 14

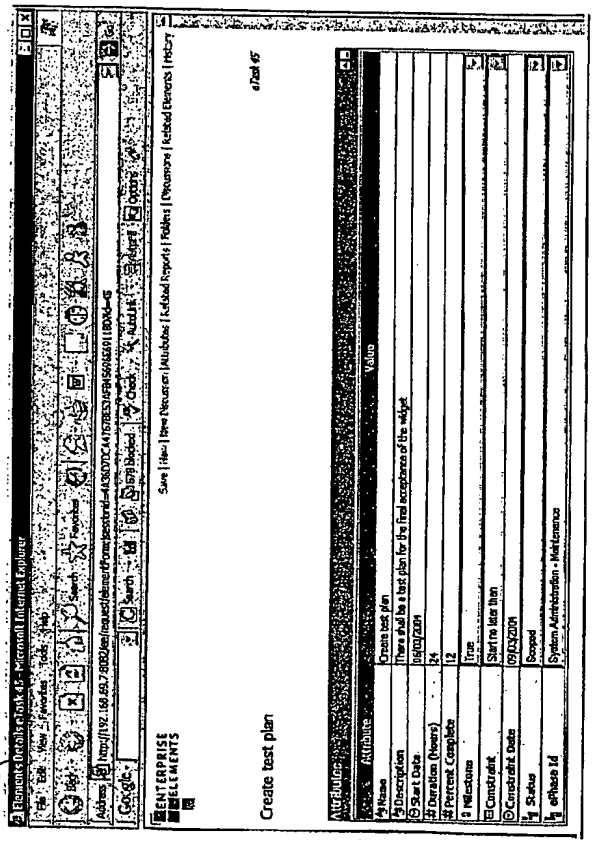

1500
Create test plan
Attributes
Value
Name
Create test plan
Description
Start Date
15/07/2011
Duration (Hours)
24
Percent Complete
12
Milestone
True
Constraint
Start no later than
Constraint Date
09/12/2011
Status
Scoped
Assignment Group
System Administration - Maintenance

Figure 15

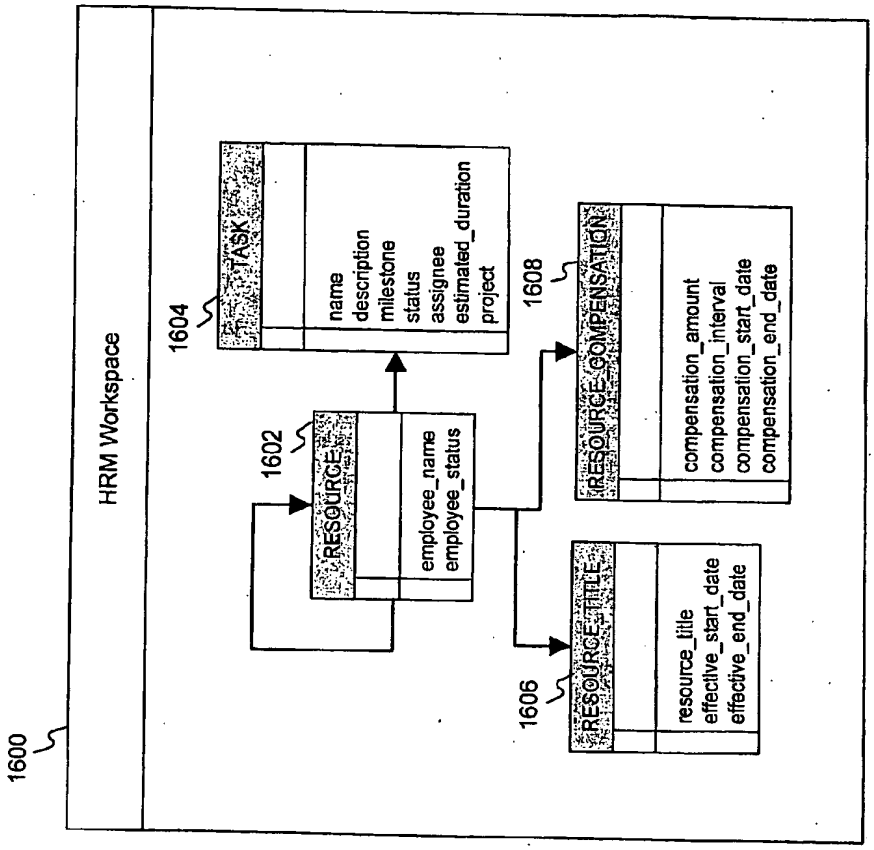

HRM Workspace
1600
TASK
name
description
milestone
status
assignee
estimated_duration
project
1604
RESOURCE
employee_name
employee_status
1602
RESOURCE_TITLE
resource_title
effective_start_date
effective_end_date
1606
RESOURCE_COMPENSATION
compensation_amount
compensation_interval
compensation_start_date
compensation_end_date
1608

Figure 16

17/17

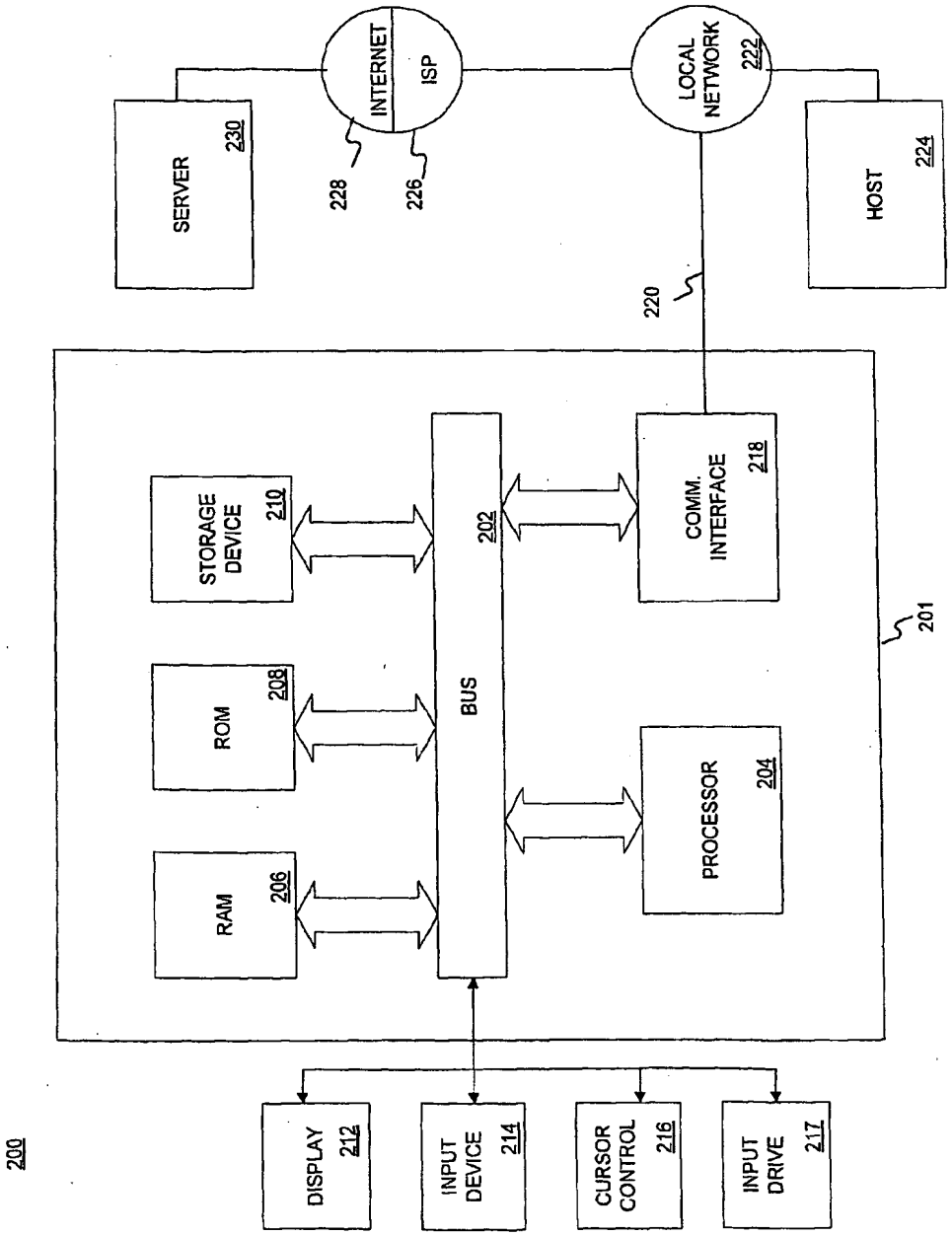


Figure 17

INTERNATIONAL SEARCH REPORT ISR

International application No
PCT/US2006/023242

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ANDRZEJ JODLOWSKI, PIOTR HABELA, JACEK PLODZIEN, KAZIMIERZ SUBIETA: "Objects and Roles in the Stack-Based Approach" PROC. OF THE 13TH INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS (DEXA 2002), AIX-EN-PROVENCE, FRANCE, LNCS, vol. 2453, 2 September 2002 (2002-09-02), pages 514-523, XP002401006 page 515, line 28 - page 516, line 13 page 518, line 18 page 520, line 5 - last line figure 3 ----- -/--	1-39

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
29 September 2006	12/10/2006

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Bykowski, Artur

Form PCT/ISA/210 (second sheet) (April 2005)

ServiceNow Ex. 1002 Page 255 of 410

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PIOTR HABELA, KAZIMIERZ SUBIETA: "Overcoming the Complexity of Object-Oriented DBMS Metadata Management" PROC. OF THE 9TH INTERNATIONAL CONFERENCE OBJECT-ORIENTED INFORMATION SYSTEMS (OOIS 2003), GENEVA, SWITZERLAND, 2 September 2003 (2003-09-02), pages 214-225, XP002401007 page 216, line 16 - last line page 220, line 2 - page 221, line 26 page 223, line 10 - page 224, line 31 figures 3,4	1-39
X	BLAHA M ET AL: "Converting OO models into RDBMS schema" IEEE SOFTWARE USA, vol. 11, no. 3, May 1994 (1994-05), pages 28-39, XP002401008 ISSN: 0740-7459 page 32, left-hand column, line 12 - page 34, left-hand column, line 12	1-39
X	SILYAN ARSOV, BORIS RACHEV: "An approach for design of the object-oriented databases with restricted natural language access" PROC. OF THE 4TH INTERNATIONAL CONFERENCE CONFERENCE ON COMPUTER SYSTEMS AND TECHNOLOGIES: E-LEARNING, SOFIA, BULGARIA, 19 June 2003 (2003-06-19), pages 147-152, XP002401009 page 148, line 8 - page 151, line 35	1,4,5,8, 9,12-14, 17,18, 21,22, 25-27, 30,31, 34,35, 38,39
A	NOEL JERKE, DARIN BEARD: "Building Database-Driven Flash Applications, chapter 5, Building a Flashy Online Poll Engine" [Online] October 2003 (2003-10), APRESS, SPRINGER-VERLAG BERLIN AND HEIDELBERG GMBH & CO. KG , BERKLEY, USA , XP002401199 ISBN: 1-59059-110-0 Retrieved from the Internet: URL:http://www.apress.com/ApressCorporate/supplement/1/186/1590591100-1450.pdf> [retrieved on 2006-09-28] page 122 - page 141	1-39

PATENT COOPERATION TREATY

PCT

INTERNATIONAL PRELIMINARY REPORT ON PATENTABILITY
(Chapter I of the Patent Cooperation Treaty)

(PCT Rule 44*bis*)

Applicant's or agent's file reference 9801.2-304	**FOR FURTHER ACTION**	See item 4 below
International application No. PCT/US2006/023242	International filing date *(day/month/year)* 14 June 2006 (14.06.2006)	Priority date *(day/month/year)* 14 June 2005 (14.06.2005)
International Patent Classification (8th edition unless older edition indicated) See relevant information in Form PCT/ISA/237		
Applicant ENTERPRISE ELEMENTS, INC.		

1. This international preliminary report on patentability (Chapter I) is issued by the International Bureau on behalf of the International Searching Authority under Rule 44 *bis*.1(a).

2. This REPORT consists of a total of 6 sheets, including this cover sheet.

   In the attached sheets, any reference to the written opinion of the International Searching Authority should be read as a reference to the international preliminary report on patentability (Chapter I) instead.

3. This report contains indications relating to the following items:

☒	Box No. I	Basis of the report
☐	Box No. II	Priority
☐	Box No. III	Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
☐	Box No. IV	Lack of unity of invention
☒	Box No. V	Reasoned statement under Article 35(2) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
☐	Box No. VI	Certain documents cited
☐	Box No. VII	Certain defects in the international application
☐	Box No. VIII	Certain observations on the international application

4. The International Bureau will communicate this report to designated Offices in accordance with Rules 44*bis*.3(c) and 93*bis*.1 but not, except where the applicant makes an express request under Article 23(2), before the expiration of 30 months from the priority date (Rule 44*bis* .2).

	Date of issuance of this report 17 December 2007 (17.12.2007)
The International Bureau of WIPO 34, chemin des Colombettes 1211 Geneva 20, Switzerland Facsimile No. +41 22 338 82 70	Authorized officer Nora Lindner e-mail: pt02.pct@wipo.int

From the
INTERNATIONAL SEARCHING AUTHORITY

To:

see form PCT/ISA/220

PCT

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY
(PCT Rule 43*bis*.1)

Date of mailing *(day/month/year)* see form PCT/ISA/210 (second sheet)	
Applicant's or agent's file reference see form PCT/ISA/220	**FOR FURTHER ACTION** See paragraph 2 below
International application No. PCT/US2006/023242	International filing date *(day/month/year)* 14.06.2006
Priority date *(day/month/year)* 14.06.2005	
International Patent Classification (IPC) or both national classification and IPC INV. G06F17/30	
Applicant ENTERPRISE ELEMENTS, INC.	

1. This opinion contains indications relating to the following items:

- ☒ Box No. I Basis of the opinion
- ☐ Box No. II Priority
- ☐ Box No. III Non-establishment of opinion with regard to novelty, inventive step and industrial applicability
- ☐ Box No. IV Lack of unity of invention
- ☒ Box No. V Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement
- ☐ Box No. VI Certain documents cited
- ☐ Box No. VII Certain defects in the international application
- ☐ Box No. VIII Certain observations on the international application

2. **FURTHER ACTION**

If a demand for international preliminary examination is made, this opinion will usually be considered to be a written opinion of the International Preliminary Examining Authority ("IPEA") except that this does not apply where the applicant chooses an Authority other than this one to be the IPEA and the chosen IPEA has notifed the International Bureau under Rule 66.1*bis*(b) that written opinions of this International Searching Authority will not be so considered.

If this opinion is, as provided above, considered to be a written opinion of the IPEA, the applicant is invited to submit to the IPEA a written reply together, where appropriate, with amendments, before the expiration of 3 months from the date of mailing of Form PCT/ISA/220 or before the expiration of 22 months from the priority date, whichever expires later.

For further options, see Form PCT/ISA/220.

3. For further details, see notes to Form PCT/ISA/220.

Name and mailing address of the ISA:	Date of completion of this opinion	Authorized Officer
European Patent Office - P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk - Pays Bas Tel. +31 70 340 - 2040 Tx: 31 651 epo nl Fax: +31 70 340 - 3016	see form PCT/ISA/210	Bykowski, Artur Telephone No. +31 70 340-4565

15/01/2008

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US2006/023242

Box No. I Basis of the opinion

1. With regard to the **language**, this opinion has been established on the basis of:

   ☒ the international application in the language in which it was filed

   ☐ a translation of the international application into , which is the language of a translation furnished for the purposes of international search (Rules 12.3(a) and 23.1 (b)).

2. With regard to any **nucleotide and/or amino acid sequence** disclosed in the international application and necessary to the claimed invention, this opinion has been established on the basis of:

   a. type of material:

      ☐ a sequence listing

      ☐ table(s) related to the sequence listing

   b. format of material:

      ☐ on paper

      ☐ in electronic form

   c. time of filing/furnishing:

      ☐ contained in the international application as filed.

      ☐ filed together with the international application in electronic form.

      ☐ furnished subsequently to this Authority for the purposes of search.

3. ☐ In addition, in the case that more than one version or copy of a sequence listing and/or table relating thereto has been filed or furnished, the required statements that the information in the subsequent or additional copies is identical to that in the application as filed or does not go beyond the application as filed, as appropriate, were furnished.

4. Additional comments:

Form PCT/ISA/237 (April 2005)

ServiceNow Ex. 1002 Page 259 of 410

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY

International application No.
PCT/US2006/023242

Box No. V Reasoned statement under Rule 43*bis*.1(a)(i) with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

1. Statement

Novelty (N)	Yes:	Claims	10-11,23-24,36-37
	No:	Claims	19,12-22,25-35,38-39
Inventive step (IS)	Yes:	Claims	
	No:	Claims	1-39
Industrial applicability (IA)	Yes:	Claims	1-39
	No:	Claims	

2. Citations and explanations

see separate sheet

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY (SEPARATE SHEET)

International application No.
PCT/US2006/023242

Re Item V.

1 Reference is made to the following document:
D1 : ANDRZEJ JODLOWSKI *et al*: "Objects and Roles in the Stack-Based Approach", PROC. OF THE 13TH INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS (DEXA 2002), AIX-EN-PROVENCE, FRANCE, LNCS vol. 2453, 2 September 2002 (2002-09-02), pages 514-523, XP002401006

2 INDEPENDENT CLAIM 1

2.1 The present application does not meet the criteria of Article 33(1) PCT, because the subject-matter of claim 1 is not new in the sense of Article 33(2) PCT.
Document D1 discloses (the references in parentheses applying to this document) a data repository, the data repository being adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets (p. 515, l. 18-20), comprising:
- an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification (p. 520, l. 5);
- an element relation module configured to store one or more relationships between the elements in the element module (p. 520, l. 5; p. 516, l. 6);
- a class module configured to define at least one class of the elements and store the class (p. 520, l. 6);
- an attribute module configured to define one or more attributes and store the attributes (p. 516, l. 11-12);
- a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class (p. 516, l. 11-12); and
- a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements (p. 515, l. 30-31; p. 520, last line).

WRITTEN OPINION OF THE INTERNATIONAL SEARCHING AUTHORITY (SEPARATE SHEET)

International application No.
PCT/US2006/023242

3 DEPENDENT CLAIMS 2-13, 15-26, 28-39
Dependent claims 2-13, 15-26, 28-39 do not contain any features which, in combination with the features of any claim to which they refer, meet the requirements of the PCT in respect of novelty and/or inventive step (Article 33(2) and (3) PCT).

4 Since the subject-matter of each of independent claims 14, 27 corresponds to the subject matter of claim 1, the same reasoning as given for claim 1 will apply mutatis mutandis.

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/023242

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ANDRZEJ JODLOWSKI, PIOTR HABELA, JACEK PLODZIEN, KAZIMIERZ SUBIETA: "Objects and Roles in the Stack-Based Approach" PROC. OF THE 13TH INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS (DEXA 2002), AIX-EN-PROVENCE, FRANCE, LNCS, vol. 2453, 2 September 2002 (2002-09-02), pages 514-523, XP002401006 page 515, line 28 - page 516, line 13 page 518, line 18 page 520, line 5 - last line figure 3 ----- -/--	1-39

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
29 September 2006	12/10/2006

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Bykowski, Artur

Form PCT/ISA/210 (second sheet) (April 2005)

ServiceNow Ex. 1002 Page 263 of 410

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/023242

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PIOTR HABELA, KAZIMIERZ SUBIETA: "Overcoming the Complexity of Object-Oriented DBMS Metadata Management" PROC. OF THE 9TH INTERNATIONAL CONFERENCE OBJECT-ORIENTED INFORMATION SYSTEMS (OOIS 2003), GENEVA, SWITZERLAND, 2 September 2003 (2003-09-02), pages 214-225, XP002401007 page 216, line 16 - last line page 220, line 2 - page 221, line 26 page 223, line 10 - page 224, line 31; figures 3,4	1-39
X	BLAHA M ET AL: "Converting OO models into RDBMS schema" IEEE SOFTWARE USA, vol. 11, no. 3, May 1994 (1994-05), pages 28-39, XP002401008 ISSN: 0740-7459 page 32, left-hand column, line 12 - page 34, left-hand column, line 12	1-39
X	SILYAN ARSOV, BORIS RACHEV: "An approach for design of the object-oriented databases with restricted natural language access" PROC. OF THE 4TH INTERNATIONAL CONFERENCE CONFERENCE ON COMPUTER SYSTEMS AND TECHNOLOGIES: E-LEARNING, SOFIA, BULGARIA, 19 June 2003 (2003-06-19), pages 147-152, XP002401009 page 148, line 8 - page 151, line 35	1,4,5,8, 9,12-14, 17,18, 21,22, 25-27, 30,31, 34,35, 38,39
A	NOEL JERKE, DARIN BEARD: "Building Database-Driven Flash Applications, chapter 5, Building a Flashy Online Poll Engine" [Online] October 2003 (2003-10), APRESS, SPRINGER-VERLAG BERLIN AND HEIDELBERG GMBH & CO. KG , BERKLEY, USA , XP002401199 ISBN: 1-59059-110-0 Retrieved from the Internet: URL:http://www.apress.com/ApressCorporate/supplement/1/186/1590591100-1450.pdf> [retrieved on 2006-09-28] page 122 - page 141	1-39

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

ServiceNow Ex. 1002 Page 264 of 410

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

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 December 2006 (28.12.2006)

PCT

(10) International Publication Number
WO 2006/138420 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/US2006/023242

(22) International Filing Date: 14 June 2006 (14.06.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/690,147 14 June 2005 (14.06.2005) US

(71) Applicant *(for all designated States except US)*: **ENTERPRISE ELEMENTS, INC.** [US/US]; 11654 Plaza America Drive, Suite 639, Reston, VA 20190-4700 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **DEWOOLFSON, Bruce, Randall** [US/US]; 46290 Summerhill Place, Potomac Falls, VA 20165 (US).

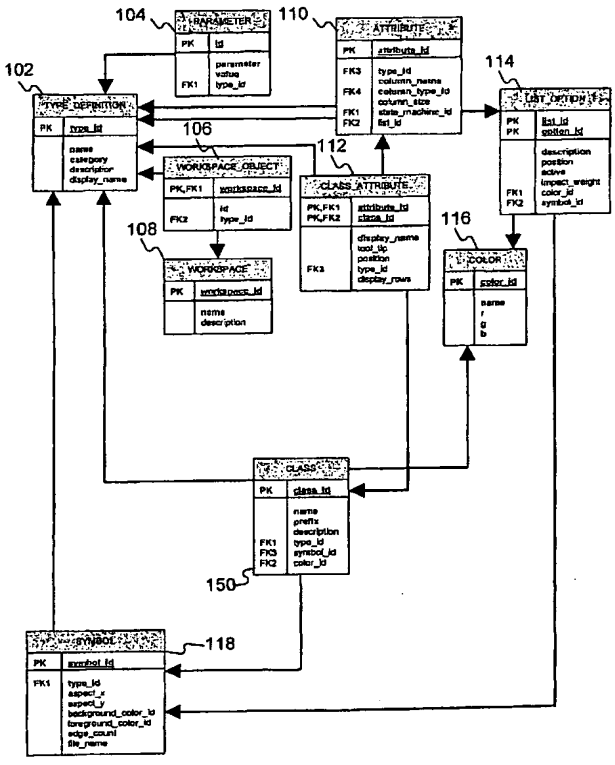
(74) Agents: **GARRETT, Arthur, S.** et al.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.l.p., 901 New York Avenue, Nw, Washington, DC 20001-4413 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,

[Continued on next page]

(54) Title: DATABASE DATA DICTIONARY



(57) Abstract: Systems and methods are provided for manipulating data sets. In accordance with one implementation, a computerized system is provided for storing, managing, indexing, interrelating, and/or retrieving data sets in a manner independent of the data model. The system includes an element module configured to store and uniquely identify elements and an element relation module configured to store relationships between the elements in the element module. The computerized system may also comprise a class module configured to store attributes of elements in a class and a type definition module configured to define the class and the attributes related to the class. The computerized system may further comprise a state machine module, the state machine module including a state machine transition module and a status module.

WO 2006/138420 A1

RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

(48) Date of publication of this corrected version:
29 November 2007

(15) Information about Correction:
see PCT Gazette No. 48/2007 of 29 November 2007

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

DATABASE DATA DICTIONARY

DESCRIPTION

Cross-Reference to Related Application(s)

[001] This application claims the benefit of priority from U.S. Provisional Application No. 60/690,147, entitled "Database Data Dictionary," filed June 14, 2005, the disclosure of which is expressly incorporated herein by reference to its entirety.

Technical Field

[002] This invention relates to the field of computer data storage. More particularly, the present invention relates to methods and systems for storing, managing, indexing, interrelating, and/or retrieving data sets.

Background Information

[003] Currently, a conventional data storage system implements its own data model according to the user interface and the business rules specification for that system. For the conventional system, the system's developers write dedicated code and user interface code that displays or manages certain data inputs such as check boxes, lists, combo boxes or other data inputs. Such systems depend upon a specific data model and lack flexibility and portability. The conventional system's dedicated code and user interface in some respects manipulate the data that the conventional system's application stores and retrieves during operation. Moreover, the conventional system stores and retrieves data locally on a local disk, such as a C drive on a Windows system. It is well known that conventional systems may also store data on a remote storage media, e.g. a file server or database.

[004] There are efforts to overcome the lack of flexibility and portability of conventional data storage systems, such as the adoption and utilization of extensible markup language for storing data (e.g., Microsoft Office® XML Open Format, OASIS Open Office XML Format, etc). Data files in open data formats (e.g., XML) facilitate data transfer and are typically stored on a desktop machine. XML data files rely on the structures defined by XML standards, which allows standardized XML parsers to read data files expressed in XML format.

[005] However, open data formats based on XML and other open standards do not interrelate and manage dynamic data sets through process and collaboration and state machines. Open data formats (e.g., XML) also do not provide for inheritance and override behaviors, collaboration or state machine process enforcement, and/or element style normalization.

[006] Accordingly, there is a need for systems and methods that store, manage, index, interrelate, and/or retrieve data sets in a manner independent of the data model.

SUMMARY

[007] In accordance with the principles of the present invention, as embodied and broadly described herein, methods, systems and computer program products are provided for manipulating data sets. The data set manipulation may be achieved through utilization of a processor and a data repository adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets.

[008] According to one embodiment, a system is provided for manipulating data sets. The system comprises a processor and a repository, the data repository including

an element module configured to store and uniquely identify a plurality of elements each identified by a unique element identification, an element relation module configured to store one or more relationships between the elements in the element module, and a class module configured to define at least one class of the elements and store the class. The data repository further includes an attribute module configured to define and store the attributes, a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class, and a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements.

[009] According to another embodiment, a method is provided for manipulating data sets. The method comprises storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, and storing one or more relationships between the elements in the data repository. The method further comprises defining at least one class of the elements and storing the class in the data repository, defining one or more attributes and storing the attributes in the data repository, defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository, and defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

[010] Another exemplary embodiment provides for a computer-readable medium containing instructions to configure a processor to perform a method for manipulating data sets. The method comprises storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, and storing one or more relationships between the elements in the data repository. The method further comprises defining at least one class of the elements and storing the class in the data repository, defining one or more attributes and storing the attributes in the data repository, defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository, and defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

[011] Additional aspects of the invention are set forth in the detailed description which follows or may be learned by practice of methods, systems, and articles of manufacture consistent with the present invention. It is understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[012] Both the foregoing general description and the following detailed description are exemplary and explanatory only. They do not restrict the invention, as claimed. Furthermore, the accompanying drawings, which are incorporated in and

constitute a part of this specification, illustrate embodiments of the invention and together with the description, serve to explain the principles of the invention. In the drawings:

[013] Figure 1 is an exemplary entity-relationship diagram of an elements metadata repository (EMR) for storing, managing, indexing, interrelating, and/or retrieving data sets, consistent with the principles of the present invention;

[014] Figure 2 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing articulated types, workspaces and workspace objects, class and class attributes, parameters, list options, etc;

[015] Figure 3 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing elements, element relations, element attributes, element history, element documents, etc;

[016] Figure 4 illustrates an exemplary diagrammatic visualization for visually selecting and relating elements using a graphical user interface (GUI);

[017] Figure 5 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing state machines, state machine transitions, state machine statuses, etc;

[018] Figure 6 illustrates an exemplary state machine diagram;

[019] Figure 7 illustrates a GUI of an exemplary diagrammatic depiction displaying congregations of requirements that are related to subcomponents of a system;

[020] Figure 8 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing tuples, named texts and metadata for named texts, etc;

[021] Figure 9 illustrates a GUI of an exemplary elements explorer;

[022] Figure 10 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing collaboration status history, polls, discussions, system components, etc;

[023] Figure 11 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing terminology, glossary, etc;

[024] Figure 12 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing emails, email details, etc;

[025] Figure 13 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing formatted dataset, SQL extensible stylesheet language transformations, etc;

[026] Figures 14 and 15 depict GUIs of exemplary editing environments that may be invoked from a displayed element;

[027] Figure 16 is a diagram depicting an exemplary Human Resource Management workspace; and

[028] Figure 17 is a diagram depicting an exemplary computer system in which systems and methods consistent with the principles of the elements metadata repository of Figure 1 may be implemented.

DETAILED DESCRIPTION

[029] Reference will now be made in detail to present embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Entity-Relationship Configuration

[030] FIG. 1 shows an entity relationship diagram 100 of an elements metadata repository (EMR) for storing, managing, indexing, interrelating, and/or retrieving data sets, consistent with principles of the present invention. In accordance with embodiments of the invention, a repository, such as the EMR, is a central place where data and metadata is stored and maintained. Metadata is data that is used to describe other data. Examples of metadata include schema, tables, indices, views, column definitions, etc. Metadata is recursive such that data about data about the data may be stored. This recursive nature may be expressed as levels, such as "M" number of levels. For example, the actual data being stored is M0. Data about that actual data (i.e., M0) may be referred to as M1. Data about the M1 layer data may be referred to as M2 and so on. In this example, the M1 and M2 layers are metadata, i.e., data about data. The EMR of FIG. 1 allows for storage of any number of data layers.

[031] Embodiments of the present invention, as shown in FIG. 2, include a class table 150, which may be analogous to a relational database table. Just as a table contains columns, a class, such as an entry in class table 150, contains attributes. In an exemplary implementation of the EMR, using database systems such as Oracle®, Microsoft SQL Server®, or MySQL, a user may create a class and its attributes by

creating a table and constructing columns within the table. In other words, the user creates a class and populates the class with attributes where attributes are conceptually other pieces of information of interest to the things that the class is trying to describe. For instance, if the class was a person, an attribute may be the person's name, birth date, or social security number. Any number of attributes can be assigned to any number of classes.

[032] Unlike a conventional relational database, the EMR extends the notion of available attributes types beyond the standard string, number, and date types. To do so, the EMR allows dynamically extendable entries in a type definition table 102. Type definition table 102 may store articulated types that may be, for example, an M4 layer while still maintaining the meta meta meta meta data language. By using the type of the person's name for an attribute, the user can declare more than what a normal relational database may store. For example, the EMR provides for a boolean type which is not normally found in a relational database. The EMR also provides for a pick list type, which stores a plurality of optional selections, and restricts actual data settings to none, or one and only one of the optional selections. The pick list type is also not found in a relational database. Moreover, the EMR is capable of storing a process or state machine type which is not found in a relational database. All these data types provide flexibility in the type of data management that can be performed.

[033] In accordance with still further embodiments of the invention, the EMR includes a parameter table 104. Parameter table 104 allows for any unforeseen attribute values to be placed upon system level components without redefining the base schema. For instance, if one type of named text is a SQL query, and the system

requires security settings on that one type of named text (further described below), then a parameter entry can be used to specify that setting instead of redefining named_text table (e.g. named_text table 140 in FIG. 8) to include an additional column, which may or may not be needed for all other named text types.

[034] Embodiments of the present invention, as shown in FIG. 2, include a workspace table 108 and a workspace_object table 106. Workspace table 108 and workspace_object table 106 provide for logical grouping of elements into subsets that are usable for particular users or purposes. For example, if an EMR user is a tester, then a "Testing" workspace may be declared. Objects, including elements, classes, named texts, and "things" that is defined by a numeric identifier (which may be every "thing") may be placed into 'Testing' workspace via entries in workspace_object table 106. The tester would be assigned to this workspace (also through workspace_object table 106), and therefore only be permitted to see the other objects in the workspace. The effect is a basic level of security and data volume reduction to small localized easily navigable subsets.

[035] Consistent with embodiments of the invention, an attribute table 110 holds the definition of each attribute used in each class in the EMR. Attribute table 110 may store the physical name and type of the data which corresponds to the database column name and column type where the actual values are stored. A class_attribute table 112 may associate a given attribute to one or more classes (in class table 150). Because class_attribute table 112 allows for one attribute to be referenced by many classes, there is attribute inheritance among the classes. Attribute inheritance among the classes is a powerful notion not found in traditional relational database management

systems (RDBMS). Class_attribute table 112 stores the logical name that may be used by the attribute when presented to the user for data manipulation such as insert or update or delete of values. The ability to reference the same attribute via multiple names depending on which class it appears in is akin to the concept of overridden inheritance, also a concept not found in RDBMS.

[036] In an embodiment of the invention, the EMR may immediately create a physical instantiation of the logical class and attribute model as soon as a class attribute definition is entered and committed into class_attribute table 112. The EMR may immediately represent the table and columns that the class and attribute definitions dictate. Moreover, the EMR's editors (see, for example, FIG. 14 and 15) may immediately conform to this definition and support full data manipulation of any values in any defined attribute. Furthermore, when certain types of attributes are attached to a class, appropriate named text entries are automatically created and tupled to the class and may be immediately available for use in the EMR. For example, if a pick list attribute type is added to a class, then queries that returns, for instance, all rows in the class and how many rows fall into each of the available pick list options, are automatically created, which allows the EMR to present the results (via charts or other presentation formats) to the users. This immediate construction of physical storage, complete editing capability, and automated reporting allows the EMR to become operational immediately upon logical schema definition.

[037] Embodiments of the present invention, as shown in FIG. 2, further include a list_option table 114. List_option table 114 may hold all possible enumerations in all possible pick list attributes, which allows for structured user interface and predictable

query structures when displaying and managing pick list data. A color table 116 may hold a record for each enumerated color referenced by any object in the EMR. A color can be declared and associated to an object such as a class, list option selection, or element, etc, and when detected during presentation, the id reference value for the color can be replaced with a swatch of the actual color or any other run time decoration as appropriate.

[038] Consistent with embodiments of the invention, the EMR includes a symbol table 118. Symbol table 118 is similar to color table 116 because it may hold a list of references to graphical objects that can be used in the system. The number and type of graphical objects are not restricted, and can be extended by addition of type definition entries but typically include standard graphic files, such as .BMP, .GIF, .JPG, .SVG, and .SVGz. Once declared in this table, these graphical objects can be referenced by any object in the system, and when the reference is detected at display time, the reference may be replaced with a rendering of the actual image. For example, reports may be constructed that show check marks, colored balls, military insignia, or any other relevant image depending on the needs of the visualization.

[039] The EMR of FIG. 1 conceptually breaks the world into an articulation of "things." The EMR places these "things" into one pile in an element table 160. An element stored in element table 160 implies that the user can find similarity between any "thing" that the user would like to discuss, report, store, or manage. Element table 160 provides the user the capability to add those "things" into a congruous pile so that the user can deal with them without regard to their exact nature. This provides an over-normalization of any data captured by the EMR into one record type called an element.

[040] FIG. 3 illustrates an exemplary element, which has a unique identifier (e.g., element ID), a name (e.g., description), and a classification (e.g., class identifier). Consistent typing of the element ID permits the user to refer to elements across the schema without having to convert the type of the element ID. In one embodiment, the element ID takes on a number format, e.g., a long value or an integer value. In this embodiment, any "thing" in the world may be uniquely identified by a number, e.g., an integer. Moreover, once any "thing" has been differentiated from nothingness, the "thing" has a name. This name may be called a description. In one embodiment, the key components of the element are the element ID and the name, as shown in element table 160. The description does not have to be unique. Some "things" may be described generally with the same name but have differences that render those things unique from each other. Another component of the element is a class identifier. The class identifier is a pointer or an indicator of the type or classification of this element in the universe. The class identifier allows the EMR to articulate the differences between types of elements that otherwise appear to be similar.

[041] Embodiments of the present invention include an element_relation table 162 for storing relationships between elements. Further to storing and managing elements in the universe, the EMR of FIG. 1 records and manages the fact that one element relates to another element in element_relation table 162. Each entry in element_relation table 162 may contain a pointer to at least two elements in elements table 160 and a type definition describing the type of relationship. Using the over-normalized model, the EMR ends up with a collection of elements and a collection of indicators of how those elements relate to each other.

[042] Embodiments of the present invention, as shown in FIG. 3, include an element_attr_c# table 164. Element_attr_c# table 164 represents a plurality of physical tables that are constructed into the base RDBMS. A physical table may be constructed in element_attr_c# table 164 for each class that is declared in class table 150. A column is inserted into this table for each class_attribute entry that specifies an attribute that should be in this class. In the physical implementation the # symbol is replaced with the identifier of the class to ensure uniqueness in table names, which is a requirement in the RDBMS but not necessarily in the class names of the element system.

[043] For each element_attr_c# table constructed, a corresponding entry in an element_history table 166 may also be constructed. Element_history table 166 records each version of each element through its lifespan in the EMR. In addition to all values of all attributes, element_history table 166 may also store indicators of the user making the change, and the time the change was made. This provides the EMR the capability to provide a complete history for all tracked elements throughout the life of the elements. This is useful for recovering a point in time configuration, or mixing historic version of various element subsets into named groups sometimes referred to as Baselines, Blocks, Revisions, Versions, Variants, or the like.

[044] Consistent with embodiments of the invention, the EMR includes an element_document table 168, which may represent a series of records that logically group a set of elements into what would traditionally be called a document. Element_document table 168 provides for formatting information that is separated from

the elements themselves, yet are important for the reconstruction for an ordered formatted display of the elements.

[045] From the collection of elements and the collection of indicators, the EMR can create visualizations for users by placing the elements in context with other elements that they relate to. FIG. 4 illustrates an exemplary visualization 400. Visualization 400 may also be called a diagrammatic visualization, which may be implemented by an elements diagram manager. In a diagrammatic visualization, the elements diagram manager selects or allows the user to select subsets of elements in the universe and place them on a drawing canvas. The elements diagram manager then arranges the elements in relation to other elements based on the conductivity rules found in element_relation table 162.

[046] In visualization 400, several elements are arranged on the canvas in proximity to each other and connected by directional lines that indicate their relation to each other, as determined by element_relation table 162. Each element in the canvas may be expanded to find and display elements in proximity to itself that is not yet on the canvas. Visualization 400 provides the user the capability to dynamically explore related elements.

[047] In addition to expansion of an element, each element in the EMR may invoke an editing screen that manages the values for the attributes of that element, based on the classification of that element. FIG. 14 displays a screen shot illustrating an exemplary editing environment 1400 that may be invoked from any displayed element, whether it appears in a diagram canvas, element explorer list, or other context.

[048] As illustrated in FIG. 14, editing environment 1400 shows several tabs, a General Tab, an Attributes tab, and a History tab. The selected tab in the example shown in FIG. 14 is the Attributes tab, which displays a listing of each attribute associated with the elements class, and an editor for the value of that attribute. In this example, the type of attribute is indicated by small icons to the left of the attribute name and the editor appears to the right of the attribute name. The order in which the attributes appear and certain display characteristics such as the number of rows of text to display in the edit field are determined by the entry in class_attribute table 112. Specialized editors can be defined and used such as date selectors, pick list combo boxes, and boolean checkboxes, all of which are invoked dynamically based on the definition of the attributes that happen to be in this class.

[049] FIG. 15 shows a screen shot of another exemplary representation 1500 of the dynamic editing capability of an element. In this embodiment, the EMR provides dynamic edit capability through web-based programming language and a standard browser without a structured programming language such as Java, as seen in editing environment 1400 (of FIG. 14). By providing web-based edit capability, the EMR allows users to edit attributes of an element by invoking one or more hyperlinks to the element when the element is referenced in other web-based reporting or outputs.

[050] State machines are a very powerful extension of normal relational database attributes or column types. As shown in FIG. 5, a state_machine table 120, a state_machine_transition table 122, and a status table 124 describe the structure and behavior of state machines. The term state machine may be used interchangeably with the terms process or workflow. A state machine (e.g., an entry in state_machine table

120) has one or more statuses (e.g., entries in status table 124) which are connected through transitions (e.g., entries in state_machine_transition table 122). The user may classify the elements by declaring one or more possible state machine statuses for each of the elements.

[051] Using the person example, if the person was an employee, then the EMR of FIG. 1 would be able to declare at least one possible status of the person. The EMR could declare the person's status in the state machine model as shown in FIG. 6, such as whether the person is a prospective employee 600, an actively hired employee 610, an employee on probation 620, an employee terminated for cause 630, etc. The EMR may restrict the transition from one status to another by limiting each status's forward transition connections. For instance, the user may first connect prospective employee status 600 to hired status 610. Next, the user may connect hired status 610 to on probation status 620. Furthermore, on probation status 620 may be connected to terminated for cause status 630 or back to hired status 610. Loops and branches are allowable in a stateful process.

[052] The EMR may enforce data behavior and constrain the user's ability to manipulate data based on the state machine model. For one or more stateful models, state machines may be applied to a data classification in a manner that forces data to follow at least one prescribed process declared by the user or a state administrator in a given enterprise. For instance, a prospective employee may not be immediately transitioned to terminated for cause because that violates the rules declared by the process. A prospective employee may only be propagated forward to available or declared next statuses, e.g., hired. Furthermore, the EMR may grant certain users the

ability to restrict the condition under which a transition is allowable, for example, only to users in the organization who are authorized to terminate an employee. Although the transition from actively hired to terminated with cause is generally allowed, there may be one or more exclusionary rules that declare that only a certain user or subset of users may transition an actively hired employee to the status of terminated with cause. Other exclusionary rules might state that the user may not transition an actively hired employee to terminated for cause unless the user has entered a cause and linked it by relation to the employee.

[053] There are many types of data models beyond employees, e.g., the Department of Defense core architectural data model (CADM), the Defense Department architectural framework (DODAF), or the Meta Objects Facility (MOF). These are examples of standardized data models for users to store data, and the models are independent of the data definition that the user wishes to implement. The user can implement multiple concurrent disparate data definitions within a single enterprise repository, such as the EMR of FIG. 1.

[054] Currently, when a system is declared, it implements its own data model according to the user interface and the business rules specification for that system. For the conventional system, the system's developers write dedicated code and user interface code that displays or manages certain data inputs such as check boxes, lists, combo boxes or other data inputs. The conventional system's dedicated code and user interface manipulate the data that the system's application stores and retrieves during operation. Moreover, the conventional system stores and retrieves data locally on a local disk, like a C drive on a Windows system.

[055] There are several drawbacks to conventional systems, one of which is that the data is typically stored in a proprietary file, i.e., a file that is a Microsoft Word® document in .doc format that contains proprietary binary code. The contents of that file are not clear unless one uses Microsoft Word® or another tool that can decode the contents. While Microsoft Word® stores textual information and formats information in a proprietary format contained within an operating system level object called a file, the EMR foregoes the file-based implementation in favor of a elements-based implementation. In the elements-based implementation, the contents of what otherwise would be a file are flexibility represented as elements. Therefore, the contents are not restricted to use by a proprietary application, but are concurrently available to all database-aware applications.

[056] In the elements-based implementation, a document would not be represented by a collection of proprietarily stored formatting and textual information. Instead, the document would be stored as an element with, e.g., one element for each sentence in the document. The elements represent a usable chunk of contextualized information, such as a sentence. In the field of ontologies, words in a sentence by themselves have no meaning. An ontology implies context. Words by themselves may have a definition, but for the words to truly convey meaning, the words need to included in at least a phrase and probably a sentence. In the elements-based approach, each of the elements conforms to the rules of generalized things in the universe, and therefore may be reused or referenced to other things in the universe, e.g., through entries in element_relation table 162. Moreover, each of the elements in its own context is also meaningful and reusable.

[057] In implementing a solution for a requirements management problem, people would typically use a conventional document to articulate thousands or hundreds or tens of requirements that they would like a system or systems to achieve. In an embodiment of the elements-based implementation in the EMR, each requirement statement would be an element that the user may associate a process to, independent of any kind of hard-coded association. The user has a choice to declare, on each element, such a process which is automatically enforced for that user.

[058] Using the EMR, an element or, in this case, a requirement, would be listed and available as generally readable in an open query-able fashion. This allows the user to organize and aggregate that requirement in more than just the context of one requirements document as the user would in a conventional document-based system. For example, the user may organize and aggregate requirements in a diagrammatic depiction 700 as shown in FIG. 7 that shows congregations of requirements that are related to specific subcomponents of the main system that the user is trying to build.

[059] Diagrammatic depiction 700 shows a subset of elements that may represent requirements as in the previous example. In the example illustrated in diagrammatic depiction 700, there are at least two (2) requirements that are common to the major system components represented in diagrammatic depiction 700. If each generally circular grouping is considered a system component and its related requirements, then the left hand component is related to the right hand component through the two elements generally in the middle of the diagram and linked simultaneously to both.

[060] For example, consider a situation where the user declares that an engine is needed in the process of building an airplane. In a file-based system, the user either declares that through the requirements themselves or through the practice of engineering the airplane, for which there are a thousand requirements that describe how the engine should mount to the wing. In the file-based system, the requirements may be in several spreadsheets or documents. The requirements may be repeated in the spreadsheet and a document, and may be repeated again in a testing suite application. Moreover, the requirements may be further repeated on the architectural drawing of the component. In the EMR of FIG. 1 there would be one instance of the requirement and it would be shared among all applications that needed to reference that instance of that requirement. Items such as a diagrammatic depiction (e.g. diagrammatic depiction 700), spreadsheet-like depiction, or documentation depiction, are now file independent. The items are just depictions of the core data set.

[061] Embodiments of the present invention include a named_text table 140, as shown in FIG. 8. Named_text table 140 provides a powerful normalization of all text that the user may refer to in a system. Text is a ubiquitous descriptor of many things, including graphics. With current techniques, there are textual representations of graphics, notably scalable vector graphic (SVG) from companies like Adobe®, which are textual descriptions that can be rendered into graphics. For example, a textual description may state "draw a red circle with a 50 pixel radius." The textual description can be translated into an actual depiction on the screen. This illustrates that text represents most things or most things can be represented by text, not only graphics through the SVG, but also any data structure through extended markup language. The

EMR takes advantage of this by declaring these sets of texts and identifying them with a name and a unique identifier. These sets of texts then become usable in the elements-based implementation in many ways.

[062] Embodiments of the present invention include a tuple table 144, as shown in FIG. 9. Tuple table 144 is similar to element_relation table 162, except that tuple table 144 has been separated to store information about the system schema as opposed to the generated user schema. Tuple table 144 relates any identified item to any other identified item and that relationship is created according to a type (e.g., type_id). Given that an entry in named_text table 140 is a thing and that the user may relate the thing to other types of named text, the EMR declares, for the user's convenience, a named_text item that is akin to a folder in an operating system hierarchy. The folder may be called, for example, "My Favorites," or "My Documents," or in this instance, "My Elements." That folder can then be referenced by the user. Next, the EMR may declare 2, 3, 5 or 100 (or any number of) sub-folders. The sub-folders may be referenced by entries in tuple table 144 to the folder and other sub-folders.

[063] The combination of named_text table 140 and tuple table 144 provides to the EMR the capability to create a recursion between named_text types that represent folders to other named_text types that represent folders in a hierarchy. In an embodiment shown in FIG. 9, the EMR may contain a folder called root (e.g. "EE Root"), another folder called "My elements" (e.g. "My Elements (MARK)" and "My Elements (RONO)"), and tuples the root entry in named_text table 140 to the "My Elements" entry in named_text table 140. In doing so, the EMR indicates through elements explorer 900

a hierarchy or a tree or an indenture of the folders. This particular implementation approximates a standard operating system file explorer, in which the user can click and drag and drop folders and other items into a folder in a tree or navigable folder set. Through the same mechanism, an element may be tupled into a folder. In the elements-based implementation, the EMR may use a folder to organize and aggregate random sets of things in the universe.

[064] The EMR may utilize entries in named_text table 140 to name structured query language queries. SQL is a textual language that describes an interaction or request to a relational database. SQL queries are the mechanism to retrieve data from the database. SQL query is a powerful notion for relational databases and may be implemented as a named_text entry. Unlike a conventional database, e.g., Oracle® or Microsoft SQL Server®, that stores user-generated SQL as a "view," the EMR stores user-generated SQL as a named_text entry. Because the user-generated SQL is an entry in named_text table 140, and since folders exist as entries in named_text table 140 or tuple table 144, the EMR may filter or query entries in named_text table 140 or tuple table 144 into one or more folders.

[065] Thus, if the SQL statement serves as a folder, the results of the SQL statement or the items returned by the query would be the contents of that folder. The SQL statement to the user is therefore a dynamic folder or a SQL folder and no tuple exists between that data set. In other words, elements in the dynamic folder can be placed into other folders. The dynamic folder is executed at run-time if the user selects the dynamic folder and the resulting elements that come back from the query are presented as if they were tupled into that folder. This is a dynamic process and the

result set may differ depending on the dynamic data changes. This dynamic process simplifies the organization and provides to the user the ability to locate data through a dynamic folder hierarchy. The dynamic folder hierarchy further includes dynamic folders that change the data set appropriately every time the user visits them.

[066] An example of the dynamic folder capability provided by the EMR is a "My Tasks" folder. Assume that task is an element classification and that user is an attribute of the task. If there are tasks in the system that are assigned to the user, the user may perform a query that directs the EMR to select one or more tasks for which the user is the assigned user. The result of that query may change over time. The administrator may assign a task to someone else, so the task may be deleted or moved to a completed state and the user's folder would only contain a current task set that matches the query as defined by a named_text of that folder. This allows for an extensible method of organizing data of interest and allowing the user to easily find and retrieve data of interest in a large, complex system.

[067] Embodiments of the present invention, as shown in FIG. 10, include a poll table 182. Poll table 182 may represent a set of identified questions that are in one of several stages of life cycle of an element or set of elements. A poll is a collaborative mechanism that restricts the responses of the participants to one or more predetermined choices which are enumerated in a poll_option table 188. The replies of the participants are recorded in a poll_reply table 184. These poll replies can be analyzed for preference among the voting community and used to record decision making processes regarding any given element or set of elements. This relation is achieved through entries in tuple table 144 (in FIG. 8).

[068] In accordance with still further embodiments of the invention, the EMR includes a discussion table 186, which is similar to poll table 182 as discussion table 186 records collaborative information regarding one or more elements in the system. However, a discussion, which may be stored as entry in discussion table 186, is not forced into a set of structured responses. Instead discussion table 186 allows free-form narrative to be recorded, and replied to by any number of users in the system.

[069] Embodiments of the present invention may further include a collaboration_status_history table 180, which may record all the states that either a discussion or poll moves through in a defined life cycle, similar to element history. Moreoever, the user and date and time is recorded with each status change.

[070] In accordance with still further embodiments of the invention, the EMR includes a system_component table 190. The EMR may register components that are programmatic in nature (e.g. Java classes), wherein the components may be executed dynamically as needed (through a process called reflection) as part of element decoration. For example, one or more Java classes may be defined and saved as a record in system_component table 190, then the Java classes may be tupled to elements. At runtime, the Java classes that are referenced may be executed when an element of the Java class is edited or viewed.

[071] Embodiments of the present invention, as shown in FIG. 11, include a terminology table 130 and a glossary 132. Terminology table 130 may store a list of terminologies that can be used in the system. Glossary table 132 may uniquely identify and define each of the terms stored in terminology table 130.

[072] In accordance with embodiments of the invention as shown in FIG. 12, the EMR includes an email table 172 and an email_detail table 170. These tables allow the EMR to populate records in the database that can programmatically be converted from rows in the RDBMS to actual emails using a known protocols such as SMTP and POP. Since the emails stored in email table 172 and email_detail table 170 are also data, the EMR may apply stateful process logic to each email - using status, state_machine, and state_machine_transition tables (FIG. 5, elements 120, 122, 124, respectively) to define the lifecycle of an email. Full email history can be maintained in the EMR and new emails can be generated either programmatically when an event occurs, or manually when a user enters an email. Individual messages may be aggregated into one larger email at a time interval specified by an administrator. The aggregated messages may also be sent on demand.

[073] In accordance with embodiments of the invention as shown in FIG. 13, the EMR includes a formatted_dataset table 176 and a SQL_XSLT table 178. Together, formatted_dataset table 176 and SQL_XSLT table 178 allow the EMR to create a named "formatted dataset," which applies formatting to a SQL result set. Formatted_dataset table 176 and SQL_XSLT table 178 may work in conjunction with named_text table 140 to relate one or more named text entries (representing SQL statements), which provide the result set(s), with another named text entry (representing XSLT or a formatting definition) which generates a final layout.

[074] To further illustrate the EMR as described above and in FIGS. 1-15, an example is provided to demonstrate the utilization and capability of the EMR, consistent with the present invention, for storing, managing, indexing, interrelating,

and/or retrieving data sets, such as large scale and dynamic data sets. One example, as shown in FIG. 16, may be Human Resource Management (HRM). Further examples may include, for instance, a vertical implementation, or schema and associated processes, that may serve the needs of a well defined market, e.g., Enterprise Architecture, Requirements Management, Six Sigma, Resource Planning, Federal Regulatory Compliance, Legal Document Management, Portfolio Management, etc.

[075] In the HRM example, it may be determined that tracking "resources" is desired. Those resources may be people, but may not necessarily be restricted to people, which is a distinction that may be made when defining the implementation. It may further be determined that the resources of interest will in fact be people, and those people will be employees of a specified company. The company may have written policies that determine employment, such as the amount of vacation time each employee may receive, a level of compensation, etc.

[076] Furthermore, it may be determined that the resources may exist in a hierarchy, e.g., have a place in an organization chart, and that tasks should be identified and assigned to these resources as needed. Steps may be taken within the EMR to support such system requirements.

[077] A workspace (e.g., HRM Workspace 1600 of FIG. 16) may optionally be entered into the ERM (e.g. workspace_object table 106 and workspace table 108 of FIGS. 1 and 2) to segregate the HRM information from other possible data sets in the EMR if desired.

[078] A schema would be logically defined that encompassed the articulated needs of the system. In this case perhaps a resource class 1602 (of FIG. 16) would be

declared with attributes of employee_name, employee_ID, and employee_status. The employee_status attribute may be of type state_machine, allowing the stateful process of employment to be declared according to the rules of the company.

[079] It may further be determined that a resource may change title and salary over time, and other classes, such as a resource_title class 1606 and a resource_compensation class 1608, would be declared each with a pointer to resource class 1602 such as resource_ID and other appropriate attributes such as resource_title, effective_start_date, and effective_end_date. Furthermore, resource_compensation class 1608 may similarly contain compensation_amount, compensation_interval, compensation_start_date, and compensation_end_date.

[080] Tasks would also be similarly defined as a task class 1604 with, for example, a name, a description, a milestone indicator, a status, an assignee, an estimated_duration, and a project indicator.

[081] To implement these classes, data would be entered into the EMR representing these structures. The order of implementation, as illustrated below, is exemplary and does not limit embodiments of the present invention.

[082] Entries may be made into class table 150 (of FIGS. 1 and 2), one each for resource, task, resource_title, and resource_compensation. Entries may be made into attribute table 110 (of FIGS. 1 and 2) for each employee_name, employee_ID, employee_status, employee_title, compensation_interval, compensation_amount, compensation_start_date, compensation_end_date, status, assignee, estimated_duration, and project.

[083] As further definition, entries may be made into state_machine table 120, state_machine_transition table 122, and status table 124 (of FIGS. 1 and 5) to declare the available selections of the status and employee_status attributes. Further entries may be placed into list_option table 114 (of FIGS. 1 and 2) to specify the available resource_titles that may be assigned to resource or resources.

[084] Entries may then be made into class_attribute table 112 (of FIGS. 1 and 2) to associate the appropriate attributes with their respective classes. This may all be done via any data insertion method available, e.g. a user interface, etc.

[085] Once these class_attribute records have been entered into class_attribute table 112, the EMR is ready for use. Users have the capability to add resources, create tasks, assign them to the resources, report on how many tasks are completed, etc. Users may also create folders to organize projects (see, e.g., FIG. 9) and view diagrams (see, e.g., FIGS. 4 and 7) of which resources are committed to which tasks without any further programmatic effort. Users may also be able to collaborate on tasks by discussing ramifications or polling a subset of users to determine specific answers to to-be-determined elements.

[086] In this example, the resource hierarchy may be constructed by linking one resource to another recursively (see, e.g., FIG. 16). This may be done either via the record editing capabilities or by using the diagramming tool (see, e.g., FIGS. 4 and 7) to draw a line from one resource to another. The diagrammatic representation of the organization chart is capable of immediately expand an individual to supervisors, peers, or subordinates. The diagrammatic representation may also show all related tasks to one or more of those individuals if desired. By providing those capabilities, the

diagrammatic representation allows for visualization of tasking overloads or underutilized resources, etc.

[087] If the EMR does not provide a specific aggregation or subsetting of tasks or resources, one or more named_text entries in named_text table 140 (of FIGS. 1 and 8) may be defined that return the appropriate results. If the default display of those results is insufficient for the business needs, then one or more formatted_dataset entries in formatted_dataset table 176 (of FIGS. 1 and 13) and SQL_XSLT entries in SQL_XSLT table 178 (of FIGS. 1 and 13) may be defined to convert the available results into any desired final output.

[088] In this example, when the user edits or inserts data in the EMR, the data may be stored in an appropriate element_attr_c# table 164 (of FIGS. 1 and 3) that was automatically constructed as part of the class definition step above. Moreover, any modifications may automatically be recorded as history in element_history table 166 (of FIGS. 1 and 3).

[089] The above-described exemplary scenario may be followed for any given schema. A full schema does not need to be defined before the EMR may be used. If for instance only resources were defined, then resource information could be entered, and organization hierarchies may be defined at a later date; the tasking information may be added without disruption to the EMR. As a further example, additional attributes may be added to the resources themselves at a later date if the attributes are deemed necessary. As another example, the EMR's user interfaces may automatically adjust to the current definition to allow seamless upgrades of the data models.

[090] Systems and methods consistent with the present invention may be implemented by computers organized in a conventional distributed processing system architecture. FIG. 17 is a block diagram illustrating a computer system 200 in which an embodiment consistent with the invention may be implemented. Computer system 200 includes a computer 201 having a bus 202 or other communication mechanism for communicating information, and a processor 204 coupled with bus 202 for processing information. Computer 201 also includes a main memory, such as random access memory (RAM) 206 or other dynamic storage device, coupled to bus 202 for storing information and instructions by processor 204. RAM 206 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 204. During operation, some or all portions of an operating system (not shown) are loaded into RAM 206. Computer 201 further includes a read only memory (ROM) 208 or other static storage device coupled to bus 202 for storing static information and instructions for processor 204. A storage device 210, such as a magnetic disk or optical disk, is provided and coupled to bus 202 for storing information and instructions.

[091] Computer 201 may be coupled via bus 202 to a display 212, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 214, including alphanumeric and other keys, is coupled to bus 202 for communicating information and command selections to processor 204. Another type of user input device is cursor control 216, such as a mouse, a trackball or cursor direction keys for communicating direction information and command selections to processor 204 and for controlling cursor movement on display 212. This input device may have two degrees

of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), which allow the device to specify positions in a plane.

[092] System 200 may further include an input drive device 217, such as a CD reader, for reading high-capacity computer-readable media, such as CD-ROMs and CDRs.

[093] The invention is related to the use of computer system 200 for manipulating data sets for, to take an example, corporate or engineering exercises. According to one implementation, systems and methods consistent with the present invention collect system information and store it in a central database in response to processor 204 executing one or more sequences of one or more instructions contained in RAM 206. Such instructions may be read into RAM 206 from a computer-readable medium via an input device such as storage device 210. Execution of the sequences of instructions contained in main memory 206 causes processor 204 to perform the process steps described herein. In an alternative implementation, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, implementations consistent with the principles of the present invention are not limited to any specific combination of hardware circuitry and software.

[094] Computer 201 also includes a communication interface 218 coupled to bus 202. Communication interface 218 provides a two-way data communication coupling to a network link 220 that is connected to local network 222. For example, communication interface 218 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 218 may be a local area

network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 218 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[095] Network link 220 typically provides data communication through one or more networks to other data devices. For example, network link 220 may provide a connection through local network 222 to host computer 224 and/or to data equipment operated by Internet Service Provider (ISP) 226. ISP 226, in turn, provides data communication services through the Internet 228. Local network 222 and Internet 228 both use electric, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 220 and through communication interface 218, which carry the digital data to and from computer system 200, are exemplary forms of carrier waves transporting the information.

[096] Computer system 200 can send messages and receive data, including program code, through the network(s), network link 220 and communication interface 218. In the Internet example, a server 230 might transmit a requested code for an application program through Internet 228, ISP 226, local network 222 and communication interface 218. The received code may be loaded in RAM 206 and executed by processor 204 as it is received. Alternatively, or in addition, it may be stored in storage device 210, or other non-volatile storage for later execution. In this manner, computer system 200 may obtain application code in the form of a carrier wave.

[097] Although computer system 200 is shown in FIG. 17 as being connectable to one server 230, those skilled in the art will recognize that computer system 200 may establish connections to multiple servers on Internet 228. Such servers may include an HTML-based Internet application, which may provide information to computer system 200 upon request in a manner consistent with the present invention.

[098] The foregoing descriptions of the invention have been presented for purposes of illustration and description. They are not exhaustive and do not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but the present invention may be implemented as a combination of hardware and software or in hardware alone. Additionally, although aspects of the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM.

WHAT IS CLAIMED IS:

1. A computerized system for manipulating data sets, comprising:

a processor; and

a data repository, the data repository being adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, comprising:

an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification;

an element relation module configured to store one or more relationships between the elements in the element module;

a class module configured to define at least one class of the elements and store the class;

an attribute module configured to define one or more attributes and store the attributes;

a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class; and

a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements.

2. The computerized system of claim 1, wherein the data repository further comprises:

a state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element.

3. The computerized system of claim 2, wherein the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

4. The computerized system of claim 1, wherein the type definition module defines a type including at least one of a boolean type, a picklist type, and a state machine type.

5. The computerized system of claim 4, wherein the data repository further comprises:

a list option module configured to enumerate one or more picklist attributes of the picklist type.

6. The computerized system of claim 2, wherein the data repository further comprises:

a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements.

7. The computerized system of claim 6, wherein the data repository further comprises:

a tuple module configured to generate and store one or more tuples, wherein each of the tuples associates a subset of the elements in accordance to one of the types associated with the subset of the elements.

8. The computerized system of claim 1, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

9. The computerized system of claim 8, wherein the data repository further comprises:

an element attribute module for each of the at least one class, the element attribute module configured to define and store the attributes of the elements in the class, wherein the element attribute module is identified using the name of the class.

10. The computerized system of claim 9, wherein the data repository further comprises:

an element history module configured to store one or more indicators of modification to the elements.

11. The computerized system of claim 10, wherein the data repository further comprises:

an element document module configured to store one or more formatting data for grouping and displaying a subset of the elements.

12. The computerized system of claim 1, wherein the processor is configured to generate a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

13. The computerized system of claim 1, wherein the processor is configured to generate a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

14. A method for manipulating data sets, comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; and

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

15. The method of claim 14, further comprising:

defining a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element, and storing the state machine type in the data repository.

16. The method of claim 15, wherein the state machine type comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

17. The method of claim 14, wherein the type comprises a type including at least one of a boolean type, a picklist type, and a state machine type.

18. The method of claim 17, further comprising:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

19. The method of claim 15, further comprising:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

20. The method of claim 19, further comprising:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

21. The method of claim 14, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

22. The method of claim 21, further comprising:

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

23. The method of claim 22, further comprising:

storing, in the data repository, one or more indicators of modification to the elements.

24. The method of claim 23, further comprising:

storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

25. The method of claim 14, further comprising:

generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

26. The method of claim 14, further comprising:

generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

27. A computer-readable medium containing instructions to configure a processor to perform a method comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; and

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

28. The computer-readable medium of claim 27, wherein the method further comprises:

defining a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element, and storing the state machine type in the data repository.

29. The computer-readable medium of claim 28, wherein the state machine type comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

30. The computer-readable medium of claim 27, wherein the type comprises a type including at least one of a boolean type, a picklist type, and a state machine type.

31. The computer-readable medium of claim 30, wherein the method further comprises:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

32. The computer-readable medium of claim 28, wherein the method further comprises:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

33. The computer-readable medium of claim 32, wherein the method further comprises:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

34. The computer-readable medium of claim 27, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

35. The computer-readable medium of claim 34, wherein the method further comprises:

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

36. The computer-readable medium of claim 35, wherein the method further comprises:

storing, in the data repository, one or more indicators of modification to the elements.

37. The computer-readable medium of claim 36, wherein the method further comprises:

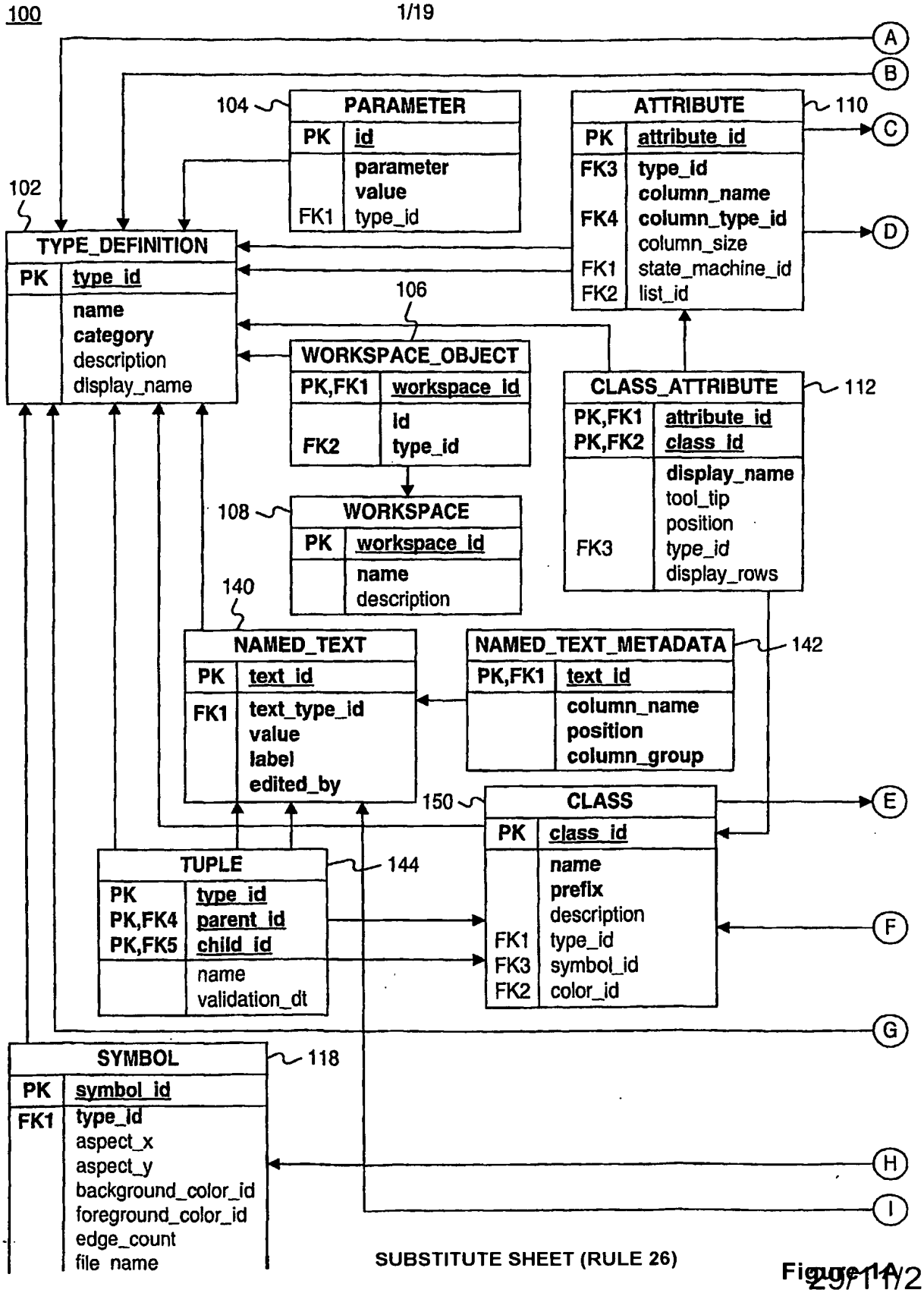
storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

38. The computer-readable medium of claim 27, wherein the method further comprises:

generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

39. The computer-readable medium of claim 27, wherein the method further comprises:

generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

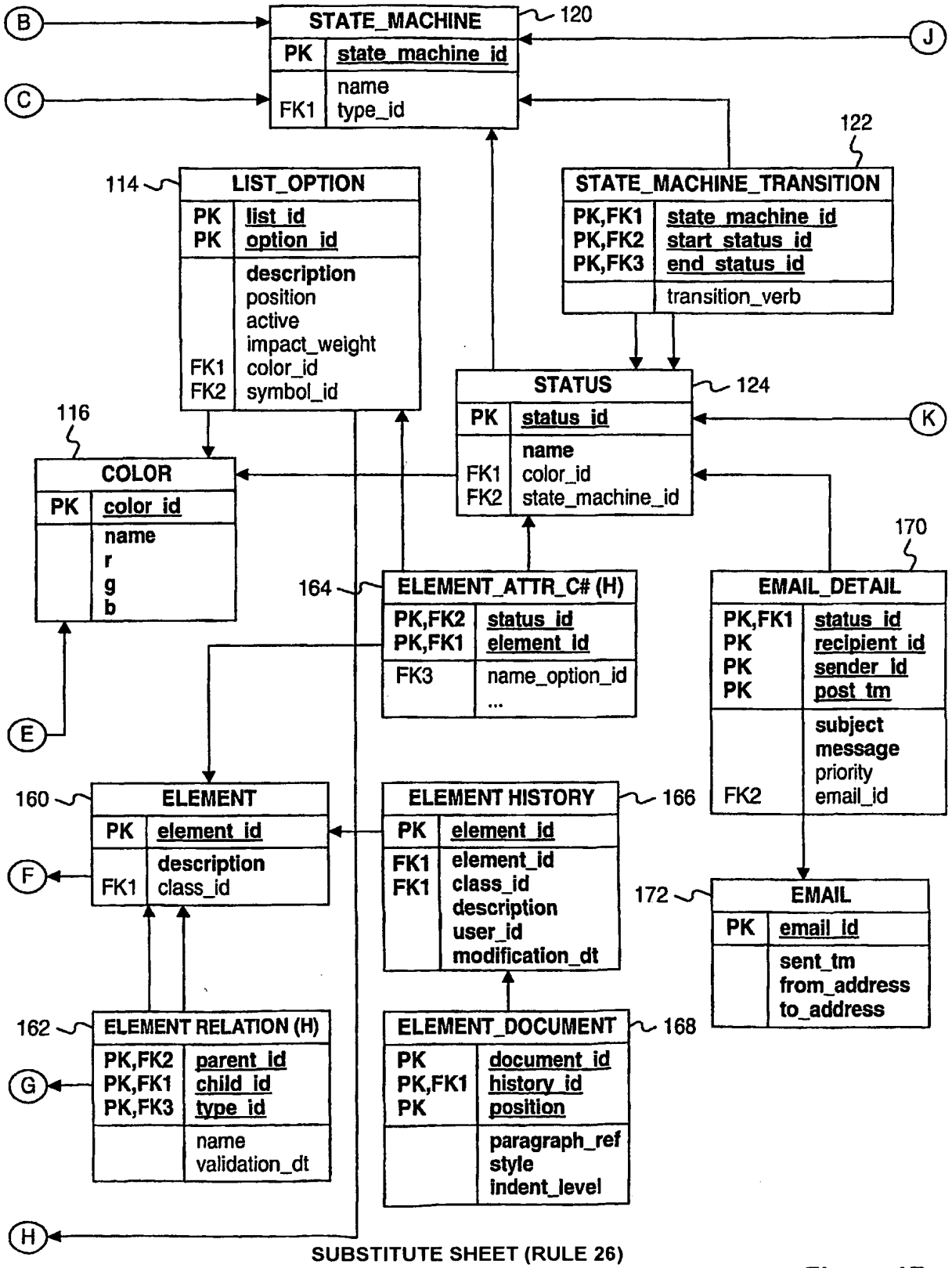


SUBSTITUTE SHEET (RULE 26)

Figure 1A

55

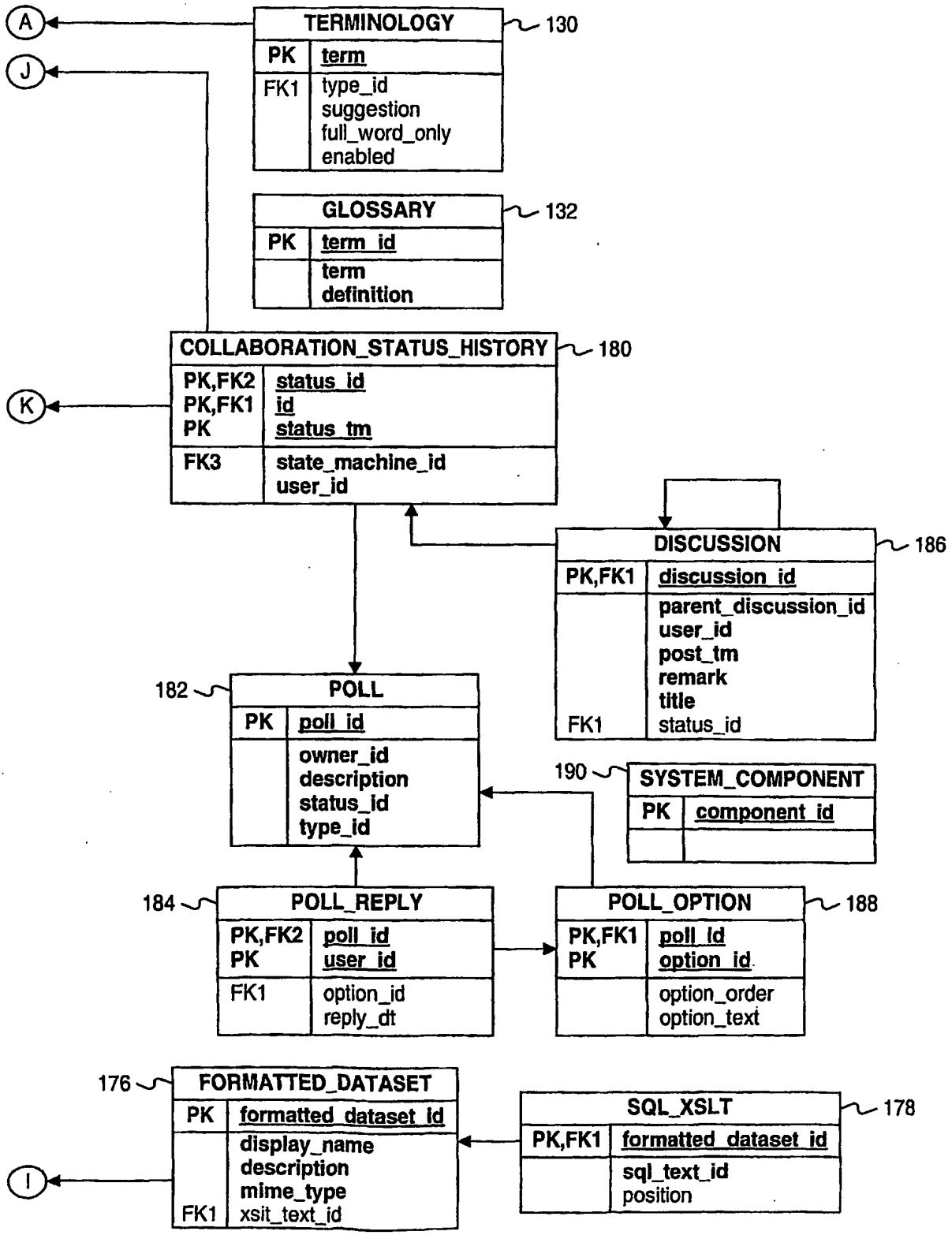
29/11/2007



SUBSTITUTE SHEET (RULE 26)

Figure 1B

ServiceNow Ex. 1002 Page 312 of 410

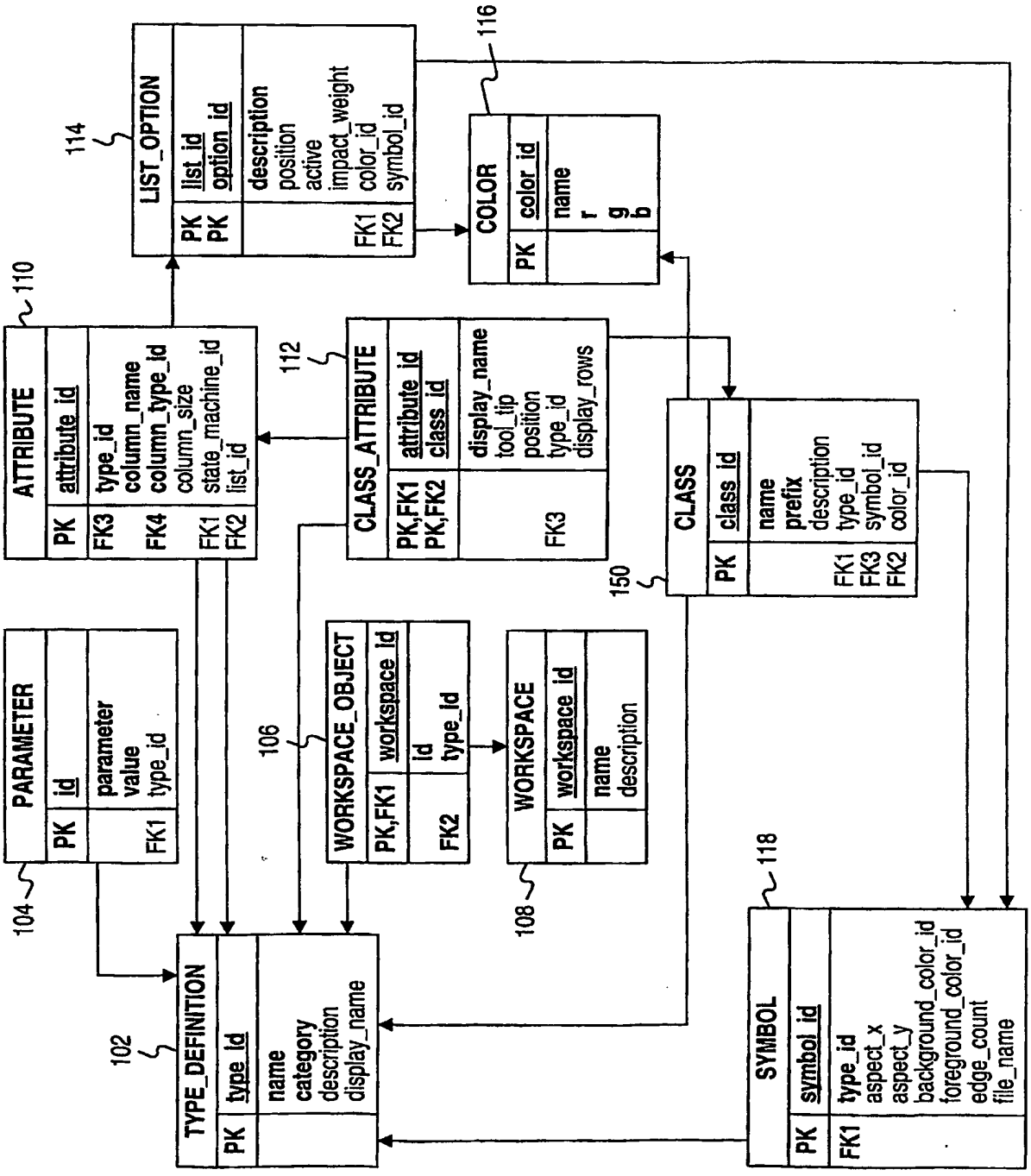


SUBSTITUTE SHEET (RULE 26)

Figure 1C

ServiceNow Ex. 1002 Page 313 of 410

Figure 2


LIST_OPTION
PK list_id
PK option_id
description
position
active
impact_weight
FK1 color_id
FK2 symbol_id
114
116
COLOR
PK color_id
name
r
g
b
110
ATTRIBUTE
PK attribute_id
FK3 type_id
column_name
FK4 column_type_id
column_size
FK1 state_machine_id
FK2 list_id
112
CLASS_ATTRIBUTE
PK,FK1 attribute_id
PK,FK2 class_id
display_name
tool_tip
position
type_id
FK3 display_rows
150
CLASS
PK class_id
name
prefix
description
FK1 type_id
FK3 symbol_id
FK2 color_id
PARAMETER
PK id
parameter
value
FK1 type_id
104
106
WORKSPACE_OBJECT
PK,FK1 workspace_id
id
FK2 type_id
WORKSPACE
PK workspace_id
name
description
108
118
SYMBOL
PK symbol_id
FK1 type_id
aspect_x
aspect_y
background_color_id
foreground_color_id
edge_count
file_name
102
TYPE_DEFINITION
PK type_id
name
category
description
display_name

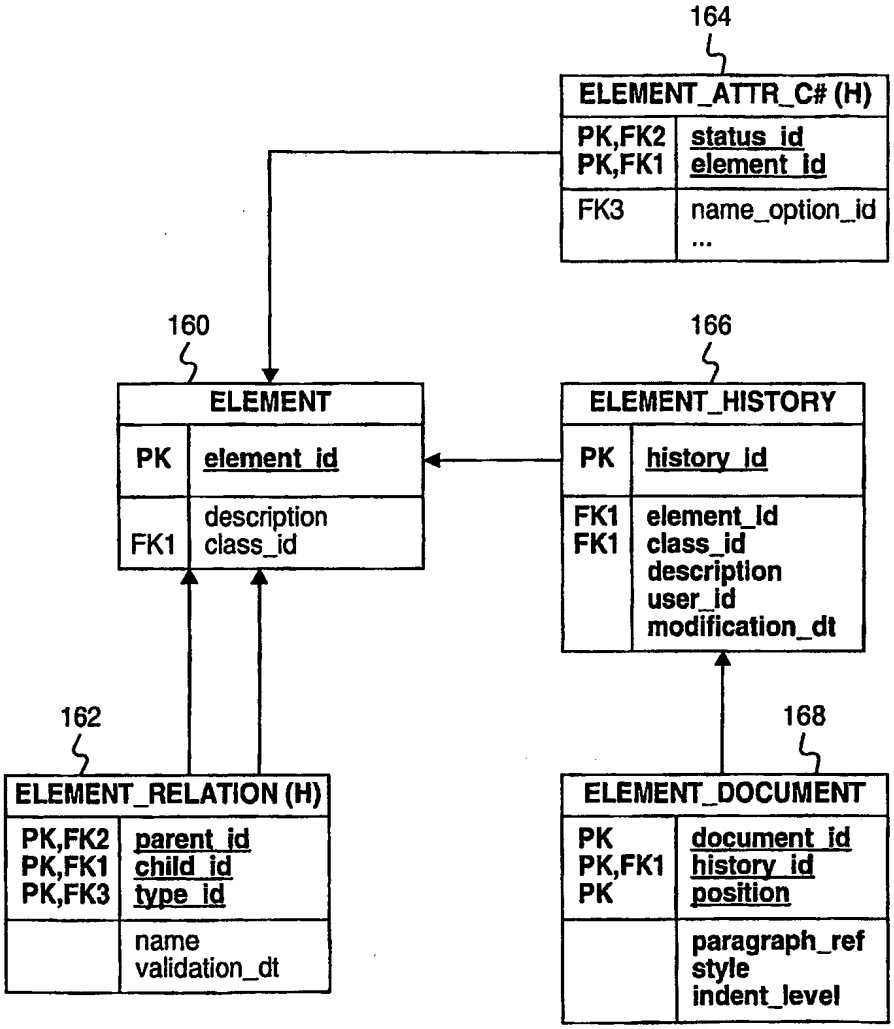

164
ELEMENT_ATTR_C# (H)
PK,FK2 status_id
PK,FK1 element_id
FK3 name_option_id
...
160
ELEMENT
PK element_id
FK1 description
class_id
166
ELEMENT_HISTORY
PK history_id
FK1 element_id
FK1 class_id
description
user_id
modification_dt
162
ELEMENT_RELATION (H)
PK,FK2 parent_id
PK,FK1 child_id
PK,FK3 type_id
name
validation_dt
168
ELEMENT_DOCUMENT
PK document_id
PK,FK1 history_id
PK position
paragraph_ref
style
indent_level

Figure 3

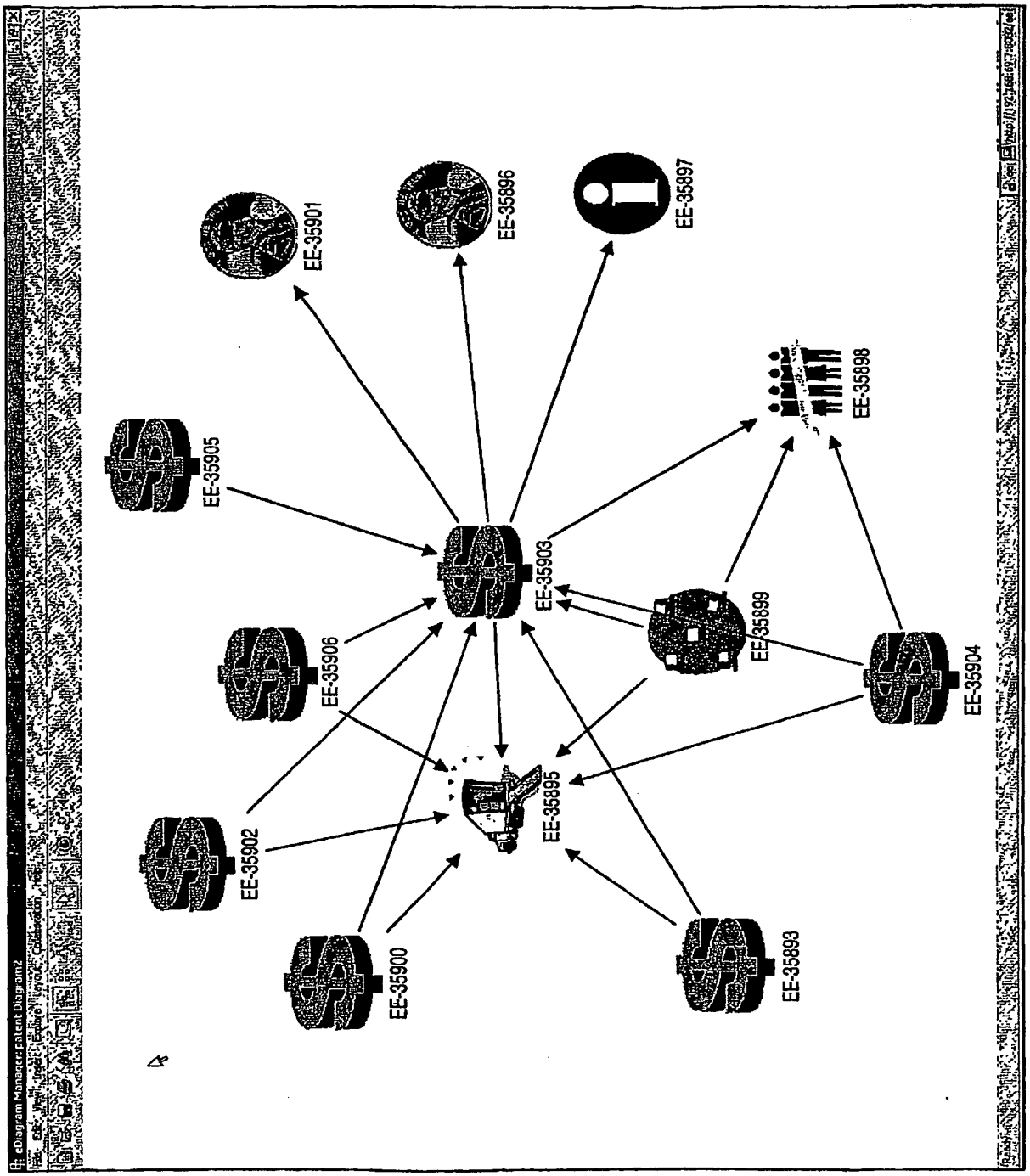

EE-35901
EE-35896
EE-35897
EE-35898
EE-35905
EE-35903
EE-35899
EE-35906
EE-35904
EE-35895
EE-35902
EE-35900
EE-35893

Figure 4

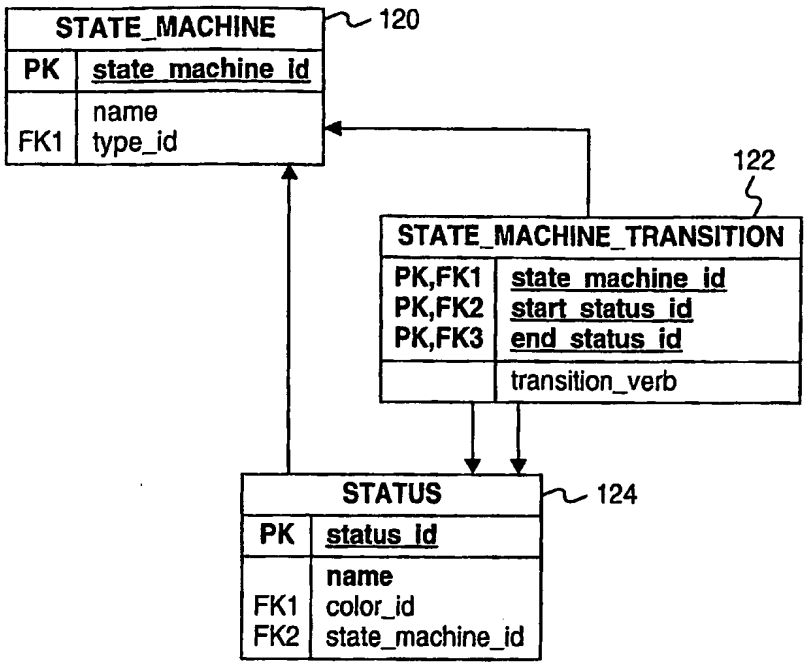

STATE_MACHINE
120
PK
state_machine_id
name
FK1
type_id
122
STATE_MACHINE_TRANSITION
PK,FK1
PK,FK2
PK,FK3
state_machine_id
start_status_id
end_status_id
transition_verb
STATUS
124
PK
status_id
name
FK1
color_id
FK2
state_machine_id

Figure 5

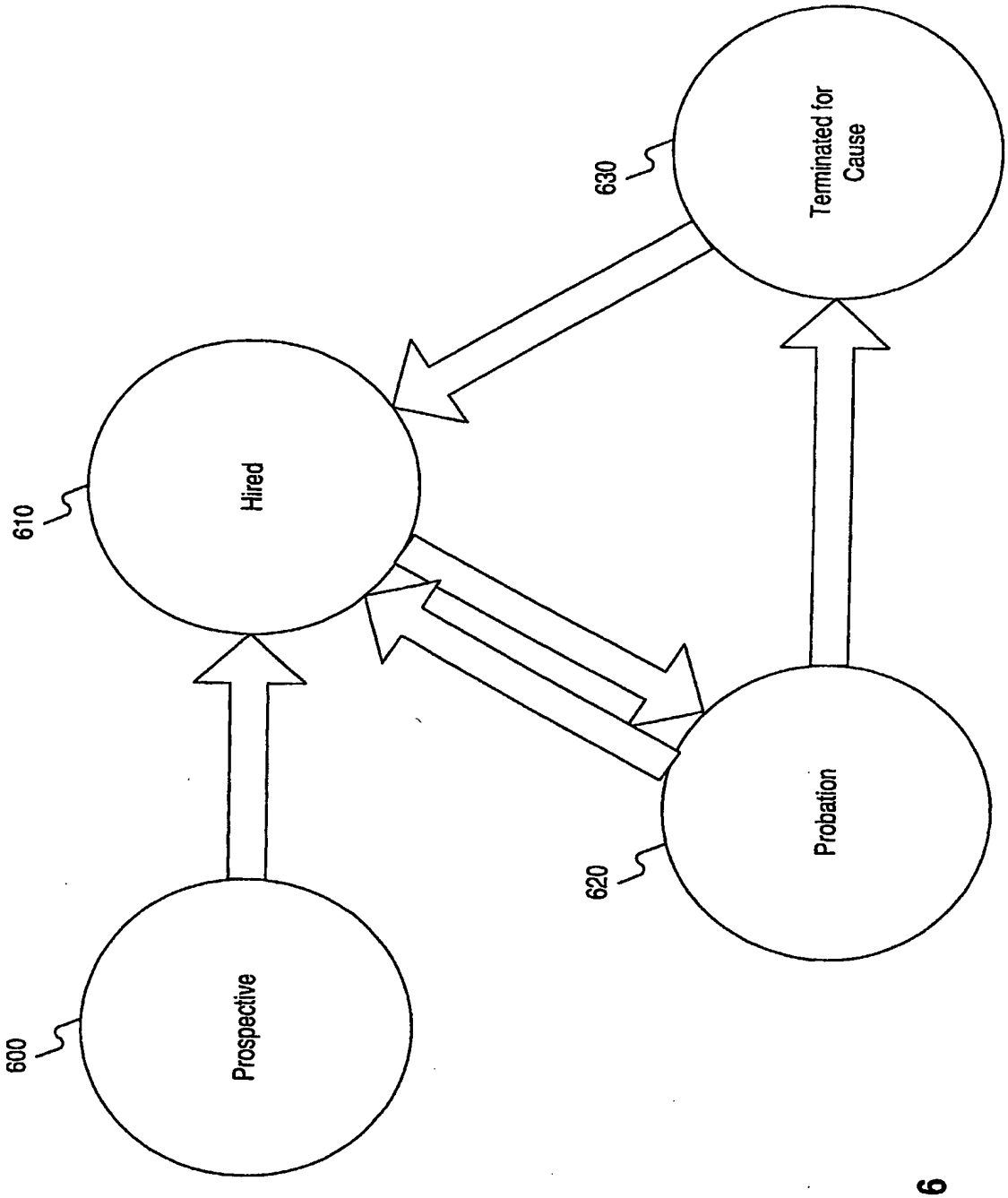


Figure 6

ServiceNow Ex. 1002 Page 318 of 410

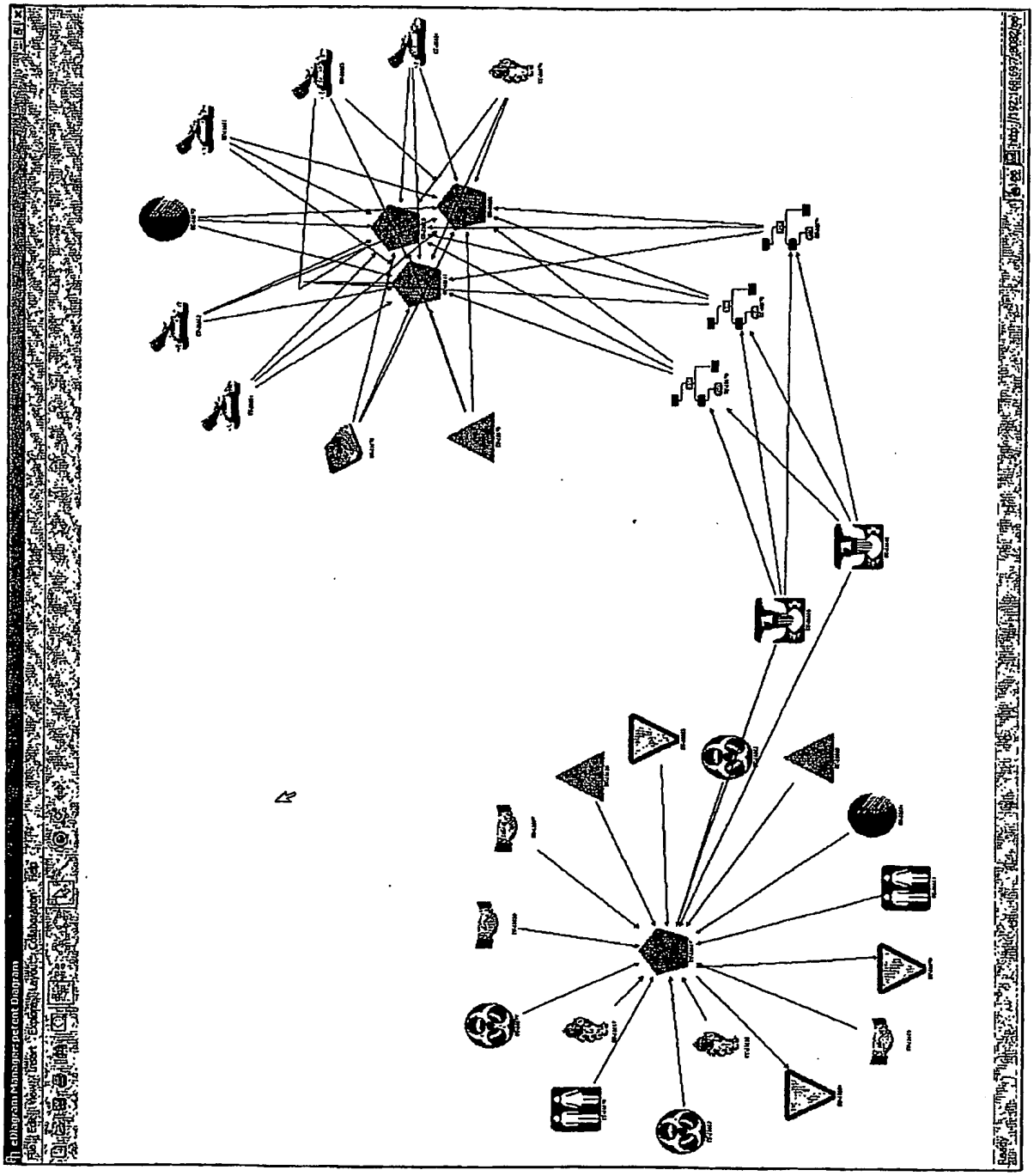


Figure 7

ServiceNow Ex. 1002 Page 319 of 410

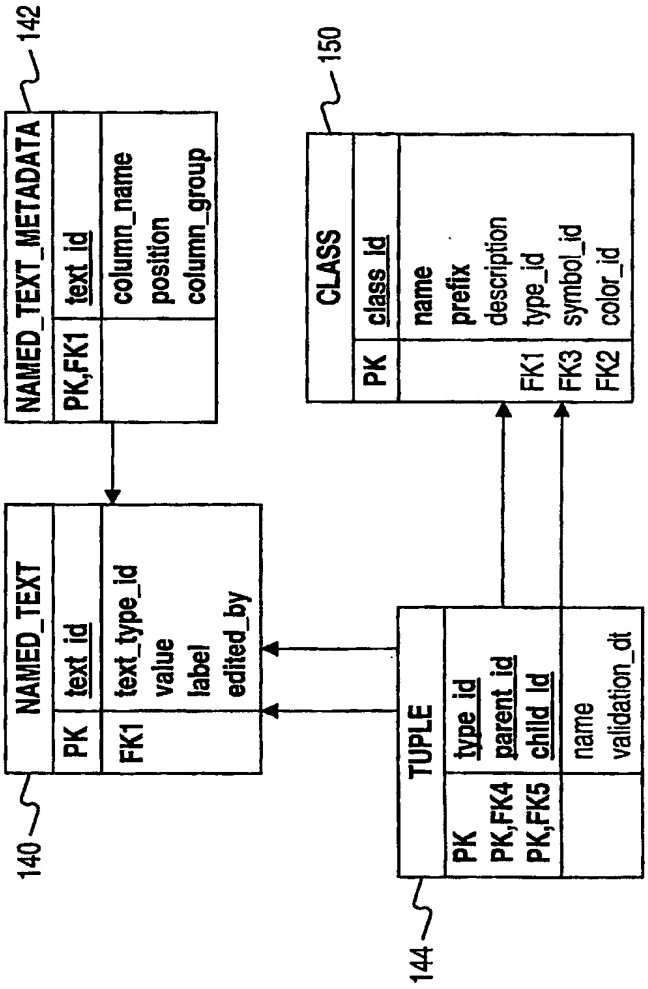


Figure 8

ServiceNow Ex. 1002 Page 320 of 410

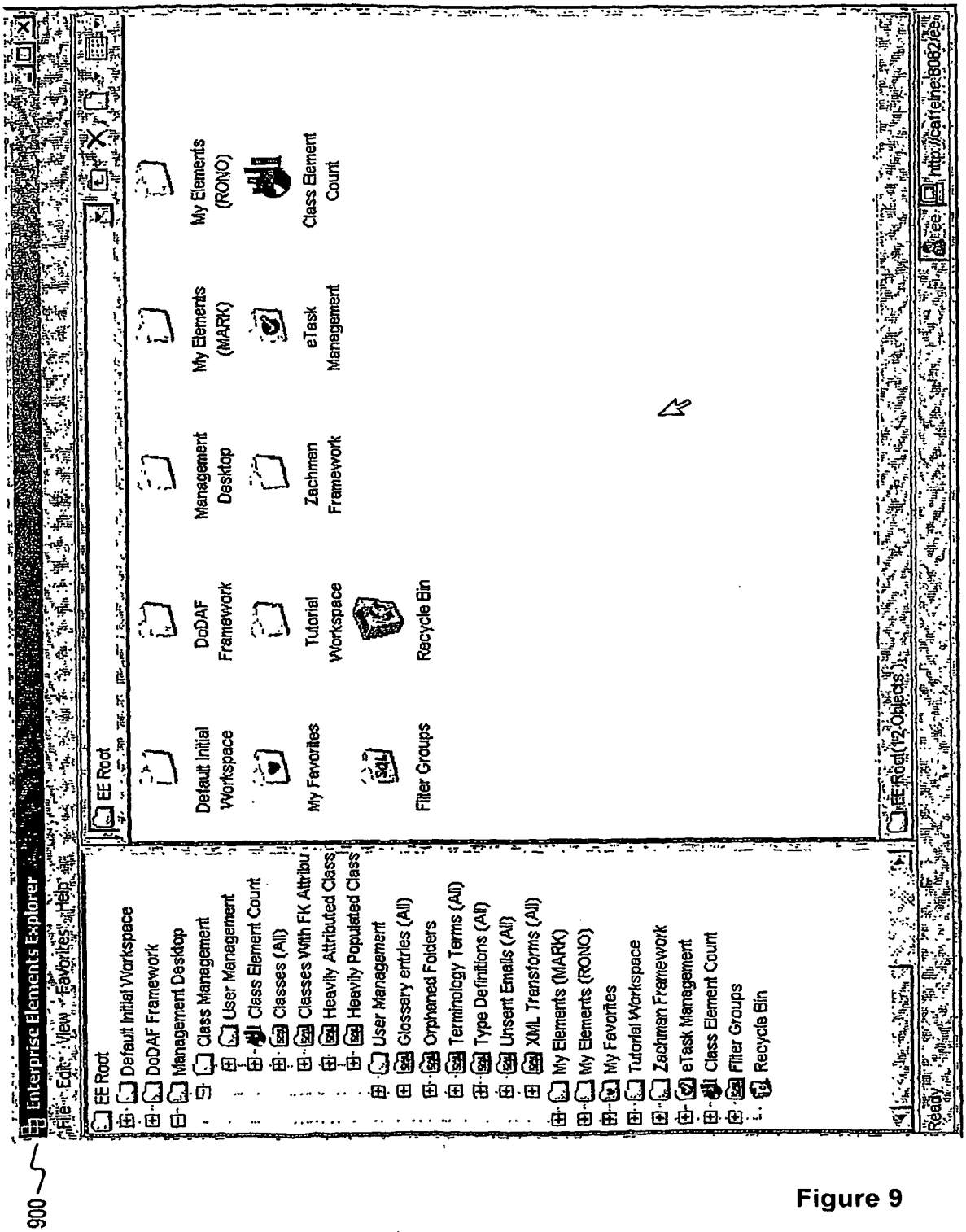

Enterprise Elements Explorer
File Edit View Favorites Help
EE Root
Default Initial Workspace
DoDAF Framework
Management Desktop
Class Management
User Management
Class Element Count
Classes (All)
Classes With FK Attribu
Heavily Attributed Class
Heavily Populated Class
User Management
Glossary entries (All)
Orphaned Folders
Terminology Terms (All)
Type Definitions (All)
Unsent Emails (All)
XML Transforms (All)
My Elements (MARK)
My Elements (RONO)
My Favorites
Tutorial Workspace
Zachman Framework
eTask Management
Class Element Count
Filter Groups
Recycle Bin
EE Root
Default Initial Workspace
DoDAF Framework
Management Desktop
My Elements (MARK)
My Elements (RONO)
My Favorites
Tutorial Workspace
Zachman Framework
eTask Management
Class Element Count
Filter Groups
Recycle Bin
EE Root (12 Objects)
Ready
http://caffeine:8082/ee
900

Figure 9

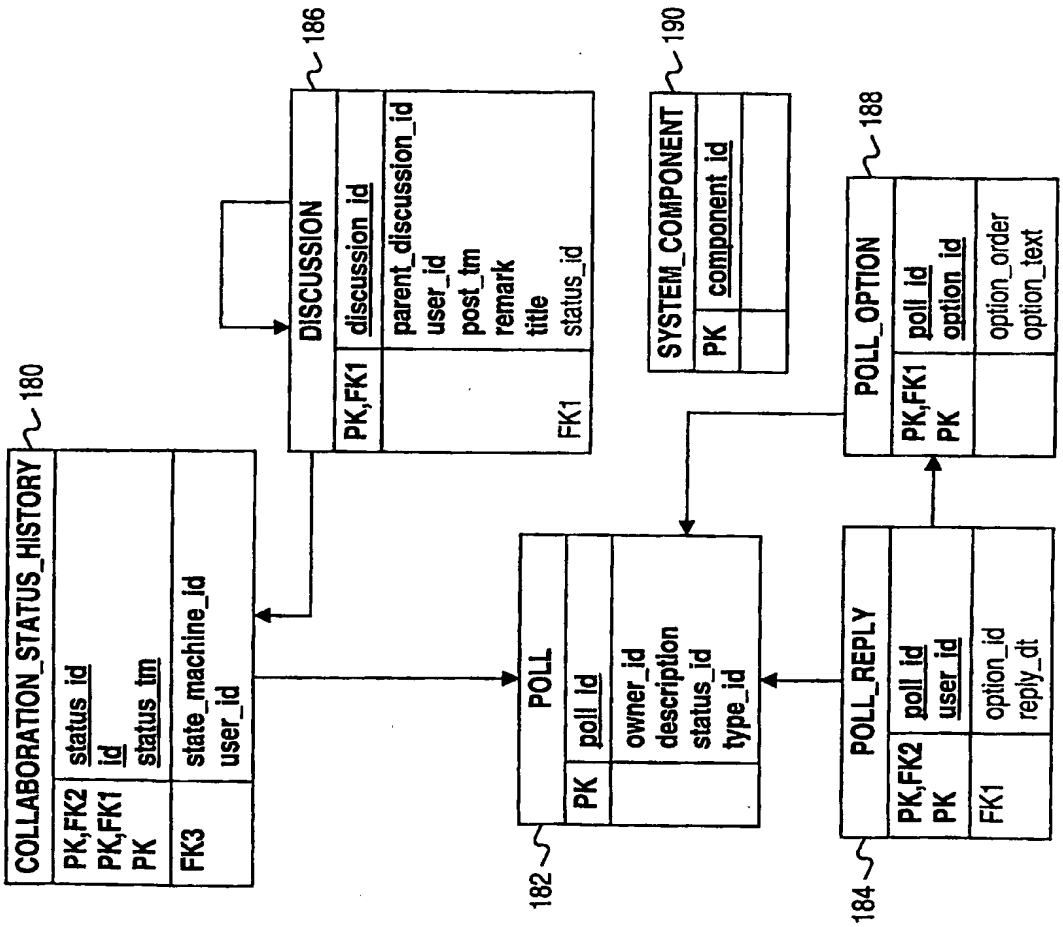

COLLABORATION_STATUS_HISTORY
180
PK,FK2 status_id
PK,FK1 id
PK status_tm
FK3 state_machine_id
user_id
DISCUSSION
186
PK,FK1 discussion_id
parent_discussion_id
user_id
post_tm
remark
title
FK1 status_id
SYSTEM_COMPONENT
190
PK component_id
POLL_OPTION
188
PK,FK1 poll_id
PK option_id
option_order
option_text
POLL
182
PK poll_id
owner_id
description
status_id
type_id
POLL_REPLY
184
PK,FK2 poll_id
PK user_id
FK1 option_id
reply_dt

Figure 10

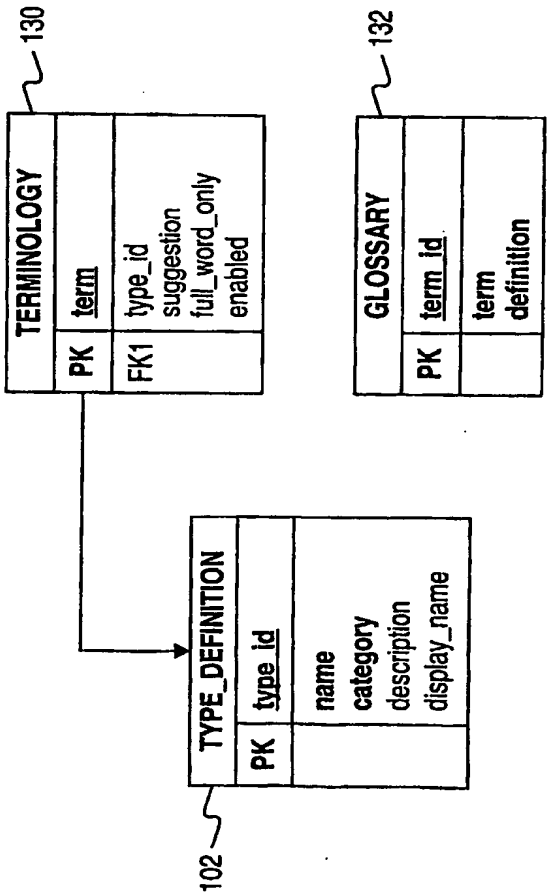

130
TERMINOLOGY
PK
term
FK1
type_id
suggestion
full_word_only
enabled
132
GLOSSARY
PK
term_id
term
definition
TYPE_DEFINITION
PK
type_id
name
category
description
display_name
102

Figure 11

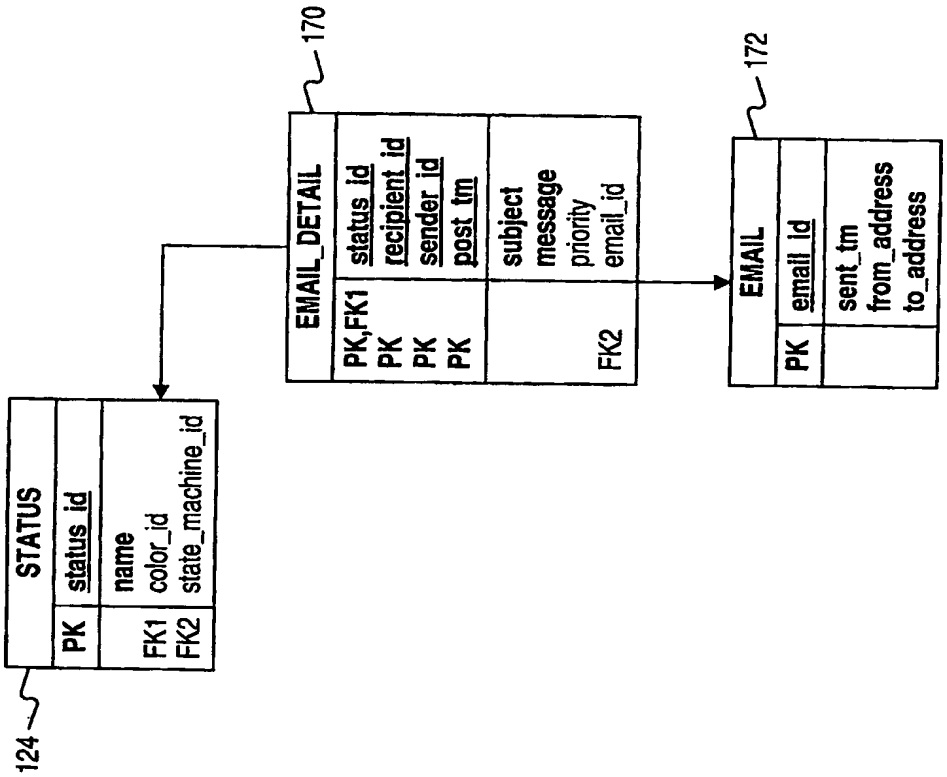

170
172
EMAIL_DETAIL
status_id
recipient_id
sender_id
post_tm
subject
message
priority
email_id
PK,FK1
PK
PK
PK
FK2
EMAIL
email_id
sent_tm
from_address
to_address
PK
STATUS
status_id
name
color_id
state_machine_id
PK
FK1
FK2
124

Figure 12

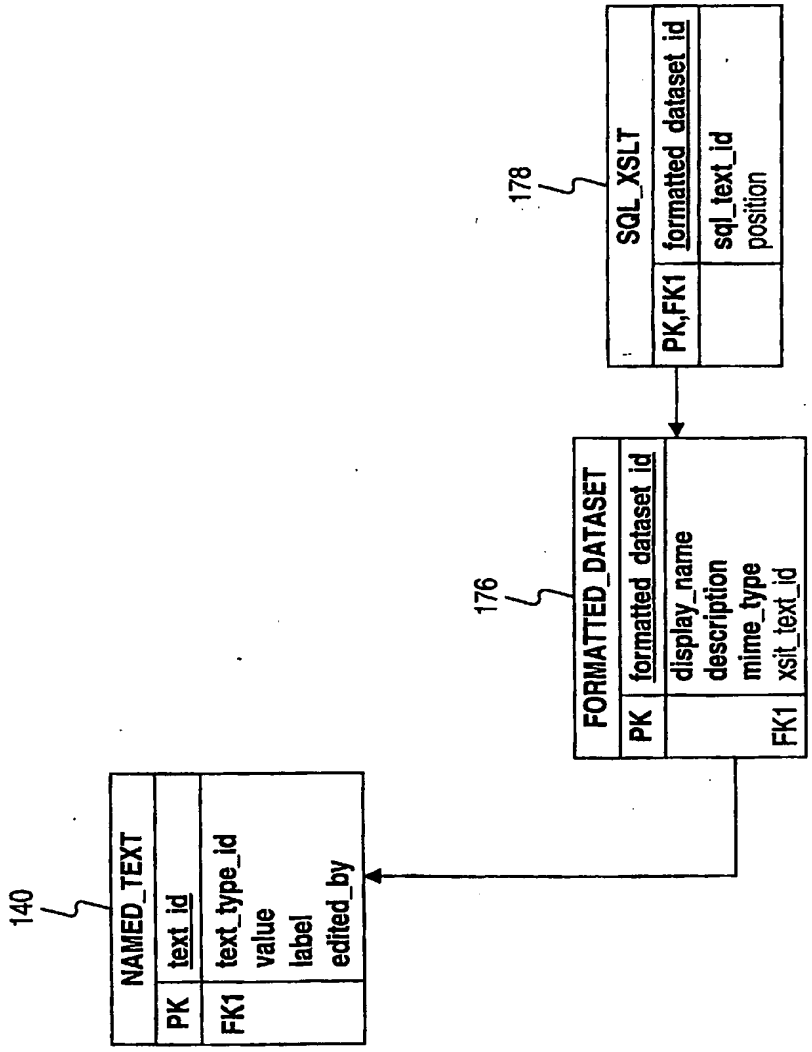

140
NAMED_TEXT
PK
text_id
FK1
text_type_id
value
label
edited_by
176
FORMATTED_DATASET
PK
formatted_dataset_id
display_name
description
mime_type
FK1
xsit_text_id
178
SQL_XSLT
PK,FK1
formatted_dataset_id
sql_text_id
position

Figure 13

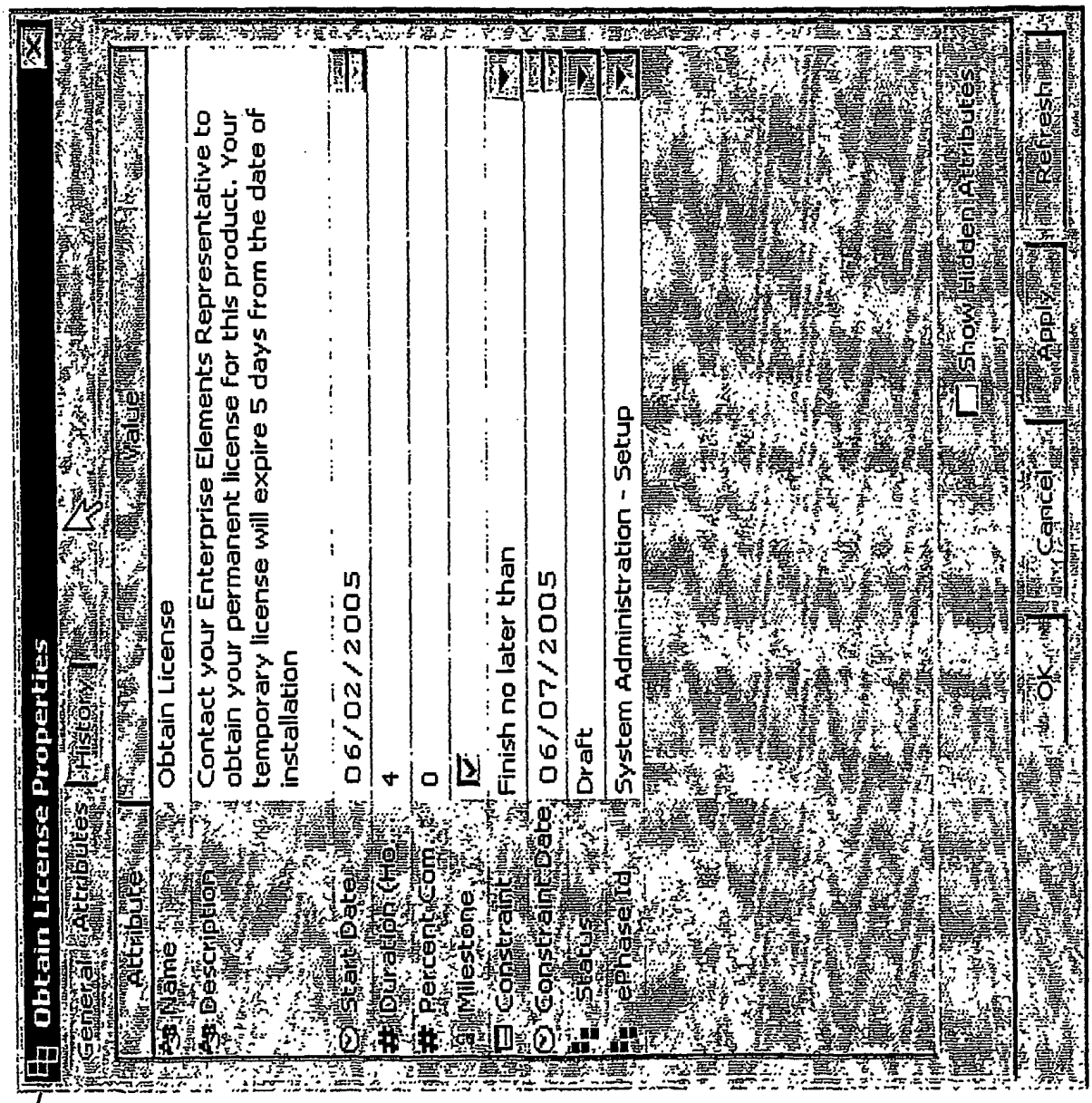

Obtain License Properties
General
Attributes
History
Attribute
Value
Name
Obtain License
Description
Contact your Enterprise Elements Representative to obtain your permanent license for this product. Your temporary license will expire 5 days from the date of installation
Start Date
06/02/2005
Duration (Ho
4
Percent Com
0
Milestone
Constraint
Finish no later than
Constrain Date
06/07/2005
Status
Draft
phaseId
System Administration - Setup
Show Hidden Attributes
OK
Cancel
Apply
Refresh
1400

Figure 14

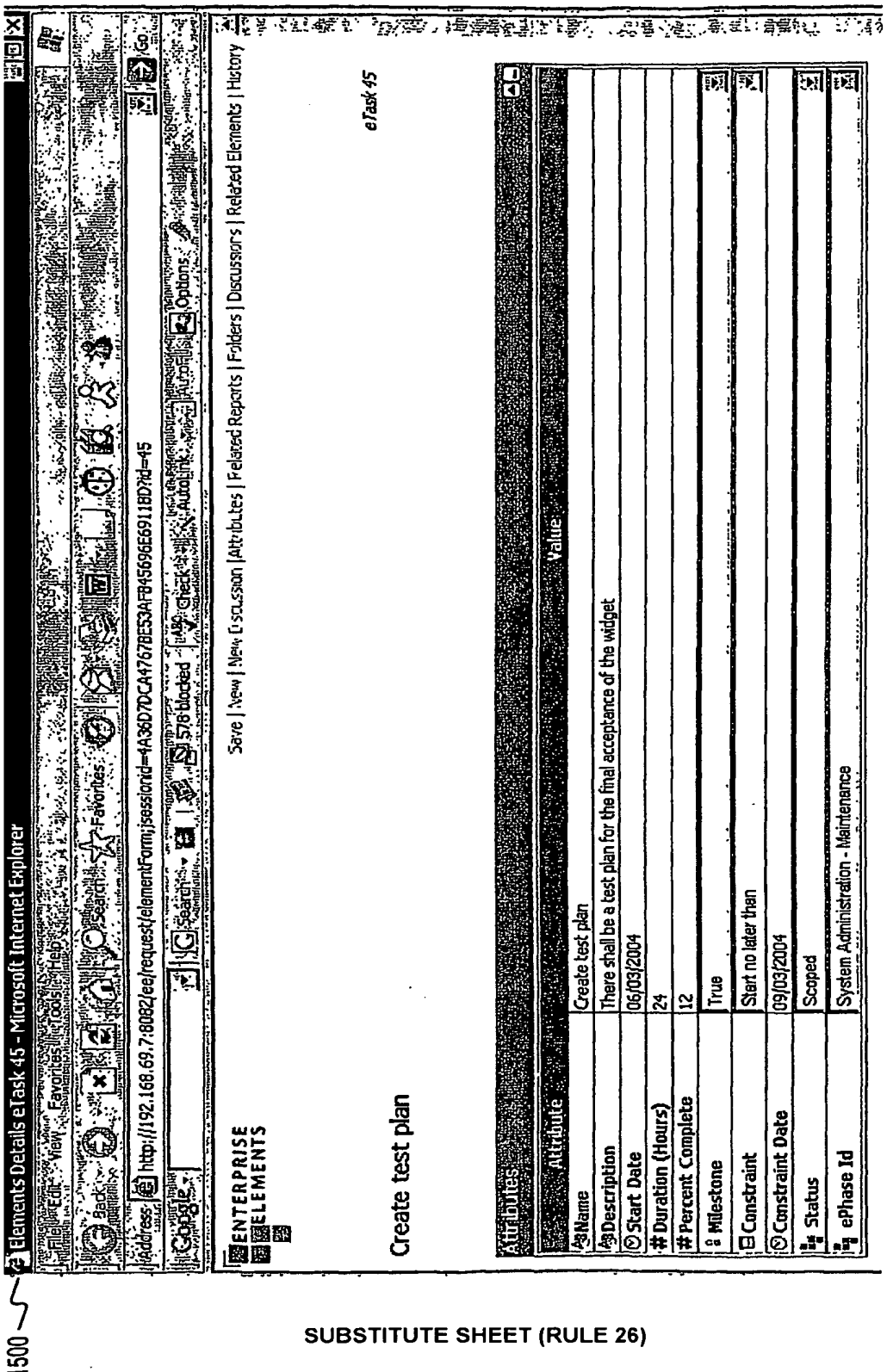

Elements Details eTask 45 - Microsoft Internet Explorer
ENTERPRISE ELEMENTS
Save | New | New Discussion | Attributes | Related Reports | Folders | Discussions | Related Elements | History
eTask 45
Create test plan
Attributes
Attribute
Value
Name
Create test plan
Description
There shall be a test plan for the final acceptance of the widget
Start Date
06/03/2004
Duration (Hours)
24
Percent Complete
12
Milestone
True
Constraint
Start no later than
Constraint Date
09/03/2004
Status
Scoped
ePhase Id
System Administration - Maintenance
1500

Figure 15

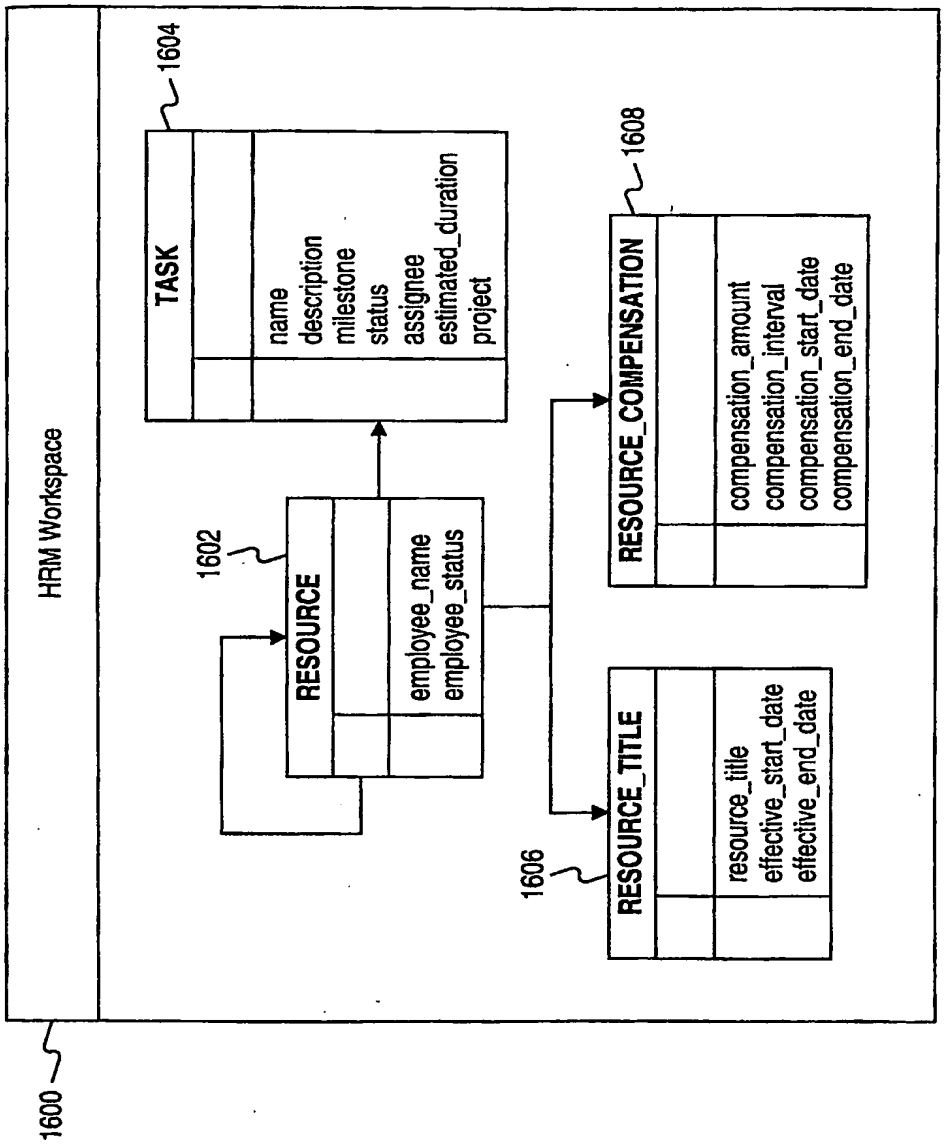

HRM Workspace
1600
1602
RESOURCE
employee_name
employee_status
1604
TASK
name
description
milestone
status
assignee
estimated_duration
project
1606
RESOURCE_TITLE
resource_title
effective_start_date
effective_end_date
1608
RESOURCE_COMPENSATION
compensation_amount
compensation_interval
compensation_start_date
compensation_end_date

Figure 16

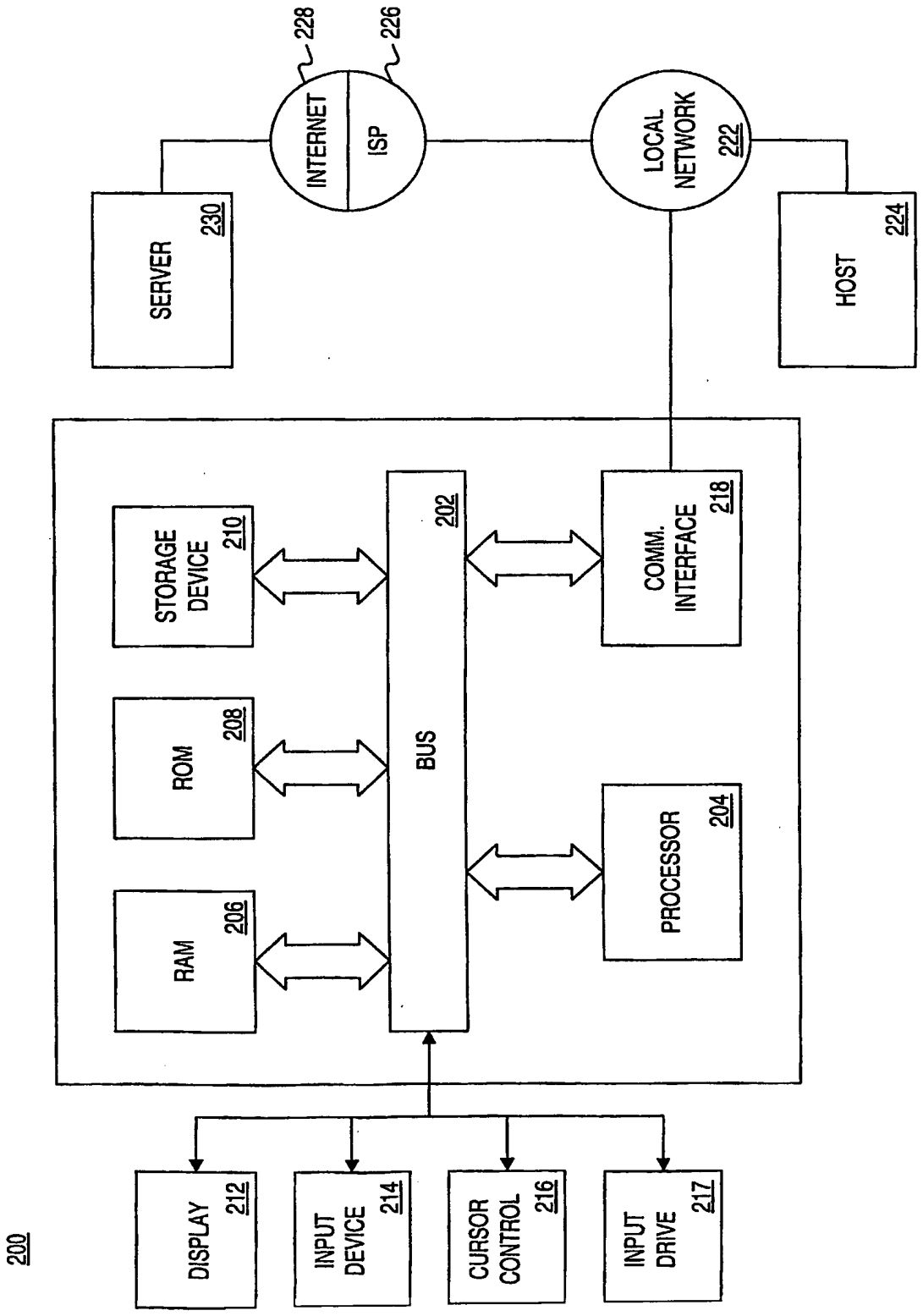

228
226
INTERNET
ISP
LOCAL NETWORK
222
SERVER
230
HOST
224
STORAGE DEVICE
210
202
COMM. INTERFACE
218
ROM
208
BUS
PROCESSOR
204
RAM
206
200
DISPLAY
212
INPUT DEVICE
214
CURSOR CONTROL
216
INPUT DRIVE
217

Figure 17

Customer No. 22,852
Attorney Docket No. 09801.0002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	
Bruce Randall DEWOOLFSON	)	
	)	Group Art Unit:
Application No.: Not Yet Assigned	)	
	)	Examiner:
Filed: December 13, 2007	)	
	)	
For: DATABASE DATA DICTIONARY	)	

U.S. NATIONAL PHASE OF INTERNATIONAL APPLICATION NO. PCT/US2006/023242

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

Sir:

INFORMATION DISCLOSURE STATEMENT UNDER 37 C.F.R. § 1.97(b)

Pursuant to 37 C.F.R. §§1.56 and 1.97(b), applicant bring to the Examiner's attention the documents listed on attached Form PTO/SB/08 and cited in the international search report. Copies of the listed non-patent literature documents are attached. Applicant respectfully requests that the Examiner consider the documents listed on attached Form PTO/SB/08 and indicate that they were considered by making an appropriate notation on this form.

This Information Disclosure Statement is being filed with the above-referenced application.

This submission does not represent that a search has been made or that no better art exists and does not constitute an admission that each or all of the listed

documents are material or constitute "prior art." If the Examiner applies any of the documents as prior art against any claim in the application and applicant determines that the cited documents do not constitute "prior art" under United States law, applicant reserves the right to present to the Office the relevant facts and law regarding the appropriate status of such documents. Applicant further reserves the right to take appropriate action to establish the patentability of the disclosed invention over the listed documents, should one or more of the documents be applied against the claims of the present application.

If there is any fee due in connection with the filing of this Statement, please charge the fee to our Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: December 13, 2007

By: _____
Ernest F. Chapman
Reg. No. 25,961

Enclosures
EFC/FPD/rac

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A SUBMISSION UNDER 35 U.S.C. 371		CUSTOMER NO. 22,852 ATTORNEY'S DOCKET NUMBER: 09801.0002 U.S. APPLICATION NO. if known, see 37 CFR 1.5(
INTERNATIONAL APPLICATION NO. PCT/US2006/023242	INTERNATIONAL FILING DATE June 14, 2006	PRIORITY DATE CLAIMED June 14, 2005
TITLE OF INVENTION DATABASE DATA DICTIONARY		
APPLICANT(S) FOR DO/EO/US Bruce Randall DEWOOLFSON		

Applicant(s) herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a submission under 35 U.S.C. 371.
3. ☐ This is an express request to begin national examination procedures (35 U.S.C. 371(f)). The submission must include items (5), (6), (9) and (21) indicated below.
4. ☐ The US has been elected (Article 31).
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c)(2)).
   a. ☒ is attached hereto (required only if not communicated by the International Bureau).
   b. ☒ has been communicated by the International Bureau.
   c. ☐ is not required, as the application was filed with the United States Receiving Office (RO/US).
6. ☐ An English language translation of the International Application as filed (35 U.S.C. 371(c)(2)).
   a. ☐ is attached hereto.
   b. ☐ has been previously submitted under 35 U.S.C. 154(d)(4).
7. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3)).
   a. ☐ are attached hereto (required only if not communicated by the International Bureau).
   b. ☐ have been communicated by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.
8. ☐ An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371 (c)(3)).
9. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
10. ☐ An English language translation of the annexes of the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

Items 11 to 20 below concern document(s) or information included:

11. ☒ Information Disclosure Statement under 37 CFR 1.97 and 1.98
12. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. ☐ A preliminary amendment.
14. ☐ An Application Data Sheet under 37 CFR 1.76.
15. ☐ A Substitute specification.
16. ☐ A power of attorney and/or change of address letter.
17. ☐ A computer-readable form of the sequence listing in accordance with PCT Rule 13*ter*.2 and 35 U.S.C. 1.821-1.825.
18. ☐ A second copy of the published International Application under 35 U.S.C. 154 (d)(4).
19. ☐ A second copy of the English language translation of the international application 35 U.S.C. 154 (d)(4).
20. ☒ Other items or information:
    a. ☒ Copy of cover page of International Publication No. WO 2006/138420 A1
    b. ☐ Copy of Notification of Missing Requirements.
    c. ☒ International Search Report from the European Patent Office
    d. ☐
    e. ☐
    f. ☐
    g. ☐

U.S. APPLICATION NO. (if known, see 37 CFR 1.5)	INTERNATIONAL APPLICATION NO. PCT/US2006/023242	ATTORNEY'S DOCKET NUMBER ☐ 09801.0002	
The following fees have been submitted:		CALCULATIONS	PTO Use Only
21. ☐ BASIC NATIONAL FEE (37 CFR 1.492(a)): **$310.00**		$310.00	
22. ☐ Examination fee (37CFR 1.492(c)) If the Written opinion prepared by ISA/US or the International preliminary examination report prepared by IPEA/US indicates all claims satisfy provisions of PCT Article 33(1)-(4) ..... **$0** All other situations..... **$210**		$210.00	
23. ☐ Search fee (37 CFR 1.492(b)) If the written opinion of the ISA/US or the International preliminary examination report prepared by IPEA/US indicates all claims satisfy provisions of PCT Article 33(1)-(4) ..... **$0** Search fee (37 CFR 1.445(a)(2)) has been paid on the international application to the USPTO as an International Searching Authority..... **$100** International Search Report prepared by an ISA other than the US and provided to the Office or previously communicated to the US by the IB..... **$410** All other situations..... **$510**		$410.00	
TOTAL of 21, 22 and 23 =		$930.00	
☐ Additional fee for specification and drawings filed in paper over 100 sheets (excluding sequence listing in compliance with 37 CFR 1.821(c) or (e) or computer program listing in an electronic medium) (37 CFR 1.492(j)). The fee is **$250** for each additional 50 sheets of paper or fraction thereof.			
Total Sheets: 63 - 100 =	Extra Sheets: /50=	Number of each additional 50 or fraction thereof (round **up** to a whole number)	Rate: x $260.00
			$
Surcharge of **$130.00** for furnishing any of the search fee, examination fee, or the oath or declaration after the date of commencement of the national stage (37 CFR 1.492(h)).			$

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE		
Total Claims	39 -20 =	19	x $50.00	$950.00	
Independent Claims	-3 =		x $210.00	$	
MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+$370.00	$	
			TOTAL OF ABOVE CALCULATIONS =	$1880.00	
☐ Applicant claims small entity status. See 37 CFR 1.27. Fees above are reduced by ½.				$	
			SUBTOTAL =	$1880.00	
Processing fee of **$130.00** for furnishing the English translation later than 30 months from the earliest priority date (37 CFR 1.492(i)). +				$	
			TOTAL NATIONAL FEE =	$1880.00	
Fee for recording the enclosed assignment (37 CFR 1.21 (h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). **$40.00** per property. +				$	
			TOTAL FEES ENCLOSED =	$1880.00	
				Amount to be refunded:	$
				charged:	$

a. ☒ This application is being filed electronically. The fees are being paid by credit card. The filing fees of $1880.00 are included.

b. ☐ Please charge my Deposit Account No. _____ in the amount of $ _____ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 06-0916 . A duplicate copy of this sheet is enclosed.

d. ☐ Fees are to be charged to a credit card or to Deposit Account No. 06-0916. **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.

NOTE: Where an appropriate time limit under 37 CFR 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the International Application to pending status.

SEND ALL CORRESPONDENCE TO:

Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.
901 New York Avenue, N.W.
Washington, DC 20001-4413

SIGNATURE

Ernest F. Chapman/Reg. No. 25,961
NAME/REGISTRATION NUMBER

DATED: December 13, 2007

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

CORRECTED VERSION

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 December 2006 (28.12.2006)

PCT

(10) International Publication Number
WO 2006/138420 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/US2006/023242

(22) International Filing Date: 14 June 2006 (14.06.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/690,147 14 June 2005 (14.06.2005) US

(71) Applicant *(for all designated States except US)*: **ENTERPRISE ELEMENTS, INC.** [US/US]; 11654 Plaza America Drive, Suite 639, Reston, VA 20190-4700 (US).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **DEWOOLFSON, Bruce, Randall** [US/US]; 46290 Summerhill Place, Potomac Falls, VA 20165 (US).

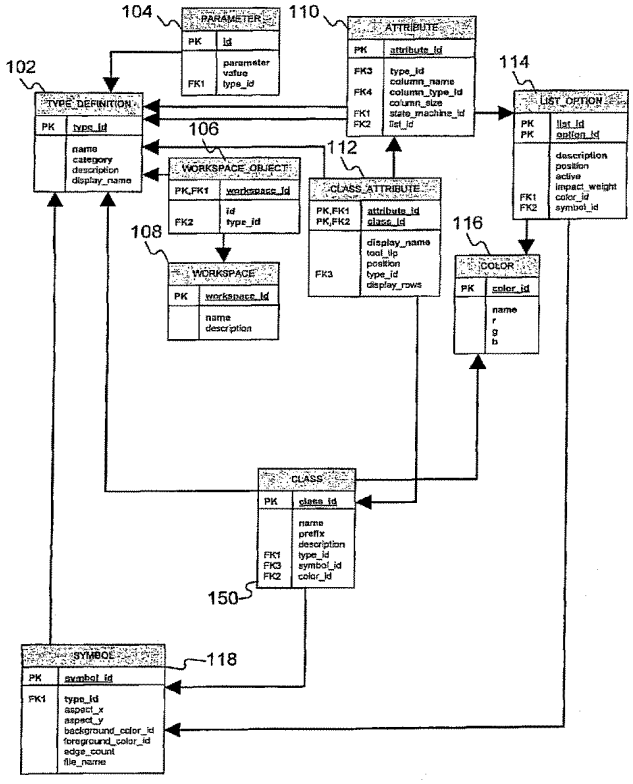
(74) Agents: **GARRETT, Arthur, S.** et al.; Finnegan, Henderson, Farabow, Garrett & Dunner, L.l.p., 901 New York Avenue, Nw, Washington, DC 20001-4413 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,

[Continued on next page]

(54) Title: DATABASE DATA DICTIONARY



(57) Abstract: Systems and methods are provided for manipulating data sets. In accordance with one implementation, a computerized system is provided for storing, managing, indexing, interrelating, and/or retrieving data sets in a manner independent of the data model. The system includes an element module configured to store and uniquely identify elements and an element relation module configured to store relationships between the elements in the element module. The computerized system may also comprise a class module configured to store attributes of elements in a class and a type definition module configured to define the class and the attributes related to the class. The computerized system may further comprise a state machine module, the state machine module including a state machine transition module and a status module.

WO 2006/138420 A1

RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

(48) Date of publication of this corrected version:
29 November 2007

(15) Information about Correction:
see PCT Gazette No. 48/2007 of 29 November 2007

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

DATABASE DATA DICTIONARY

DESCRIPTION

Cross-Reference to Related Application(s)

[001] This application claims the benefit of priority from U.S. Provisional Application No. 60/690,147, entitled "Database Data Dictionary," filed June 14, 2005, the disclosure of which is expressly incorporated herein by reference to its entirety.

Technical Field

[002] This invention relates to the field of computer data storage. More particularly, the present invention relates to methods and systems for storing, managing, indexing, interrelating, and/or retrieving data sets.

Background Information

[003] Currently, a conventional data storage system implements its own data model according to the user interface and the business rules specification for that system. For the conventional system, the system's developers write dedicated code and user interface code that displays or manages certain data inputs such as check boxes, lists, combo boxes or other data inputs. Such systems depend upon a specific data model and lack flexibility and portability. The conventional system's dedicated code and user interface in some respects manipulate the data that the conventional system's application stores and retrieves during operation. Moreover, the conventional system stores and retrieves data locally on a local disk, such as a C drive on a Windows system. It is well known that conventional systems may also store data on a remote storage media, e.g. a file server or database.

[004] There are efforts to overcome the lack of flexibility and portability of conventional data storage systems, such as the adoption and utilization of extensible markup language for storing data (e.g., Microsoft Office® XML Open Format, OASIS Open Office XML Format, etc). Data files in open data formats (e.g., XML) facilitate data transfer and are typically stored on a desktop machine. XML data files rely on the structures defined by XML standards, which allows standardized XML parsers to read data files expressed in XML format.

[005] However, open data formats based on XML and other open standards do not interrelate and manage dynamic data sets through process and collaboration and state machines. Open data formats (e.g., XML) also do not provide for inheritance and override behaviors, collaboration or state machine process enforcement, and/or element style normalization.

[006] Accordingly, there is a need for systems and methods that store, manage, index, interrelate, and/or retrieve data sets in a manner independent of the data model.

SUMMARY

[007] In accordance with the principles of the present invention, as embodied and broadly described herein, methods, systems and computer program products are provided for manipulating data sets. The data set manipulation may be achieved through utilization of a processor and a data repository adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets.

[008] According to one embodiment, a system is provided for manipulating data sets. The system comprises a processor and a repository, the data repository including

ServiceNow Ex. 1002 Page 337 of 410

an element module configured to store and uniquely identify a plurality of elements each identified by a unique element identification, an element relation module configured to store one or more relationships between the elements in the element module, and a class module configured to define at least one class of the elements and store the class. The data repository further includes an attribute module configured to define and store the attributes, a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class, and a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements.

[009] According to another embodiment, a method is provided for manipulating data sets. The method comprises storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, and storing one or more relationships between the elements in the data repository. The method further comprises defining at least one class of the elements and storing the class in the data repository, defining one or more attributes and storing the attributes in the data repository, defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository, and defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

ServiceNow Ex. 1002 Page 338 of 410

[010] Another exemplary embodiment provides for a computer-readable medium containing instructions to configure a processor to perform a method for manipulating data sets. The method comprises storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, and storing one or more relationships between the elements in the data repository. The method further comprises defining at least one class of the elements and storing the class in the data repository, defining one or more attributes and storing the attributes in the data repository, defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository, and defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

[011] Additional aspects of the invention are set forth in the detailed description which follows or may be learned by practice of methods, systems, and articles of manufacture consistent with the present invention. It is understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[012] Both the foregoing general description and the following detailed description are exemplary and explanatory only. They do not restrict the invention, as claimed. Furthermore, the accompanying drawings, which are incorporated in and

constitute a part of this specification, illustrate embodiments of the invention and together with the description, serve to explain the principles of the invention. In the drawings:

[013] Figure 1 is an exemplary entity-relationship diagram of an elements metadata repository (EMR) for storing, managing, indexing, interrelating, and/or retrieving data sets, consistent with the principles of the present invention;

[014] Figure 2 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing articulated types, workspaces and workspace objects, class and class attributes, parameters, list options, etc;

[015] Figure 3 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing elements, element relations, element attributes, element history, element documents, etc;

[016] Figure 4 illustrates an exemplary diagrammatic visualization for visually selecting and relating elements using a graphical user interface (GUI);

[017] Figure 5 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing state machines, state machine transitions, state machine statuses, etc;

[018] Figure 6 illustrates an exemplary state machine diagram;

[019] Figure 7 illustrates a GUI of an exemplary diagrammatic depiction displaying congregations of requirements that are related to subcomponents of a system;

ServiceNow Ex. 1002 Page 340 of 410

[020] Figure 8 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing tuples, named texts and metadata for named texts, etc;

[021] Figure 9 illustrates a GUI of an exemplary elements explorer;

[022] Figure 10 shows a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing collaboration status history, polls, discussions, system components, etc;

[023] Figure 11 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing terminology, glossary, etc;

[024] Figure 12 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing emails, email details, etc;

[025] Figure 13 depicts a subset of the exemplary entity-relationship diagram of the EMR, illustrating data tables for storing formatted dataset, SQL extensible stylesheet language transformations, etc;

[026] Figures 14 and 15 depict GUIs of exemplary editing environments that may be invoked from a displayed element;

[027] Figure 16 is a diagram depicting an exemplary Human Resource Management workspace; and

[028] Figure 17 is a diagram depicting an exemplary computer system in which systems and methods consistent with the principles of the elements metadata repository of Figure 1 may be implemented.

ServiceNow Ex. 1002 Page 341 of 410

DETAILED DESCRIPTION

[029] Reference will now be made in detail to present embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

Entity-Relationship Configuration

[030] FIG. 1 shows an entity relationship diagram 100 of an elements metadata repository (EMR) for storing, managing, indexing, interrelating, and/or retrieving data sets, consistent with principles of the present invention. In accordance with embodiments of the invention, a repository, such as the EMR, is a central place where data and metadata is stored and maintained. Metadata is data that is used to describe other data. Examples of metadata include schema, tables, indices, views, column definitions, etc. Metadata is recursive such that data about data about the data may be stored. This recursive nature may be expressed as levels, such as "M" number of levels. For example, the actual data being stored is M0. Data about that actual data (i.e., M0) may be referred to as M1. Data about the M1 layer data may be referred to as M2 and so on. In this example, the M1 and M2 layers are metadata, i.e., data about data. The EMR of FIG. 1 allows for storage of any number of data layers.

[031] Embodiments of the present invention, as shown in FIG. 2, include a class table 150, which may be analogous to a relational database table. Just as a table contains columns, a class, such as an entry in class table 150, contains attributes. In an exemplary implementation of the EMR, using database systems such as Oracle®, Microsoft SQL Server®, or MySQL, a user may create a class and its attributes by

creating a table and constructing columns within the table. In other words, the user creates a class and populates the class with attributes where attributes are conceptually other pieces of information of interest to the things that the class is trying to describe. For instance, if the class was a person, an attribute may be the person's name, birth date, or social security number. Any number of attributes can be assigned to any number of classes.

[032] Unlike a conventional relational database, the EMR extends the notion of available attributes types beyond the standard string, number, and date types. To do so, the EMR allows dynamically extendable entries in a type definition table 102. Type definition table 102 may store articulated types that may be, for example, an M4 layer while still maintaining the meta meta meta meta data language. By using the type of the person's name for an attribute, the user can declare more than what a normal relational database may store. For example, the EMR provides for a boolean type which is not normally found in a relational database. The EMR also provides for a pick list type, which stores a plurality of optional selections, and restricts actual data settings to none, or one and only one of the optional selections. The pick list type is also not found in a relational database. Moreover, the EMR is capable of storing a process or state machine type which is not found in a relational database. All these data types provide flexibility in the type of data management that can be performed.

[033] In accordance with still further embodiments of the invention, the EMR includes a parameter table 104. Parameter table 104 allows for any unforeseen attribute values to be placed upon system level components without redefining the base schema. For instance, if one type of named text is a SQL query, and the system

ServiceNow Ex. 1002 Page 343 of 410

requires security settings on that one type of named text (further described below), then a parameter entry can be used to specify that setting instead of redefining named_text table (e.g. named_text table 140 in FIG. 8) to include an additional column, which may or may not be needed for all other named text types.

[034] Embodiments of the present invention, as shown in FIG. 2, include a workspace table 108 and a workspace_object table 106. Workspace table 108 and workspace_object table 106 provide for logical grouping of elements into subsets that are usable for particular users or purposes. For example, if an EMR user is a tester, then a "Testing" workspace may be declared. Objects, including elements, classes, named texts, and "things" that is defined by a numeric identifier (which may be every "thing") may be placed into 'Testing' workspace via entries in workspace_object table 106. The tester would be assigned to this workspace (also through workspace_object table 106), and therefore only be permitted to see the other objects in the workspace. The effect is a basic level of security and data volume reduction to small localized easily navigable subsets.

[035] Consistent with embodiments of the invention, an attribute table 110 holds the definition of each attribute used in each class in the EMR. Attribute table 110 may store the physical name and type of the data which corresponds to the database column name and column type where the actual values are stored. A class_attribute table 112 may associate a given attribute to one or more classes (in class table 150). Because class_attribute table 112 allows for one attribute to be referenced by many classes, there is attribute inheritance among the classes. Attribute inheritance among the classes is a powerful notion not found in traditional relational database management

ServiceNow Ex. 1002 Page 344 of 410

systems (RDBMS). Class_attribute table 112 stores the logical name that may be used by the attribute when presented to the user for data manipulation such as insert or update or delete of values. The ability to reference the same attribute via multiple names depending on which class it appears in is akin to the concept of overridden inheritance, also a concept not found in RDBMS.

[036] In an embodiment of the invention, the EMR may immediately create a physical instantiation of the logical class and attribute model as soon as a class attribute definition is entered and committed into class_attribute table 112. The EMR may immediately represent the table and columns that the class and attribute definitions dictate. Moreover, the EMR's editors (see, for example, FIG. 14 and 15) may immediately conform to this definition and support full data manipulation of any values in any defined attribute. Furthermore, when certain types of attributes are attached to a class, appropriate named text entries are automatically created and tupled to the class and may be immediately available for use in the EMR. For example, if a pick list attribute type is added to a class, then queries that returns, for instance, all rows in the class and how many rows fall into each of the available pick list options, are automatically created, which allows the EMR to present the results (via charts or other presentation formats) to the users. This immediate construction of physical storage, complete editing capability, and automated reporting allows the EMR to become operational immediately upon logical schema definition.

[037] Embodiments of the present invention, as shown in FIG. 2, further include a list_option table 114. List_option table 114 may hold all possible enumerations in all possible pick list attributes, which allows for structured user interface and predictable

ServiceNow Ex. 1002 Page 345 of 410

query structures when displaying and managing pick list data. A color table 116 may hold a record for each enumerated color referenced by any object in the EMR. A color can be declared and associated to an object such as a class, list option selection, or element, etc, and when detected during presentation, the id reference value for the color can be replaced with a swatch of the actual color or any other run time decoration as appropriate.

[038] Consistent with embodiments of the invention, the EMR includes a symbol table 118. Symbol table 118 is similar to color table 116 because it may hold a list of references to graphical objects that can be used in the system. The number and type of graphical objects are not restricted, and can be extended by addition of type definition entries but typically include standard graphic files, such as .BMP, .GIF, .JPG, .SVG, and .SVGz. Once declared in this table, these graphical objects can be referenced by any object in the system, and when the reference is detected at display time, the reference may be replaced with a rendering of the actual image. For example, reports may be constructed that show check marks, colored balls, military insignia, or any other relevant image depending on the needs of the visualization.

[039] The EMR of FIG. 1 conceptually breaks the world into an articulation of "things." The EMR places these "things" into one pile in an element table 160. An element stored in element table 160 implies that the user can find similarity between any "thing" that the user would like to discuss, report, store, or manage. Element table 160 provides the user the capability to add those "things" into a congruous pile so that the user can deal with them without regard to their exact nature. This provides an over-normalization of any data captured by the EMR into one record type called an element.

ServiceNow Ex. 1002 Page 346 of 410

[040] FIG. 3 illustrates an exemplary element, which has a unique identifier (e.g., element ID), a name (e.g., description), and a classification (e.g., class identifier). Consistent typing of the element ID permits the user to refer to elements across the schema without having to convert the type of the element ID. In one embodiment, the element ID takes on a number format, e.g., a long value or an integer value. In this embodiment, any "thing" in the world may be uniquely identified by a number, e.g., an integer. Moreover, once any "thing" has been differentiated from nothingness, the "thing" has a name. This name may be called a description. In one embodiment, the key components of the element are the element ID and the name, as shown in element table 160. The description does not have to be unique. Some "things" may be described generally with the same name but have differences that render those things unique from each other. Another component of the element is a class identifier. The class identifier is a pointer or an indicator of the type or classification of this element in the universe. The class identifier allows the EMR to articulate the differences between types of elements that otherwise appear to be similar.

[041] Embodiments of the present invention include an element_relation table 162 for storing relationships between elements. Further to storing and managing elements in the universe, the EMR of FIG. 1 records and manages the fact that one element relates to another element in element_relation table 162. Each entry in element_relation table 162 may contain a pointer to at least two elements in elements table 160 and a type definition describing the type of relationship. Using the over-normalized model, the EMR ends up with a collection of elements and a collection of indicators of how those elements relate to each other.

ServiceNow Ex. 1002 Page 347 of 410

[042] Embodiments of the present invention, as shown in FIG. 3, include an element_attr_c# table 164. Element_attr_c# table 164 represents a plurality of physical tables that are constructed into the base RDBMS. A physical table may be constructed in element_attr_c# table 164 for each class that is declared in class table 150. A column is inserted into this table for each class_attribute entry that specifies an attribute that should be in this class. In the physical implementation the # symbol is replaced with the identifier of the class to ensure uniqueness in table names, which is a requirement in the RDBMS but not necessarily in the class names of the element system.

[043] For each element_attr_c# table constructed, a corresponding entry in an element_history table 166 may also be constructed. Element_history table 166 records each version of each element through its lifespan in the EMR. In addition to all values of all attributes, element_history table 166 may also store indicators of the user making the change, and the time the change was made. This provides the EMR the capability to provide a complete history for all tracked elements throughout the life of the elements. This is useful for recovering a point in time configuration, or mixing historic version of various element subsets into named groups sometimes referred to as Baselines, Blocks, Revisions, Versions, Variants, or the like.

[044] Consistent with embodiments of the invention, the EMR includes an element_document table 168, which may represent a series of records that logically group a set of elements into what would traditionally be called a document. Element_document table 168 provides for formatting information that is separated from

ServiceNow Ex. 1002 Page 348 of 410

the elements themselves, yet are important for the reconstruction for an ordered formatted display of the elements.

[045] From the collection of elements and the collection of indicators, the EMR can create visualizations for users by placing the elements in context with other elements that they relate to. FIG. 4 illustrates an exemplary visualization 400. Visualization 400 may also be called a diagrammatic visualization, which may be implemented by an elements diagram manager. In a diagrammatic visualization, the elements diagram manager selects or allows the user to select subsets of elements in the universe and place them on a drawing canvas. The elements diagram manager then arranges the elements in relation to other elements based on the conductivity rules found in element_relation table 162.

[046] In visualization 400, several elements are arranged on the canvas in proximity to each other and connected by directional lines that indicate their relation to each other, as determined by element_relation table 162. Each element in the canvas may be expanded to find and display elements in proximity to itself that is not yet on the canvas. Visualization 400 provides the user the capability to dynamically explore related elements.

[047] In addition to expansion of an element, each element in the EMR may invoke an editing screen that manages the values for the attributes of that element, based on the classification of that element. FIG. 14 displays a screen shot illustrating an exemplary editing environment 1400 that may be invoked from any displayed element, whether it appears in a diagram canvas, element explorer list, or other context.

ServiceNow Ex. 1002 Page 349 of 410

[048] As illustrated in FIG. 14, editing environment 1400 shows several tabs, a General Tab, an Attributes tab, and a History tab. The selected tab in the example shown in FIG. 14 is the Attributes tab, which displays a listing of each attribute associated with the elements class, and an editor for the value of that attribute. In this example, the type of attribute is indicated by small icons to the left of the attribute name and the editor appears to the right of the attribute name. The order in which the attributes appear and certain display characteristics such as the number of rows of text to display in the edit field are determined by the entry in class_attribute table 112. Specialized editors can be defined and used such as date selectors, pick list combo boxes, and boolean checkboxes, all of which are invoked dynamically based on the definition of the attributes that happen to be in this class.

[049] FIG. 15 shows a screen shot of another exemplary representation 1500 of the dynamic editing capability of an element. In this embodiment, the EMR provides dynamic edit capability through web-based programming language and a standard browser without a structured programming language such as Java, as seen in editing environment 1400 (of FIG. 14). By providing web-based edit capability, the EMR allows users to edit attributes of an element by invoking one or more hyperlinks to the element when the element is referenced in other web-based reporting or outputs.

[050] State machines are a very powerful extension of normal relational database attributes or column types. As shown in FIG. 5, a state_machine table 120, a state_machine_transition table 122, and a status table 124 describe the structure and behavior of state machines. The term state machine may be used interchangeably with the terms process or workflow. A state machine (e.g., an entry in state_machine table

ServiceNow Ex. 1002 Page 350 of 410

120) has one or more statuses (e.g., entries in status table 124) which are connected through transitions (e.g., entries in state_machine_transition table 122). The user may classify the elements by declaring one or more possible state machine statuses for each of the elements.

[051] Using the person example, if the person was an employee, then the EMR of FIG. 1 would be able to declare at least one possible status of the person. The EMR could declare the person's status in the state machine model as shown in FIG. 6, such as whether the person is a prospective employee 600, an actively hired employee 610, an employee on probation 620, an employee terminated for cause 630, etc. The EMR may restrict the transition from one status to another by limiting each status's forward transition connections. For instance, the user may first connect prospective employee status 600 to hired status 610. Next, the user may connect hired status 610 to on probation status 620. Furthermore, on probation status 620 may be connected to terminated for cause status 630 or back to hired status 610. Loops and branches are allowable in a stateful process.

[052] The EMR may enforce data behavior and constrain the user's ability to manipulate data based on the state machine model. For one or more stateful models, state machines may be applied to a data classification in a manner that forces data to follow at least one prescribed process declared by the user or a state administrator in a given enterprise. For instance, a prospective employee may not be immediately transitioned to terminated for cause because that violates the rules declared by the process. A prospective employee may only be propagated forward to available or declared next statuses, e.g., hired. Furthermore, the EMR may grant certain users the

ServiceNow Ex. 1002 Page 351 of 410

ability to restrict the condition under which a transition is allowable, for example, only to users in the organization who are authorized to terminate an employee. Although the transition from actively hired to terminated with cause is generally allowed, there may be one or more exclusionary rules that declare that only a certain user or subset of users may transition an actively hired employee to the status of terminated with cause. Other exclusionary rules might state that the user may not transition an actively hired employee to terminated for cause unless the user has entered a cause and linked it by relation to the employee.

[053] There are many types of data models beyond employees, e.g., the Department of Defense core architectural data model (CADM), the Defense Department architectural framework (DODAF), or the Meta Objects Facility (MOF). These are examples of standardized data models for users to store data, and the models are independent of the data definition that the user wishes to implement. The user can implement multiple concurrent disparate data definitions within a single enterprise repository, such as the EMR of FIG. 1.

[054] Currently, when a system is declared, it implements its own data model according to the user interface and the business rules specification for that system. For the conventional system, the system's developers write dedicated code and user interface code that displays or manages certain data inputs such as check boxes, lists, combo boxes or other data inputs. The conventional system's dedicated code and user interface manipulate the data that the system's application stores and retrieves during operation. Moreover, the conventional system stores and retrieves data locally on a local disk, like a C drive on a Windows system.

ServiceNow Ex. 1002 Page 352 of 410

[055] There are several drawbacks to conventional systems, one of which is that the data is typically stored in a proprietary file, i.e., a file that is a Microsoft Word® document in .doc format that contains proprietary binary code. The contents of that file are not clear unless one uses Microsoft Word® or another tool that can decode the contents. While Microsoft Word® stores textual information and formats information in a proprietary format contained within an operating system level object called a file, the EMR foregoes the file-based implementation in favor of a elements-based implementation. In the elements-based implementation, the contents of what otherwise would be a file are flexibility represented as elements. Therefore, the contents are not restricted to use by a proprietary application, but are concurrently available to all database-aware applications.

[056] In the elements-based implementation, a document would not be represented by a collection of proprietarily stored formatting and textual information. Instead, the document would be stored as an element with, e.g., one element for each sentence in the document. The elements represent a usable chunk of contextualized information, such as a sentence. In the field of ontologies, words in a sentence by themselves have no meaning. An ontology implies context. Words by themselves may have a definition, but for the words to truly convey meaning, the words need to included in at least a phrase and probably a sentence. In the elements-based approach, each of the elements conforms to the rules of generalized things in the universe, and therefore may be reused or referenced to other things in the universe, e.g., through entries in element_relation table 162. Moreover, each of the elements in its own context is also meaningful and reusable.

ServiceNow Ex. 1002 Page 353 of 410

[057] In implementing a solution for a requirements management problem, people would typically use a conventional document to articulate thousands or hundreds or tens of requirements that they would like a system or systems to achieve. In an embodiment of the elements-based implementation in the EMR, each requirement statement would be an element that the user may associate a process to, independent of any kind of hard-coded association. The user has a choice to declare, on each element, such a process which is automatically enforced for that user.

[058] Using the EMR, an element or, in this case, a requirement, would be listed and available as generally readable in an open query-able fashion. This allows the user to organize and aggregate that requirement in more than just the context of one requirements document as the user would in a conventional document-based system. For example, the user may organize and aggregate requirements in a diagrammatic depiction 700 as shown in FIG. 7 that shows congregations of requirements that are related to specific subcomponents of the main system that the user is trying to build.

[059] Diagrammatic depiction 700 shows a subset of elements that may represent requirements as in the previous example. In the example illustrated in diagrammatic depiction 700, there are at least two (2) requirements that are common to the major system components represented in diagrammatic depiction 700. If each generally circular grouping is considered a system component and its related requirements, then the left hand component is related to the right hand component through the two elements generally in the middle of the diagram and linked simultaneously to both.

ServiceNow Ex. 1002 Page 354 of 410

[060] For example, consider a situation where the user declares that an engine is needed in the process of building an airplane. In a file-based system, the user either declares that through the requirements themselves or through the practice of engineering the airplane, for which there are a thousand requirements that describe how the engine should mount to the wing. In the file-based system, the requirements may be in several spreadsheets or documents. The requirements may be repeated in the spreadsheet and a document, and may be repeated again in a testing suite application. Moreover, the requirements may be further repeated on the architectural drawing of the component. In the EMR of FIG. 1 there would be one instance of the requirement and it would be shared among all applications that needed to reference that instance of that requirement. Items such as a diagrammatic depiction (e.g. diagrammatic depiction 700), spreadsheet-like depiction, or documentation depiction, are now file independent. The items are just depictions of the core data set.

[061] Embodiments of the present invention include a named_text table 140, as shown in FIG. 8. Named_text table 140 provides a powerful normalization of all text that the user may refer to in a system. Text is a ubiquitous descriptor of many things, including graphics. With current techniques, there are textual representations of graphics, notably scalable vector graphic (SVG) from companies like Adobe®, which are textual descriptions that can be rendered into graphics. For example, a textual description may state "draw a red circle with a 50 pixel radius." The textual description can be translated into an actual depiction on the screen. This illustrates that text represents most things or most things can be represented by text, not only graphics through the SVG, but also any data structure through extended markup language. The

ServiceNow Ex. 1002 Page 355 of 410

EMR takes advantage of this by declaring these sets of texts and identifying them with a name and a unique identifier. These sets of texts then become usable in the elements-based implementation in many ways.

[062] Embodiments of the present invention include a tuple table 144, as shown in FIG. 9. Tuple table 144 is similar to element_relation table 162, except that tuple table 144 has been separated to store information about the system schema as opposed to the generated user schema. Tuple table 144 relates any identified item to any other identified item and that relationship is created according to a type (e.g., type_id). Given that an entry in named_text table 140 is a thing and that the user may relate the thing to other types of named text, the EMR declares, for the user's convenience, a named_text item that is akin to a folder in an operating system hierarchy. The folder may be called, for example, "My Favorites," or "My Documents," or in this instance, "My Elements." That folder can then be referenced by the user. Next, the EMR may declare 2, 3, 5 or 100 (or any number of) sub-folders. The sub-folders may be referenced by entries in tuple table 144 to the folder and other sub-folders.

[063] The combination of named_text table 140 and tuple table 144 provides to the EMR the capability to create a recursion between named_text types that represent folders to other named_text types that represent folders in a hierarchy. In an embodiment shown in FIG. 9, the EMR may contain a folder called root (e.g. "EE Root"), another folder called "My elements" (e.g. "My Elements (MARK)" and "My Elements (RONO)"), and tuples the root entry in named_text table 140 to the "My Elements" entry in named_text table 140. In doing so, the EMR indicates through elements explorer 900

ServiceNow Ex. 1002 Page 356 of 410

a hierarchy or a tree or an indenture of the folders. This particular implementation approximates a standard operating system file explorer, in which the user can click and drag and drop folders and other items into a folder in a tree or navigable folder set. Through the same mechanism, an element may be tupled into a folder. In the elements-based implementation, the EMR may use a folder to organize and aggregate random sets of things in the universe.

[064] The EMR may utilize entries in named_text table 140 to name structured query language queries. SQL is a textual language that describes an interaction or request to a relational database. SQL queries are the mechanism to retrieve data from the database. SQL query is a powerful notion for relational databases and may be implemented as a named_text entry. Unlike a conventional database, e.g., Oracle® or Microsoft SQL Server®, that stores user-generated SQL as a "view," the EMR stores user-generated SQL as a named_text entry. Because the user-generated SQL is an entry in named_text table 140, and since folders exist as entries in named_text table 140 or tuple table 144, the EMR may filter or query entries in named_text table 140 or tuple table 144 into one or more folders.

[065] Thus, if the SQL statement serves as a folder, the results of the SQL statement or the items returned by the query would be the contents of that folder. The SQL statement to the user is therefore a dynamic folder or a SQL folder and no tuple exists between that data set. In other words, elements in the dynamic folder can be placed into other folders. The dynamic folder is executed at run-time if the user selects the dynamic folder and the resulting elements that come back from the query are presented as if they were tupled into that folder. This is a dynamic process and the

result set may differ depending on the dynamic data changes. This dynamic process simplifies the organization and provides to the user the ability to locate data through a dynamic folder hierarchy. The dynamic folder hierarchy further includes dynamic folders that change the data set appropriately every time the user visits them.

[066] An example of the dynamic folder capability provided by the EMR is a "My Tasks" folder. Assume that task is an element classification and that user is an attribute of the task. If there are tasks in the system that are assigned to the user, the user may perform a query that directs the EMR to select one or more tasks for which the user is the assigned user. The result of that query may change over time. The administrator may assign a task to someone else, so the task may be deleted or moved to a completed state and the user's folder would only contain a current task set that matches the query as defined by a named_text of that folder. This allows for an extensible method of organizing data of interest and allowing the user to easily find and retrieve data of interest in a large, complex system.

[067] Embodiments of the present invention, as shown in FIG. 10, include a poll table 182. Poll table 182 may represent a set of identified questions that are in one of several stages of life cycle of an element or set of elements. A poll is a collaborative mechanism that restricts the responses of the participants to one or more predetermined choices which are enumerated in a poll_option table 188. The replies of the participants are recorded in a poll_reply table 184. These poll replies can be analyzed for preference among the voting community and used to record decision making processes regarding any given element or set of elements. This relation is achieved through entries in tuple table 144 (in FIG. 8).

ServiceNow Ex. 1002 Page 358 of 410

[068] In accordance with still further embodiments of the invention, the EMR includes a discussion table 186, which is similar to poll table 182 as discussion table 186 records collaborative information regarding one or more elements in the system. However, a discussion, which may be stored as entry in discussion table 186, is not forced into a set of structured responses. Instead discussion table 186 allows free-form narrative to be recorded, and replied to by any number of users in the system.

[069] Embodiments of the present invention may further include a collaboration_status_history table 180, which may record all the states that either a discussion or poll moves through in a defined life cycle, similar to element history. Moreoever, the user and date and time is recorded with each status change.

[070] In accordance with still further embodiments of the invention, the EMR includes a system_component table 190. The EMR may register components that are programmatic in nature (e.g. Java classes), wherein the components may be executed dynamically as needed (through a process called reflection) as part of element decoration. For example, one or more Java classes may be defined and saved as a record in system_component table 190, then the Java classes may be tupled to elements. At runtime, the Java classes that are referenced may be executed when an element of the Java class is edited or viewed.

[071] Embodiments of the present invention, as shown in FIG. 11, include a terminology table 130 and a glossary 132. Terminology table 130 may store a list of terminologies that can be used in the system. Glossary table 132 may uniquely identify and define each of the terms stored in terminology table 130.

ServiceNow Ex. 1002 Page 359 of 410

[072] In accordance with embodiments of the invention as shown in FIG. 12, the EMR includes an email table 172 and an email_detail table 170. These tables allow the EMR to populate records in the database that can programmatically be converted from rows in the RDBMS to actual emails using a known protocols such as SMTP and POP. Since the emails stored in email table 172 and email_detail table 170 are also data, the EMR may apply stateful process logic to each email - using status, state_machine, and state_machine_transition tables (FIG. 5, elements 120, 122, 124, respectively) to define the lifecycle of an email. Full email history can be maintained in the EMR and new emails can be generated either programmatically when an event occurs, or manually when a user enters an email. Individual messages may be aggregated into one larger email at a time interval specified by an administrator. The aggregated messages may also be sent on demand.

[073] In accordance with embodiments of the invention as shown in FIG. 13, the EMR includes a formatted_dataset table 176 and a SQL_XSLT table 178. Together, formatted_dataset table 176 and SQL_XSLT table 178 allow the EMR to create a named "formatted dataset," which applies formatting to a SQL result set. Formatted_dataset table 176 and SQL_XSLT table 178 may work in conjunction with named_text table 140 to relate one or more named text entries (representing SQL statements), which provide the result set(s), with another named text entry (representing XSLT or a formatting definition) which generates a final layout.

[074] To further illustrate the EMR as described above and in FIGS. 1-15, an example is provided to demonstrate the utilization and capability of the EMR, consistent with the present invention, for storing, managing, indexing, interrelating,

and/or retrieving data sets, such as large scale and dynamic data sets. One example, as shown in FIG. 16, may be Human Resource Management (HRM). Further examples may include, for instance, a vertical implementation, or schema and associated processes, that may serve the needs of a well defined market, e.g., Enterprise Architecture, Requirements Management, Six Sigma, Resource Planning, Federal Regulatory Compliance, Legal Document Management, Portfolio Management, etc.

[075] In the HRM example, it may be determined that tracking "resources" is desired. Those resources may be people, but may not necessarily be restricted to people, which is a distinction that may be made when defining the implementation. It may further be determined that the resources of interest will in fact be people, and those people will be employees of a specified company. The company may have written policies that determine employment, such as the amount of vacation time each employee may receive, a level of compensation, etc.

[076] Furthermore, it may be determined that the resources may exist in a hierarchy, e.g., have a place in an organization chart, and that tasks should be identified and assigned to these resources as needed. Steps may be taken within the EMR to support such system requirements.

[077] A workspace (e.g., HRM Workspace 1600 of FIG. 16) may optionally be entered into the ERM (e.g. workspace_object table 106 and workspace table 108 of FIGS. 1 and 2) to segregate the HRM information from other possible data sets in the EMR if desired.

[078] A schema would be logically defined that encompassed the articulated needs of the system. In this case perhaps a resource class 1602 (of FIG. 16) would be

ServiceNow Ex. 1002 Page 361 of 410

declared with attributes of employee_name, employee_ID, and employee_status. The employee_status attribute may be of type state_machine, allowing the stateful process of employment to be declared according to the rules of the company.

[079] It may further be determined that a resource may change title and salary over time, and other classes, such as a resource_title class 1606 and a resource_compensation class 1608, would be declared each with a pointer to resource class 1602 such as resource_ID and other appropriate attributes such as resource_title, effective_start_date, and effective_end_date. Furthermore, resource_compensation class 1608 may similarly contain compensation_amount, compensation_interval, compensation_start_date, and compensation_end_date.

[080] Tasks would also be similarly defined as a task class 1604 with, for example, a name, a description, a milestone indicator, a status, an assignee, an estimated_duration, and a project indicator.

[081] To implement these classes, data would be entered into the EMR representing these structures. The order of implementation, as illustrated below, is exemplary and does not limit embodiments of the present invention.

[082] Entries may be made into class table 150 (of FIGS. 1 and 2), one each for resource, task, resource_title, and resource_compensation. Entries may be made into attribute table 110 (of FIGS. 1 and 2) for each employee_name, employee_ID, employee_status, employee_title, compensation_interval, compensation_amount, compensation_start_date, compensation_end_date, status, assignee, estimated_duration, and project.

ServiceNow Ex. 1002 Page 362 of 410

[083] As further definition, entries may be made into state_machine table 120, state_machine_transition table 122, and status table 124 (of FIGS. 1 and 5) to declare the available selections of the status and employee_status attributes. Further entries may be placed into list_option table 114 (of FIGS. 1 and 2) to specify the available resource_titles that may be assigned to resource or resources.

[084] Entries may then be made into class_attribute table 112 (of FIGS. 1 and 2) to associate the appropriate attributes with their respective classes. This may all be done via any data insertion method available, e.g. a user interface, etc.

[085] Once these class_attribute records have been entered into class_attribute table 112, the EMR is ready for use. Users have the capability to add resources, create tasks, assign them to the resources, report on how many tasks are completed, etc. Users may also create folders to organize projects (see, e.g., FIG. 9) and view diagrams (see, e.g., FIGS. 4 and 7) of which resources are committed to which tasks without any further programmatic effort. Users may also be able to collaborate on tasks by discussing ramifications or polling a subset of users to determine specific answers to to-be-determined elements.

[086] In this example, the resource hierarchy may be constructed by linking one resource to another recursively (see, e.g., FIG. 16). This may be done either via the record editing capabilities or by using the diagramming tool (see, e.g., FIGS. 4 and 7) to draw a line from one resource to another. The diagrammatic representation of the organization chart is capable of immediately expand an individual to supervisors, peers, or subordinates. The diagrammatic representation may also show all related tasks to one or more of those individuals if desired. By providing those capabilities, the

ServiceNow Ex. 1002 Page 363 of 410

diagrammatic representation allows for visualization of tasking overloads or underutilized resources, etc.

[087] If the EMR does not provide a specific aggregation or subsetting of tasks or resources, one or more named_text entries in named_text table 140 (of FIGS. 1 and 8) may be defined that return the appropriate results. If the default display of those results is insufficient for the business needs, then one or more formatted_dataset entries in formatted_dataset table 176 (of FIGS. 1 and 13) and SQL_XSLT entries in SQL_XSLT table 178 (of FIGS. 1 and 13) may be defined to convert the available results into any desired final output.

[088] In this example, when the user edits or inserts data in the EMR, the data may be stored in an appropriate element_attr_c# table 164 (of FIGS. 1 and 3) that was automatically constructed as part of the class definition step above. Moreover, any modifications may automatically be recorded as history in element_history table 166 (of FIGS. 1 and 3).

[089] The above-described exemplary scenario may be followed for any given schema. A full schema does not need to be defined before the EMR may be used. If for instance only resources were defined, then resource information could be entered, and organization hierarchies may be defined at a later date; the tasking information may be added without disruption to the EMR. As a further example, additional attributes may be added to the resources themselves at a later date if the attributes are deemed necessary. As another example, the EMR's user interfaces may automatically adjust to the current definition to allow seamless upgrades of the data models.

ServiceNow Ex. 1002 Page 364 of 410

[090] Systems and methods consistent with the present invention may be implemented by computers organized in a conventional distributed processing system architecture. FIG. 17 is a block diagram illustrating a computer system 200 in which an embodiment consistent with the invention may be implemented. Computer system 200 includes a computer 201 having a bus 202 or other communication mechanism for communicating information, and a processor 204 coupled with bus 202 for processing information. Computer 201 also includes a main memory, such as random access memory (RAM) 206 or other dynamic storage device, coupled to bus 202 for storing information and instructions by processor 204. RAM 206 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 204. During operation, some or all portions of an operating system (not shown) are loaded into RAM 206. Computer 201 further includes a read only memory (ROM) 208 or other static storage device coupled to bus 202 for storing static information and instructions for processor 204. A storage device 210, such as a magnetic disk or optical disk, is provided and coupled to bus 202 for storing information and instructions.

[091] Computer 201 may be coupled via bus 202 to a display 212, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 214, including alphanumeric and other keys, is coupled to bus 202 for communicating information and command selections to processor 204. Another type of user input device is cursor control 216, such as a mouse, a trackball or cursor direction keys for communicating direction information and command selections to processor 204 and for controlling cursor movement on display 212. This input device may have two degrees

ServiceNow Ex. 1002 Page 365 of 410

of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), which allow the device to specify positions in a plane.

[092] System 200 may further include an input drive device 217, such as a CD reader, for reading high-capacity computer-readable media, such as CD-ROMs and CDRs.

[093] The invention is related to the use of computer system 200 for manipulating data sets for, to take an example, corporate or engineering exercises. According to one implementation, systems and methods consistent with the present invention collect system information and store it in a central database in response to processor 204 executing one or more sequences of one or more instructions contained in RAM 206. Such instructions may be read into RAM 206 from a computer-readable medium via an input device such as storage device 210. Execution of the sequences of instructions contained in main memory 206 causes processor 204 to perform the process steps described herein. In an alternative implementation, hard-wired circuitry may be used in place of or in combination with software instructions to implement the invention. Thus, implementations consistent with the principles of the present invention are not limited to any specific combination of hardware circuitry and software.

[094] Computer 201 also includes a communication interface 218 coupled to bus 202. Communication interface 218 provides a two-way data communication coupling to a network link 220 that is connected to local network 222. For example, communication interface 218 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 218 may be a local area

ServiceNow Ex. 1002 Page 366 of 410

network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented. In any such implementation, communication interface 218 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[095] Network link 220 typically provides data communication through one or more networks to other data devices. For example, network link 220 may provide a connection through local network 222 to host computer 224 and/or to data equipment operated by Internet Service Provider (ISP) 226. ISP 226, in turn, provides data communication services through the Internet 228. Local network 222 and Internet 228 both use electric, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 220 and through communication interface 218, which carry the digital data to and from computer system 200, are exemplary forms of carrier waves transporting the information.

[096] Computer system 200 can send messages and receive data, including program code, through the network(s), network link 220 and communication interface 218. In the Internet example, a server 230 might transmit a requested code for an application program through Internet 228, ISP 226, local network 222 and communication interface 218. The received code may be loaded in RAM 206 and executed by processor 204 as it is received. Alternatively, or in addition, it may be stored in storage device 210, or other non-volatile storage for later execution. In this manner, computer system 200 may obtain application code in the form of a carrier wave.

ServiceNow Ex. 1002 Page 367 of 410

[097] Although computer system 200 is shown in FIG. 17 as being connectable to one server 230, those skilled in the art will recognize that computer system 200 may establish connections to multiple servers on Internet 228. Such servers may include an HTML-based Internet application, which may provide information to computer system 200 upon request in a manner consistent with the present invention.

[098] The foregoing descriptions of the invention have been presented for purposes of illustration and description. They are not exhaustive and do not limit the invention to the precise form disclosed. Modifications and variations are possible in light of the above teachings or may be acquired from practicing of the invention. For example, the described implementation includes software but the present invention may be implemented as a combination of hardware and software or in hardware alone. Additionally, although aspects of the present invention are described as being stored in memory, one skilled in the art will appreciate that these aspects can also be stored on other types of computer-readable media, such as secondary storage devices, like hard disks, floppy disks, or CD-ROM; a carrier wave from the Internet or other propagation medium; or other forms of RAM or ROM.

ServiceNow Ex. 1002 Page 368 of 410

WHAT IS CLAIMED IS:

1. A computerized system for manipulating data sets, comprising:

a processor; and

a data repository, the data repository being adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets, comprising:

an element module configured to store and uniquely identify a plurality of elements, wherein each of the elements is identified by a unique element identification;

an element relation module configured to store one or more relationships between the elements in the element module;

a class module configured to define at least one class of the elements and store the class;

an attribute module configured to define one or more attributes and store the attributes;

a class attribute module configured to define and store one or more class-attribute associations between at least one of the attributes and the class; and

a type definition module configured to define and store one or more types of the class, the attributes related to the class, and the relationships between the elements.

ServiceNow Ex. 1002 Page 369 of 410

2. The computerized system of claim 1, wherein the data repository further comprises:

a state machine module, the state machine module including a state machine transition module and a status module, the state machine module configured to define and store a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element.

3. The computerized system of claim 2, wherein the state machine module defines and stores at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

4. The computerized system of claim 1, wherein the type definition module defines a type including at least one of a boolean type, a picklist type, and a state machine type.

5. The computerized system of claim 4, wherein the data repository further comprises:

a list option module configured to enumerate one or more picklist attributes of the picklist type.

ServiceNow Ex. 1002 Page 370 of 410

6. The computerized system of claim 2, wherein the data repository further comprises:

a named text module configured to store textual representations of query, control, and display mechanisms that relate to the elements.

7. The computerized system of claim 6, wherein the data repository further comprises:

a tuple module configured to generate and store one or more tuples, wherein each of the tuples associates a subset of the elements in accordance to one of the types associated with the subset of the elements.

8. The computerized system of claim 1, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

9. The computerized system of claim 8, wherein the data repository further comprises:

an element attribute module for each of the at least one class, the element attribute module configured to define and store the attributes of the elements in the class, wherein the element attribute module is identified using the name of the class.

ServiceNow Ex. 1002 Page 371 of 410

10. The computerized system of claim 9, wherein the data repository further comprises:

an element history module configured to store one or more indicators of modification to the elements.

11. The computerized system of claim 10, wherein the data repository further comprises:

an element document module configured to store one or more formatting data for grouping and displaying a subset of the elements.

12. The computerized system of claim 1, wherein the processor is configured to generate a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

13. The computerized system of claim 1, wherein the processor is configured to generate a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

ServiceNow Ex. 1002 Page 372 of 410

14. A method for manipulating data sets, comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; and

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

15. The method of claim 14, further comprising:

defining a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element, and storing the state machine type in the data repository.

ServiceNow Ex. 1002 Page 373 of 410

16. The method of claim 15, wherein the state machine type comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

17. The method of claim 14, wherein the type comprises a type including at least one of a boolean type, a picklist type, and a state machine type.

18. The method of claim 17, further comprising:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

19. The method of claim 15, further comprising:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

20. The method of claim 19, further comprising:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

21. The method of claim 14, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

ServiceNow Ex. 1002 Page 374 of 410

22. The method of claim 21, further comprising:

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

23. The method of claim 22, further comprising:

storing, in the data repository, one or more indicators of modification to the elements.

24. The method of claim 23, further comprising:

storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

25. The method of claim 14, further comprising:

generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

26. The method of claim 14, further comprising:

generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

ServiceNow Ex. 1002 Page 375 of 410

27. A computer-readable medium containing instructions to configure a processor to perform a method comprising:

storing and uniquely identifying a plurality of elements in a data repository, each of the elements identified by a unique element identification, wherein the data repository is adapted to process, retrieve, and store data contained in the data sets and one or more layers of metadata of the data in the data sets;

storing, in the data repository, one or more relationships between the elements;

defining at least one class of the elements and storing the class in the data repository;

defining one or more attributes and storing the attributes in the data repository;

defining one or more class-attribute associations between at least one of the attributes and the class and storing the class-attribute associations in the data repository; and

defining one or more types of the class, the attributes related to the class, and the relationships between the elements, and storing the types in the data repository.

28. The computer-readable medium of claim 27, wherein the method further comprises:

defining a state machine type, the state machine type storing a first status of one element and a transition to a second status of the one element, and storing the state machine type in the data repository.

ServiceNow Ex. 1002 Page 376 of 410

29. The computer-readable medium of claim 28, wherein the state machine type comprises at least one forward transition connection, the forward transition connection restricting the transition from the first status to the second status of the one element.

30. The computer-readable medium of claim 27, wherein the type comprises a type including at least one of a boolean type, a picklist type, and a state machine type.

31. The computer-readable medium of claim 30, wherein the method further comprises:

enumerating one or more picklist attributes of the picklist type and storing the enumerated picklist attributes in the data repository.

32. The computer-readable medium of claim 28, wherein the method further comprises:

storing, in the data repository, textual representations of query, control, and display mechanisms that relate to the elements.

33. The computer-readable medium of claim 32, wherein the method further comprises:

generating one or more tuples, each of the tuples associating a subset of the elements in accordance to one of the types associated with the subset of the elements, and storing the tuples in the data repository.

ServiceNow Ex. 1002 Page 377 of 410

34. The computer-readable medium of claim 27, wherein each of the class-attribute associations is identified by a logical name and a name of the class.

35. The computer-readable medium of claim 34, wherein the method further comprises:

defining, for each of the at least one class, a set of the attributes of the elements in the class, wherein the set is identified using the name of the class, and storing the set in the data repository.

36. The computer-readable medium of claim 35, wherein the method further comprises:

storing, in the data repository, one or more indicators of modification to the elements.

37. The computer-readable medium of claim 36, wherein the method further comprises:

storing, in the data repository, one or more formatting data for grouping and displaying a subset of the elements.

38. The computer-readable medium of claim 27, wherein the method further comprises:

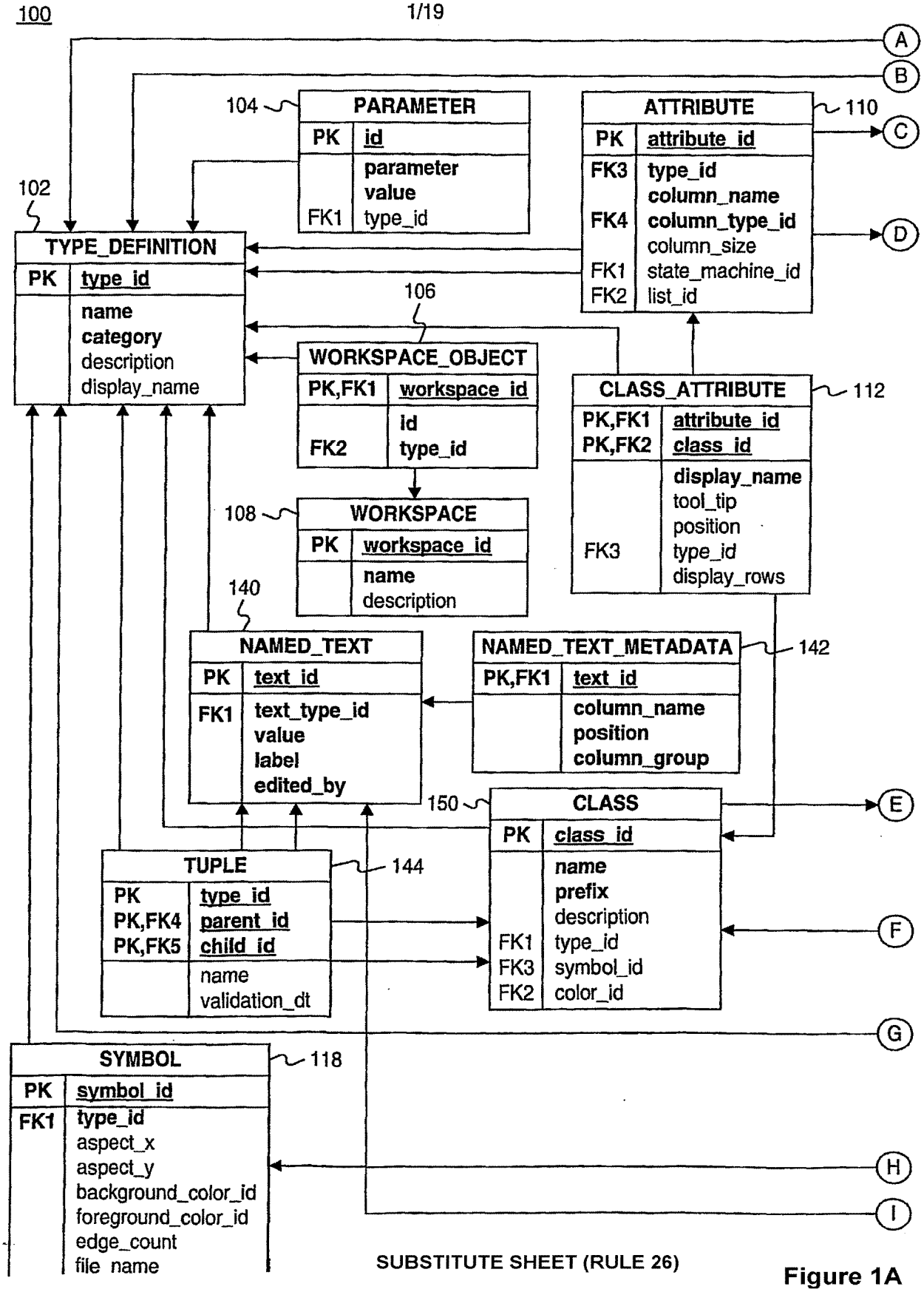
generating a data model of the elements based on the data in the data sets and the layers of metadata stored in the data repository.

ServiceNow Ex. 1002 Page 378 of 410

39. The computer-readable medium of claim 27, wherein the method further comprises:

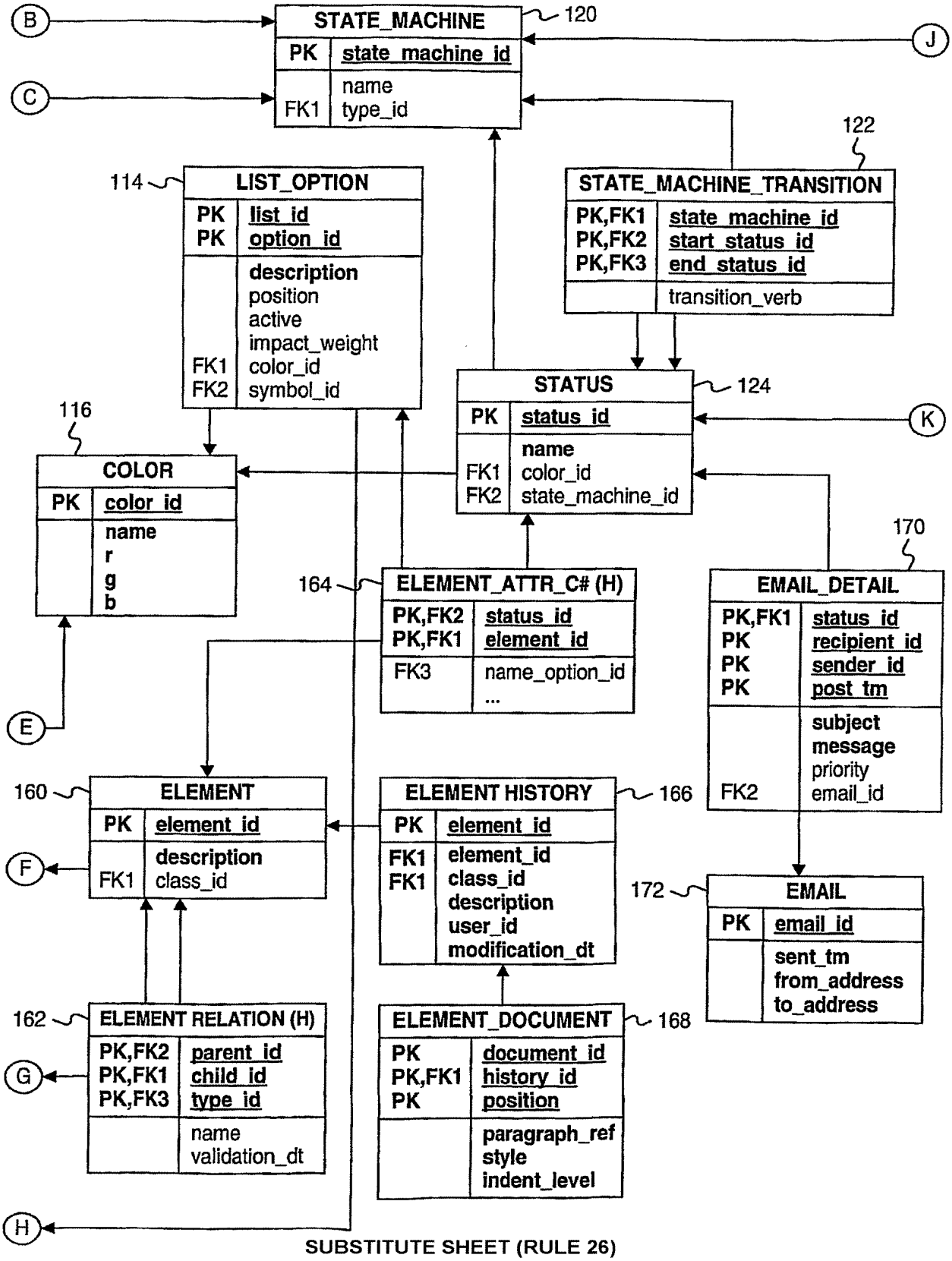
generating a dynamic user-interface for accessing and manipulating the elements based on the data in the data sets and the layers of metadata stored in the data repository.

ServiceNow Ex. 1002 Page 379 of 410



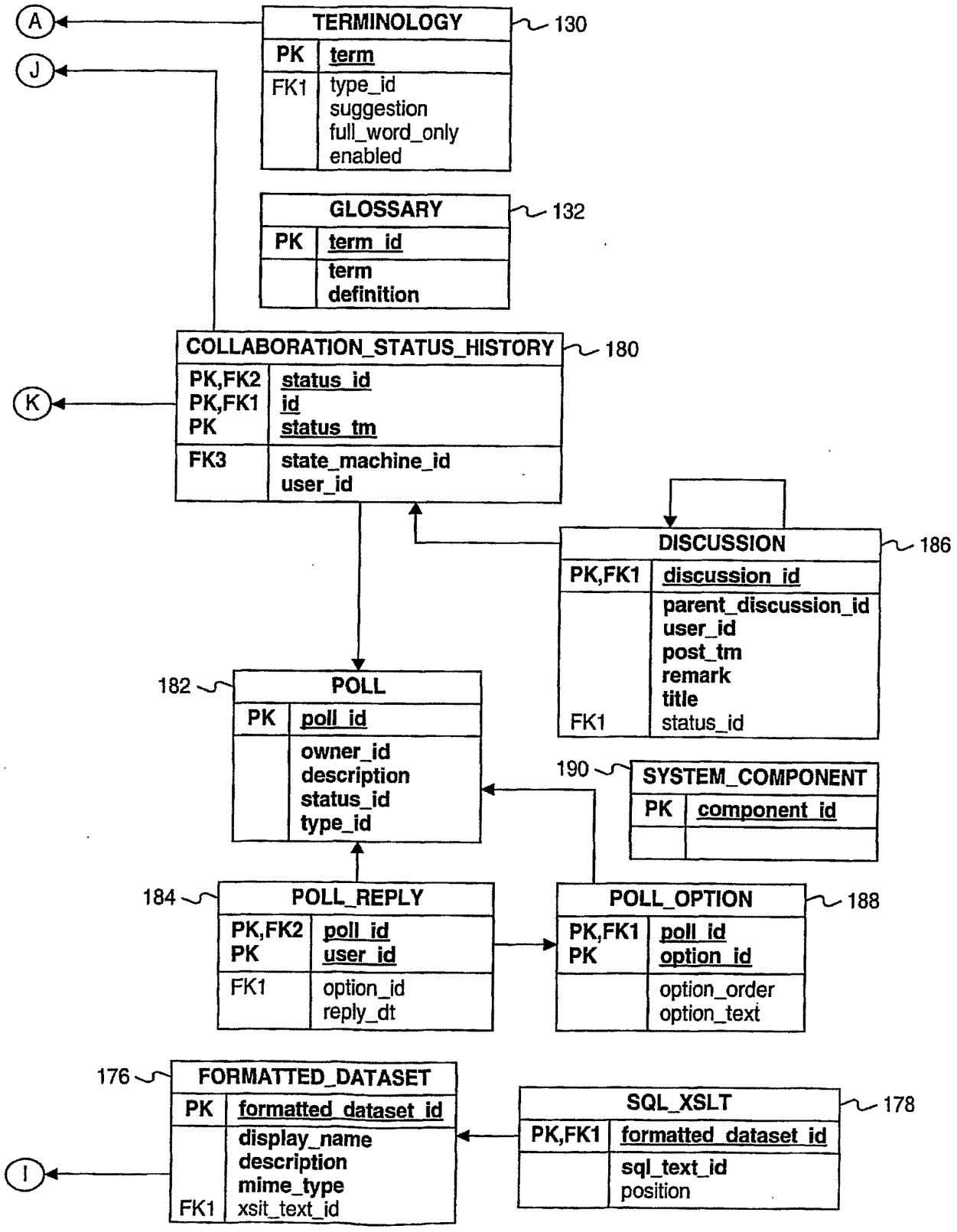
SUBSTITUTE SHEET (RULE 26)

Figure 1A



SUBSTITUTE SHEET (RULE 26)

Figure 1B



SUBSTITUTE SHEET (RULE 26)

Figure 1C

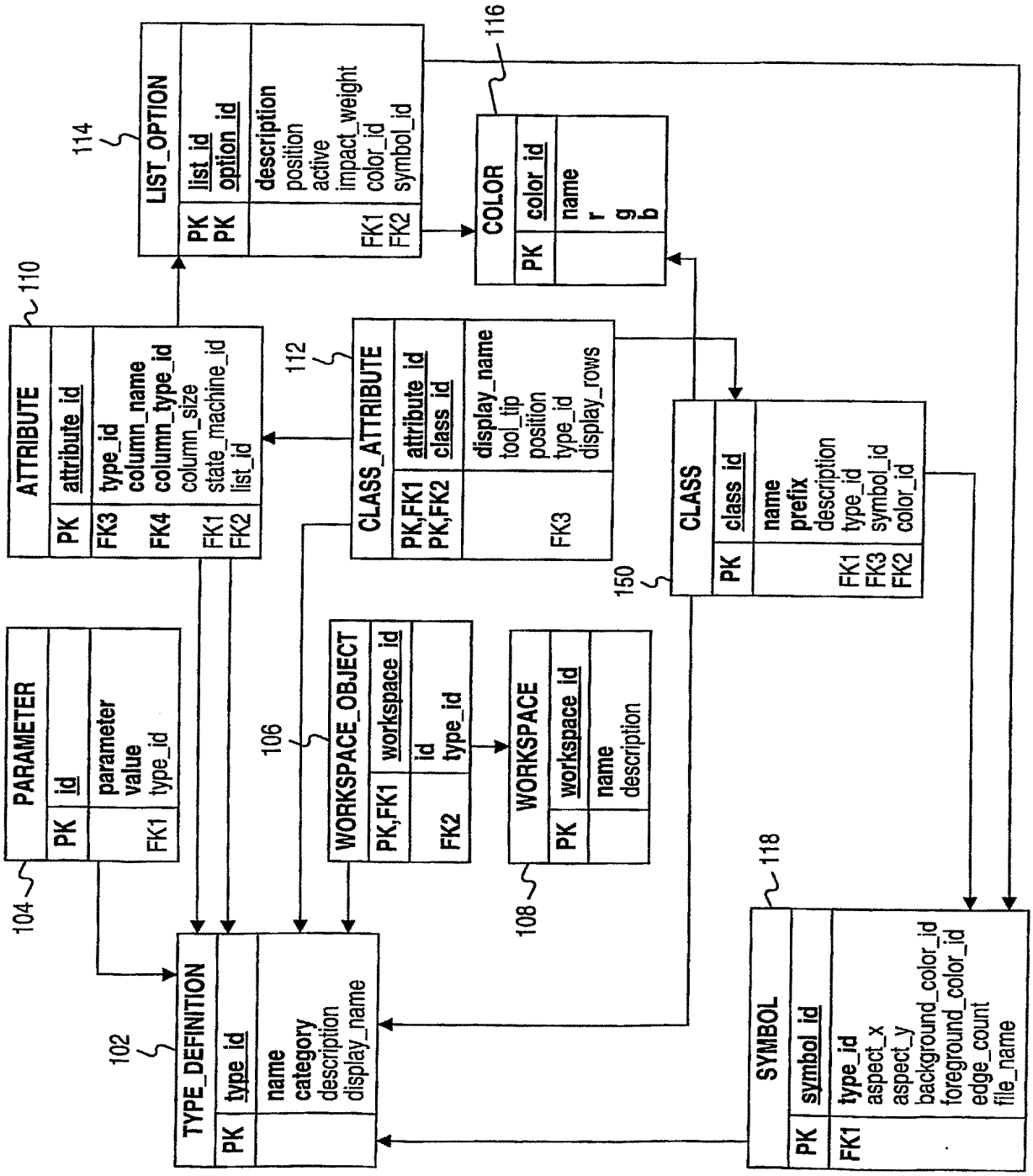

TYPE_DEFINITION
PK type_id
name
category
description
display_name
102
ATTRIBUTE
PK attribute_id
FK3 type_id
FK4 column_name
column_type_id
column_size
state_machine_id
FK1 list_id
FK2
110
PARAMETER
PK id
parameter
value
FK1 type_id
104
WORKSPACE_OBJECT
PK,FK1 workspace_id
id
FK2 type_id
106
WORKSPACE
PK workspace_id
name
description
108
LIST_OPTION
PK list_id
PK option_id
description
position
active
impact_weight
FK1 color_id
FK2 symbol_id
114
COLOR
PK color_id
name
r
g
b
116
CLASS_ATTRIBUTE
PK,FK1 attribute_id
PK,FK2 class_id
display_name
tool_tip
position
type_id
display_rows
FK3
112
CLASS
PK class_id
name
prefix
description
type_id
symbol_id
color_id
FK1
FK3
FK2
150
SYMBOL
PK symbol_id
FK1 type_id
aspect_x
aspect_y
background_color_id
foreground_color_id
edge_count
file_name
118

Figure 2

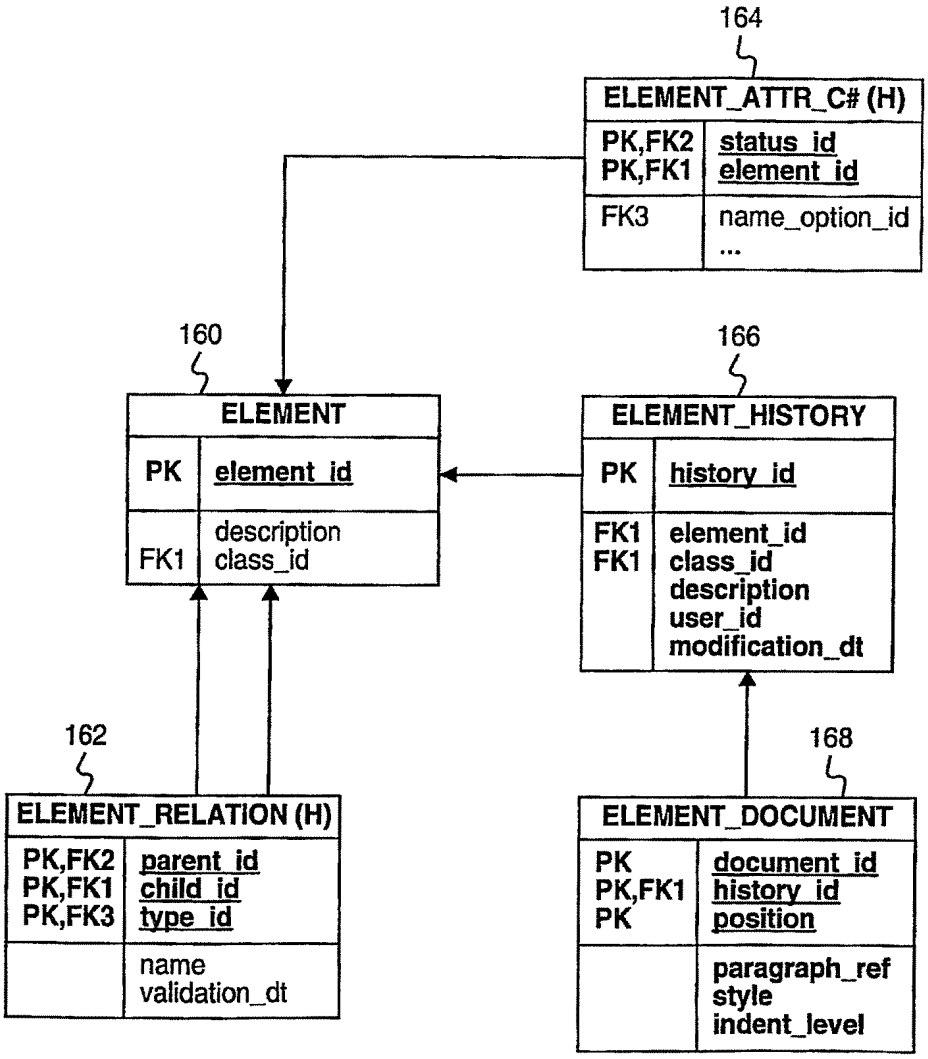

164
ELEMENT_ATTR_C# (H)
PK,FK2 status_id
PK,FK1 element_id
FK3 name_option_id
...
160
ELEMENT
PK element_id
FK1 description
class_id
166
ELEMENT_HISTORY
PK history_id
FK1 element_id
FK1 class_id
description
user_id
modification_dt
162
ELEMENT_RELATION (H)
PK,FK2 parent_id
PK,FK1 child_id
PK,FK3 type_id
name
validation_dt
168
ELEMENT_DOCUMENT
PK document_id
PK,FK1 history_id
PK position
paragraph_ref
style
indent_level

Figure 3

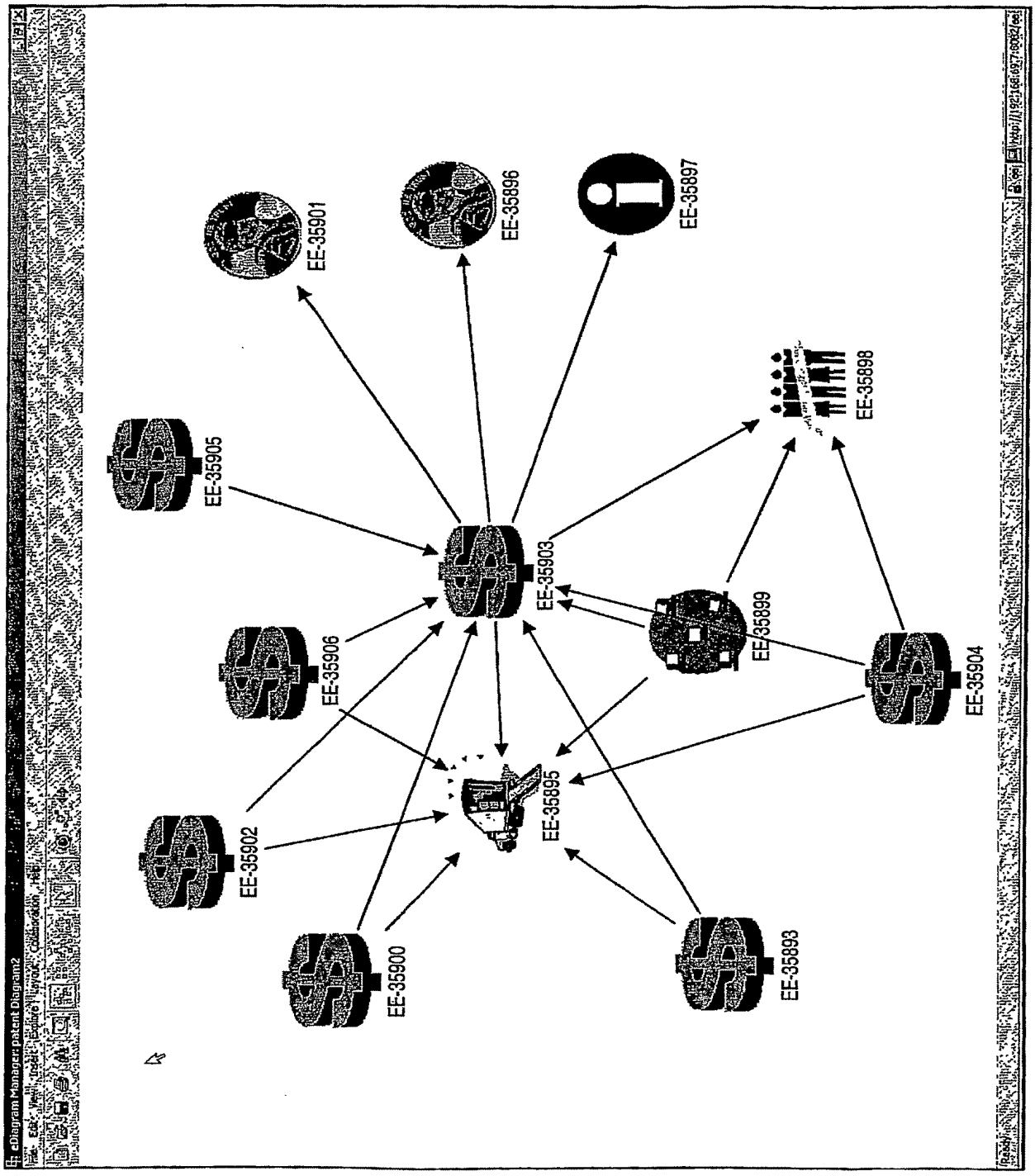

EE-35901
EE-35896
EE-35897
EE-35905
EE-35898
EE-35903
EE-35906
EE-35899
EE-35904
EE-35895
EE-35902
EE-35900
EE-35893

Figure 4

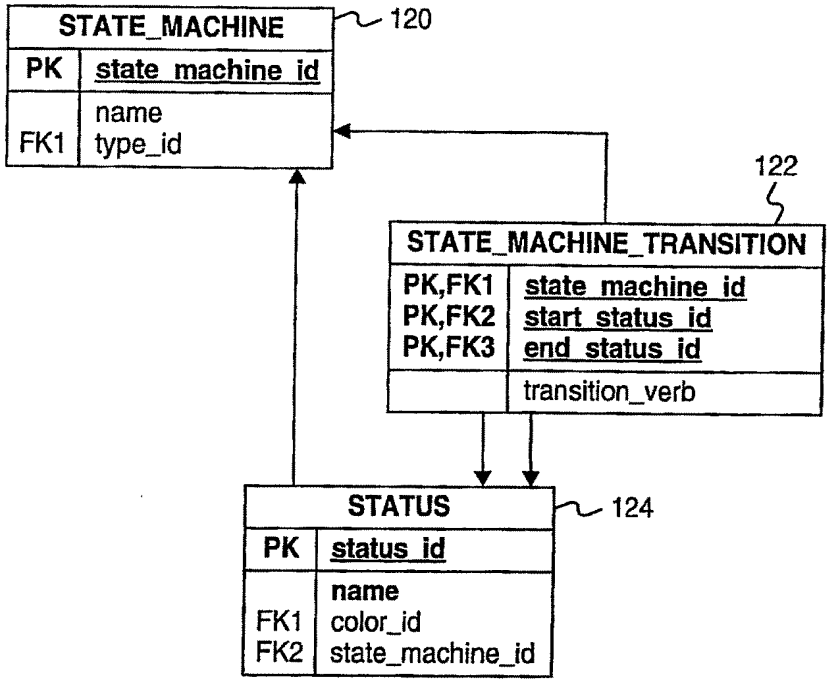


Figure 5

ServiceNow Ex. 1002 Page 386 of 410

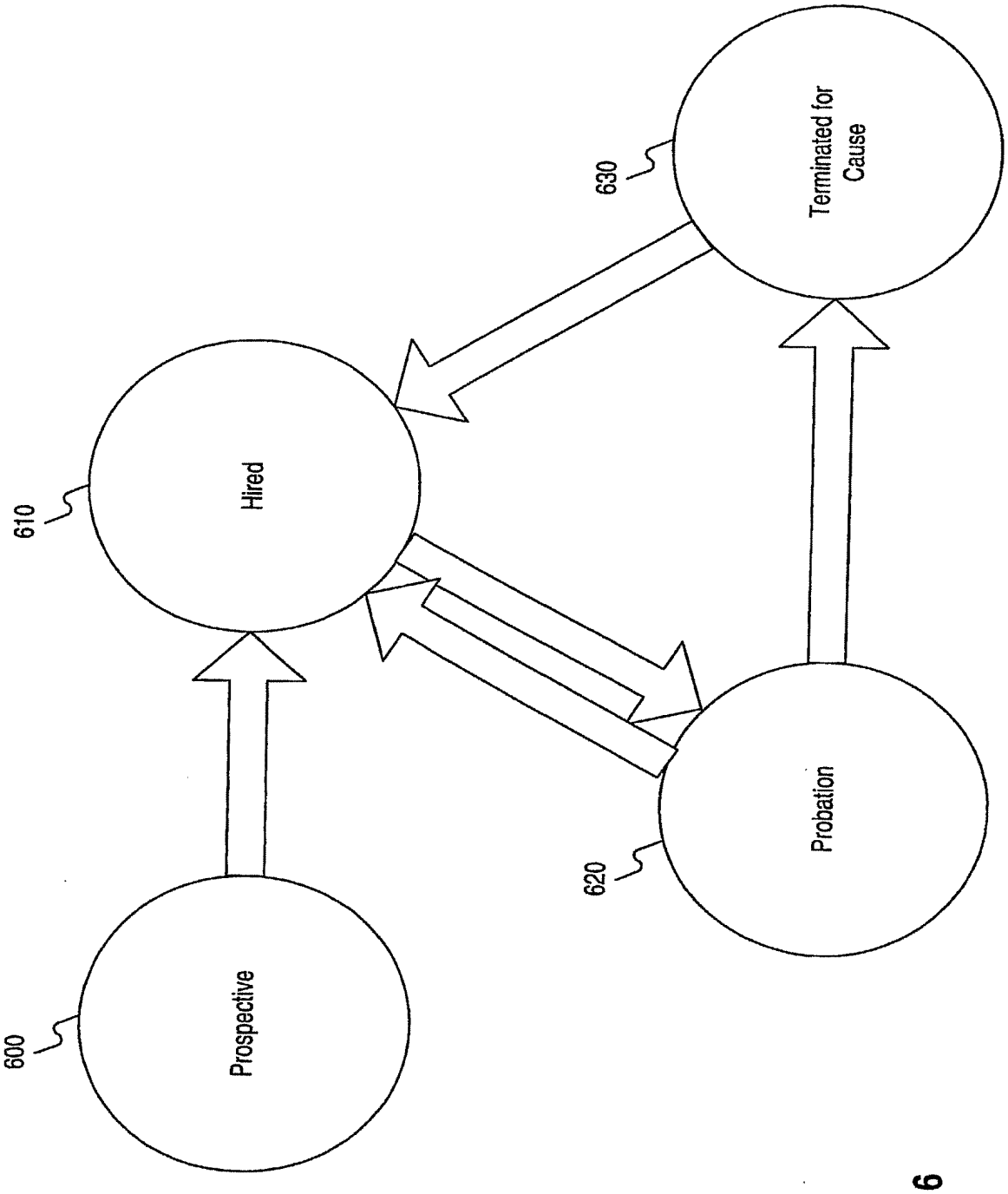


Figure 6

ServiceNow Ex. 1002 Page 387 of 410

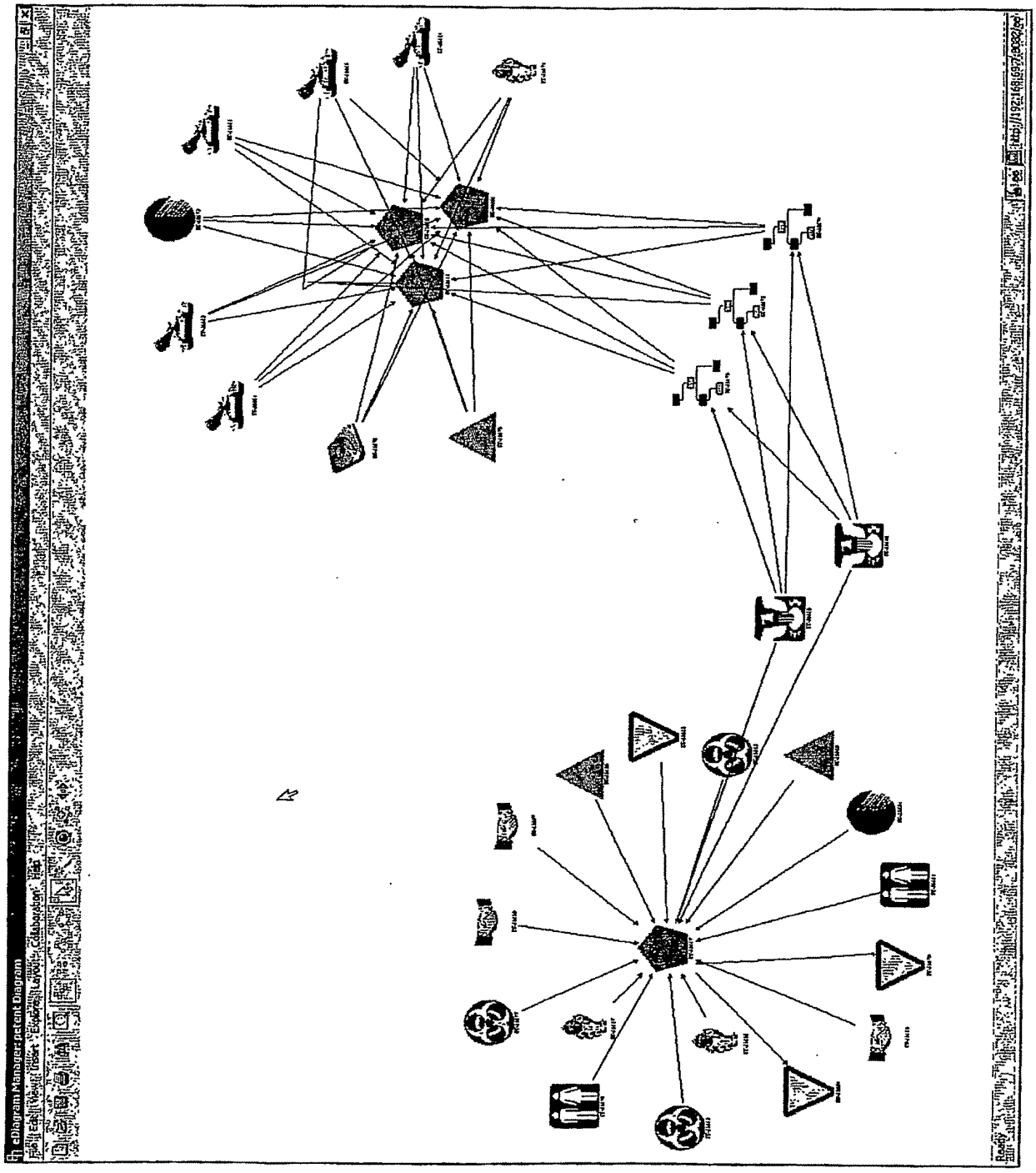


Figure 7

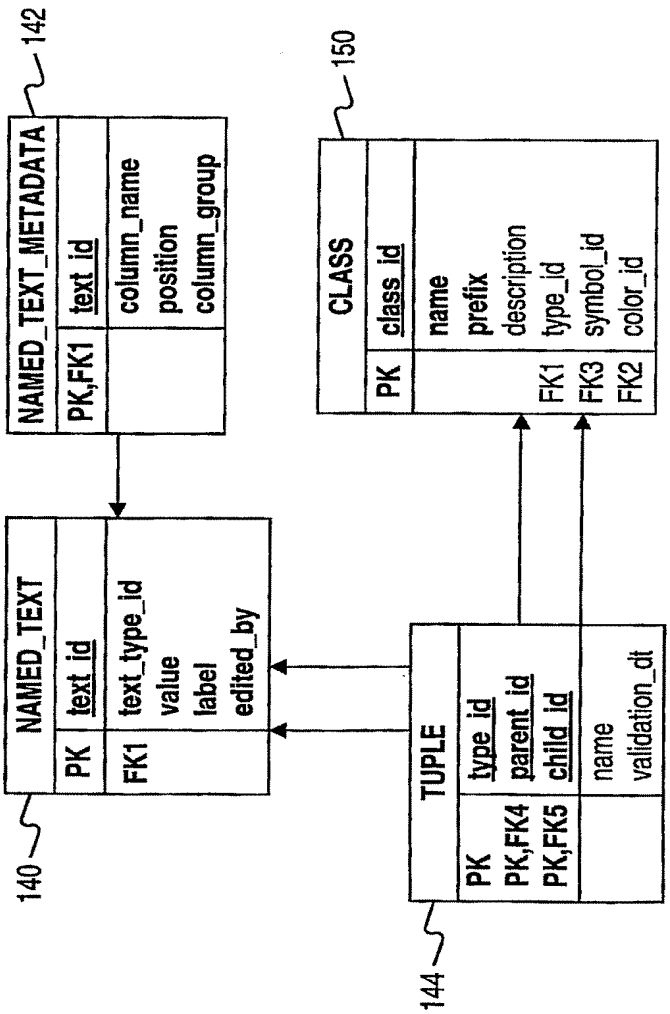

NAMED_TEXT_METADATA
142
PK,FK1 text_id
column_name
position
column_group
CLASS
150
PK class_id
name
prefix
description
FK1 type_id
FK3 symbol_id
FK2 color_id
NAMED_TEXT
140
PK text_id
FK1 text_type_id
value
label
edited_by
TUPLE
144
PK type_id
PK,FK4 parent_id
PK,FK5 child_id
name
validation_dt

Figure 8

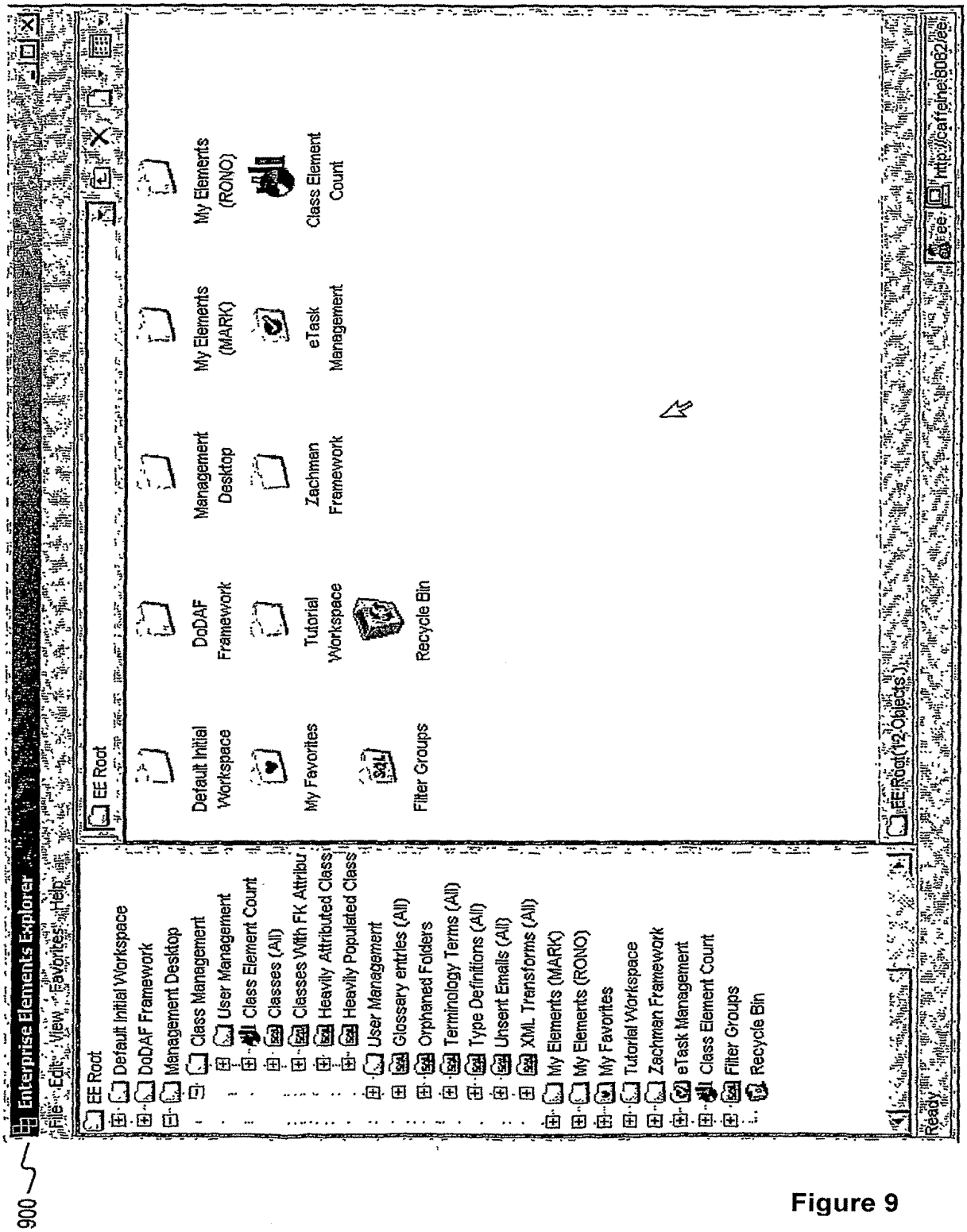

900
Enterprise Elements Explorer
File Edit View Favorites Help
EE Root
Default Initial Workspace
DoDAF Framework
Management Desktop
Class Management
User Management
Class Element Count
Classes (All)
Classes With FK Attribu
Heavily Attributed Class
Heavily Populated Class
User Management
Glossary entries (All)
Orphaned Folders
Terminology Terms (All)
Type Definitions (All)
Unsent Emails (All)
XML Transforms (All)
My Elements (MARK)
My Elements (RONO)
My Favorites
Tutorial Workspace
Zachman Framework
eTask Management
Class Element Count
Filter Groups
Recycle Bin
EE Root
Default Initial Workspace
DoDAF Framework
Management Desktop
My Elements (MARK)
My Elements (RONO)
My Favorites
Tutorial Workspace
Zachman Framework
eTask Management
Class Element Count
Filter Groups
Recycle Bin
EE Root (12 Objects)
Ready
Free
http://caffeine:8082/ee

Figure 9

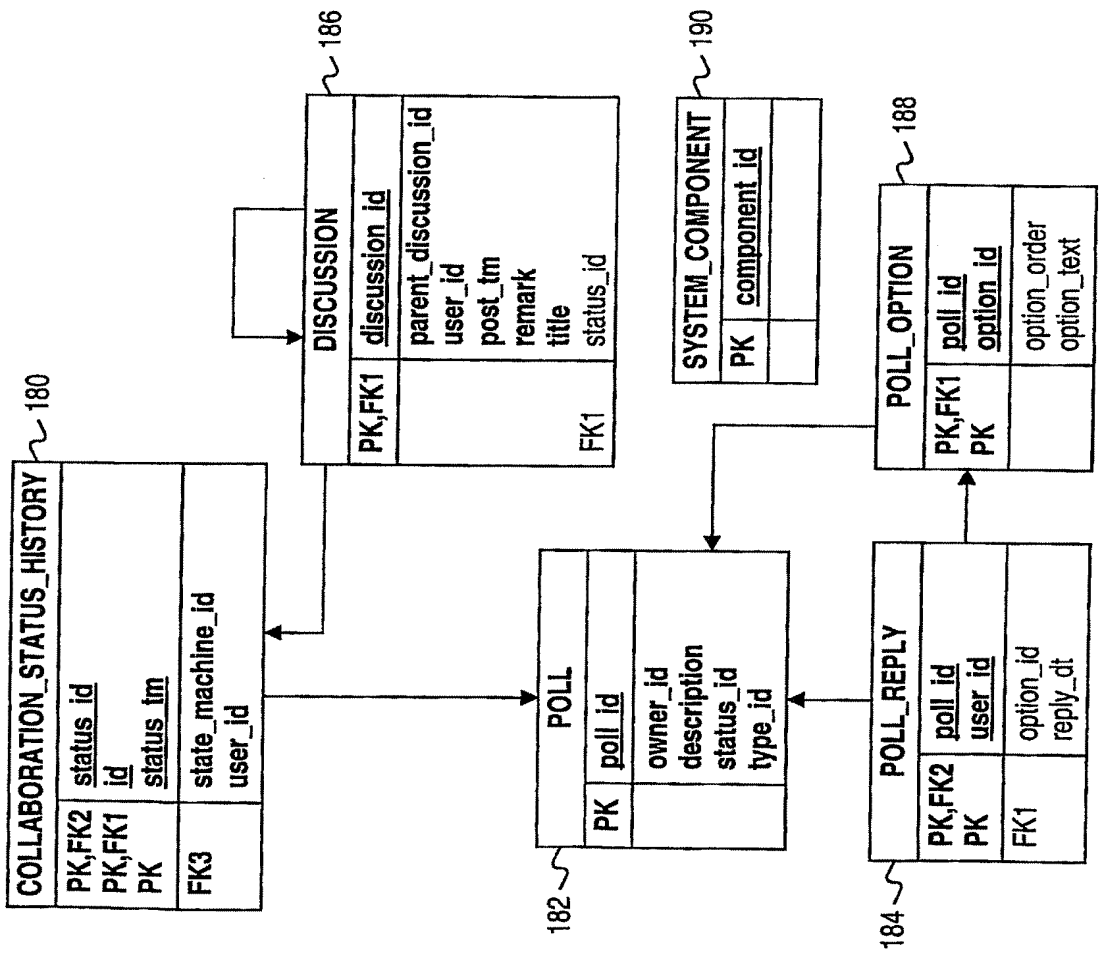

COLLABORATION_STATUS_HISTORY
180
PK,FK2 status_id
PK,FK1 id
PK status_tm
FK3 state_machine_id
user_id
DISCUSSION
186
PK,FK1 discussion_id
parent_discussion_id
user_id
post_tm
remark
title
FK1 status_id
SYSTEM_COMPONENT
190
PK component_id
POLL_OPTION
188
PK,FK1 poll_id
PK option_id
option_order
option_text
POLL
182
PK poll_id
owner_id
description
status_id
type_id
POLL_REPLY
184
PK,FK2 poll_id
PK user_id
FK1 option_id
reply_dt

Figure 10

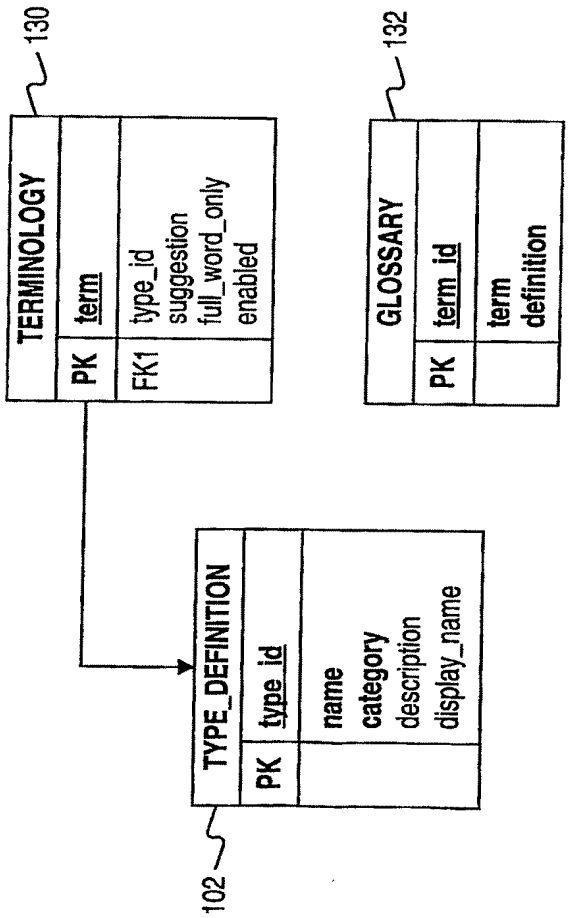

130
TERMINOLOGY
PK
term
FK1
type_id
suggestion
full_word_only
enabled
132
GLOSSARY
PK
term_id
term
definition
TYPE_DEFINITION
PK
type_id
name
category
description
display_name
102

Figure 11

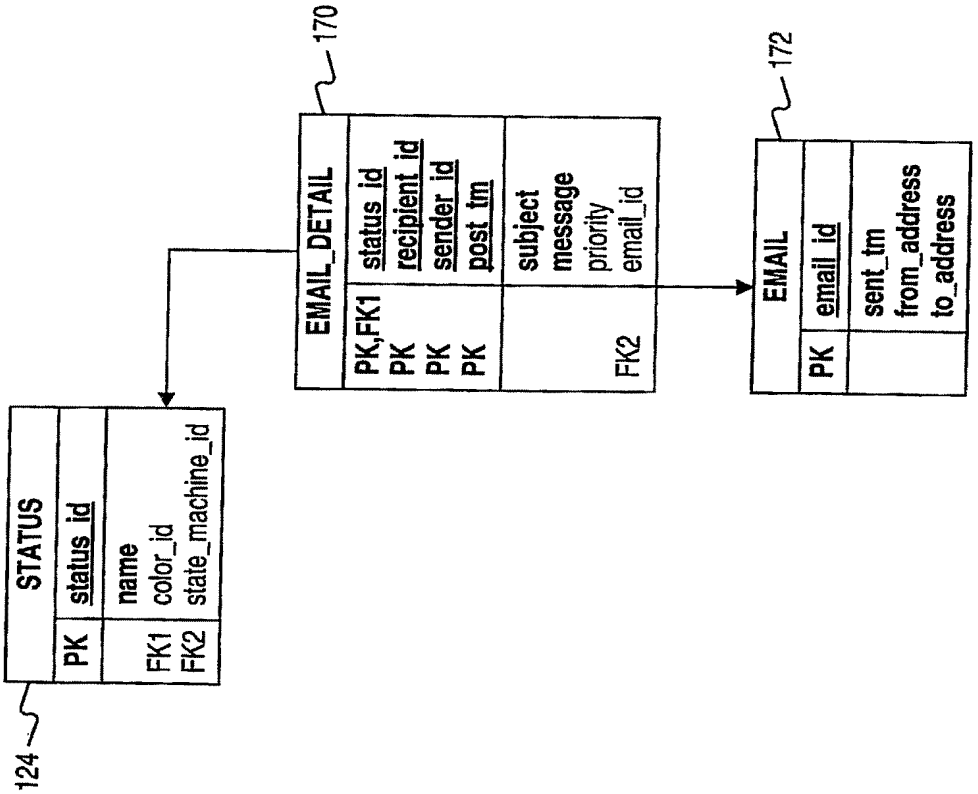

STATUS
PK status_id
name
FK1 color_id
FK2 state_machine_id
124
EMAIL_DETAIL
PK,FK1 status_id
PK recipient_id
PK sender_id
PK post_tm
subject
message
priority
FK2 email_id
170
EMAIL
PK email_id
sent_tm
from_address
to_address
172

Figure 12

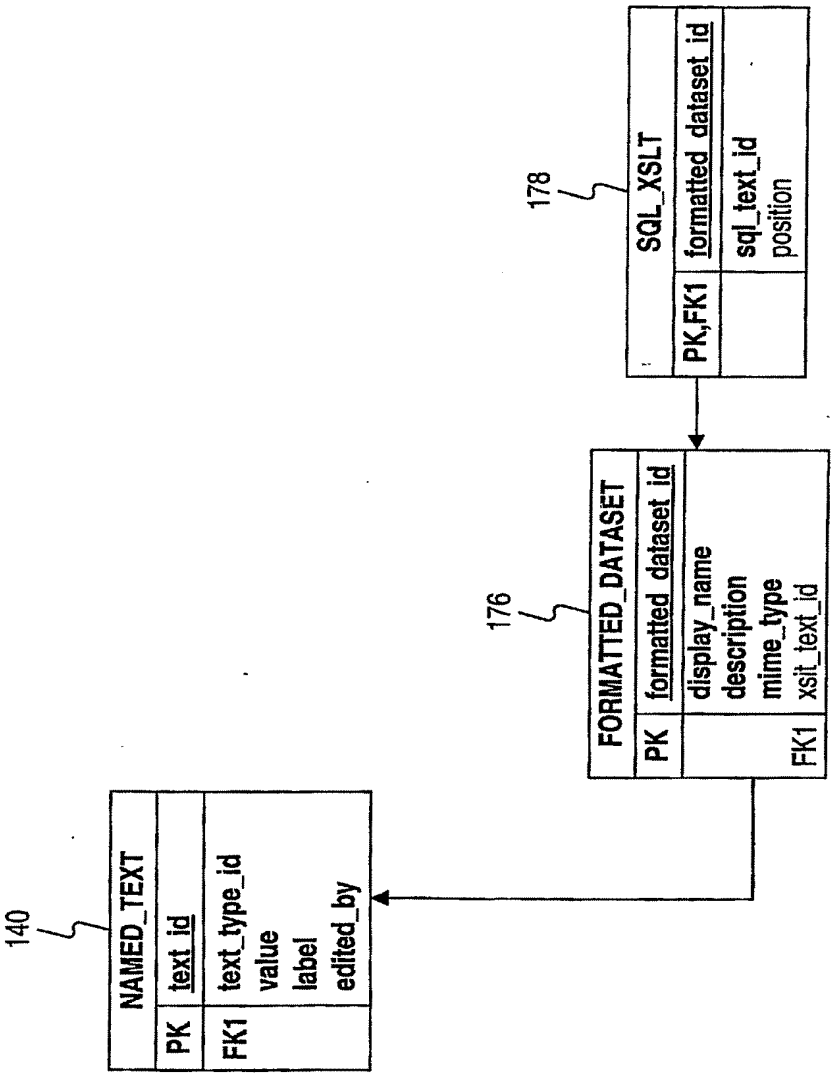

140
NAMED_TEXT
PK
text_id
FK1
text_type_id
value
label
edited_by
176
FORMATTED_DATASET
PK
formatted_dataset_id
display_name
description
mime_type
FK1
xsit_text_id
178
SQL_XSLT
PK,FK1
formatted_dataset_id
sql_text_id
position

Figure 13

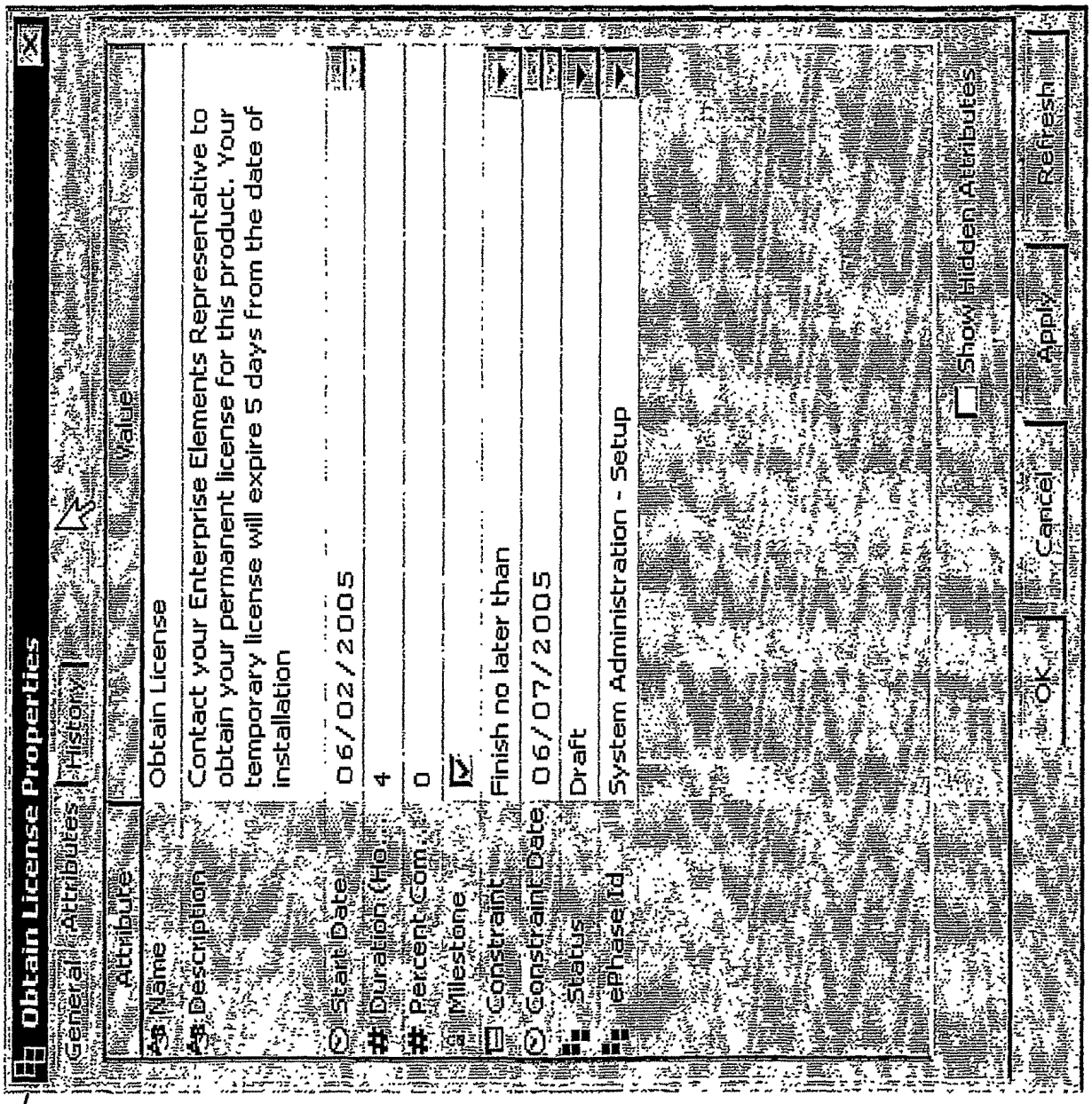

1400
Obtain License Properties
General
Attributes
History
Attribute
Value
Name
Obtain License
Description
Contact your Enterprise Elements Representative to obtain your permanent license for this product. Your temporary license will expire 5 days from the date of installation
Start Date
06/02/2005
Duration (Ho...
4
Percent_Com...
0
Milestone
Constraint
Finish no later than
Constraint Date
06/07/2005
Status
Draft
ePhaseId
System Administration - Setup
Show Hidden Attributes
OK
Cancel
Apply
Refresh

Figure 14

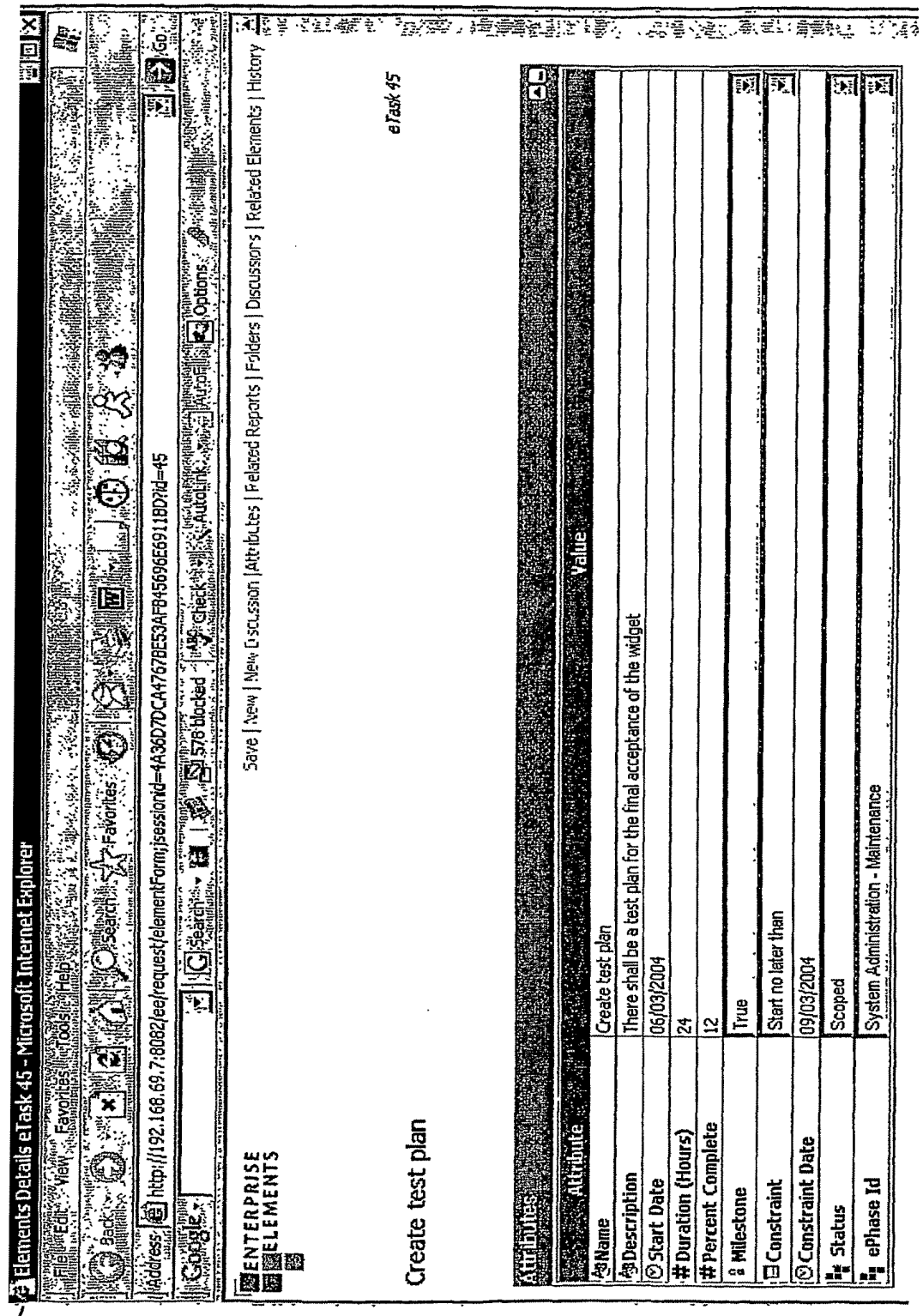

1500
Elements Details eTask 45 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://192.168.69.7:8082/ee/request/elementForm;jsessionid=4A36D7DCA4767BE53AFB456966E6911BD?id=45
Google Search 578 blocked Check AutoLink AutoFill Options
ENTERPRISE ELEMENTS
Save | New | New Discussion | Attributes | Related Reports | Folders | Discussions | Related Elements | History
Create test plan
eTask 45
Attributes
Attribute
Value
Name
Create test plan
Description
There shall be a test plan for the final acceptance of the widget
Start Date
06/03/2004
Duration (Hours)
24
Percent Complete
12
Milestone
True
Constraint
Start no later than
Constraint Date
09/03/2004
Status
Scoped
ePhase Id
System Administration - Maintenance

Figure 15

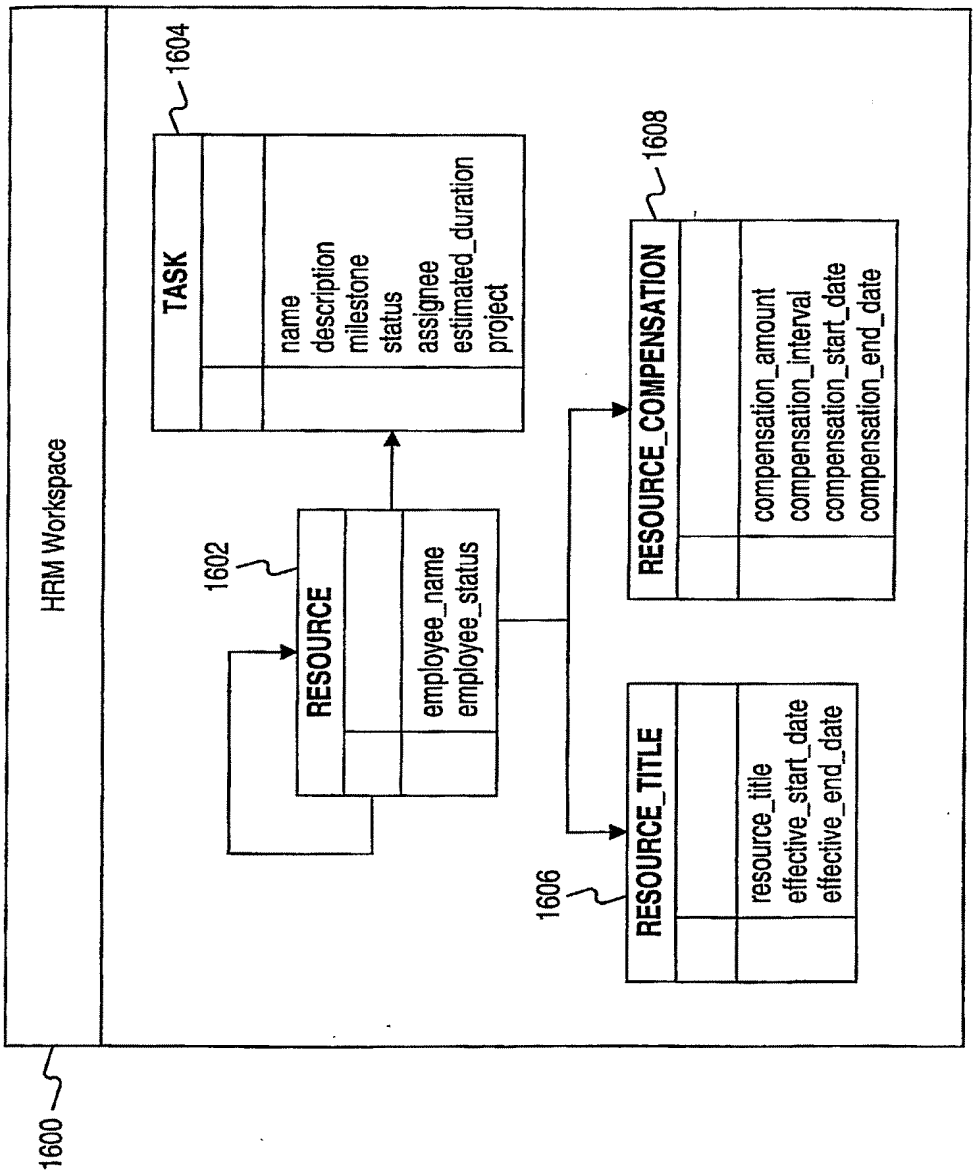

HRM Workspace
1600
1602
RESOURCE
employee_name
employee_status
1604
TASK
name
description
milestone
status
assignee
estimated_duration
project
1606
RESOURCE_TITLE
resource_title
effective_start_date
effective_end_date
1608
RESOURCE_COMPENSATION
compensation_amount
compensation_interval
compensation_start_date
compensation_end_date

Figure 16

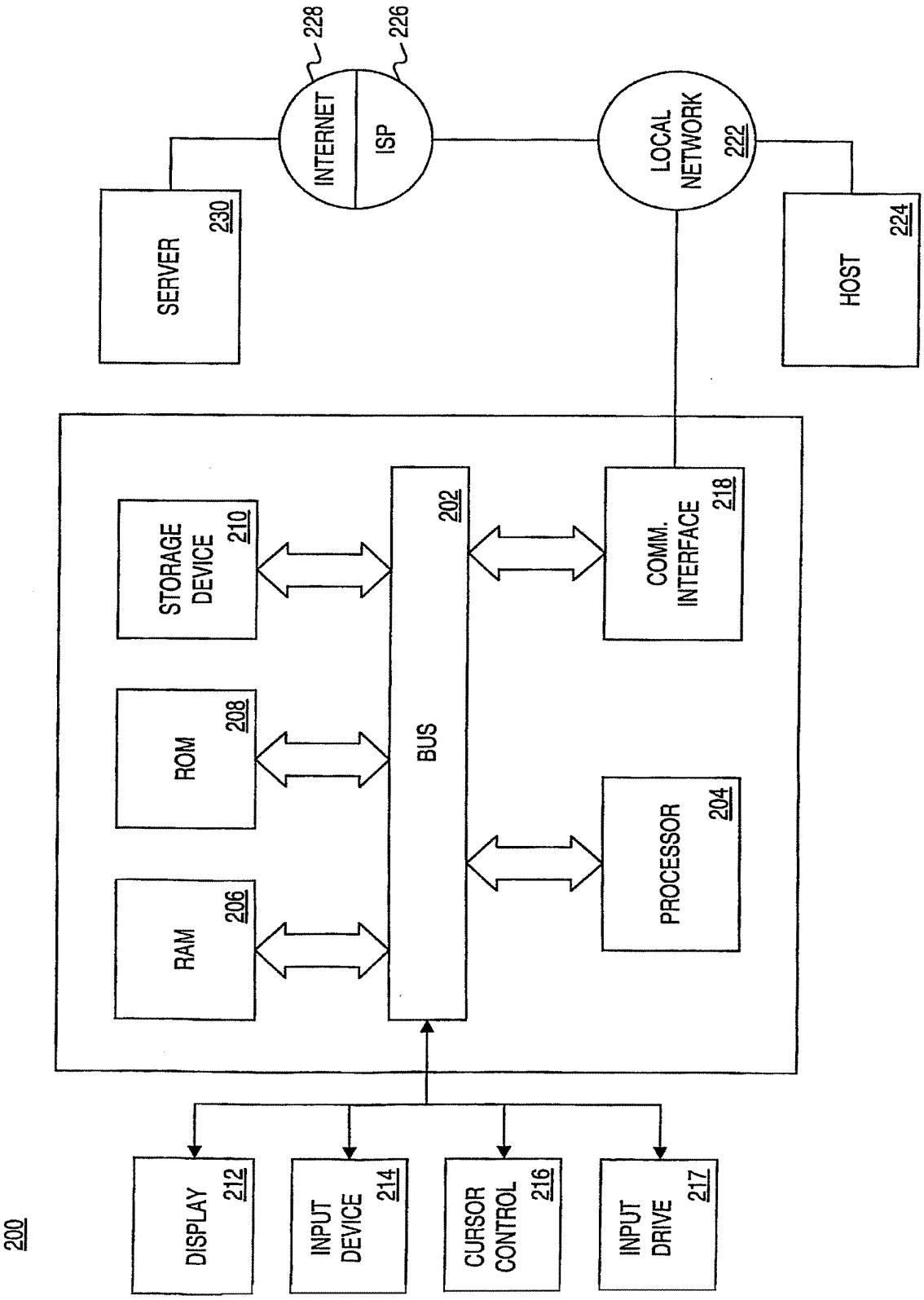


Figure 17

ServiceNow Ex. 1002 Page 398 of 410

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/023242

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	ANDRZEJ JODLOWSKI, PIOTR HABELA, JACEK PLODZIEN, KAZIMIERZ SUBIETA: "Objects and Roles in the Stack-Based Approach" PROC. OF THE 13TH INTERNATIONAL CONFERENCE ON DATABASE AND EXPERT SYSTEMS APPLICATIONS (DEXA 2002), AIX-EN-PROVENCE, FRANCE, LNCS, vol. 2453, 2 September 2002 (2002-09-02), pages 514-523, XP002401006 page 515, line 28 - page 516, line 13 page 518, line 18 page 520, line 5 - last line figure 3 ----- -/--	1-39

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
29 September 2006	12/10/2006

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Bykowski, Artur

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/023242

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	PIOTR HABELA, KAZIMIERZ SUBIETA: "Overcoming the Complexity of Object-Oriented DBMS Metadata Management" PROC. OF THE 9TH INTERNATIONAL CONFERENCE OBJECT-ORIENTED INFORMATION SYSTEMS (OOIS 2003), GENEVA, SWITZERLAND, 2 September 2003 (2003-09-02), pages 214-225, XP002401007 page 216, line 16 - last line page 220, line 2 - page 221, line 26 page 223, line 10 - page 224, line 31 ; figures 3,4	1-39
X	BLAHA M ET AL: "Converting OO models into RDBMS schema" IEEE SOFTWARE USA, vol. 11, no. 3, May 1994 (1994-05), pages 28-39, XP002401008 ISSN: 0740-7459 page 32, left-hand column, line 12 - page 34, left-hand column, line 12	1-39
X	SILYAN ARSOV, BORIS RACHEV: "An approach for design of the object-oriented databases with restricted natural language access" PROC. OF THE 4TH INTERNATIONAL CONFERENCE CONFERENCE ON COMPUTER SYSTEMS AND TECHNOLOGIES: E-LEARNING, SOFIA, BULGARIA, 19 June 2003 (2003-06-19), pages 147-152, XP002401009 page 148, line 8 - page 151, line 35	1,4,5,8, 9,12-14, 17,18, 21,22, 25-27, 30,31, 34,35, 38,39
A	NOEL JERKE, DARIN BEARD: "Building Database-Driven Flash Applications, chapter 5, Building a Flashy Online Poll Engine" [Online] October 2003 (2003-10), APRESS, SPRINGER-VERLAG BERLIN AND HEIDELBERG GMBH & CO. KG , BERKLEY, USA , XP002401199 ISBN: 1-59059-110-0 Retrieved from the Internet: URL:http://www.apress.com/ApressCorporate/supplement/1/186/1590591100-1450.pdf> [retrieved on 2006-09-28] page 122 - page 141	1-39

DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I hereby declare that: my residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

DATABASE DATA DICTIONARY

the specification of which
☐ is attached and/or
☐ was filed on as United States Application Serial No. and/or
☒ was filed on June 14, 2006 as PCT International Application No. PCT/US2006/023242 and
was amended on (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above. I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. § 119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate or § 365(a) of any PCT international application(s) designating at least one country other than the United States, listed below and have also identified below, any foreign application(s) for patent or inventor's certificate, or any PCT International application(s) having a filing date before that of the application(s) of which priority is claimed:

Country	Application Number	Date of Filing	Priority Claimed Under 35 U.S.C. 119
			☐ YES ☐ NO
			☐ YES ☐ NO

I hereby claim the benefit under 35 U.S.C. § 119(e) of any United States provisional application(s) listed below:

Application Number	Date of Filing
60/690,147	June 14, 2005

I hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s) or § 365(c) of any PCT International application(s) designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application(s) in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR § 1.56 which became available between the filing date of the prior application(s) and the national or PCT International filing date of this application:

Application Number	Date of Filing	Status (Patented, Pending, Abandoned)

I hereby appoint the following attorney and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P., CUSTOMER NUMBER 22,852.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are 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.

Full Name of First Inventor Bruce Randall DeWoolfson	Inventor's Signature	Date 12/10/2007
Residence 46290 Summerhill Place, Potomac Falls, VA 20165		Citizenship US
Post Office Address 46290 Summerhill Place, Potomac Falls, VA 20165		

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A SUBMISSION UNDER 35 U.S.C. 371		CUSTOMER NO. 22,852 ATTORNEY'S DOCKET NUMBER: 09801.0002 U.S. APPLICATION NO. if known, see 37 CFR 1.5(
INTERNATIONAL APPLICATION NO. PCT/US2006/023242	INTERNATIONAL FILING DATE June 14, 2006	PRIORITY DATE CLAIMED June 14, 2005
TITLE OF INVENTION DATABASE DATA DICTIONARY		
APPLICANT(S) FOR DO/EO/US Bruce Randall DEWOOLFSON		

Applicant(s) herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a submission under 35 U.S.C. 371.
3. ☐ This is an express request to begin national examination procedures (35 U.S.C. 371(f)). The submission must include items (5), (6), (9) and (21) indicated below.
4. ☐ The US has been elected (Article 31).
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c)(2)).
   a. ☒ is attached hereto (required only if not communicated by the International Bureau).
   b. ☒ has been communicated by the International Bureau.
   c. ☐ is not required, as the application was filed with the United States Receiving Office (RO/US).
6. ☐ An English language translation of the International Application as filed (35 U.S.C. 371(c)(2)).
   a. ☐ is attached hereto.
   b. ☐ has been previously submitted under 35 U.S.C. 154(d)(4).
7. ☒ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3)).
   a. ☐ are attached hereto (required only if not communicated by the International Bureau).
   b. ☐ have been communicated by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☒ have not been made and will not be made.
8. ☐ An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371 (c)(3)).
9. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
10. ☐ An English language translation of the annexes of the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

Items 11 to 20 below concern document(s) or information included:

11. ☒ Information Disclosure Statement under 37 CFR 1.97 and 1.98
12. ☐ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
13. ☐ A preliminary amendment.
14. ☐ An Application Data Sheet under 37 CFR 1.76.
15. ☐ A Substitute specification.
16. ☐ A power of attorney and/or change of address letter.
17. ☐ A computer-readable form of the sequence listing in accordance with PCT Rule 13*ter*.2 and 35 U.S.C. 1.821-1.825.
18. ☐ A second copy of the published International Application under 35 U.S.C. 154 (d)(4).
19. ☐ A second copy of the English language translation of the international application 35 U.S.C. 154 (d)(4).
20. ☒ Other items or information:
    a. ☒ Copy of cover page of International Publication No. WO 2006/138420 A1
    b. ☐ Copy of Notification of Missing Requirements.
    c. ☒ International Search Report from the European Patent Office
    d. ☐
    e. ☐
    f. ☐
    g. ☐

U.S. APPLICATION NO. (if known, see 37 CFR 1.5)	INTERNATIONAL APPLICATION NO. PCT/US2006/023242	ATTORNEY'S DOCKET NUMBER ☐ 09801.0002	
The following fees have been submitted:		CALCULATIONS	PTO Use Only
21. ☐ BASIC NATIONAL FEE (37 CFR 1.492(a)): **$310.00**		$310.00	
22. ☐ Examination fee (37CFR 1.492(c)) If the Written opinion prepared by ISA/US or the International preliminary examination report prepared by IPEA/US indicates all claims satisfy provisions of PCT Article 33(1)-(4) ..... **$0** All other situations..... **$210**		$210.00	
23. ☐ Search fee (37 CFR 1.492(b)) If the written opinion of the ISA/US or the International preliminary examination report prepared by IPEA/US indicates all claims satisfy provisions of PCT Article 33(1)-(4) ..... **$0** Search fee (37 CFR 1.445(a)(2)) has been paid on the international application to the USPTO as an International Searching Authority..... **$100** International Search Report prepared by an ISA other than the US and provided to the Office or previously communicated to the US by the IB..... **$410** All other situations..... **$510**		$410.00	
TOTAL of 21, 22 and 23 =		$930.00	
☐ Additional fee for specification and drawings filed in paper over 100 sheets (excluding sequence listing in compliance with 37 CFR 1.821(c) or (e) or computer program listing in an electronic medium) (37 CFR 1.492(j)). The fee is **$250** for each additional 50 sheets of paper or fraction thereof.			
Total Sheets: 63 - 100 =	Extra Sheets: /50=	Number of each additional 50 or fraction thereof (round **up** to a whole number)	Rate: x $260.00
			$
Surcharge of **$130.00** for furnishing any of the search fee, examination fee, or the oath or declaration after the date of commencement of the national stage (37 CFR 1.492(h)).			$

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE		
Total Claims	39 -20 =	19	x $50.00	$950.00	
Independent Claims	-3 =		x $210.00	$	
MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+$370.00	$	
			TOTAL OF ABOVE CALCULATIONS =	$1880.00	
☐ Applicant claims small entity status. See 37 CFR 1.27. Fees above are reduced by ½.				$	
			SUBTOTAL =	$1880.00	
Processing fee of **$130.00** for furnishing the English translation later than 30 months from the earliest priority date (37 CFR 1.492(i)). +				$	
			TOTAL NATIONAL FEE =	$1880.00	
Fee for recording the enclosed assignment (37 CFR 1.21 (h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). **$40.00** per property. +				$	
			TOTAL FEES ENCLOSED =	$1880.00	
				Amount to be refunded:	$
				charged:	$

a. ☒ This application is being filed electronically. The fees are being paid by credit card. The filing fees of $1880.00 are included.

b. ☐ Please charge my Deposit Account No. _____ in the amount of $ _____ to cover the above fees. A duplicate copy of this sheet is enclosed.

c. ☒ The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. 06-0916 . A duplicate copy of this sheet is enclosed.

d. ☐ Fees are to be charged to a credit card or to Deposit Account No. 06-0916. **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.

NOTE: Where an appropriate time limit under 37 CFR 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the International Application to pending status.

SEND ALL CORRESPONDENCE TO:

Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.
901 New York Avenue, N.W.
Washington, DC 20001-4413

SIGNATURE

Ernest F. Chapman/Reg. No. 25,961
NAME/REGISTRATION NUMBER

DATED: December 13, 2007

Substitute for form 1449A/PTO				Complete if Known	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT				Application Number	
				Filing Date	December 13, 2007
				First Named Inventor	Bruce Randall DEWOOLFSON
				Art Unit	
(Use as many sheets as necessary)				Examiner Name	
Sheet	1	of	1	Attorney Docket Number	09801.0002

U.S. PATENTS AND PUBLISHED U.S. PATENT APPLICATIONS

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

Note: Copies of the U.S. Patent Documents are not Required in IDS filed after October 21, 2004

FOREIGN PATENT DOCUMENTS

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

NON PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No.[1]	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	Translation[6]
		JODLOWSKI, ET AL., "Objects and Roles in the Stack-Based Approach", XP-002401006, Proc. of the 13th International Conference on Database and Expert Systems Applications (DEXA 2002), Vol. 2453, pages 514-523, (September 2, 2002)	
		HABELA, ET AL., "Overcoming the Complexity of Object-Oriented DBMS Metadata Management", XP-002401007, Proc. of the 9th International Conference Object-Oriented Information Systems (00IS 2003), pages 214-225, (September 2, 2003)	
		BLAHA, ET AL., "Converting OO Models into RDBMS Schema", XP-002401008, IEEE Software USA, Vol. 11, No. 3, pages 28-39, (May 1994)	
		ARSOV, ET AL., "An Approach for Design of the Object-Oriented Databases with Restricted Natural Language Access", XP-002401009, Proc. of the 4th International Conference Conference on Computer Systems and Technologies: E-Learning, pages 147-152, (June 19, 2003)	
		JERKE, ET AL., "Building Database-Driven Flash Applications", XP-002401199, APRESS, SPRINGER-VERLAG BERLIN AND HEIDELBERG GMBH & CO. KG, pages 119-170, (October 2003)	

Examiner Signature		Date Considered	

Electronic Patent Application Fee Transmittal

Application Number:	
Filing Date:	
Title of Invention:	DATABASE DATA DICTIONARY
First Named Inventor/Applicant Name:	Bruce Randall DEWOOLFSON
Filer:	Ernest F. Chapman/ROSIE CREMEANS
Attorney Docket Number:	09801.0002

Filed as Large Entity

U.S. National Stage under 35 USC 371 Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
National Stage Fee	1631	1	310	310
Natl Stage Search Fee - Report provided	1642	1	410	410
National Stage Exam - all other cases	1633	1	210	210
Pages:				
Claims:				
Claims in excess of 20	1615	19	50	950
Miscellaneous-Filing:				
Petition:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**1880**

Electronic Acknowledgement Receipt

EFS ID:	2586329
Application Number:	11917410
International Application Number:	PCT/US06/23242
Confirmation Number:	6234
Title of Invention:	DATABASE DATA DICTIONARY
First Named Inventor/Applicant Name:	Bruce Randall DEWOOLFSON
Customer Number:	22852
Filer:	Ernest F. Chapman/ROSIE CREMEANS
Filer Authorized By:	Ernest F. Chapman
Attorney Docket Number:	09801.0002
Receipt Date:	13-DEC-2007
Filing Date:	
Time Stamp:	14:37:54
Application Type:	U.S. National Stage under 35 USC 371

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$1880
RAM confirmation Number	8864
Deposit Account	060916
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)

1	Transmittal of New Application	0002-TRANS.pdf	110266 9318d969e0b5ba61d473c7d327e987b3a6ff73f7	no	2
Warnings:					
Information:					
2		0002-SPEC-DRAW-ISR.pdf	3761418 1078b5363f7bfe9f8fc878c9d05584b4f0970189	yes	67

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Application Data Sheet	1	2
	Specification	3	35
	Claims	36	46
	Drawings-only black and white line drawings	47	65
	Documents submitted with 371 Applications	66	67

Warnings:					
Information:					
3	Oath or Declaration filed	0002-DEC.pdf	183015 5fb5b3a6b036c3690581a1a8d9e6795547142fa3	no	2
Warnings:					
Information:					
4		0002-IDS.pdf	103997 8b202e214b6efd7483ab23b56cdda719d7ab367a	yes	3

	Multipart Description/PDF files in .zip description		
	Document Description	**Start**	**End**
	Information Disclosure Statement Letter	1	2
	Information Disclosure Statement (IDS) Filed	3	3

Warnings:					
Information:					
5	NPL Documents	0002-NPL1.pdf	722847 0b60338e680277cef3dfff39a4840e5987dee962	no	6
Warnings:					
Information:					

6	NPL Documents	0002-NPL2.pdf	2259158 2bcafcffab744cc8c0c6b444db650cb7ed129d2c	no	52
Warnings:					
Information:					
7	NPL Documents	0002-NPL3.pdf	1135449 6761df958f52a373703cab820a1ec3ed130e3650	no	12
Warnings:					
Information:					
8	NPL Documents	0002-NPL4.pdf	915277 0022e8f7c1c04a526b3d232a56b661f0da105067	no	10
Warnings:					
Information:					
9	NPL Documents	0002-NPL5.pdf	831727 08802fa0d05e2049561478fcdaf56f20237fd2b7	no	12
Warnings:					
Information:					
10	Fee Worksheet (PTO-06)	fee-info.pdf	8526 9ed632b8f4c6c2c8939a7b0453c8e6aaa926938b	no	2
Warnings:					
Information:					
			Total Files Size (in bytes):	10031680	

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.